Creating a Gamified Tutorial for Socio-Technical Security Requirements Engineering Education

Author:
Daniel Alami Cabezas (4238737)
d.alamicabezas@students.uu.nl

Supervisors:
1st Supervisor
dr. Fabiano Dalpiaz
F.Dalpiaz@uu.nl

2nd Supervisor
dr. Joske M. Houtkamp
J.M.Houtkamp@uu.nl

M.Sc. Business Informatics

Universiteit Utrecht

Thesis Report (April, 2017)

Abstract

Thanks to the advent of interactive technologies, education institutions are looking for innovative teaching methods to increase the engagement and reach of students. Besides the uprise of MOOCs, gamification has been shown to produce positive results when it comes to increasing people's engagement and interest in conducting tasks. Unfortunately, the application and benefits of these technologies in teaching software engineering (SE) and requirements engineering (RE) -the latter being our area of focus- remain largely unexplored.

In this thesis, we introduce the STS-Tooltorial, an interactive gamified platform that executes within a security requirements modeling tool and helps learners apprehend the STS-ml language and basic notions about security requirements.

In addition, we present the framework and design principles that underpinned the design of our gamified interactive tutorial, including best practices for gamifying learning experiences and instructional design considerations.

Finally, we report a two-country (Spain, Netherlands), two-population (postgraduate students, IT professionals) early evaluation focused on the platform's effectiveness and usability.

Table of Contents

Glossary of Abbreviations

ICT Information and Communications Technology

ILO Intended Learning Outcome

IS Information Systems or Information Science(s)

MOOC Massive Open Online Course

NIST National Institute of Standards and Technology

OCL Object Constraint Language

PBL Points, Badges, and Leaderboards

PDD Process-Deliverable Diagram

PIT Personal Investment Theory

QuaSMOD Quality of Security Requirements Models

RE Requirements Engineering

ROI Return on Investment

RQ Research Question

SE Software Engineering

SDT Self-Determination Theory

SRE Security Requirements Engineering

List of Figures

List of Tables

Acknowledgements

After over a year of sometimes constant and other times not-so-constant dedication, these pages cannot be devoid of expressions of gratitude.

I would primarily like to thank my first supervisor for his utmost unconditional feedback and for being flexible, adaptive, and supportive throughout the whole process. I have come to greatly appreciate his relentless dedication to teaching and shaping the spirit of young graduates. Thank you, Fabiano.

My thanks go next to my employers at International Business Machines, S.A (IBM Spain) for giving me room and space to THINK creatively about the challenges of gamification as well as see through the completion of this research. In the final stages of this journey they also provided us with participants for conducting our empirical evaluation.

The student evaluation could not have been possible without the help of eight MBI students from Utrecht University. Thank you guys and girls, there are no words for expressing my gratitude. I know still owe you an Amazon gift card: it will eventually come, I promise.

Finally, I am grateful to María for being the best partner one could ever dream. Thank you.

I. Foundation

In this block, we motivate the raison d'être of this thesis and describe the research method that underpins our research. In the first chapter, we begin by exploring the importance of (Security) Requirements Engineering, and propose a series of research goals and questions with the aim of improving education in this field. Subsequently, we justify the scientific and societal contributions of our research. Starting from these goals and questions, the second chapter describes the research methods and strategies that have been followed throughout this thesis.

1. Introduction

The pervasiveness of Information and Communications Technology (ICT) cannot be denied. The modern world has become so reliant on Information Systems (IS) that a seemingly small failure (e.g., malicious data entry allowed in a web form) can have negative, and even disastrous consequences (e.g., leakage of personal information). Failure to get the right requirements that fit the intended purpose of a software system is seen as a key cause of many project failures, ranging from pure project cancellation to budget, scope, schedule, or quality issues (Charette, 2005).

For that reason, Requirements Engineering (RE) stands as a crucial activity for creating high-quality software and a vital component in successful project development (Pohl, 2010). RE sets out to tackle three main challenges associated with requirements: defects, their cost, and the consequences that such defects entail. Defects are often introduced in the RE phase of software development. For instance, Wiegers (2001) notes that the percentage of defects originating during requirements engineering is 50%[1]. Such defects are also costly. Several studies have noted that reworking requirements costs 50% to 80% of total project effort (Jones, 1986; Wiegers, 2001). Once fielded, they cost 10 to 100 times as much to correct than if they had been detected during the design or development stages (McConnell, 2001). Finally, requirements defects snowball, affecting activities that follow in the lifecycle (e.g., architecture, development, implementation, testing, or reuse). As a rule of thumb, the longer a defect exists, the harder and costlier it is to fix.

A crucial quality requirement of any software system is security. Failing to consider security requirements early in the development process also has costly implications. From a monetary perspective, for instance, code fixes performed after releasing software can result in 30 times the cost of fixes performed during the design phase (NIST, n.d.). Similarly, the cost of a data breach for an organization is $3.8 million, with an average cost per record of information stolen of $154 (Ponemon, 2015). In 2002, the National Institute of Standards and Technology (NIST) estimated that security and reliability software faults cost the economy $59.5 billion annually. Many of these breaches occur because systems are designed and engineered considering security as an afterthought (Stallings & Brown, 2012).

[1] The figures reported in pages 15 and 16 regarding the effects of "bad" requirements, security breaches, and benefits of RE are hard-to-quantify estimations and, as such, need to be interpreted with caution.

In light of this unsettling reality, it is not surprising that security is viewed as an important problem in practice and, naturally, information security has garnered the interest of industry and academia in recent years. Organizations are devoting resources to tackle security challenges. But how can organizations handle this situation? Soo Hoo (2001) observes that the Return on Investment (ROI) of introducing early secure engineering practices ranges from 12 to 21 percent, with the highest return occurring when they are introduced during the design phase (i.e., security by design). Since then, several studies have highlighted the importance of including security during the early (and all) stages of the systems development process (Mead, Hough, & Stehney, 2005; McGraw, 2006). These efforts fall under the domain of Security Requirements Engineering (SRE), a research discipline that emerged more than a decade ago in response to the monetary expenditures associated with (bad) security (Dubois & Mouratidis, 2010).

For ICT-driven environments, the quality of the (security) requirements engineering process is of critical importance. The design of a high-quality information system is a complex task that, among other things, requires experience as well as rigorous technical and business skills. One of the such critical skills is capturing requirements into formal or semi-formal representations: models.

Models are used for understanding the world around us, as well as for creating new artifacts. The modeling activity is at the core of system design. Conceptual modeling is seen as a key activity to reduce the complexity of a problem domain and integrate multiple perspectives in the system design process (e.g., business and technical expertise), thereby fostering stakeholder communication (Wand, Monarchi, Parsons, & Woo, 1995). Moreover, such models help analysts to record and precisely state requirements, capture design decisions, generate usable work products, explore multiple solutions economically, master complex systems, and increase analysts understanding of the domain (Wand et al., 1995; Bezivin et al., 2009).

Interestingly, as **Section 3.1** reveals, most SRE approaches advocate the use of modeling languages to create models of the system. In the case of SRE, models allow to precisely document and analyze security requirements together with design requirements (Basin, Clavel, & Egea, 2009). Consequently, in the context of SRE, it follows that developing, manipulating, and understanding models is an important skill to master. We observe that mastering this skill is responsibility of the analyst during his requirements engineering education. However, in line with modern educational theories, this responsibility also has to be shared by the educator to ensure an effective learning process.

However, teaching conceptual modeling is not easy. Factors underlying the quality of a conceptual model range from the understanding that the modeler has about the modeling language, its rules, semantics, and constructs, to knowledge about the application domain to be modeled (Nelson, Poels, Genero, & Piattini, 2012). Irrespective of the language, the modeling activity is predominantly based on personal experience and tacit knowledge, and it is difficult to transfer to novice modelers. Consequently, teaching these skills and knowledge becomes a challenging undertaking. While high-quality conceptual models are seen as a key contributor to the quality of information systems (ISO 25000), teaching the necessary skills becomes a problematic for a plethora of reasons. Sedrakyan, Snoeck, and Poelmans (2014) identify the following factors:

- ***High cognitive load***: due to the difficulty of the task, novice modelers usually follow a procedural (i.e., one-task-at-a-time) approach that often results in overlooked aspects with regard to the quality of the model itself (e.g., missing requirements). Experts, on the other hand, simultaneously switch between model generation and cross-validation activities (e.g., detection of conceptual errors or omitted requirements), with the latter comprising nearly 80% of the design activity (Wang & Brooks, 2007).
- ***Insufficient domain knowledge***: lack of intensive trial-and-error in the classroom is identified as the major limitation in novices' modeling experience (p. 368). This is further complicated by the fact that application domain knowledge (i.e., familiarity with the context) is a strong predictor of task performance and model quality, at least in contexts where requirements are not fully specified (Hambrick & Engle, 2002; Dhillon & Dasgupta, 2011).
- ***Lack of tool support***: Barjis et al. (2012) point to the absence of technical artifacts such as computer-assisted learning tutorials as a major factor to the lack of preparedness and skills of students and professionals.
- ***Inexistent validation procedures***: the tacit nature of the conceptual modeling activity makes it hard to define established validation procedures, and thus is harder to learn (c.f. Shanks, Tansley, & Weber, 2003)
- ***Complex modeling tools***: industry tools are often packed with an overwhelmingly amount of features, which hinder the learning process (Siau &

> Loo, 2006). Most academic tools, on the other hand, are complex because either their user interface is complicated, or are packed with features that are sophisticated and hard to understand.

Based on these observations, it can be concluded that educating novice modelers is an important endeavor that must be met with educational interventions. In the next subsection we offer a glimpse of other current problems surrounding this issue, and position our work in relation to these.

1.1 Problem Statement

Education related to (SRE) modeling has recently garnered the interest of academia, but there are several open challenges that touch upon the use of tools, model quality, and engagement (Börstler, Kuzniarz, Alphonce, Sanders, & Smialek, 2012). There are few tools suited for educational purposes. Also, the lack of an accepted definition of model quality hinders its teaching. Another issue that surfaces is: how can modeling be taught, while keeping learners engaged?

Taking SRE as our focus, this thesis explores how to effectively teach a particular SRE method: the Socio-Technical Systems (STS) method, a goal-oriented and model-driven approach for designing secure socio-technical systems (Dalpiaz, Paja, & Giorgini, 2016). We choose to focus on this method because of its unique perspective to SRE, namely, its focus on understanding security issues not only from the technical, but also from the organizational and social contexts from which systems operate, i.e., from a socio-technical systems perspective (Trist, 1981; Sommerville et al., 2011). Socio-technical systems pervade our lives, and failing to acknowledge this fact during the design of (secure) systems can have severe consequences, such as lower end-user acceptance and less value to stakeholders (Baxter & Sommerville, 2011). In this regard, STS is thus one of the few SRE approaches concerned with the interactions between humans, organizations, and (software) subsystems. A recently developed method, STS is interesting due to its powerful expressive capabilities, adherence with standards, clear representation of assets, and operationalizability (Dalpiaz et al., 2016). The pragmatic benefits of this approach combined with its recency, make STS an enticing method to learn for people (e.g., students or professionals) who want to sharpen their security requirements skills and differentiate themselves from the competition.

Currently, STS is introduced in (under)graduate courses as part of IS curricula. A book on this topic has been recently published by its authors (Dalpiaz et al., 2016).

However, and returning to our original observation, how could this model-driven method be taught (and learned) in a more innovative and engaging way?

1.2 Research Objectives and Research Questions

The previous question has driven our research, and is materialized into the following two main goals of this thesis:

G1	**Improving** the **learning experience** and **increase** learner **engagement** while learning socio-technical security requirements engineering, compared to other traditional approaches

G2	Devising an assessment framework and guidelines for **determining** and **improving** the **quality** of security requirements models

A suitable way of tackling the first goal is by creating a gamification-powered interactive tutorial system that presents the STS Method in a holistic way. The theoretical underpinnings of gamification are later described in **Section 3.4**, but, for now, a practical definition is "the use of game design elements in non-game contexts" (Deterding, Dixon, Khaled, & Nacke, 2011, p. 9).

Thus, from **G1**, the following research question and sub-questions are formulated:

RQ1	How can gamification be suitably applied in the context of SRE to improve the learning experience and engagement of learners?

- SQ1 Which are intended learning outcomes (ILOs) of (socio-technical) security requirements engineering for first-time learners?
- SQ2 What aspects can be gamified to improve the effectiveness of an interactive tutorial?
- SQ3 How can an interactive tutorial for teaching a socio-technical security requirements method (viz., STS) be designed?
- SQ4 To what degree do gamified tutorials show any promise for SRE education?

The second goal, **G2**, aims to tackle the problem identified in the previous section regarding the lack of a formal definition of model quality, as well as strategies modelers can employ to improve security requirements models. Therefore, this leads to the second research question and associated sub-question.

RQ2 How can the quality of security requirements models be appraised and improved?

- **SQ5** What does quality of security requirements models mean?
- **SQ6** How can the quality of security requirements models be appraised?
- **SQ7** What improvement strategies can be executed to improve the quality of security requirements models?

1.3 Research Contributions

This section outlines the relevance of the problem employing two perspectives, namely: a scientific perspective that discusses the contribution of this thesis to the research community, and a social viewpoint that stresses the impact of this work to society.

1.3.1 Scientific Contribution

From a scientific stance, the contribution of this thesis can be viewed through two lenses, as it contributes to the body of knowledge on gamification (of learning) and (security) requirements engineering.

First, the overview of literature on innovative teaching methods represents the first comprehensive analysis that has been done to date in the academic literature. Researchers can draw on this review to get a perspective of what advances and limitations in didactics can be found in the RE and conceptual modeling areas. Similarly, scholars or professionals can use the list of guidelines for 1) designing interactive tutorials, 2) gamifying learning experiences, and 3) game elements compiled in this thesis for their own research endeavors.

Second, the gamified interactive tutorial represents the first empirical account of applying gamification in the context of conceptual modeling and one of the earliest in requirements engineering. From a research perspective, the empirical evaluation of the tutorial sheds light on whether gamification holds any promise in the case of security requirements engineering and in teaching conceptual modeling. The research approach and its limitations outlined in this thesis serve as a foundation for replicating the adoption of gamification in other domains of the educational sphere, and ascertaining its effectiveness. For example, the choice of gamification elements is greatly informed by well-established theories such as self-determination theory and cognitive evaluation theory.

Third, a new quality framework for conceptual modeling in the context of security requirements has been developed drawing on multiple theories. From a research viewpoint, the artifact can be used in studies that investigate the mechanisms that lead to successful conceptual model applications of secure socio-technical systems. These studies might in turn inform novice conceptual modelers on how to develop better security requirements models, and ensure that they perceive the quality of such improvements. We therefore contend that our preliminary framework can serve as a complementary instrument for evaluating the quality of conceptual models from a socio-pragmatic viewpoint. Further, the framework is a step forward towards finally corroborating the relationship between the quality of conceptual models and actual system quality.

To further augment our reach, all but the last contribution (i.e., quality framework) have been submitted for publication in a peer-reviewed venue and is included in Appendix V.

1.3.2 Social Contribution

From a societal perspective, the results of this research are valuable to a number of parties. The discussion below is structured around the main deliverables.

The gamified tutorial can be beneficial for Software Engineering (SE) practitioners as well as educators and organizations. First, the empirical evaluation sheds on the benefits and tradeoffs associated with the development of gamified tutorials. In this sense, software designers can make informed decisions when developing these, and the scientific community and other interested parties have more empirical knowledge of their effectiveness. Second, educators can resort to this research to find insights on the suitability of interactive tutorials. We hope that this research serves as a groundwork and sparks the interest of educators in applying innovative methods to enrich the training of the future generation of professionals. Similarly, organizations can make use of the tool or apply it in other domains for training professionals. In our case, designing socio-technical systems with a focus on security concerns in the early stages of software design undoubtedly has a strong societal and economic impact, especially now that organizations are increasingly facing complex privacy and security issues. At a minimum, it should help to shed light on what approaches can be followed in the design of systems to improve the current situation around security concerns.

With regard to the proposed framework, when applied in practice it can be used to evaluate and compare the perceived quality of alternative models or refined versions of the same model (or parts of it). The framework aims at being easy to use and applicable in daily practice, and stresses the importance of ensuring stakeholder satisfaction. Indeed, (internal/external) stakeholders are unlikely to sign-off a model if they are not fully committed to and agree with it. In this sense, the strategies laid out for improving the quality dimensions represent one of the first accounts on how to practically apply such a framework. Moreover, the measurement instrument can be used to gauge stakeholder perceptions on the quality of (improved) versions of models, and can greatly inform decision-making activities such as when to stop improving a model. Therefore, we argue that the framework represents a stepping stone towards improving the quality of security requirements models for (socio-technical) systems. Finally, note that learners can also make use of the framework to better understand the factors underlying the quality of such models.

1.4 Thesis outline

This chapter ends with a glimpse of the remainder of the thesis. After having introduced the problem statement and the objectives of this researcher, **Chapter 2** delves into the research approach followed during the study. Subsequently, **Chapters 3** and **4** lay out the theoretical and conceptual framework underpinning our research. **Chapter 5** and **6** describe the design and evaluation of the main deliverable of this thesis. **Chapter 7** sheds light on the results of the study, the implications of which are discussed in **Chapter 8**. Finally, **Chapter 9** elaborates on the conclusions and provides further research directions. This report is supplemented by five Appendixes (I-V). Worthy of noting here is **Appendix V**, which contains a scientific publication to be presented at a major international conference that provides a gentle introduction to key studies conducted as part of this thesis.

2. Research Method

The goals outlined in **Section 1.2** have a direct impact on the research methods and strategies that are appropriate to achieve them. The research process starts out with several literature reviews on the domains of interest, followed by interviews with domain experts. Although there is overlap with respect to some of the research methods applied during the different phases (e.g., survey research), the purpose of the methods and the nature of the phases change as we approach the final objective, from an exploratory to a design-based, although also exploratory, approach.

Given that the main deliverables is an Information Science (IS) artifact, the research is to be conducted over the umbrella of design science research. This approach is particularly useful when the solution to a problem entails the design of abstract artifacts (Vaishnavi & Kuechler, 2004, 2007; Wieringa, 2014). There has been some controversy over design science. In one of its earliest accounts, Takeda, Veerkamp, Tomiyama, and Yoshikawa, (1990) defined design research as consisting of design cycles with the following steps: awareness, suggestion, development, evaluation, and conclusion. Moreover, it was only deemed appropriate for projects of an exploratory nature. In the context of IS, scientists reformulated the method and included evaluation and validation activities at the core of the method (Hevner, March, Park, & Ram, 2004). Indeed, authors such as Peffers, Tuunanen, Rothenberger, and Chatterjee (2007) stress the importance of creating artifacts that solve real-world problems, and caution researchers to consider the explicit applicability of research solutions (cf. Wieringa (2014), who argues that validation of implemented artifacts are beyond the scope of design science). We strongly agree with this proposition, and argue that a good combination of theoretical *and* practical contributions is necessary to have an impact on the research fields under study. We attempt to provide these contributions with the development of a gamified tutorial and, as an appendix, a quality framework for security requirements models[2].

The discussion of the research approach can be structured in terms of the two main deliverables of this thesis and their associated research objectives. We explain the research approach by means of a rational problem decomposition based on the engineering cycle (EC), as outlined by Wieringa (2014). The EC consists of the following phases:

[2] Please observe that the quality framework represents an de-prioritized work product of this research due to time constraints. It is included in this thesis as an appendix (see **Appendix IV**).

- Problem investigation
- Treatment design
- Treatment validation
- Treatment implementation
- Implementation evaluation

In this thesis, we only tackle the first three phases the cycle, and use two cycles (viz., **EC1** and **EC2**) for each of the two main deliverables. For the sake of completeness, we draw connections between these phases and those proposed by Takeda et al. (1990) when appropriate (denoted by brackets). To enhance traceability and readability, the specific research techniques for each deliverable and phase are shown in *italics*.

Problem investigation [Problem awareness]

The design of a gamified, interactive tutorial is a practical design problem. The first engineering cycle, **EC1**, is a rational problem decomposition for the design of this IS artifact. **EC1** starts with a problem investigation. The associated tasks are the definition of the thesis motivations and goals, the research approach (covered in this chapter), and a review of the state of the art to position our research. The supportive research technique is desktop research and, in particular, *literature reviews* on 1) the landscape of Security Requirements Engineering, and 2) innovative teaching methods to underpin the need for such an artifact. These activities result in the first deliverable, which is presented in the Introduction and in **Sections 3.1** and **3.2**.

Similarly, the design of a quality framework for the conceptual modeling of security requirements is a theoretical problem with practical implications (**EC2**). We stress the applicability aspect of such an artifact (i.e., to create a useful framework that practitioners can regularly use when designing or evaluating these models). **EC2** also starts with a problem investigation, whose main tasks are 1) to generate a brief overview of the state of the art on the appraisal of conceptual model quality, and to gather experts' views on the aspects that are important when determining whether a security requirements model is of good quality. The supportive research techniques are a *literature review* and *survey research*, respectively. The results of this phase are captured in **Appendix IV**.

Treatment design [Suggestion and development]

After a deep understanding of the context has been sought, the second phase of both **EC1** and **EC2** comprises the design of their respective artifacts.

The design of the gamified tutorial requires an extensive review of the state of the art on gamification elements that can be used in learning contexts as well as the educational objectives that such a tutorial should tackle. To that end, several *literature reviews* are conducted on interactive tutorials (**Section 3.3**), gamification elements and best practices for gamifying learning experiences (**Section 3.4**), and educational objectives for conceptual modeling. Given the almost non-existent literature on the latter area, we supplement the search with ***semi-structured interviews with subject-matter experts*** to identify an initial list of learning outcomes. Drawing on this body of knowledge, a first version of the tutorial is then designed and constructed. The supportive research technique followed during the creation of the prototype is a simplified version of the ***goal-directed design process*** (Cooper, Reimann, Cronin, & Noessel, 2014). The results of this phase for **EC1** are presented in **Chapter 5**, in which a gamified interactive tutorial integrated within the STS-Tool is introduced: the STS-Tooltorial.

Despite the previous phase in **EC2** reveals that there have been multiple attempts to determine the quality of conceptual models, we decide to design a framework for the specific case of security requirements models. Before its design, a task within this phase is to motivate the need for it. As a result of this phase, the Quality of Security Requirements Models (QuaSMOD) framework is officially introduced in **Appendix IV**.

Treatment validation [Evaluation]

After the design of the STS-Tooltorial, a tentative empirical evaluation is conducted using students and IT professionals as subjects. Since the platform consists of several gamified elements, motivational affordances, and many types of media that work together to create a positive learning experience, it was difficult to craft a suitable comparison group. This fact, coupled with time constraints and the difficulty in finding a sufficient sample size to draw appropriate conclusions, led us to opt for a ***quasi-experiment design*** that favored pragmatism and qualitative analysis over scientific rigorousness. The empirical and inferential nature of the experiment necessitates of a research cycle (i.e., **R1**), which comprises the investigation of the research problem (where goals and hypotheses are stated), research scope and design (where the quasi-experiment is planned and the required instruments are constructed), execution of the evaluation, and analysis of results (including a defense and analysis of threats to validity) (Wieringa, 2014). **Chapters 6** and **7** present our empirical evaluation.

With regard to **EC2,** the QuaSMOD framework is weakly evaluated by theoretically justifying the validity of the approach. An additional complete engineering cycle is required to further refine and evaluate the artifact.

Research management [Conclusion]

Once **EC1** comes to an end, the STS-Tooltorial can be deployed for teaching aspects of security requirements engineering for socio-technical systems using the STS Method and its associated language. The effectiveness of the tool can be further evaluated in other academic contexts (e.g., different universities, longer research timeframe, refined versions) and implemented in organizational settings. These activities represent areas of future research. Lastly, upon completion of **EC2**, practitioners and students can use the QuaSMOD framework to structure discussions around the quality of security requirements models. The framework lends itself to a more thorough empirical validation (e.g., with students) as well as an evaluation of its validity and usefulness in practice.

Aside from the phases outlined in this section, the engineering and design cycles must also be managed. This includes activities such as project planning, selection of research problem, stakeholder communication, and ensuring that design problems and knowledge questions are formulated and tackled in a methodologically sound way (Wieringa, 2014). In this regard, the project planning is reflected in the Process-Deliverable Diagram (PDD) in **Appendix I**. Furthermore, for traceability purposes, Table 1 in the appendix breaks down and summarizes the relationship between the research questions, the research method, its associated tasks, and their estimated duration.

Lastly, it is worth mentioning that the original research idea was proposed by the first supervisor of this thesis, and jointly (re)formulated and executed with the help of the author and second supervisor. Finally, the communication activities are achieved by publicly disseminating our work with this (electronic) thesis document and publishing associated material to an international conference (**Appendix V**).

2.1 Semi-structured Interviews on Teaching Experiences, Learning Outcomes, and Quality Domains

The answers to research sub-questions SQ1 and SQ6 are based on a combination of survey research and literature review. While the introduction of **Chapter 3** elaborates on the literature review process, this section explicates the approach followed for conducting the semi-structured interviews.

Owing to the exploratory nature of this research, where little scientific work has been conducted hitherto, several interviews were conducted to get a deeper understanding of the research context. In particular, the following questions were sought:

- What teaching experiences, practices and heuristics can be extracted from teaching SRE and modeling?
- What are the ILOs for teaching security requirements modeling in socio-technical systems?
- What quality domains and aspects should the QuaSMOD have?

The identification of quality domains for our framework and the ILOs for the interactive tutorial (which are inextricably linked to teaching experiences) called for a research method that allowed domain experts to freely discuss these topics and the researcher to inquire follow-up in-depth questions. The need for spontaneous and non-rigid communication rendered the use of fully structured interviews impossible for this research (Galletta & Cross, 2013). Similarly, the need for extracting insights about the three areas outlined above was suggestive of the inappropriateness of employing unstructured questions. Therefore, a semi-structure interview approach was preferred over the other two options to allow for some degree of comparison between respondents and keep the interviews focused on the topics at hand (Corbetta, 2003).

Selection process

A combination of judgement and convenience sampling was used in the recruitment of key informants in the fields of SRE and socio-technical systems. To get more nuanced perspectives, special attention was put in recruiting participants from academia.

As far as academia is concerned, the selection of interviewees was driven by their research contributions to the field under study (i.e., judgement sampling). This includes, for instance, the original authors of the STS Method as well as other similar methods used for modeling security requirements (e.g., Secure Tropos, iStar).

Interviewees with industry links comprise early-adopters of the practice of socio-technical security requirements engineering, and were therefore selected based on a convenience sample provided by the first supervisor of this thesis.

A total of 16 participants were contacted, of which finally seven agreed to interview (i.e., a decent 43% response rate). Table 2 captures the list of participants in the order in which they were interviewed along with their company/university affiliation.

Table 2 - Anonymized list of SRE experts that were interviewed

Interviewee ID	Background	Years of Teaching Experience	Affiliation
1	Academia	6	University of Trento
2	Academia	12	University of Gothenburg
3	Academia	29	University of Bergen
4	Industry & Academia	10	Technical University of Munich
5	Academia	30	Carnegie Mellon University
6	Industry & Academia	>15	University of Brighton
7	Industry	10	International Business Machines

As can be seen, the sample was biased towards academia. This is not necessarily bad, as they were most helpful in providing teaching experiences to inform the creation of instructional material for the STS-Tooltorial (our main deliverable).

Interview structure

As stated before, the objective of the interview was threefold. Firstly, to gain a deeper understanding of the educational context and extract relevant lessons learned, teaching experiences on security requirements modeling were sought from those respondents that had an educational background. Secondly, we tapped on the interviewees' knowledge to identify learning outcomes for the creation of the tutorial. The third objective was to stimulate those experts to share what domains affected the quality of socio-technical security requirements models

With the aim of having a structured approach to the interview process, an interview protocol was created (see **Appendix II**). The protocol helped to ensure that interviews were consistent and the most relevant questions were posed in a proper way, while allowing sufficient room for discussion. Having an interview protocol is helpful when there are time constraints involved and thus research questions need to be prioritized (Harrell & Bradley, 2009), as was in this instance. The protocol consisted of the following sections: 1) introduction, 2) interviewee background, 3) unguided questions, 4) validation, and 5) conclusion. An overview of the sections is provided below.

The researcher, research aims, and interview structure were presented in the introduction. In the interviewee background section, warm-up questions were posed to get the participant comfortable and determine in greater detail her expertise and work context (i.e., industry or academia).

Subsequently, in the unguided section of the interview respondents were asked to share teaching experiences (if applicable), identify learning outcomes relevant for the tutorial, and pinpoint quality aspects for a framework that could be potentially used to appraise the quality of socio-technical security requirements models. In all cases there was ample time for discussion and clarification of concepts. Every time a new aspect was identified for any of the latter two areas, the respondent was probed to make a case for it (e.g., why the notion of *completeness* was important in a requirements model). Also, a record was kept for each of the areas to identify the number of times an experience or aspect was mentioned.

Triangulation and number of interviews

Originally, we thought that the interviews could also be used to validate both the list of quality domains and learning outcomes identified through literature and past interviews up until that moment (i.e., validation section). In the former case, the list of quality domains could have been introduced and respondents asked to comment on their completeness, correctness, and whether (and how) they could be used for gauging the quality of models. Similarly, respondents could have been inquired about the correctness and completeness of the list of learning outcomes. Observe that this is a form of data triangulation that strengthens the findings empirically derived from the interviews (Jick, 1979). However, after the pilot and second real interview, we decided to discard this section, since it was too difficult for the domain experts to comment on these aspects. For example, not all of the participants were acquainted with the STS Method (i.e., too specific questions), nor had sufficient experience in conceptual modeling.

Finally, the conclusion part offered the opportunity to wrap up the interview and discuss any open issues.

A total of seven interviews were conducted over a period of three weeks. Each lasted between 45 and 60 minutes, depending on the background of the participants (e.g., respondents without teaching experience were not inquired about their lessons learned). The number of interviews was a priori determined based on practical considerations (e.g., time constraints, available sampling frame), given that this was but a step in a series of research activities.

II. Background

After having introduced the research problem and the objectives of this thesis, this block presents the theoretical and conceptual frameworks that underpin our research. In the third chapter, we stand on the shoulders of past and recent giants to get a clearer overview of the research fields under study and thus expand our knowledge of the world. Relying on this newly-acquired knowledge, in the fourth chapter we leap into the unknown to see further with the help of our conceptual framework, a guiding light used to illuminate the way.

3. Theoretical Framework

This chapter sets out to offer an overview of the current state of the art in the fields of inquiry of this thesis, to wit: SRE, teaching methods, interactive tutorials, and gamification (of learning). In finding relevant literature, a flexible approach was preferred over a highly-structured and systematic method. This was regarded as appropriate because the literature review was used to 1) get an overview of the fields, and 2) inform the design of the IS artifacts (i.e., STS-Tooltorial and QuaSMOD framework).

In particular, we opted for the approach to literature review in IS suggested by Webster and Watson (2002), which is based on snowballing. Specifically, both backward (i.e. following reference lists) and forward (i.e. papers that cited a relevant paper) procedures were followed. The choice stems from the fact that a sufficient number of relevant papers were provided by the thesis supervisor, and that the review included fairly general terms that would have generated much more noise if a systematic database-search approach had been followed (Jalali & Wohlin, 2012). Moreover, given that this study explores several large academic disciplines, flexibility was deemed as an aspect to consider. Further, the choice is supported by the fact that actual conclusions drawn from literature are not highly dependent on whether database searches or snowballing is adopted (Jalali & Wohlin, 2012, p. 36).

The snowballing technique was continued until we deemed that the collected literature was sufficiently extensive (i.e., saturation was reached) (Eisenhardt, 1989). Evidence of this was obtained when similar viewpoints kept appearing (e.g., same proposal of game elements across sources) and no new insights were obtained.

To avoid bias, the inclusion and exclusion criteria were defined before conducting the review. As inclusion criteria, we opted for peer-reviewed, scientific publications considered to be of interest with regard to the research objectives. This criterion was relaxed when academic literature was found to be lacking and thus an internet search for grey literature was also necessary. Conversely, the exclusion criterion was set to “studies that present guidelines or best practices when these have not been empirically verified”. The reason behind this criterion was that, insofar as possible, the design of the STS-Tooltorial had to be designed building upon effective and empirical guidelines.

3.1 Review on Security Requirements Engineering (SRE)

This section presents an overview of the different approaches to SRE that can be found in academic literature. Rather than giving specific details about each of the approaches, the objective of this chapter is to briefly survey the landscape of Security Requirements Engineering and position the underlying topic of the main deliverable of this thesis. The chapter ends with an in-depth overview of the selected approach for the purposes of this research, i.e., the STS Method.

To better explain and situate the approaches, we introduce a fictitious running example that is used throughout this chapter to create models:

NovoX is the intelligence service of the planet Askoni. Despite being situated at the periphery of the galaxy (more precisely, at the Rolmson province), Askoni is one of the most vital planets for the Galactic Empire. In particular, NovoX is responsible for galactic monitoring, collection, and processing of information and data for planetary intelligence and rebel counterintelligence purposes.

Although the Empire appears stable and powerful, it is currently facing many economic, social, and political conflicts that threaten its hegemony. The root of all problems is NovaVisione, an until now dormant organization that, for decades, has been slowly deteriorating the influence of the Empire. The Empire has a vast web of informers that help to counteract the actions of NovaVisione by feeding information to NovoX. The analysts at NovoX are responsible for generating actionable intelligence to aid viceroys and members of the Galactic Council in making strategic decisions.

3.1.1 SQUARE Method

SQUARE consists of a 9-step process for eliciting, categorizing, and prioritizing security requirements from the early stages of the development life cycle (Mead et al., 2005). Rather than providing specific tools or technique, the method attempts to provide generic guidance on the steps necessary to create secure systems and applications. A brief overview of the steps is provided below:

1. Agree on key security definitions. The objective is to create a document of agreed definitions that stakeholders can consult during the SQAURE process.
2. Identify security goals. The objective is to generate an overarching business goal for the project at hand and several security goals that will later on guide the prioritization of security requirements. Goals can be elicited by asking

questions such as: “what assets stakeholder X wants to protect?”, “what (possibly conflicting) goals have the different stakeholders?”.

3. Develop artifacts to support security requirements definition. Such artifacts may be misuse cases, abuses cases, Secure Tropos models, anti-goal models, STS-models, etc.
4. Perform risk assessment. The objective is to identify the threats and vulnerabilities that a system may face (along with their likelihood and impact), so that appropriate responses can be taken (e.g., in the form of security requirements)
5. Select elicitation techniques that are suitable for the client organization and project at hand.
6. Elicit security requirements. The objective is to generate a first security requirements document using the elicitation techniques selected in step 5.
7. Categorize requirements. The objective is to categorize requirements so that their subsequent prioritization becomes easier. SQUARE recommends classifying requirements as essential, non-essential, system level, software level, or as architectural constraints, although this is not fixed.
8. Prioritize requirements. The objective is to have an ordered list of requirements prioritized by some criteria (e.g., cost-value, importance, satisfaction, risk).
9. Inspect requirements. The final step concerns the validation of requirements to identify defects before delivering the final security requirements document.

SQUARE has been used both for teaching purposes (Mead, Shoemaker, & Ingalsbe, 2009) as well as in industry (Mead et al., 2005).

3.1.2 CORAS Method

CORAS is a model-driven method for analyzing security risks (Braber, Hogganvik, Lund, Stølen, & Vraalsen, 2007). It consists of seven steps:

1. Introductory meeting between analysts and the client to ascertain the overarching goals of the latter.
2. Meetings with various client representatives so that analysts can present their understanding and clarify insights from what has been learned so far based on the first meeting and desk research.

3. Involved a more precise description of the target to be analyzed, including assumptions being made.
4. An expert workshop is organized to identify unwanted incidents, threats, vulnerabilities, and threat scenarios.
5. An expert workshop is organized to estimate probability and impact values for each of the unwanted incidents identified in the previous step.
6. Presentation of the first risk picture to the client, where corrections will probably be made based on feedback.
7. Treatment identification session. Treatments are assessed with regard to cost and benefit before a definite plan is created.

Throughout steps 2 to 7 the (UML-based) CORAS security risk modeling language is used as the main technique for communication and interaction purposes.

3.1.3 UML Extensions

This subsection reviews two of the most popular UML extensions to security requirements: UMLsec and SecureUML.

UMLsec

UMLsec is an extension of UML proposed by Jürjens (2002) for modeling security-related aspects. These aspects are analyzed by modeling the behavior of an adversary and modeling security information using UML stereotypes. The default profile offered by UMLsec includes security properties such as confidentiality, secure information flow, non-repudiation, and fair exchange, among others.

To illustrate UMLsec, Figure 1 and 2 show a use case and activity diagrams capturing the interactions between the Information Unit at NovoX and a Secret informer. Fair exchange (i.e., information for money) is ensured by adding the *<<fair exchange>>* stereotype to the system. This prevents both parties from cheating. The activity diagram also contains the tags {start} and {stop} to enforce a constraint. Specifically, after a {start} state (i.e., giving information) is reached, eventually one of the {stop} states will also be reached (i.e., thank the informer or get the money back). This means that NovoX is guaranteed to receive its money back when the information provided by an informer is not of good quality.

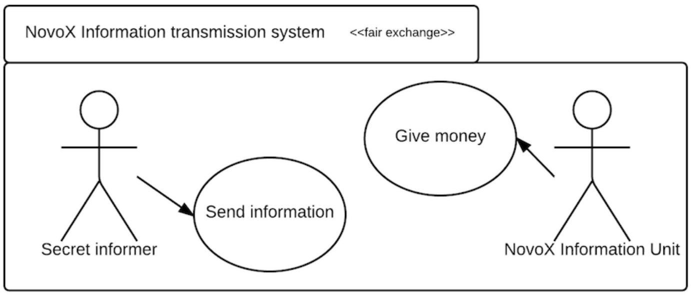


Figure 1 - Use cases with UMLsec

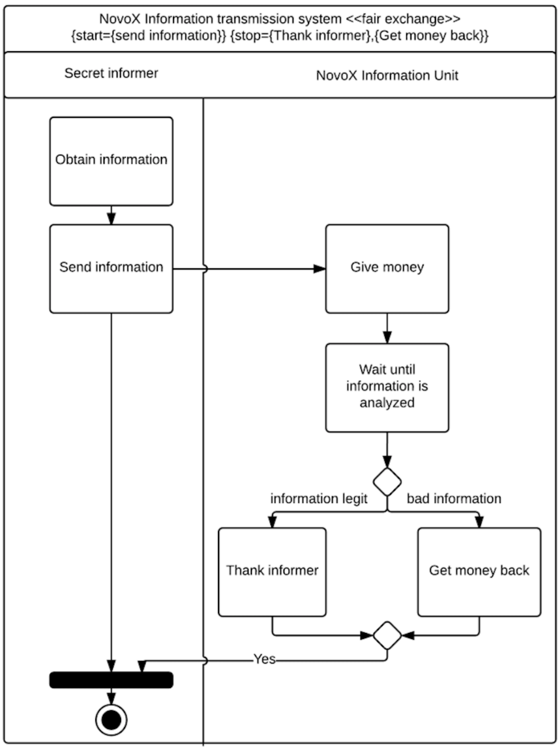


Figure 2 - Activity diagrams with UMLsec

SecureUML

Lodderstedt, Basin, and Doser (2002) introduced SecureUML, a modeling language based on UML that allows for the modeling of role-based access control policies and the subsequent integration into a model-driven software development process. SecureUML models are helpful for devising and enforcing control mechanisms once a technical design of the system has been suggested. Similar to UMLsec, SecureUML uses the concept of stereotypes to define its basic syntactic elements, which are as follows:

- **User**: refers to any person that interacts with the system
- **Role**: a function within an organization or system that describes the responsibilities and permissions assigned to users associated with that role.

- **Permission**: association class between a role and a resource that describes the operations that the role can execute upon the resource.
- **Authorization constraints**: expresses a precondition imposed on every call to an operation of a particular resource. In other words, they indicate the conditions that must be met before permission is granted. Constraints are expressed using OCL.

The use of SecureUML is illustrated in Figure 3. In the example, there are two roles (Imperial Viceroy and Galactic Council) and one resource (Intelligence report). The Intelligence report has two operations that allow the retrieval of its contents in two ways: a sanitized (i.e., blacked out) version and access to the raw document. The following permissions are granted to the roles:

- NovoX enforces a need-to-know policy whereby Imperial Viceroys (such as Viceroy McKinsey) can retrieve the raw report only when the intelligence contained in it affects their planet. In all other cases, they only have access to a sanitized version of the document. This is reflected in the viceroyRetrieval permission and its associated OCL constraint.
- Conversely, members of the Galactic Council (such as Lord Ferd) can always access the raw reports, as they need as much information as possible to maintain the status quo. This is reflected in the councilRetrieval permission.

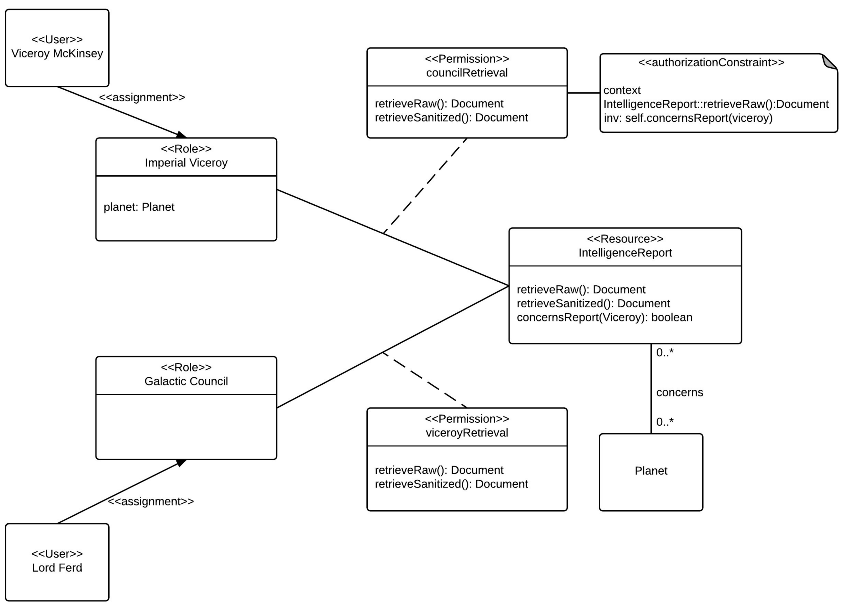


Figure 3 - Defining policies with SecureUML

3.1.4 Use Case Extensions

A use case is a list of steps that defines the interactions between an actor (e.g., a user) and the system under consideration to accomplish a goal (Jacobson, 1992). Use cases can be represented using textual, structural, and visual modeling techniques (e.g., use case diagrams in UML) (Jacobson, 1992). Since the latter is of interest to this thesis, from now the term use case is used to refer to its diagrammatic representation. In the context of security requirements, three adaptations of use cases exist: abuse cases, misuse cases, and security cases. We review them in turn.

Abuse cases

McDermott and Fox (1999) adapted UML use case models to analyze security requirements through the introduction of abuse case models. An abuse case represents a "complete interaction between a system and one or more actors, where the results of the interaction are harmful to the system, one of the actors, or one of the stakeholders in the system" (p. 59). Abuse cases do not incorporate new notational elements. To avoid confusion, they are kept separate from regular use cases.

Figure 4 shows an abuse case where the actor NovoX may compromise the decision-making abilities and security of the Galactic Empire by disseminating wrong intelligence reports and disclosing confidential information to either members of NovaVisione or unauthorized public servants (e.g., viceroys without a need-to-know). Similarly, NovaVisione may threaten the security of the socio-technical system by bribing intelligence officers at NovoX.

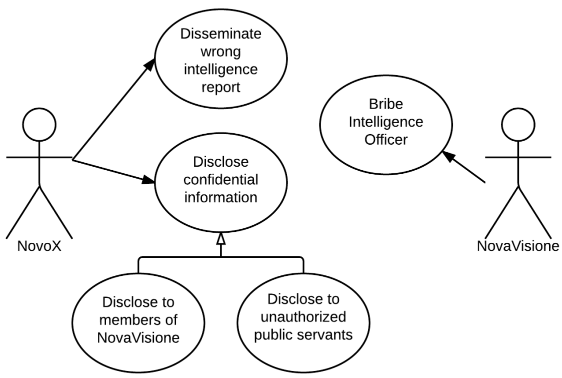


Figure 4 - Abuse cases

Misuse cases

Similarly, Sindre and Opdahl (2005) inverted the notion of use cases to create misuse cases. Misuse cases describe functionalities or situations that a system should prevent from occurring, as they pose a threat to the system. Misuse cases are represented with a black rectangle to distinguish them from regular use cases, and are initiated by (possibly inadvertent) misusers. The diagrams originally introduced two new relations: the *prevents* relation indicates that a misuse case prohibits the execution of the related use case; the *detects* relation specifies that a use case can be used to discover (and possibly prevent) the execution of a misuse case (Sindre & Opdahl, 2005). Alexander (2002) further included relations such as: *mitigates,* which indicates that a use case can mitigate the success of a misuse case; *threatens*, which specifies that a misuse case can threaten a use case.

Figure 5 exemplifies the use of misuse cases in the context of our running example. NovoX's duties include analyzing the quality of the information sent by Secret informers, paying them for it, and disseminating intelligence reports to interested parties (e.g., viceroys). Three misusers are identified: NovoX's information analyst, Secret informer, and NovaVisione.

Information analysts have the misuse case of conspiring with informers to split the money received for illegitimate intelligence, which threatens NovoX's use case of analyzing information.

In turn, a Secret informer can transmit information of bad quality for the sole purpose of instantly receiving illegitimate money in return. In this situation, NovoX can temporarily withhold the money transfer to mitigate this attack, provided that the Secret informer is identified as an untrustworthy source (e.g., a new informer). Finally, NovaVisione always tries to eavesdrop information originating from key planets like Askoni in an attempt to thwart the influence of the Empire. To mitigate this attack, NovoX can employ a quantum key distribution algorithm to create a secure communication channel.

Observe that once misuse cases are identified, new use cases (e.g., security requirements) can be derived to mitigate them. Examples in Figure 5 are the quantum key distribution algorithm and the temporary withholding of money.

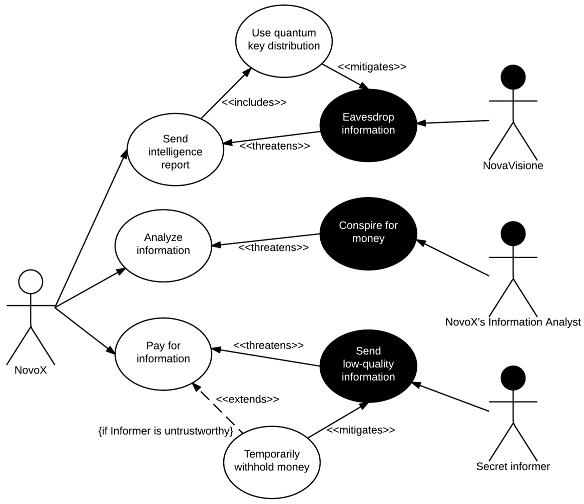


Figure 5 - Misuse cases

Security use cases

Security use cases can be used in conjunction with misuse cases to analyze and specify security requirements (Firesmith, 2003). The objective is to analyze the assets and services that need protection, the threats to which these are vulnerable, and the required security requirements and mechanisms needed to protect such assets and services.

Figure 6 uses two of the uses cases presented in Figure 5 and extends them with security use cases (situated in the middle between use and misuse cases). To securely send intelligence reports, the diagram includes security uses cases to ensure privacy and integrity of both data and communications. The diagram also has a security use case to ensure non-repudiation of analysis results, with the aim of deterring NovoX's information analysts from colluding with the enemy. The resulting three security use cases specify requirements that protect NovoX from two security threats involving attacks by either NovoX's employees or NovaVisione.

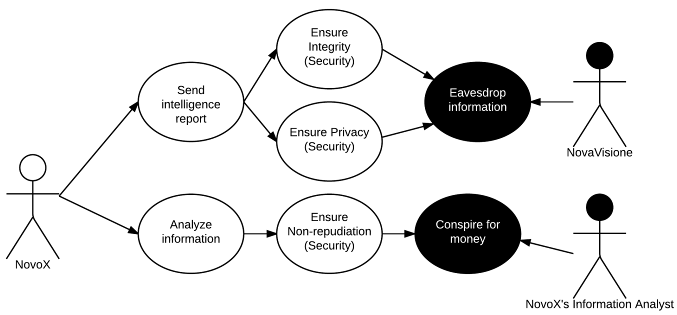


Figure 6 - Security use cases

The approaches reviewed so far are rooted on fairly standardized RE methods and techniques, and assume a system-oriented standpoint. In other words, they model the behaviors that a system should adopt, or the control mechanisms that such a system should afford. However, they fail to support the modeling and analysis of security considerations at an organizational (or, more generally, socio-technical) level.

Early security requirements is also associated with the creation of agent- or goal-oriented security requirements models. These models heighten the discussion of security by ascribing intentionality to the actions performed by social actors. Methods and techniques that fall within this category are anti-goals, Secure Tropos, Secure i*, and STS (presented at the end of this chapter). The approaches propose to analyze security from a broad socio-technical systems perspective.

3.1.5 Anti-goals (KAOS Modeling)

Lamsweerde (2004) proposed a goal-oriented framework based on KAOS (Dardenne, Lamsweerde, & Fickas, 1993) for generating and resolving obstacles (i.e., anti-goals) to security goal satisfaction. Security requirements can be generated after following three steps: 1) specification of patterns for security goals, 2) derivation of anti-models (consisting of obstacles, also known as anti-goals) that threaten such specification, and 3) derivation of alternative countermeasures to such threats and definition of new requirements by selecting the alternatives that best meet quality requirements.

In step 2, anti-goals are refined to form a threat tree, whose leaf nodes represent vulnerabilities or anti-requirements. As an example, Figure 7 contains an anti-goal model consisting of a hierarchy of goals.

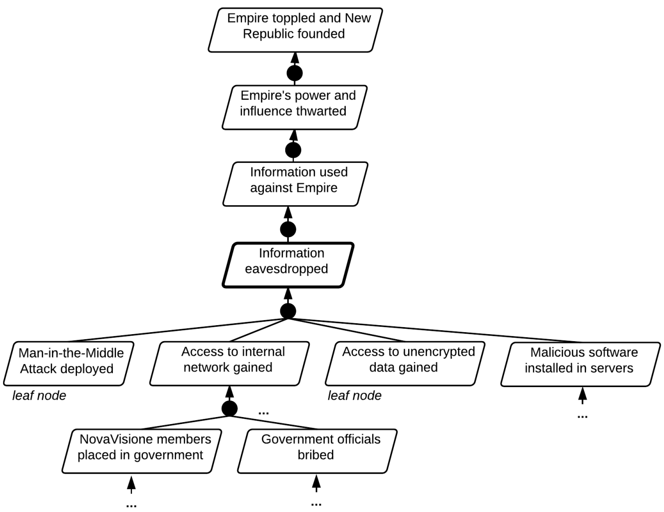


Figure 7 - Anti-goal models

The anti-goal Information Eavesdropped is expanded into parent goals by asking "why?" questions, and expanded into sub-goals by asking "how?" questions. The model shows that the motive behind NovaVisione's desire to eavesdrop is to topple the Empire, along with mechanisms through which it may do so. Further analysis would entail the creation of countermeasures against the leaf nodes (i.e., Man-in-the-middle attack and Access to unencrypted data gained), and expansion of non-leaf nodes.

Therefore, anti-goal models are helpful in capturing the rationale behind an attack, and the ways attackers might compromise a system.

3.1.6 Secure Tropos

Tropos is an agent-oriented software development method that incorporates many of the concepts of the i* modeling framework (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004). Secure Tropos is an extension of Tropos to the field of security requirements. However, there currently exist two approaches that go by the name of Secure Tropos: the work by Giorgini, Massaci, Mylopoulos, and Zannone, (2004, 2005), and the one by Mouratidis and Giorgini (2007).

The Secure Tropos by Giorgini et al. (2004) allows requirements engineers to capture trust and security requirements. It introduces the concepts of ownership, trust, and delegation within a normal functional requirements model. From that, security and trust requirements can be derived. Their work is a seminal paper that states the importance of considering security (and trust) concerns early in the development process at an organizational level. Moreover, it offers a model-driven approach for considering security and trust in the context of functional requirements. The refined version of their Secure Tropos (Giorgini et al., 2005) supports the necessity of delegating services and permissions for pragmatic reasons even when trust relationships do not exist. In other words, they acknowledge that a system can still be secure when accountability is enforced through monitoring. Moreover, the extended framework makes a distinction between trust and delegations of permissions, and trust and delegations of execution.

An example of Secure Tropos by Giorgini et al. (2004) is depicted in Figure 8. The Galactic Council is the legislative body of the Empire whose goals are to preserve peace, negotiate alliances, and obtain intelligence reports for decision-making purposes. Being burdened with such daunting objectives, the Council is not capable of performing the latter goal, so its execution is trusted to NovoX (*Te*, *De*). NovoX, in turn, (*P*) provides the service of generating intelligence reports. Similarly, NovoX delegates and trusts the activity of analyzing information to its Information Analysts, which is one of the tasks associated with the creation of intelligence reports.

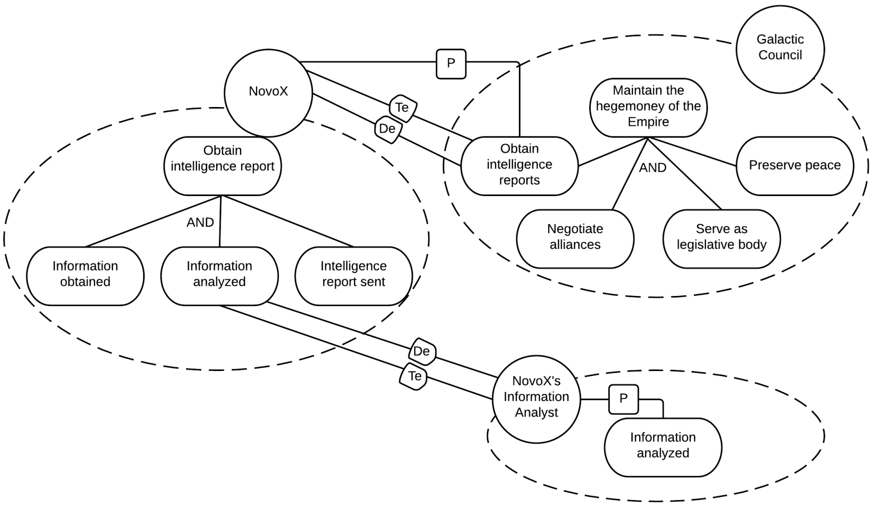


Figure 8 - Secure Tropos (also known as SI*) by Giorgini et al. (2004, 2005)

The second proposal of Secure Tropos was proposed by Mouratidis and Giorgini (2007). The authors extended the Tropos method to model security concerns. Their Secure Tropos introduces the following security concepts:

- Security constraint, defined as a "restriction related to security issues, which can influence the analysis and design of the system under development" (p. 8)
- Secure dependency, which describes one or many security constraints that ought to be fulfilled for a dependency to be satisfied.
- Secure entity, used to represent a secure goal, task, or resource.

The modeling aspects of Secure Tropos by Mouratidis and Giorgini (2007) are exemplified in Figure 9. The figure features two types of models borrowed from i* and Tropos, namely: the strategic dependency diagram, which captures the interactions between actors of the system; and the strategic rationale diagram, in which the internal rationales of agents are made explicit.

The former diagram shows that Viceroys depend on NovoX for obtaining intelligence reports, and that NovoX depends on trusted Secret informers for gathering information. It also features three (security) constraints: 1) viceroys can depend on NovoX for obtaining intelligence reports only if the intelligence concerns a planet under their ruling, 2) secret informers request NovoX that their identity be kept anonymous when providing information, and 3) when fulfilling the goal of obtaining information, secret informers must provide information of high quality.

The diagram at the bottom depicts the intentions of actor NovoX. The goal Obtain intelligence report is AND-decomposed into three regular goals, concerned with the intelligence cycle (i.e., obtain, analyze, and disseminate), and one security goal with a positive contribution towards achieving the security constraint Informer's identity anonymized.

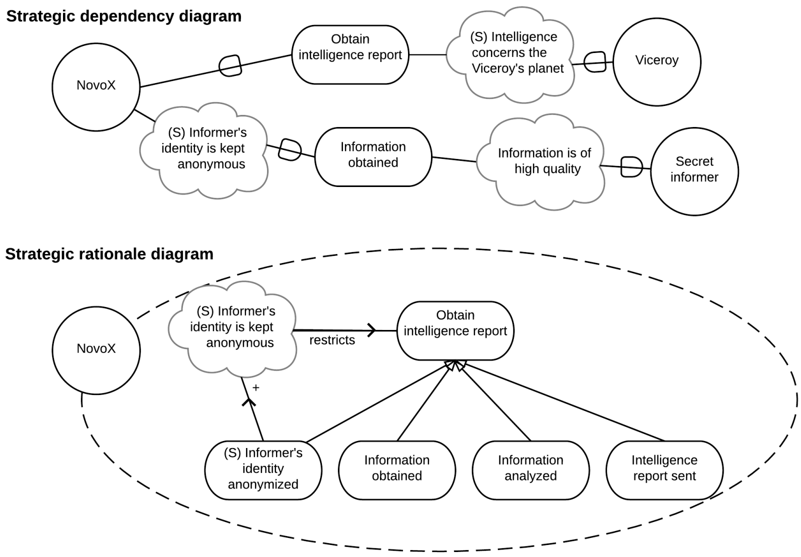


Figure 9 - Secure Tropos by Mouratidis and Giorgini (2007)

3.1.7 Secure i*

Secure i* (Liu, Yu, & Mylopoulos, 2003) is an extension of Yu's i* modeling framework for modeling an analyzing security tradeoffs. The framework focuses on the social interactions among actors and the systems that act on their behalf. It introduces security and privacy goals, and uses dependency analysis to validate if the system is secure. Secure i* also focuses on the alignment of security requirements with other requirements, as the authors acknowledge that there may be situations in which requirements conflict with each other. The framework comprises of the following activities:

- Actor identification: the objective is to identify all the actors involved in the system.
- Goal/task identification: for each actor, that which s/he wants to achieve is identified.

- Dependency identification: the objective is to identify the relationships between actors.
- Attacker analysis: each identified actor so far is treated as an attacker, and their (possible) malicious intents are examined.
- Dependency vulnerability analysis: seeks to identify those dependency relationships (created in Step 3) that are vulnerable to attacks, and the effects that such a compromise may entail up in the dependency chain.
- Countermeasure analysis: solutions are formulated for tackling the identified vulnerabilities based on the attackers' capacities.

The first three activities refer to the requirements analysis process already supported by i*. Also, note that the activities (and models) can be iteratively refined.

3.1.8 STRIDE

STRIDE is a threat classification scheme pioneered by Microsoft (Microsoft, 2005). The threat model allows for the discussion of potential threats in a structured way. STRIDE is an acronym that stands for the six types of threats shown in the left column of Table 3.

For each of these threats, Hernan, Lambert, Ostwald, and Shostack (2006) identify a series of security services that go beyond the widely-recognized CIA triad (i.e., confidentiality, integrity, and availability) (Whitman & Mattord, 2007). These are shown in the right column of the table.

Table 3 - Mapping of threats to security properties that guard against them. Adapted from Hernan et al. (2006)

Threat	Security properties
Spoofing identity	Authentication
Tampering with data	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Daniel of service	Availability
Elevation of privileges	Authorization

STRIDE is used during the threat identification activity within Microsoft's threat modeling process (2007), which consists of the following activities (Microsoft, 2007) :

- Identify security objectives
- Survey the application
- Decompose it
- Identify threats
- Identify vulnerabilities

3.1.9 Security Patterns

In the domain of software engineering, a pattern refers to a general reusable solution to a commonly occurring problem within a given context. Patters are used in areas such as software architecture (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996; Buschmann, Henney, Schmidt, 2007), user interface and interaction design (Cooper et al., 2014), game design (Bjork & Holopainen, 2004), and security (Schumacher, Fernandez, Hybertson, & Buschmann, 2005).

Patterns are usually captured in a catalog, to which designers can resort when looking to satisfy a specific (security) goal. Based on their level of abstraction, patterns can be divided into architectural and design patterns. As with any kinds of patterns, security patterns become especially helpful when designing the technical aspects of a secure system, as they serve as best practice solutions to typical security problems (Schumacher et al., 2005).

3.1.10 Extensions of Problem Frames

Problem frames are a means to describe software development problems invented by Jackson (2001). They are defined as "a kind of pattern that defines an intuitively identifiable problem class in terms of its context and the characteristics of its domains, interfaces and requirement" (p. 76). Several authors have suggested using (extensions of) problem frames for dealing with security aspects. In this subsection we review two of the most salient approaches: abuse frames and SEPP.

Abuse frames (Lin, Nuseibeh, Ince, Jackson, & Moffett, 2003) are an adaptation of the problem frames approach by Jackson for the early analysis of security threats and security requirements derivation. They introduce the notation of anti-requirement, defined as the intention of a malicious user that subverts an existing requirement. Security threats are modeled as abuse frames, which consist of the following elements:

- A machine, which contains the vulnerabilities that a malicious user exploits to attack the system.
- A victim, which represents the assets under attack.
- A malicious user and an anti-requirement, which together constitute the threat agent.

Figure 10 depicts a threat described by a generic abuse frame diagram (left) and an instantiation of it within the context of our running example (right). The example shows a situation in which a Viceroy illegitimately requests the retrieval of the entire contents of an intelligence report that does not concern him, thereby breaking the need-to-know principle. As a result of this request, the Intelligence Management System retrieves the report and exposes its content to the Viceroy, who may use it for personal purposes (e.g., blackmailing a member of the Galactic Council).

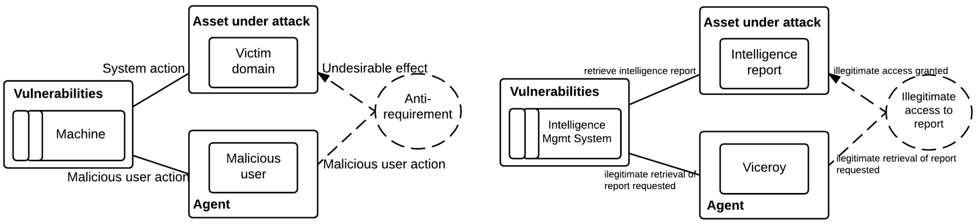


Figure 10 - Abuse frames

An interesting combination of patterns and problem frames can be found in the work of Hatebur, Heisel, and Schmidt (2007), who propose a security engineering process based on patterns. In particular, the central idea is that security problem frames are stored in a catalog and retrieved based on the security requirements of the system at hand. These abstract problem frames are in turn associated with concretized solution approaches.

3.1.11 Attack Trees

Attack trees are a systematic method for identifying ways in which an asset may be attacked (Schneier, 1999). The models bear close resemblance to the goal-oriented methods reviewed in this chapter, as attacks are also refined in AND/OR decompositions (Moore, Ellison, & Linger, 2001). Here, however, the focus is on elaborating the possible ways in which an attack may unfold.

Figure 11 shows an example of an attack tree. The tree starts from a root, which represents an event that, if occurs, can have negative consequences. In this case, such event refers to the theft of secrets by NovaVisione. The covert organization can do so by either (i.e., OR decomposition) intercepting communications, accessing the servers at NovoX, or recruiting the help of insiders. Some possible intrusion scenarios may be {Find disposal locations, Inspect dumpster, Monitor InterPlaNet communications for leakage}, or {Blackmail government official}, among many others.

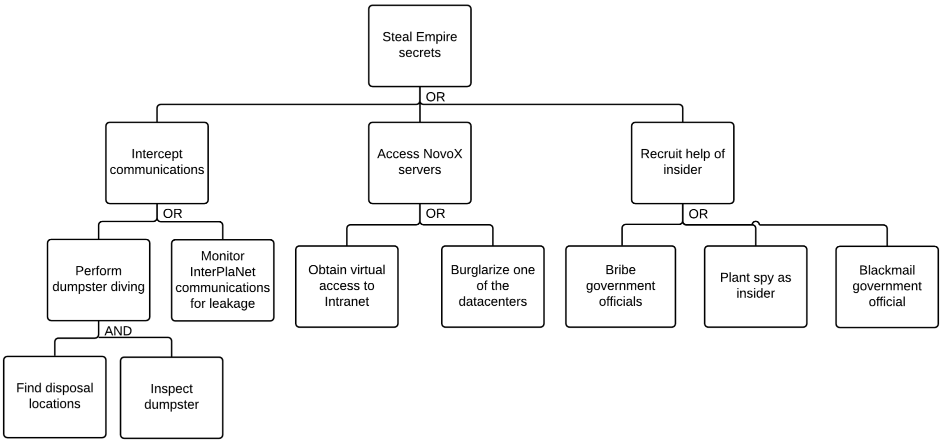


Figure 11 - Attack trees

3.1.12 Extensions of BPMN

There have been several attempts to extend BPMN (Omg.org, 2015), the de-facto standard for modeling business processes, with the intent of including security concerns. This subsection reviews some of these extensions.

In 2007, Rodríguez, Fernández-Medina, and Piattini proposed the incorporation of security requirements into business process diagrams from the perspective of a business analyst. The authors propose a notation based on graphical concepts to represent security semantics. Their meta-model includes five types of requirements: non-repudiation, attack harm detention, integrity, privacy, and access control.

Similarly, SecureBPMN is an approach proposed by SAP that extends the notation of BPMN with a security language that allows for the specification of security aspects such as role-based access control, separation of duties, binding of duty, and need-to-know (Brucker, Hang, Lückemeyer, & Ruparel, 2012).

SecBPMN is a framework that allows security engineers to define business processes with security concepts and procedural security policies (Salnitri, Dalpiaz, & Giorgini, forthcoming). It consists of two main components: the SecBPMN-ml modeling language and the SecBPMN-Q query language. While the former allows for the modeling of business processes with security concepts, the latter is concerned with procedural security policies (Salnitri, Dalpiaz, & Giorgini, 2014). The SecBPMN language includes eight security annotations, to wit: accountability, auditability, authenticity, availability, confidentiality, integrity, non-repudiation, and privacy. The new version of the framework, SecBPMN2 (Salnitri, Paja, & Giorgini, 2015), includes three new security annotations (i.e., separation of duties, bind of duties, and non-delegation), and three new relations.

As an example, Figure 12 features the expressive capabilities of SecBPMN. The diagram depicts one of the possible interactions between a Secret informer and NovoX. The task of a Secret informer is to leverage his network to obtain information and send it to NovoX. When such information is sent, the informer is paid provided that he does not qualify as an untrusted source (in which case, the money is temporarily withhold). Then, analysts at NovoX analyze the information and, if they regard it as high-quality, produce actionable intelligence in the form of a report. Otherwise, the informer is flagged as untrustworthy and the money for this transaction is permanently withhold. Please refer to the BPMN specification (Omg.org, 2015) for details about the notational elements of the language employed in the diagram.

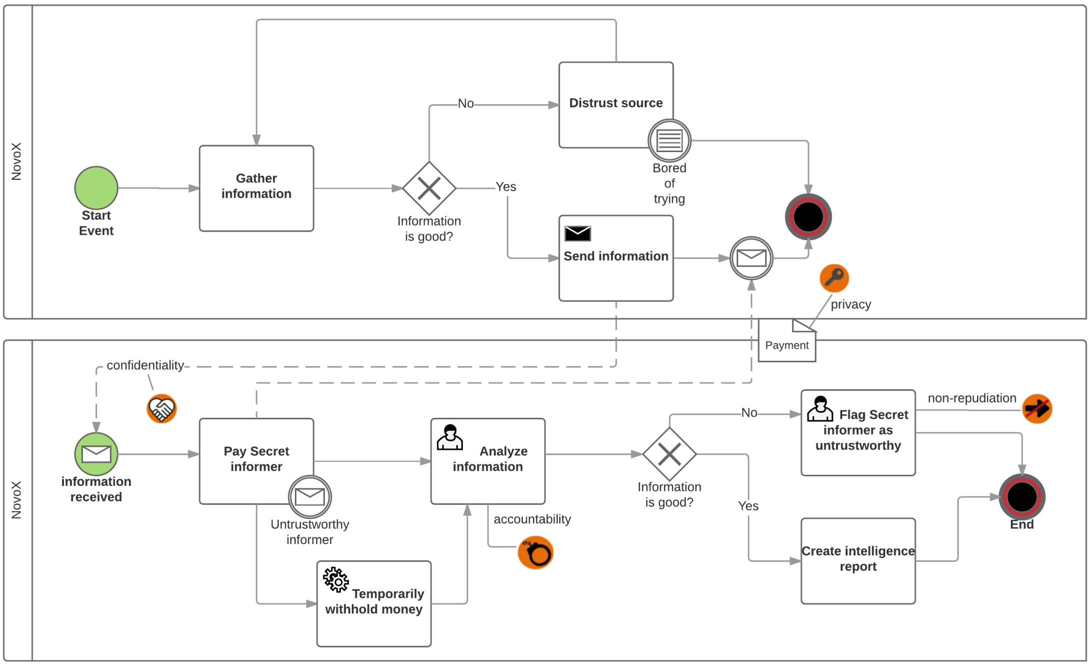


Figure 12 - Defining secure business processes with SecBPMN

Of relevance in this diagram are the security concepts modeled with SecBPMN-ml. The language uses annotations to specify such concepts. The diagram exemplifies three of them: 1) confidentiality of the message flow, indicating that information should not be disclosed to unauthorized parties (e.g., NovaVisione), 2) accountability of the analysis conducted by NovoX's information analysts (to avoid collusion), and 3) privacy of the payment document with regard to to the identity of the informer.

Note that a deadlock situation may occur when the Secret informer keeps waiting from the money that will never arrive because of his submission of low-quality information. In spite of this, the example serves to illustrate the expressiveness of the SecBPMN-ml when analyzing security in detailed processes.

3.1.13 Other Approaches

This subsection contains a collection of approaches that deal with security requirements in the early stages of systems development, but either do not offer or elaborate ways through which security aspects may be modeled. Compared to SQUARE or CORAS, these approaches are less established and thus not treated in separate sections. They are nonetheless included here for completeness purposes.

Risk-Based Security Requirements Engineering Framework

Mayer, Rifaut, and Dubois (2005) propose a risk-based Security Requirements Engineering framework based on four key iterative steps:

- Analysis of context and identification of assets. IT assets are modeled using architectural modeling techniques, and business goals are modeled using the goal-oriented language i*.
- Determination of security goals associated with those assets.
- Security requirements elicitation using i* models.
- Selection of countermeasures (i.e., security solutions) suited to security requirements.

These steps are underpinned by an independent risk analysis activity. The greatest contribution of the framework is that it helps in structuring SRE activities. However, as this chapter has shown, there are more suitable extensions of i* for modeling security requirements, with more expressiveness than what i* offers.

Security policies and requirements using GBRAM

Anton and Earp (2000) propose to use the goal-based requirements analysis method (GRAM) (Anton, 1996) to formulate security and privacy policies for e-commerce systems, and operationalize those into system requirements. The steps (represented with ovals) and artifacts/information sources (denoted by rectangles) are described in

Figure 13. Their approach assumes that goals are specified in natural language, and complemented with scenarios.

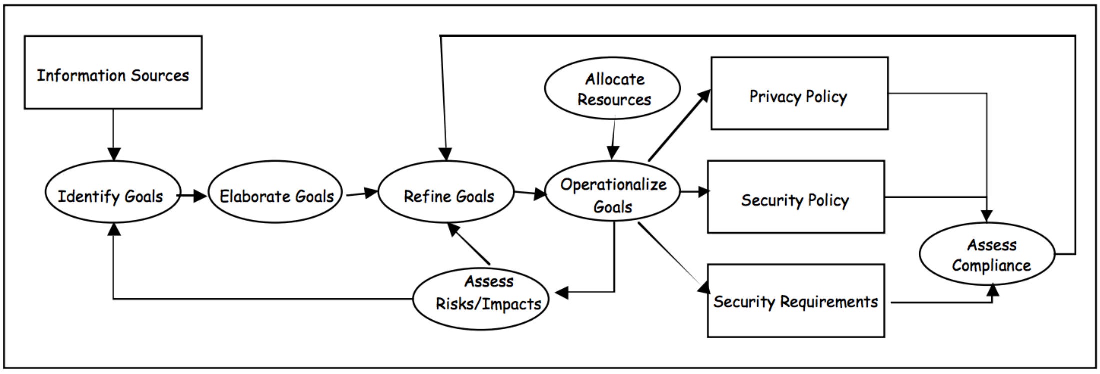


Figure 13 - Overview of GBRAM. Extracted from Anton and Earp (2000)

Security Requirements Engineering Framework (SREF)

Based on their previous work on core security requirements artifacts (Moffett, Haley, & Nuseibeh, 2004), Haley, Laney, Moffett, and Nuseibeh (2008) present a framework for requirements elicitation and analysis based on four iterative activities:

- Identify functional requirements, so that a representation of the system context is created.
- Identify security goals. This is achieved by identifying candidate assets, selecting management principles (e.g., separation of duties, least privilege), and conducting a threat analysis.
- Identify security requirements. The authors equate security requirements to constraints imposed on functional requirements. Therefore, such requirements are stated in text inside a functional requirement. Problem diagrams (similar to problem frames) are used to
- Construct and evaluate satisfaction arguments, to verify that security requirements are satisfied by the system. Satisfaction arguments consist of a formal logic outer argument (e.g., predicate, temporal, or propositional logics), and an inner language expressed in structured natural language.

Since the authors equate security requirements with constraints, their approach falls short and is more constrained than methods such as Secure Tropos, STS, or SI*.

Security Requirements Engineering Process (SREP)

SERP is an asset-based, risk-driven, and reuse-based method for the establishment of security requirements in the development of information systems (Mellado, Fernández-Medina, & Piattini, 2006, 2007). It combines the activities of SQUARE and the reuse-based approach to determining security requirements by Sindre, Firesmith, and Opdahl (2003), while also integrating the Common Criteria standard into the development process. Note that Common Criteria is an ISO standard that provides objective assurance that the specification, implementation, and evaluation activities related to a secure software product have been performed in a rigorous way. The depth and rigor of the assurance activities can vary depending on the desired confidence level (ISO/IEC 15408:2009).

The authors propose the creation of a security resources repository to support the reuse of security requirements (expressed in UMLsec, security use cases, or plain text), assets, threats (expressed as misuse cases or attack trees), and countermeasures.

CLASP

The Comprehensive, Lightweight Application Security Process (CLASP) is a set of process consisting of 24 activities and 7 best practices designed to help development teams to consider security aspects early in the development cycle (OWASP, 2016).

On a high level, the best practices identified by CLASP are the following: institute awareness programs, perform application assessments, capture security requirements, implement secure development practices, build vulnerability remediation procedures, define and monitor metrics, and publish operational security guidelines. Taken together, the best practices cover the entire software life cycle.

For each best practice, several role-based activities are defined. For instance, the third practice (i.e., capture security requirements), comprises of activities such as: identifying a global security policy, identifying resources and trust boundaries, creating misuse cases.

OWASP is better used once the technical system-to-be is being designed or developed. Therefore, it can be used in conjunction with methods such as Secure Tropos and STS, as the latter are concerned with high-level security requirements and interactions of intentional actors within a socio-technical system.

3.1.14 STS Method

The last approach reviewed in this section is that which is at the core of this research: the Socio-Technical Security (STS) method (Dalpiaz et al., 2016). In a similar fashion to the goal-oriented methods introduced earlier in this section, the central tenet of STS is that the design of secure software systems must extend beyond the discussion of technical aspects to also include aspects such as the organizational and social contexts. The method is innovative in that it adopts the perspective of socio-technical systems, acknowledging the importance that such systems have nowadays. Special focus is put on the interactions between social actors, and the objectives that such actors have in the system.

The STS Method can thus be used to design secure socio-technical systems during the early stages of a systems development process. STS is model-driven, given that one of its main activities revolves around the creation of models to represent the security requirements of the system-to-be. The models are created using the STS modeling language (STS-ml), which has all the necessary concepts and relationships to express security requirements. Moreover, the process of constructing such models is supported by a computer tool, called STS-Tool. The tool is a graphical modeling environment that ensures inter-view consistency, possesses reasoning capabilities that simplify the validation process and allows analysts to create security requirements specifications from the envisaged models. Observe that it is within this tool that our gamified tutorial is embedded.

Figure 14 depicts an overview of the activities and deliverables of STS. After performing an initial requirements elicitation, three main activities are conducted: social modeling, information modeling, and authorization modeling. In the first activity, the STS Method starts with an analysis of the socio-technical context, which comprises the identification of stakeholders, the assets they wish to protect, their security needs, as well as the interactions among actors and the threats that such actors face. The resulting deliverable is a social view model, which is iteratively refined until no further refinements are deemed necessary. Subsequently, the information modeling phase comprises the identification of the information that stakeholders own, as well as how this information is structured and materialized in documents. The outcome of this phase is the information view model. In the last modeling phase, permissions and prohibitions between stakeholders are specified in the authorization view model. These relationships express who can use documents that materialize specific information, for what purpose (e.g., to satisfy a certain objective), and how (i.e., read, modify, produce, transfer).

These three views together constitute the STS model. These different, albeit complementary, views may contain inconsistencies or conflicts among security requirements. Thus, the next step is to conduct an (automated) analysis on the STS model to detect such problems. The results of this analysis are provided by the STS-Tool, and they help analysts to fix the errors and conflicts in the corresponding views.

Finally, the last activity concerns the generation of a security requirements specification based on an error-free STS model. Observe that it is possible for a specification to contain unresolved conflicts among requirements, given that in practice it may not be feasible or necessary to resolve all conflicts (e.g., from a cost-benefit point of view). Such decisions should be made in the context of the requirements validation activity by attending to the perspectives of the relevant stakeholders.

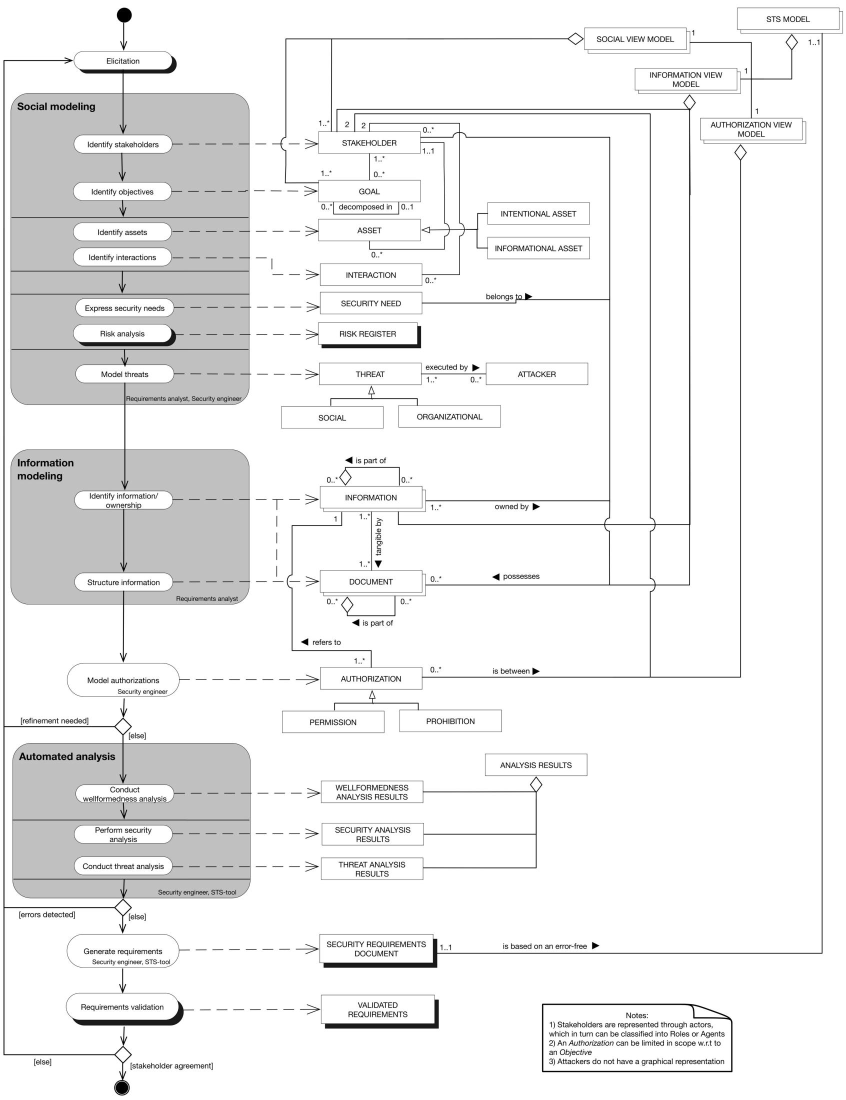


Figure 14 - PDD of the STS Method

Note. *Examples of the views of the STS-ml can be obtained by downloading a copy of the STS-Tooltorial and taking part in the educational activities of the tutorial.*

3.1.15 Conclusions

This section has presented a review of the state-of-the-art in security requirements engineering and, in particular, socio-technical SRE methods such as Secure Tropos and STS. One of the key conclusions extracted from this review is that all the approaches share a common denominator: the principle of considering security early in the software or systems development process (i.e., security by design). However, they also differ in their interpretation of what security requirements constitute. Goal-oriented methods tend to focus on analyzing the intentions and relationships of actors within a social setting in order to extract high-level requirements. Conversely, techniques such as use case extensions, UMLsec, and abuse frames adopt a technical viewpoint and solely focus on the elicitation of security requirements for technical systems. Even more, approaches such as SecBPMN and SecureUML target the late stages of SRE, once the technical systems or detailed organizational processes have been devised.

Another observation is that the approaches can be situated at different abstraction levels. Roughly speaking, they can be classified as: techniques, methods, or frameworks (i.e., super-methods). Specifically, while some of the approaches are just techniques (e.g., attack trees), others can be regarded as methods with prescribed activities and associated modeling techniques (e.g., STS). These methods, in turn, can be integrated within Security Requirements Engineering frameworks (e.g., SQUARE or SREF). Therefore, rather than competing against each other, the approaches can be combined during the design, development, and maintenance of a (socio-technical) system to cover the full systems life cycle. This diversity of approaches is just but a reflection of the multi-faceted nature of the field of SRE.

After having reviewed the SRE landscape, the following section broadens the discussion by looking at the super-field of Requirements Engineering and the field of Software Engineering to review and find inspiration on the state-of-the-art of innovative teaching methods.

3.2 Innovative Teaching Methods in Software and Requirements Engineering

The topic of education has attracted the interest of the scientific communities of both RE and SE. Owing to this growing interest, several conferences and workshops have been organized in the last decade. Examples are the Conference on Software Engineering Education and Training (CSEE&T), the ICSE Software Engineering Education and Training Track (SEET), and the workshop on Requirements Engineering Education and Training (REET). In the context of conceptual modeling, the Symposium on Conceptual Modeling Education (SCME) has been held four times. Moreover, the international iStar Teaching workshop (iStarT) was held last year for the first time. These interdisciplinary venues have called for the active cooperation among related scientific fields. With the aim of informing our research, the following subsections give an overview of innovative teaching methods with a subsequent focus on the use of computer-based tools.

3.2.1 Overview of Teaching Methods

As far as RE teaching is concerned, innovative methods used to enrich traditional classroom training reported in these venues range from improvisation theater (Hoffmann & Weißbach, 2014); role playing (Zowghi & Paryani, 2003; Liang & Graaf, 2010), project simulations (Damian, Hadwin, & Al-Ani, 2006), and case-based courses with either virtual (Gabrysiak, Giese, Seibel, & Neumann, 2010; Beus-Dukic, 2011) or real stakeholder involvement (Sikkel & Daneva, 2011; Gabrysiak, Giese, & Seibel, 2011; Penzenstadler, Mahaux, & Heymans, 2013; Mich, 2014). These methods and techniques are reviewed in turn.

Improvisational (improv) theater sessions, separately popularized by Spolin (1999) and Johnstone (2012), consist of training games where the objective is to learning and having fun. Improv theater is about interactiveness, being spontaneous, and encouraging failure to foster learning (Hoffman & Weißbach, 2014). The use of improv theater in software engineering was first reported by Mahaux and Maiden (2008), who used it to improve the outcomes of the RE process by spurring team-based innovation and stakeholder communication. Recently, Hoffman and Weißbach (2014) have explored its use to teach communication and soft skills to novice requirements engineers. Each game has a set of learning outcomes and is moderated by an experienced coach who reflects on the actions of the players and enforces certain rules.

Role playing is a group dynamics technique where students change their behavior to assume a specific role. In the context of RE education, Zowghi and Paryani (2003) originally introduced it as a complement to traditional approaches, with a focus on uncovering RE issues that arise in real settings. Their work was later picked up by Liang and Graaf (2010), who explored the use of role playing during a course project, which entailed the completion and delivery of a software requirements specification document template. Specifically, students were separated in groups (i.e., developer group and customer group) and each student had a specific role depending on the team to which they were assigned (e.g., team leader and requirements engineer for the developer team, and customer and domain expert for the customer team). Over the course of several weeks, the teams interacted in several meetings, through which students were able to better experience typical RE issues and situations.

Before that, the fictitious customer-developer role proposed by Liang and Graaf (2010) had already been explored to a larger extent by Damian et al. (2006). In a joint effort with three universities located in Canada, Australia, and Italy, they used project simulations as a way to teach geographically distributed software development (GSD) skills to (under)graduate students. Emphasis was put on requirements management and stakeholder communication through computer-mediated software.

The last method concerns what is known as case-based courses, in which students work on a large non-trivial project. Involving (real) stakeholders has been argued to be one of the most effective means to teach SE (and RE) skills, as students can directly model professional practice in a seemingly real setting and reflect on their actions (Penzenstadler et al., 2013). For instance, Gabrysiak et al. (2011) and Penzenstadler et al. (2013) independently reported on the win-win situation that arises for researchers, students, and industry from having real stakeholders for teaching requirements elicitation, specification, and validation. In the domain of conceptual modeling, Sikkel and Daneva (2011) showed how critical reflection skills of students can be developed by involving stakeholders. Specifically, by creating an initial class diagram and then talking with a more knowledgeable stakeholder, who challenged their assumptions, students better understood that there is not a unique solution for any given problem, and that a system must be socially constructed. In the case of requirements analysis, Mich (2014) reported that involving real stakeholders for eliciting requirements and validating (i.e., presenting) a business requirements document increased students' motivation and quality of projects.

However, due to logistic and organizational constraints, collaboration with real stakeholders is a difficult undertaking. To allow for some realistic experience of client-developer interaction, authors such as Gabrysiak et al. (2010) and Beus-Dukic (2011) have examined the use of virtual and role-played stakeholders during the elicitation, specific, and validation phases of a non-trivial project. Their initial results show that virtual stakeholders can be a cost-effective and scalable way to elicit realistic experiences when real stakeholders cannot be involved.

In almost all cases, the material that is taught using these approaches does not go deep into specific methods, techniques or tools. Instead, there seems to be a predominant focus in the teaching of soft skills. At first, this may come off as striking given that RE-specific knowledge is an important aspect of the requirements engineer toolbox. However, Herrmann (2013) conducted an analysis of 200 IT-related job advertisements from a German job portal and found that industry gave more importance to soft skills (92%) (the top three being teamwork, knowledge of English, and communication skills), whereas RE knowledge was only mentioned on 34% of the advertisements. Out of those that mentioned RE knowledge, the most demanded competency was modeling methods acumen (43%). Given that only one of the reviewed approaches focused on modeling, the conclusion is clear: there exists a gap of innovative approaches for teaching conceptual modeling (assuming we are not victims of the law of small numbers).

So far, we have reviewed overarching teaching methods, without any considerations of the mechanism used to deliver such training. Thus, the next section shifts the focus to the use of computer-based tools and, particularly, games (and derivations thereof) in the classroom.

3.2.2 Use of Computer-based Tools in the Classroom

In the reviewed literature, the use of computer-based tools has predominantly focused on the use of software-based feedback agents and simulation. For instance, Merten, Schäfer, and Bürsner (2012) investigated the integration inside an existing RE tool of a system that provides pro-active advice based on a knowledge base consisting of rules and best practices. The system guides practitioners in various RE activities. More recently, Sedrakyan et al. (2014) and Sedrakyan and Snoeck (2015) have studied the application of simulation-based learning for teaching conceptual modeling. Their prototype allows students to receive automated feedback, highlighting parts of the model that are erroneous, so that they engage in a trial-and-error self-regulated learning process. In both instances, the effectiveness of the prototypes was evaluated by conducting pre-post control group experiments, and the specific features of the prototype and attitudes towards use were measured using surveys. Given the suitability and similarity of their evaluation approach to that envisioned for our gamified tutorial, their experimental setup has partly informed our experimental design.

One of the earliest accounts of the potential benefits of introducing computer games for serious purposes such as education and training can be found in the work by Swartout and van Lent (2003). This insight caught up the interest of software engineering faculty scholars, who saw games as a way of increasing the motivation and engagement of students. Among the early pioneers in considering games in the context of SE coursework were Navarro and van der Hoek (2004, 2009), Wang, Øfsdahl, and Mørch-Storstein (2007, 2008), Smith and Gotel (2008), and Connolly, Stansfield, and Hainey (2007). Their works are reviewed below.

Navarro and van der Hoek (2004, 2009) proposed an educational software engineering simulation game, SimSE. This single-player game allows students to practice the activity of managing different types of software engineering processes. In particular, each student takes the role of a project manager and must guide a team of developers to successfully complete a virtual project. In a multi-site evaluation, SimSE was shown to successfully teach the intended learning outcomes and was considered engaging by students. However, the authors caution that SimSE should be treated as a complementary element to other teaching methods (p. 329).

Similarly, Wang et al. (2007, 2008) explored the opportunities of gamifying the lectures of a software architecture course. They introduced Lecture Quiz, a multiplayer quiz game aimed at promoting student engagement and variation in teaching methods. Their evaluation showed that the game contributed to increased learning and student satisfaction.

Smith and Gotel (2008) created RE-O-Poly, a modified version of the famous Monopoly game whose main objective is to teach eight basic RE practices to undergraduate students. The authors argue that the game can be used by novice organizations which need a fast and cost-effective to introduce RE. Unfortunately, its development has been discontinued since 2009.

Finally, the game proposed by Connolly et al. (2007) can be argued to be one of the most refined games seen in SE literature.The authors propose a multi-player scenario-based simulation game where a team of students (either university students or professionals) have to deliver and manage a number of software projects. Each player is assigned a role (e.g., system analyst, developer, team leader) and tasked with a series of responsibilities. In addition, a so-called facilitator can intervene during the game to pose challenges to the team so that they can experience the eventualities of software projects. Some of these interventions are: call a team meeting, change requirements during development, reduce the number of resources assigned to a project, etc.

In the area of conceptual modeling, to our knowledge there has only been one attempt to apply games or game-like elements in both undergraduate and graduate curricula. Oliveira, Werneck, Leite, and Cysneiros (2015) introduced a Monopoly-based game to teach constructs of ERi*c, a technique based on the i* and NFR languages.

Overall, while there is a considerable amount of literature that points at the use of games for teaching aspects of SE and RE, the same cannot be said about the use of gamification. Indeed, we found a dearth of studies examining the possibilities of gamification as applied to learning and, specifically, (interactive) tutorials. It is thus important to note the empirical and knowledge gap that exists in the current literature of gamification in software engineering in general and conceptual modeling in particular. At this point it is timely to mention that gamification (our area of focus) and (serious) games are related but nonetheless different concepts. In this sense, **Section 3.4** reviews relevant literature on gamification (of learning) and lays out the foundation for the gamification component of the STS-Tooltorial.

3.2.3 Conclusions and Limitations

The last decade has seen a growing interest in the application of innovative learning methods, techniques, and tools for RE and SE curricula. Despite the proliferation of these different methods, little emphasis has been put on evaluating their empirical effectiveness using robust approaches such as experiments. At the very best, evaluations are conducted using surveys, but the majority only inquire about psychological outcomes (e.g., student satisfaction with the experience). While the authors of the methods report positive experiences and results, one can question whether this situation is a reflection of their effectiveness or a result of mere-exposure effect (Zajonc, 2001). In general, several methodological limitations could be identified during this literature review: 1) sample sizes are usually small (N=20 or lower) or not reported, 2) validated psychometric instruments are not used, 3) evaluations lack experimental control groups and are solely based on user evaluation or personal experiences, and 4) evaluation timeframes are often very short, which may have significantly confounded the results of the studies (e.g., positive experiences due to mere-exposure effect). On the longer term, the practical benefits and generalizability (i.e., from students to professionals) of those artifacts that have a practitioner orientation should be investigated.

This literature review has explicitly focused on venues where SE and RE scholars gather to address issues related to education. Thus, this review has limitations with regard to reporting the research done in other conceptually- or theoretically- similar disciplines related to innovative teaching methods and gamification (i.e., selection bias). There is the possibility that there are studies that inquire about similar phenomena of interest, but discussed them in different terms, and thus were impossible to find within the time constraints of this project. Here we relied on snowballing techniques and had as inclusion criteria the aforementioned selected venues, which inevitably limited the amount of gathered papers. Note, however, that these venues are the main publication targets for requirements engineering education research (Ouhbi, Idri, Fernández-Alemán, Toval, 2015). Therefore, the present section presents a close look at the research done in the topic of innovative teaching methods for RE and conceptual modeling in particular. Due to the lack of standardization in the field (e.g., different research methods, lack of valid measurements), performing a formal meta-analysis was rendered impractical. This is worsened because multiple studies employed qualitative methods. As the research on the effectiveness of teaching methods progresses, special consideration should be given to ensure the comparability of studies. Thus, we urge that future studies focus on the execution of proper inferential studies, by e.g., conducting experiments, using validated psychometric instruments, or ensuring an adequate sample size.

3.3 Interactive Tutorials and Design Considerations

This section gives an overview of best practices that can be leveraged during the design of interactive tutorials. Before delving into the specific guidelines, it is important to draw a distinction between the different possible tutorial formats that can be found, and where exactly the STS-Tooltorial falls. Broadly speaking, tutorials can be classified along four dimensions. The first three are borrowed from the work of Fernquist, Grossman, and Fitzmaurice (2011).

- **Scope**: tutorials can be task centric, feature centric, content (i.e., workflow), or method centric.
- **Interactivity**: they can be passively consumed, active, or reactive.
- **Integration**: they can be provided through physical guides, on-line, or in-application/tool.
- **Gamification**: tutorials can be gamified, or not gamified.

Based on this classification, the STS-Tooltorial is a gamified, method, reactive, in-tool tutorial. Note that, to our knowledge, this research represents the first account of such tutorial in literature.

Most of the implementations focus on tutorials that solely deal with features present in an application, that is, ways to increase software learnability. As Figure 15 shows, the simplest forms of interactive tutorials are guided tours and overlays. They are employed to orient first-time users (Cooper et al., 2014).

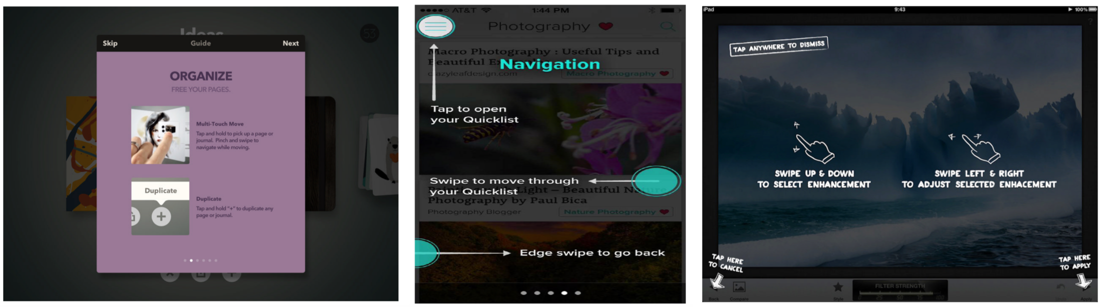


Figure 15 - Examples of interactive tutorials: guided tours and overlays (Cooper et al., 2014)

More refined tutorials are those that support the workflow of users for longer periods. Examples of this can be found in Dropbox, GamiCAD, and Ribbon Hero 2. Dropbox uses a tutorial that consists of tasks that must be completed to obtain extra space. GamiCAD is a gamified interactive tutorial system for novice users of AutoCAD (Li, Grossman, & Fitzmaurice, 2012). Similarly, Ribbon Hero 2 is a video game that gradually exposes novice users to the features in Microsoft Office (Microsoft.com, 2011). Even more, there are others that, adopting a holistic approach, have a stronger educational component that goes beyond the features within the application. These tutorials attempt to teach concepts that can presumably have an effect on the learners' cognitive processes and real-world behavior. An example would be the STS-Tooltorial.

When creating a system that offers instructional material, it is important to draw on past research where such an undertaking has been attempted or addressed. Guidelines for creating computerized (interactive) training tutorials can be traced back to more than three decades ago, with the novel works of Anderson, Carroll, and their respective colleagues. What follows is an overview of empirical research studies along with relevant conclusions.

After a decade of research on intelligent tutor development based on computer tutoring theory Anderson, Corbett, Koedinger, & Pelletier (1995) extracted the following principles (only those relevant are shown here): 1) provide instruction in the problem-solving context, 2) promote an abstract understanding of the problem solving context, 3) minimize working memory load, 4) provide immediate feedback on errors, and 5) facilitate successive approximations to the target skill.

A well-known theory is the minimalist instruction approach pioneered by Carroll (1990), distilled after his observations on the effectiveness of traditional systematic self-instruction. This theory is especially relevant for the design of computer-based training (i.e., tutorials). It becomes even more important in our context, given the limited amount of screen real estate within the application. Owing to the varying interpretations of misconceptions regarding minimalism (Carroll & van der Meij, 1996), it is worth looking at different perspectives with the hope of triangulating.

From the point of view of Kearsley (as cited in Patsula, 1999), the theory posits that: 1) all learning activities should be meaningful and self-contained, 2) activities should exploit a learner's previous experience and knowledge, 3) learners should be given realistic projects as quickly as possible, 4) instruction should permit self-directed reasoning and improvising by increasing the number of active learning activities, 5) training activities should provide for error recognition and recovery, and 6) there should be a close linkage between the training and the actual task.

Minimalist theorists van der Meij and Carroll (1995, p. 21) summarized key principles and heuristics for minimalist instruction design. We reproduce them here due to their relevance for interactive tutorials:

- **Principle 1**: Choose an action-oriented approach.
  - **Heuristic 1.1**: Provide an immediate opportunity to act.
  - **Heuristic 1.2**: Encourage and support exploration and innovation.
  - **Heuristic 1.3**: Respect the integrity of the user's activity.
- **Principle 2**: Anchor the tool in the task domain.
  - **Heuristic 2.1**: Select or design instructional activities that are real tasks.
  - **Heuristic 2.2**: The components of the instruction should reflect the task structure.
- **Principle 3**: Support error recognition and recovery.
  - **Heuristic 3.1**: Prevent mistakes whenever possible.
  - **Heuristic 3.2**: Provide error information when actions are error prone or when correction is difficult.

    - **Heuristic 3.3**: Provide error information that supports detection, diagnosis, and recovery.
    - **Heuristic 3.4**: Provide on-the-spot error information.
- **Principle 4**: Support reading to do, study and locate.
    - Heuristic 4.1: Be brief, don't spell out everything.
    - Heuristic 4.2: Provide closure for chapters.

In his book review, Horn (1999, p.1) cautions that minimalism may not be the right approach for advanced training or reference documentation. He further distills the following nine principles:

1. Use real tasks for the training exercises and let users select their own tasks.
2. Get the learner started on real tasks fast by eliminating almost all front-end orientational material.
3. Guide learners' reasoning, exploring and improvising with questions and other hints.
4. Design the materials so that they can be read in any order in so far as possible.
5. Help learners to coordinate training materials and software by providing landmarks for normal or error situations
6. Focus early attention in the training materials on enabling the learner to recognize and recover from errors.
7. Engage the learner's prior knowledge in introducing novel concepts. Use familiar office tasks, language and metaphors.
8. Consider using the learning situation, as opposed to practical on-the-job examples, for learning examples, exercises and explorations.
9. Aim for optimizing learning designs by repeated testing and avoiding the temptation to systematize approaches into checklists.

Beyond the concept of minimalist design, there are some studies worth mentioning.

In a pioneering study, Carroll and Carrithers (1984) evaluated a so-called training wheels design. The design is based on the progressive disclosure concept, whereby users are initially limited to a small set of features vs. unlimited designs. Moreover, attempted departures from a correct action path are blocked. Their evaluation showed that the former design enhanced learnability, because users were less frustrated and exposed to fewer features.

In the context of interactive multimedia, Park and Hannafin (1993) extract 20 empirically-based principles and implications for the design of learning systems based on psychological, pedagogical, and technological foundations. These are reproduced in Table 4.

Table 4 - Principles and implications for the design of learning systems (Park & Hannafin, 1993)

Principle	Implications	How it will be applied
Related prior knowledge is the single most powerful influence in mediating subsequent learning	Layer information to accommodate multiple levels of complexity and accommodate differences in related prior knowledge	Educational content becomes progressively more difficult as the user advances in the tutorial (in agreement with gamification principles)
New knowledge becomes increasingly meaningful when integrated with existing knowledge (meaning is developed by the learner, not placed into the learner)	Embed structural aids to facilitate selection, organization, and integration; embed activities that prompt learners to generate their own unique meaning	Not applicable in the context of the STS-Tooltorial
Learning is influenced by the supplied organization of concepts to be learned	Organize lesson segments into internally consistent idea units	Educational content is segmented in small pieces of coherent information (in agreement with gamification principles)
Knowledge to be learned needs to be organized in ways that reflect differences in learner familiarity with lesson content, the nature of the learning task, and assumptions about the structure of knowledge	Linkages between and among nodes need to reflect the diverse ways in which the system will be used	A hierarchical approach is used for organizing the instructional content, as it is assumed that learners have little prior knowledge
Knowledge utility improves as processing and understanding deepen	Provide opportunities to reflect critically on learning and to elaborate knowledge; encourage learners to articulate strategies prior to, during, and subsequent to interacting with the environment	Not applicable due to the highly structured nature of the tutorial
Knowledge is best integrated when unfamiliar concepts can be related to familiar concepts	Use familiar metaphors both in conveying lesson content and designing the system interface	Visual interface metaphors are used when applicable to lessen processing demands (e.g., cards for selecting topics)
Learning improves as the number of complementary stimuli used to represent learning content increases	Present information using multiple, complementary symbols, formats and perspectives	Textual information is supplemented with pictures Several forms of feedback are given (e.g., auditive, color highlight, textual)

Principle	Implications	How it will be applied
Learning improves as the amount of invested mental effort increases	Embed activities that increase the perceived demand characteristics of both the media and learning activities	Gamification is used to increase engagement and need to win Time-based exercises
Learning improves as competition for similar cognitive resources decreases and declines as competition for the same resources increases	Structure presentations and interactions to complement cognitive processes and reduce the complexity of the processing task	Principles of minimalist instruction and minimalist interface design are employed
Transfer improves when knowledge is situated in authentic contexts	Anchor knowledge in realistic contexts and settings	The tutorial is embedded within the STS modeling tool Educational content is anchored in a meaningful and relevant context through a fictional story
Knowledge flexibility increases as the number of perspectives on a given topic increases and the conditional nature of the knowledge is understood	Provide methods that help learners acquire knowledge from multiple perspectives and cross-reference knowledge in multiple ways	Not applicable within the scope of the project
Knowledge of details improves as instructional activities are more explicit, while understanding improves as the activities are more integrative	Differentiate orienting activities for forthcoming information based upon desired learning: provide organizing activities for information already received	Objectives and expectation are stated before beginning a mission
Feedback increases learning important lesson content, and decreases incidental learning	Provide opportunities to respond and receive feedback but avoid excessive response focusing when incidental learning is expected	Feedback is constantly provided based on user responses
Shifts in attention improve the learning of related concepts	Differentiate important information through cosmetic amplification, repetition, and recasting to direct learners' attention	Changes in color or fonts are used to highlight relevant parts within a model or the tutorial itself Arrows are used to prompt to specific places in a model based on context
Learners become confused and disoriented when procedures are complex, insufficient, or inconsistent	Provide clearly defined procedures for navigating within the system and accessing on-line support	Tutorial is designed using established web UI guidelines (e.g., clear navigation, error recovery) Onboarding tutorial helps first time users

Principle	Implications	How it will be applied
Visual representations of lesson content and structure improve the learner's awareness of both the conceptual relationships and procedural requirements of a learning system	Provide concept maps and other graphical aids to help learners understand, locate, and navigate within interactive learning systems	Orientation is improved by structuring the lesson content in missions
Individuals vary widely in their need for guidance	Provide tactical, instructional, and procedural assistance	The figure of game master is used to assist learners and orchestrates the gaming/ learning process Instructional and tactical assistance is given through context-sensitive feedback (e.g., video tutorials, links between the tutorial and current state of the model canvas). Procedural assistance is given through the onboarding tutorial upon first time use
Learning systems are most efficient when they adapt to relevant individual differences	Interactive multimedia must adapt dynamically to both learner and content characteristics	Tailored feedback is given based on personality traits
Metacognitive demands are greater for loosely structured learning environments than for highly structured ones	Provide prompts and self-check activities to aid the learner in monitoring comprehension and adapting individual learning strategies	Not applicable, as the tutorial is highly structured
Learning is facilitated when system features are functionally self-evident, logically organized, easily accessible, and readily deployed	Employ screen design and procedural conventions that require minimal cognitive resources, are familiar or can be readily understood, and are consonant with learning requirements	Some attention is paid to interaction design and HCI concerns during the design of the tutorial

Plaisant and Shneiderman (2005) argue that video tutorials have become an increasingly prevalent source of information for users, and put forward 10 guidelines for creating recorded demonstrations: 1) provide procedural or instructional information rather than conceptual information, 2) keep segments short, 3) ensure that tasks are clear, 4) coordinate demonstrations with text, 5) use spoken narration, 6) be faithful to the actual user interface, 7) highlight areas to guide attention, 8) give users control, and 9) keep file sizes small, and 10) strive for universal usability.

Another area concerns the application of contextual assistance. Its benefits were originally showed by Anderson, Boyle, Farrell, and Reiser (1987). Nowadays, the pervasive way of offering assistance is through the use of tooltips (i.e., short textual information). The main task of tooltips is to help users locate functionality, although they do not increase understanding (Ehret, 2002). More recently, Grossman and Fitzmaurice (2010) have proposed ToolClips, a form of contextual video assistance (i.e., video tutorials) for learning software functionality without interrupting flow. Their evaluation showed that ToolClip users significantly completed more tasks successfully than those who used traditional help techniques.

3.3.1 Guidelines for Designing Interactive Tutorials

This section ends with a general overview of the guidelines for creating interactive tutorials extracted from the literature reviewed. These guidelines are summarized in Table 5. In those instances were different authors give similar guidelines, a new one combining them has been created.

Table 5 - Guidelines for designing interactive tutorials

ID	Guideline	References
1	Provide instruction in the problem-solving context; anchor the tool in the task domain	(Anderson et al., 1995; Patsula, 1999; van der Meij & Carroll, 1995; Park & Hannafin, 1993)
2	Promote an abstract understanding of the problem solving context	(Anderson et al., 1995)
3	Minimize working memory load and complement cognitive processes by structuring presentations and interactions	(Anderson et al., 1995; Park & Hannafin, 1993)
4	Provide on-the-spot, immediate error information that supports detection, diagnosis, and recovery	(Anderson et al., 1995; Patsula, 1999; van der Meij & Carroll, 1995; Horn, 1999)
6	Facilitate successive approximations to the target skill	(Anderson et al., 1995)
7	Provide an immediate opportunity to act	(Patsula, 1999; van der Meij & Carroll, 1995; Horn, 1999)
8	Select or design instructional activities that are real tasks; anchor knowledge in realistic contexts and settings	(Patsula, 1999; van der Meij & Carroll, 1995; Horn, 1999; Park & Hannafin, 1993)
9	Engage the learner's prior knowledge in introducing novel concepts. Use familiar language and metaphors both in conveying lesson content and designing the system interface	(Patsula, 1999; Horn, 1999; Park & Hannafin, 1993)
10	Guide learners' reasoning, exploring and improvising with questions and other hints	(Patsula, 1999; Horn, 1999)

ID	Guideline	References
11	Design the materials so that they can be read in any order in so far as possible	(Horn, 1999)
12	Help learners to coordinate training materials and software by providing landmarks for normal or error situations	(Horn, 1999)
13	Aim for optimizing learning designs by repeated testing and avoiding the temptation to systematize approaches into checklists.	(Horn, 1999)
14	Layer information to accommodate multiple levels of complexity and accommodate differences in related prior knowledge	(Park & Hannafin, 1993)
15	Embed structural aids to facilitate selection, organization, and integration; embed activities that prompt learners to generate their own unique meaning	(Park & Hannafin, 1993)
16	Organize lesson segments into internally consistent, meaningful, and self-contained units	(Patsula, 1999; van der Meij & Carroll, 1995; Park & Hannafin, 1993)
17	Linkages between and among nodes need to reflect the diverse ways in which the system will be used	(Park & Hannafin, 1993)
18	Provide opportunities to reflect critically on learning and to elaborate knowledge; encourage learners to articulate strategies prior to, during, and subsequent to interacting with the environment	(Park & Hannafin, 1993)
19	Present information using multiple, complementary symbols, formats and perspectives	(Park & Hannafin, 1993)
20	Embed activities that increase the perceived demand characteristics of both the media and learning activities	(Park & Hannafin, 1993)
21	Provide methods that help learners acquire knowledge from multiple perspectives and cross-reference knowledge in multiple ways	(Park & Hannafin, 1993)
22	Differentiate orienting activities for forthcoming information based upon desired learning: provide organizing activities for information already received	(Park & Hannafin, 1993)
23	Provide opportunities to respond and receive feedback but avoid excessive response focusing when incidental learning is expected	(Park & Hannafin, 1993)
24	Differentiate important information through cosmetic amplification, repetition, and recasting to direct learners' attention	(Park & Hannafin, 1993)
25	Provide clearly defined procedures for navigating within the system and accessing on-line support	(Park & Hannafin, 1993)
26	Provide concept maps and other graphical aids to help learners understand, locate, and navigate within interactive learning systems	(Park & Hannafin, 1993)
27	Provide tactical, instructional, and procedural assistance	(Park & Hannafin, 1993)

ID	Guideline	References
28	Interactive multimedia must adapt dynamically to both learner and content characteristics	(Park & Hannafin, 1993)
29	Provide prompts and self-check activities to aid the learner in monitoring comprehension and adapting individual learning strategies	(Park & Hannafin, 1993)
30	Be brief, don't spell out everything (this table is not a good example of it)	(van der Meij & Carroll, 1995)
31	Apply the progressive disclosure principle (e.g., training wheels design)	(Carroll & Carrithers, 1984)
32	The design of video tutorials should comply to guidelines such as: 1) provide procedural or instructional information rather than conceptual information, 2) keep segments short, 3) ensure that tasks are clear, and 4) coordinate demonstrations with text	(Plaisant & Shneiderman 2005)
33	Offer contextual assistance through (animated) tooltips	(Anderson et al., 1987; Ehret, 2002; Grossman & Fitzmaurice, 2010)

Beyond the guidelines, the creation of an IT artifact cannot be devoid of giving due consideration to the fields of interaction design, user experience design, and human-computer interaction (albeit some of the reviewed literature already points in that direction). However, observe that the specific focus of this section was on gathering guidelines that specifically refer to the design of interactive tutorials. To that end, **Chapter 5** delves into interaction design considerations when reporting the process by which the STS-Tooltorial was designed. Moreover, these guidelines are further complemented by the best practices for gamifying learning experiences put forward in **Section 3.4.6**.

3.4 The Gamification Landscape: Taming the Beast

3.4.1 Overview of Gamification

Despite the interest and popularity that the term gamification has garnered over the last year in both industry (Kim, 2008) and academia (Deterding et al., 2011), the scientific community has not yet agreed on a definition. One of the reasons is that the areas that gamification touches upon have been under scrutiny for several decades and by different research communities. This has greatly contributed to the lack of preciseness about the term, and consequent agreement among scholars. Initial accounts of early use of gamification can be traced back to the 80s. That decade saw a surge of interest in drawing on game design and game elements to derive heuristics for creating enjoyable interfaces. In his seminal paper, Malone (1980) identified challenge, fantasy, and curiosity as the key elements to create enjoyable interfaces. More recently, Deterding et al. (2011) have inextricably linked gamification to the fields of user experience (UX) and interaction design (IxD).

So far, scholarly articles have adopted two main ways of defining gamification, with a preference for the first as reflected by the number of citations:

1. The use of game elements in non-game contexts (Deterding et al., 2011).
2. A process of providing (motivational) affordances for gameful experiences which support the customers' overall value creation (Huotari & Hamari, 2012).

Rather than opting for one approach, this study sets out to reunite both perspectives. Specifically, while the first definition is preferred, in that the definition is not tied to a particular domain (e.g., service marketing), we borrow the term affordance from the second definition, which has been traditionally used in the field of human-computer interaction (Gibson, 1977). Motivational affordances are designed with the intent of addressing the motivational needs of users and affecting their psychological needs, which in turn are responsible for (partially) invoking favorable attitudes towards the goal or behavior of interest (Zhang, 2008; Jung et al. 2010). This stresses the idea that it is the users who voluntarily act with the system and its affordances in order to meet their needs. Hence, the second definition puts more value on the subjective nature of gamification by focusing on the user experience rather than game mechanics.

While a discussion on the philosophical perspectives on game and play are beyond the scope of this thesis (see Deterding et al., 2011 for a good overview of this issue), it is important to stress that gamification and games are not the same.

Gamification is best understood as the mere inclusion of game design elements (e.g., interface or design patterns, game mechanics, principles, or models) in either digital or non-digital systems, so that they may afford gameful experiences. Conversely, games are built with the intention of enacting a full-fledged game experience. Of course, whether a gamified system is experienced as being gameful, playful, or instrumental, is subject to the interpretation of the user, and even more, the socially constructed meaning that a group attaches to the experience (Consalvo, 2009). These are the characteristics that, according to Deterding et al. (2011) set apart gamified systems from games. Thus, a clear distinction must be made between game design and gamification.

3.4.2. Rethinking Gamification

Gamification has inevitably found its way into the public discourse and, as a result, has been repeatedly used and abused by marketing and sales departments. Coupled with the pervasiveness of shallow implementations being regarded as gamification, which do not go beyond offering meaningless rewards (viz., Points, Badges, and Leaderboards, or PBL) for uninteresting activities, it is not surprising to see that the term has garnered a negative perception among critics, some of whom have come to regard it as digital snake oil (Juul, 2011).

The prevalent framing of gamification, perhaps popularized by the local search and discovery service Foursquare, as nothing more than the use of PBL to drive behavior has negative consequences. For one, it disregards several decades of empirical research on the negative consequences and limited effectiveness of such kind of incentive systems (see e.g., Kohn, 1999). Secondly, this worrisome conception of gamification falls short when it comes to drawing on theories of game psychology and enjoyment. Finally, a disregard for the ethics of design and the complexities of design and motivation seem to have settled in the public mind (see the following subsection for a depiction of the ugly side gamification).

As a result of these misconceptions, this situation has led Deterding (2014) to urge the community to rethink what gamification actually stands for. In his article, he presents six critiques of the current view on gamification (in part once championed by him), along with six proposals to rethink it. Given the importance of this work for the topic of this thesis, we present and discuss five of them (for a discussion of the sixth point, ethics, see **Section 3.4.3**):

1. From objects to contexts. The discussion of gamification, Deterding argues, should encapsulate the contexts in which users come to interact with a system. He identifies three basic context that mediate the interaction effects with any gamified system:

- **Autonomy**: gameplay should constitute a voluntary activity (Huotari & Hamari, 2012), otherwise it can harm the affective experience and performance of people (Heeter et al., 2011; Mollick & Rothbard, 2014). This is specially relevant in utilitarian contexts (e.g., workplace). The reason is that, in line with self-determination theory (see **Section 3.4.4**), a perceived lack of autonomy undermines intrinsic motivation and enjoyment (Deci & Ryan, 2012). Perceived in the sense that the interpretation of whether any type of feedback (i.e., verbal or material) can either be interpreted as controlling (and thus autonomy-thwarting) or informational (and thus competence-enhancing) is dependent upon the receiver of such feedback (Deci & Ryan, 1985; Deci & Ryan, 2011). Consequently, in utilitarian contexts (e.g., education and workplace), gamification systems may decrease motivation and performance when their use is perceived as an imposition.

- **Situational norms**: In games, finding exploits to bent its rules (i.e., gaming the system) is a welcomed activity. Finding the limit on such practices, especially in a work setting, becomes a difficult problem to manage: gamified systems do not have a clear way of showing which actions are permissible, and which ones go beyond the rules. This is in contrast to "games proper", where the community itself enforces fair play and collective enjoyment by the implicit institutionalization of social norms (Koven, 2013).
- **Embarrassment**: The sociologist Goffman (1956) argues that social norms are effected by feelings of embarrassment and shame, which in turn regulate behavior. If the non-game context in which a gamified system is deployed does not regard play as appropriate, people will refrain from engaging with the system in a gameful way to avoid feeling embarrassed.

These situation-specific contexts, Deterding (2014) contends, reflect that a change of perspective must take place, i.e., from merely designing a (gamified) software system to also considering the social and organizational aspects in which such systems operate. In other words (and in connection with the topic of this thesis), gamified systems should follow a socio-technical systems design practice.

2. From game elements to motivational design. The current discourse on gamification narrowly focuses on creating gameful and (to a less extent) playful experiences, when the ultimate goal is to motivate user behavior. Therefore, bringing the definition closer to that of Huotari and Hamari (2012), Deterding suggests extolling the goal of gamification to that of affording experiences through motivational affordances . This position situates gamification as a subset of motivational, or persuasive, design (Fogg, 2003). The implications are clear: gamification might not be the solution for any given motivational design challenge. Moreover, in those cases were gamification is rightfully deemed as appropriate, designers should draw on fields such as persuasive design to inform their designs. This constant conceptual repositioning, we contend, begs the question of whether gamification is finally finding its way as an academic discipline, or whether it can be subsumed to other, more established domains.

3. From deterministic cause-effect game design elements to affordances. The current conception of gamification is that, if we add certain game elements, we will automatically and deterministically cause a certain motivational effect in the users of a gamified system. In other words, that the experience of a dreadful activity will become more appealing (and possibly more productive) just by adding so-called game design elements, which deterministically are known to produce one exact motivational experience. This vision is far from reality: 1) as outlined in point 1, different people can interpret a stimulus (e.g., game element) in different ways, 2) a game element may provide different motivational affordances and vice versa (Antin & Churchill, 2011), and 3) motivational affordances emerge as a result of the holistic relation between a user, object, and environment. This implies that gamification ought to be considered as a component of the overall user experience. We hope to create a greater shared understanding on these issues with the conceptual model that represents these dynamics (depicted in Figure 16).

4. From patterns to lenses. This entails shifting from the prevalent pattern-based gamification design focused on game elements such as PBL, which is in conflict with the affordance viewpoint on motivation, towards one that promotes game design. Deterding (2013) argues that patterns are domain-specific and that, detached from its context, offer no meaningful guidance. For example, applying the "glowing choice" pattern (Chou, 2015) does not automatically generate a desired experience (i.e., users feeling that they have to click on it). Instead, promising approaches are the use of design lenses (Schell, 2009) and playcentric design (Fullerton, 2014). Specifically, a gamified system should be designed considering the motivational experiences that need to be afforded, placing emphasis on prototyping, testing, and adapting it until one can ensure that such experiences are actually afforded.

5. From tokenism to critical transformation. Contemporary gamification attempts to modify a behavior to some perceived better behavior without tackling the root cause for that behavior. Ultimately, Deterding (2014) argues that the purpose of gamification must be reconsidered by focusing on the real problem (i.e., well-being, or extending human capabilities). This requires explicit understanding of the users and contexts in which a gamified system will interact, and calls for approaches such as "participatory design, sustainability, and systems thinking" (p. 323).

Based on this analysis, it is apparent that the new paradigm of gamification set forth by Deterding requires much more work than originally envisioned. All these propositions make perfect sense, yet they may come across as a very difficult undertaking. Compare all the effort it would take to design a gamified system that holistically affords certain motivational experiences, with due consideration to game aspects, from a socio-technical systems stance, vs. simply adding a layer of game elements (e.g., PBL) on top of an already (possibly badly designed) system. Indeed, we can draw no further conclusion but the one that Kohn posited for pop behaviorism (of which the prevalent perspective on gamification is arguably an extension): "[rewards] (in our case, game elements) are seductively simple to apply". Common sense is not common practice, and this may explain why many academic studies and gamification implementations opt for the easy path. However, ignoring this critique puts us in an uncomfortable spot. One can blatantly ignore these critiques and look at the current perspective for confirmatory bias, or aim at doing gamification right.

In this thesis, we consciously decided to predominantly follow the once-proposed and pervasive game element-based approach that Deterding and others have come to criticize. This was decided after giving due consideration to the following aspects: 1) limited scope and time budget for the project, 2) the tutorial is to be integrated within an existing tool, 3) lack of (human) resources that the holistic approach reviewed in this section would necessitate. However, an attempt is made to take a middle-ground position by reckoning with some of Deterding's (2013) considerations. Note that the contents of this section will surface again when we examine the results and limitations of the study. Thus, for the purposes of the empirical evaluation, the study is restricted to the evaluation of the effectiveness of including game design elements into an interactive tutorial, which is itself part of a tool for the conceptual modeling of security requirements.

Having reviewed the current perspective on gamification, the next section turns it attention to the issues of gamification.

3.4.3. The Ugly Face of Gamification

This section highlights some of the criticisms and issues that have been raised with regard to gamification, focusing on those that are most relevant to this study.

Generally speaking, gamification can be situated as another component of the pervasive socio-technical transformation that is currently taking place, that is: people are increasingly (and perhaps inadvertently) relying on software to track, analyze, and automate their lives. As such, gamification poses a series of challenges that span areas as diverse as economics, privacy, data rights, culture, and ethics.

From an economics perspective, much of the hype of gamification has been attributed to the money-making interests of capitalists, who have repeatedly used the term to fit their needs (Rey, 2015). Beyond the goal of wealth accumulation, Rey contends in her post-Fordist analysis of gamification, capitalist institutions have appropriated games and play to enable exploitation of workers. In light of this, other authors such as Koven urge us to reconsider the single most important purpose of games: to have fun by playing together. Fun, he argues, should prevail over any other instrumental objectives, which are usually the reason why gamified systems are deployed (e.g., money, information, productivity, or training). To this, we argue that whenever a term catches the interest of the public, marketeers and (thought) leaders will jump on the bandwagon and try monetize on it. Surely, by no means is this limited to gamification. Moreover, given that gamification has predominantly found its way into the digital world (and thus the IT industry), it is subject to the hype cycle. Although it may be a buzzword, this does not imply that gamification cannot be established as a formal field of inquiry.

As far as privacy and security is concerned, authors such as Whitson (2015) have come to regard gamification as a de-facto surveillance and control mechanism that can potentially be accessed by third parties (e.g., governments). This is specially the case for systems that engage in behavior-tracking activities. Another salient aspect refers to ownership of the data generated by these applications. In many cases, Andrews (2015) notes, users may be unaware that data is being gathered and distributed to third parties. Therefore, she proposes an informed consent model that gives ultimate control of the data to the end users of gamified systems.

From a cultural standpoint, Khaled (2015) has explored how contemporary gamification implementations tend to be biased towards particular cultural contexts, where there exists a focus on certain interpersonal dynamics (e.g., competition, differentiation, hierarchy). Given that such gamified systems are utilized into the real world, it implies that they may not be ineffective out of these achievement- and power-focused contexts (e.g., Asian cultures place group over individual identity) because users would be culturally alienated. Thus, she proposes a more inclusive design approach that considers culture during the design process. For the purposes of this thesis, the STS-Tooltorial will be designed for cultures that foster egalitarianism, intellectual autonomy, and harmony, a schema that coincides with most West European countries (Schwartz, 2006). This can be achieved by allowing learners to pursue the learning tasks in an individualistic way, providing normative feedback, and allowing people to help each other by exchanging/liking comments.

A last area of concern is that of ethics. As repeatedly stated, one of the core aims of gamification is to motivate people to change their perceptions and attitudes towards certain activities. Because of this, early criticisms deemed gamification as inherently manipulative and coercive (Bogost, 2011). In the context of organizational settings, Shahri, Hosseini, Phalp, Taylor, and Ali (2014) examined its perils for employees. Some of the issues raised were: tension in the workplace (e.g., as a result of leaderboards), gamification as a monitoring device, freedom of information and data usage without the employee's consent, gamification as exploitationware, and behavior manipulation through a carrot-and-stick approach. Indeed, some of these concerns can be extrapolated beyond the enterprise. To counteract this criticism, Deterding (2014) proposes imbuing ethics into the design of gamified applications, to which he refers as eudaimonic (i.e., "the good life") design. In Deterding's words: "[ethical] gamifcation would thus mean (a) being a potential tool for "positive design" actively supporting human flourishing, (b) a practice performed virtuously, excellently in itself, and (c) something that realises, furthers, or is at least congruent with living a good life with others" (p. 321).

As a concluding remark, it is worth summarizing this section in dinah boyd's words: "gamification is a modern-day form of manipulation. And like all cognitive manipulation, it can help people and it can hurt people. And we will see both" (Anderson & Rainie, 2012). As is the case with other disciplines (e.g., persuasive design), we also ascribe to the view that designers are effectively materializing intentionality and morality through their designs (Berdichevsky & Neuenschwander, 1999; Johnson, 2006; Verbeek 2006). It is thus the designer's responsibility to deeply reflect on the intentionality of gamified systems and the ethical consequences that stem from the interactions of humans and machines.

3.4.4. Theoretical Foundations of Gamification: Motivational Dynamics

For the sake of completeness, this chapter ends with a short overview of the main theoretical underpinnings of motivational dynamics for gamification that are relevant for this thesis. As noted by Deterding (2011), a promising approach for conceptualizing (meaningful) gamification is by drawing on need satisfaction theories, with a focus on boosting intrinsic motivation. The leading theory in this regard is Self-Determination Theory (SDT) (Deci & Ryan, 2011). SDT argues that humans seek out and engage in activities that promise to satisfy motivational needs. According to SDT, these needs are: autonomy (presented in **Section 3.4.2**), competence, and relatedness. An activity is thus intrinsically motivating if it satisfies these needs. This work has been further repackaged under the so-called *motivation trifecta* popularized by Pink (2011), which consists of three needs (viz., autonomy, mastery, and purpose).

A sub-theory of SDT, Cognitive Evaluation Theory (CET), explains that need satisfaction depends on how people interpret the functional significance of external events. If an event is interpreted as being informational about one's competence, it boosts intrinsic motivation. However, if the event is perceived as controlling, it thwarts autonomy and thus reduces intrinsic motivation. This explains why attaching extrinsic motivations to a reward (e.g., money) or giving controlling feedback can undermine intrinsic motivation (provided that the subject was intrinsically motivated) and possibly lead to a worse task performance (Deci, Koestner, & Ryan, 1999).

Giving that playing games is an intrinsically motivating activity, it is important to draw on this research while designing and evaluating gamified systems (like ours) that attempt to increase, or at least not diminish, intrinsic motivation in subjects that use them in non-game contexts. These theories have informed the design of the gamification component of the STS-Tooltorial.

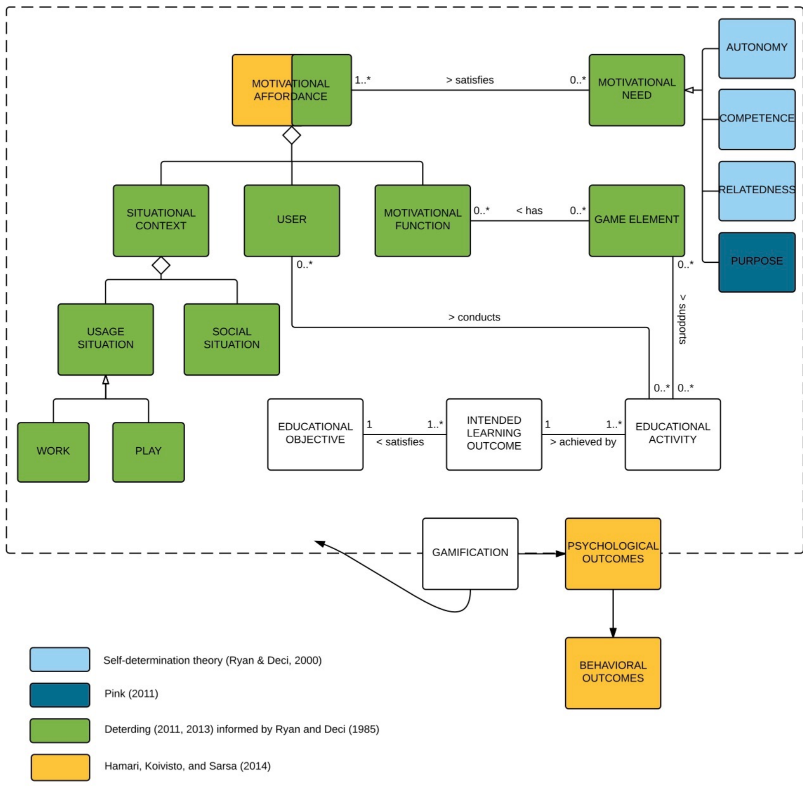


Figure 16 - Meta-model of gamification and its relation to learning

3.4.5 Motivational Affordances and Game Design Elements

After having reviewed the gamification landscape and decided to follow an approach largely based on game design elements, this section gives an extended account of the different elements that are used to gamify an experience. We take a broad focus by first examining the different types of game design elements reported in the literature, and then converge on a specific set of elements that are of interest to this project.

As mentioned in **Section 3.4.2**, the classical approach to gamification consists of using points, badges, and leaderboards. These components are pervasively used because they are relatively easy to implement and due to their apparent (short-term) effectiveness. Aside from these, a myriad of game (design) elements can be found in literature.

In a review of empirical studies, Hamari, Koivisto, and Sarsa (2014) identified the following elements: PBL, story/theme, clear goals, feedback, rewards, progress, and challenge.

In the context of education and learning, Nah, Zeng, & Telaprolu (2014) distilled eight game design elements: PBL, levels, prizes and rewards, progress bars, storyline, and feedback. In their literature review, they observed that these elements have an impact on the engagement, enjoyment, and motivation of learners. Similarly, González and Area (2014) arrive at a similar proposal of game elements: collection, points, leaderboards, levels, status, feedback, achievements, and epic meaning. More recently, the systematic mapping study conducted by Dicheva, Dichev, Agre, and Angelova (2015) pinpointed the typical game elements found in gamified educational applications: points, progress bars, virtual goods, badges, leaderboards, levels, avatars, and countdown clock.

The pitfall of these proposals is that they heavily rely on elements of an extrinsic nature. To find elements that appeal on an intrinsic level, we have to turn our attention to industry. Gamification experts such as Chou (2015) have identified a plethora of game techniques. While some of them resemble those identified in literature (e.g., PBL), others are situated at a lower abstraction level and thus are more nuanced. More interesting is that these techniques are classified into what Chou identifies as behavioral core drives. In his Octalysis gamification framework, he lays out how the different techniques tap into eight specific human drives (see Figure 17). The following table depicts the drives along with example situations/behaviors that satisfy them.

Table 6 - Description of behavioral drives and examples

Drive	Description	Examples
Epic meaning & Calling	In play when a person believes he is doing greater than himself or was chosen to take an action	Contributing to Wikileaks
Development & Accomplishment	Need for making progress, developing skills, achieving mastery, and overcoming challenges	Receiving money for good performance
Empowerment of creativity & Feedback	Expressed when people are engaged in a creative process where they repeatedly figure new things out and try different combinations (i.e., "play"), while receiving feedback	Playing Legos
Ownership & Possession	Motivation arises because people feel like they own or control something	Collecting stamps
Social influence & Relatedness	Incorporates social elements that motivate people (e.g., competition, cooperation, mentorship, social acceptance, social feedback)	Abiding to the norms established by a group
Unpredictability & Curiosity	Drive of being engaged because the immediate future cannot be predicted	Gambling
Loss & Avoidance	Motivation to avoid something negative from happening	Buying a limited and exclusive one-time promotion
Scarcity & Impatience	Drive of wanting something because it is rare, exclusive, or hard to attain (see Cialdini's scarcity principle, 2006)	Designers platform Dribble uses scarcity to encourage high-quality submissions

These drives compel us to behave in a certain way, and also are of different nature. Specifically, the right side of the octagon consists of drives closely associated with creativity, self-expression, and social dynamics. They tap into intrinsic motivation, as the activity itself is rewarding on its own. Conversely, those placed in the left side of the octagon are commonly associated with logic, calculation, and ownership. Moreover, they rely on extrinsic motivation. i.e., people feel motivated because they want to obtain something in return (e.g., a goal or reward). Viewed from another perspective, a predominant use of three drives found at the top of the framework is referred to as "White Hat" gamification, because they are associated with positive emotions. Conversely, a system that mostly relies on unpredictability, avoidance, and scarcity is an example of "Black Hat" gamification, because these tend to elicit negative feelings (Chou, 2015).

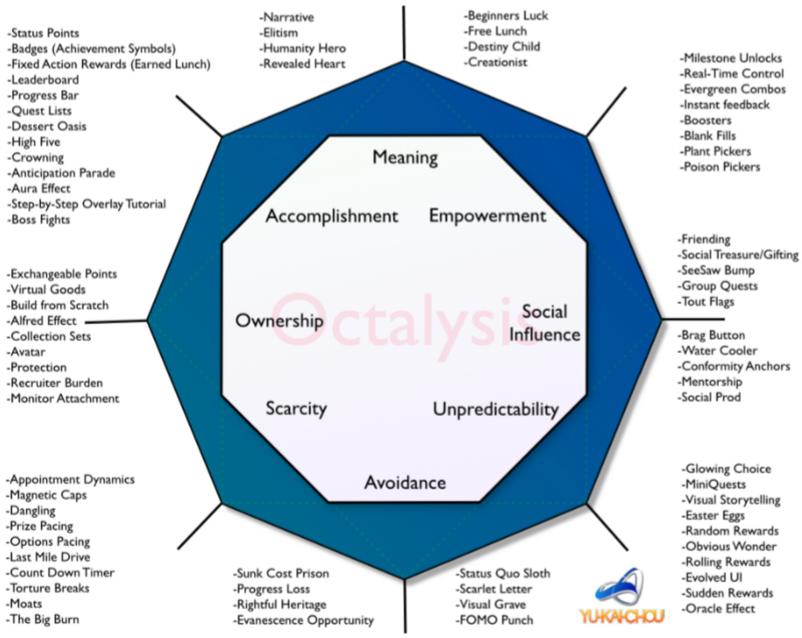


Figure 17 - Octalysis Framework. Extracted from Chou (2015)

The game elements proposed by academics and industry experts that are of interest to this research are captured in Table 7. For each element, a description along with the theoretical underpinnings and expected motivational affordances is also provided.

Table 7 - Overview of game elements

Game element	Description	Theoretical underpinning	Expected (motivational) affordances	References
Point system	Points are granted to reward users upon task completion	SDT, self-efficacy, goal setting, Personal Investment Theory (PIT), social comparison	Feedback, competition, sense of achievement, and positive emotions upon task completion	(Kapp, 2012; Nah et al., 2014; González & Área, 2014; Chou, 2015)

Game element	Description	Theoretical underpinning	Expected (motivational) affordances	References
Badge	Virtual goods with a possibly virtual representation awarded upon the completion of certain activities	SDT, self-efficacy, PIT, goal setting, PIT, social comparison	Used for goal-setting, instruction, reputation, and group identification	(Kapp, 2012; Antin & Churchill, 2011; Nah et al., 2014; González & Área, 2014)
Leaderboard	Ranking of users based on specific criteria	SDT, self-efficacy, PIT, goal setting, social comparison	Foster competition and increase recognition	(Kapp, 2012; Nah et al., 2014; Chou, 2015)
Level/Mission	Used to keep game space manageable and give a sense of progression	SDT, self-efficacy, goal-setting, PIT	Reward, status, competition, achievements	(Kapp, 2012; Nah et al., 2014; González & Área, 2014)
Progress bar	Used to track overall goal progression within a certain context	SDT, self-efficacy	Feedback, achievement	(Nah et al., 2014; Chou, 2015)
Story/ Narrative	Used to add meaning, provide context, and guide action. Comprises the elements of characters, plot, tension, and resolution	SDT, self-efficacy	Foster learning, immersion, attitude change	(Kapp, 2012; Nah et al., 2014; González & Área, 2014; De Schutter & Abeele, 2014; Chou, 2015)
Avatar	Graphical representation of a user profile that users can customize to their preferences	PIT, social comparison	Encourage self-expression, ownership	(Muntean, 2011; Nah et al., 2014)
Easter egg	Surprises hidden within a system in response to some undocumented user's command	-	Can increase excitement and encourage creative play	(Kapp, 2012; Chou, 2015)
Time restriction	Countdown clock used during the completion of a specific task	-	Add a sense of urgency to the task at hand and motivate action	(Kapp, 2012; Dicheva et al., 2015)

Game element	Description	Theoretical underpinning	Expected (motivational) affordances	References
Onboarding	Used for introducing users to the system	self-efficacy	Increase odds of buy-in	(Nah et al., 2014; De Schutter & Abeele, 2014; Chou, 2015)
Verbal/visual/ sound effects	Used to tell users about the state of the system, possibly as a result of their actions	SDT, self-efficacy	Feedback	(Li et al., 2012)
Animated feedback	Animated visual aid used to help users locate something in a system	SDT, self-efficacy	Feedback	(Li et al., 2012)
Conformity anchor	The system informs the user how close a user is to the social norm	SDT, self-efficacy, normative social influence	Feedback, social	(Chou, 2015)
Social prod	Action of minimal effort to create a social interaction (e.g., "Like")	SDT, self-efficacy	Social	(Chou, 2015)
Game master	Observes and orchestrates the learning process. Assists or influences the behavior of learners	SDT, self-efficacy	Feedback, social	(Tychsen, Hitchens, Brolund, & Kavakli 2005; Wendel, Göbel, & Steinmetz, 2012)
Chat room	Live chat where users can interact with each other	SDT	Social	(Muntean, 2011)
Activity feed	Used for displaying what activities and rewards users have completed in the system	SDT	Foster a sense of community and lead to feelings of recognition	(Muntean, 2011)

Several studies have argued that the inclusion of these game design elements can foster overall engagement (Kapp, 2012; Nah et al., 2014; Chou, 2015; Dicheva et al., 2015), besides of the specific motivational affordances that each element affords. From a higher level of abstraction, we can also distinguish gamification design principles (Deterding, 2011). The next section sheds light on guidelines to consider during the design of gamified educational applications.

3.4.6 Best Practices for Gamifying Learning Experiences

Apart from the game design elements, the literature review on gamified education helped distill several guidelines and best practices that can be used to inform the design of our gamified interactive tutorial (captured in Table 8). To conclude this section, below we provide a brief overview of relevant studies.

At a higher level, Kapp (2012) distinguishes between story, characters, recognition, chance, replayability, time, aesthetics, and continual feedback.

Muntean (2011) examined the possibilities of gamifying e-learning and distilled a series of guidelines. The following are relevant in the context of our study: 1) the educational material should be divided in small pieces of coherent content (i.e., cascading information theory), and each piece should be followed by an evaluation step (e.g., quizzes), 2) a system should be made as social as possible, 3) prosocial behavior should be rewarded, 4) learners should receive bonuses for accomplishing difficult asks, and 5) feedback must be constantly offered to inform learners' progression.

O'Donovan, Gain, and Marais (2013) gamified a university course using an online management tool with significant success in terms of course marks, lecturer appraisal, lecture attendance. They observed that including quizzes had the greatest impact on learning. Further, they emphasize that: 1) in case of having a storyline, this should be integrated with the assignments to provide meaningful context for learners, 2) the system should be social (e.g., using *guilds* to promote teamwork), and 3) rewards should be given for helping out in the community to increase belongingness and support.

De Schutter and Abeele (2014) designed and evaluated an online gamified application for undergraduate students named Gradequest. Based on their experiences, they posit that it is important to: 1) know the target audience, 2) provide freedom of choice, 3) use real-world names for challenges, 4) beware of the physical location, and 5) communicate that even if it is a game-like system, the tasks will be challenging.

In academia, Dicheva et al. (2015) found that the most widely used principles in scholarly publications are: visual status, social engagement (i.e., competition), freedom of choice, freedom to fail, and rapid feedback. Other principles include the use of goals, personalization, and unlocking content.

Table 8 - Guidelines for gamifying learning experiences

Guideline	Description	References
Freedom of choice	Allow for freedom of choice. E.g., learners should be able to choose the order or speed of the challenges to be completed, or what goals to pursue. At a minimum, give the *feeling* of freedom	(O'Donovan et al., 2013; De Schutter & Abeele, 2014; Dicheva et al., 2015; Chou, 2015)
Freedom to fail	Adopt the freedom to fail principle: poor task performance should not incur in penalties. E.g., learners should be able to retake quizzes	(O'Donovan et al., 2013; Dicheva et al., 2015)
Baby steps	Divide and present the educational content in small pieces of coherent information	(Muntean, 2011)
I know this!	Conduct an evaluation step (e.g., exercises, quizzes) after presenting educational content	(Muntean, 2011)
Together is better	Make the system as social as possible to motivate students through peer pressure or comparison with other students	(Muntean, 2011; O'Donovan et al., 2013)
Boost it!	Give special bonuses after learners complete difficult tasks	(Muntean, 2011; De Schutter & Abeele, 2014)
The good Samaritan	Compensate students not only for academic achievement but also prosocial behavior	(Muntean, 2011; O'Donovan et al., 2013)
Crystal clear	Offer immediate feedback and inform learners of their progression within the tutorial (e.g., progression bars. Frequent and immediate feedback leads to greater learning effectiveness and engagement	(Muntean, 2011; Kapp, 2012; Dicheva et al., 2015)
Ethical designer	Design with ethics in mind: e.g., full transparency and opt-in principles	(Deterding, 2014; Chou, 2015)
Know your users	Target population must be studied. This always holds, regardless of whether gamification is considered or not	(De Schutter & Abeele, 2014)
The road ain't easy	Communicate that the training will be challenging	(De Schutter & Abeele, 2014)
Location, location, location	The location in which users engage with the application matters	(De Schutter & Abeele, 2014)

4. Conceptual Framework

After having thoroughly studied the literature on gamification (of learning), the following chapter establishes the conceptual framework that has guided the design and evaluation of the educational and gamified component of the STS-Tooltorial.

Thus far, the relationship between gamification and conceptual modeling has remained largely unexplored. Our literature review did not uncover any studies measuring the relationship between these two concepts. However, as **Section 3.4** revealed, the relationship between gamification and engagement, and its subsequent effect on task performance, has garnered the interest of many scholars, who have undergone significant empirical work.

With this in mind, the following sections build upon some of those studies to provide a theoretical basis for our research model. In particular, **Section 4.1** introduces the concepts of interest and elaborates upon the relationships and hypotheses among them. Subsequently, **Section 4.2** introduces the conceptual model that is used as a basis for the experimental evaluation of the STS-Tooltorial.

4.1 Relationships and Hypotheses

4.1.1 The Gamified Teaching Trifecta: Gamification, Engagement, and Task Performance

The literature review on gamification has shown that it can be used to motivate and engage people. Moreover, universities and teaching institutions are in need of finding novel approaches for motivating students, as manifested by the myriad of academic venues that address this topic. Given that the use of gamification as a didactic teaching method remains a fairly unexplored territory (at least, in the context of information and computing sciences), it seems reasonable to contribute to the research gap by exploring its use to increase student engagement and motivation. Therefore, the following hypothesis is formulated:

> **H1**: *Learners exposed to a gamified interactive tutorial will report high levels of engagement.*

However, the benefits of engagement also ought to be considered. Relevant for this research is its effect on task performance. Engagement can take many forms. A well-known definition in the context of the workplace is the one by Schaufeli, Salanova, Gónzalez-Romá, and Bakker (2002). They define engagement as "a positive, fulfilling, work-related state of mind that is characterized by vigor, dedication, and absorption" (p.74). Vigor is characterized by high levels of energy, the willingness to put effort into the task at hand, and persistence even when problems arise. Dedication is characterized by a strong involvement in the job at hand, accompanied by enthusiasm, significance, inspiration, and pride. Finally, Schaufeli et al. (2002) define absorption as being fully immersed in one's work. Absorption (a component of engagement) is closely associated to the concept of flow (Csikszentmihalyi, 1990). Flow refers to a state of optimal experience that is characterized by focused attention, clear mind, mind and body unison, effortless concentration, loss of self-consciousness, complete control, distortion of time, and intrinsic enjoyment. Flow is seen as a motivating force for excellence, and should therefore foster performance. In light of this, one should expect that engaged learners perform better than their counterparts.

Indeed, there are several studies that point at the relationship between the construct of engagement and performance. In their meta-analysis, Harter, Schmidt, and Hayes (2002) found that employee engagement and satisfaction were associated with business outcomes. This finding was validated in a follow-up study (Harter, Schmidt, Killham, & Agrawal, 2013). In learning environments, student engagement has also been positively linked with desirable learning outcomes such as grades or critical thinking (Carini, Kuh, & Klein, 2006). By combining these theoretical findings, the following hypothesis is formulated and expected to hold:

> **H2**: *The higher the engagement of learning during their participation in the gamified tutorial, the higher their performance will be.*

4.1.2 Acceptance

The last area of inquiry refers to technology acceptance. If we take the broad understanding of technology as knowledge (Layton, 1974), technology can either refer to the (gamified) interactive tutorial or the STS Method itself (and STS-Tool by extension).

A well-established model that explains and serves to evaluate the acceptance of a technology is the Technology Acceptance Model (TAM) (Venkatesh & Davis, 2000). Rather than considering all the constructs proposed by the TAM, we solely focus on one, to wit: behavioral intention to use. In this context, two concepts are of interest. The first refers to the intention to use the system itself (i.e., the STS-Tool). More importantly, we are interested in the learners' behavioral intention to learn more about

SRE and the STS Method as a result of being exposed to the gamified tutorial. For if we can argue that the tutorial sparks the interest of learners in the subject of socio-technical security requirements modeling, then the intervention should be regarded as successful. Therefore, we hypothesize the following:

- **H3**: *Learners exposed to a gamified interactive tutorial will report a positive satisfaction towards the use of the system.*
- **H4**: *Deploying a gamified interactive tutorial positively influences intention to learn more about SRE and the STS Method.*

4.2 Conceptual Model

Based on the concepts, relationships, and hypotheses set forth in the previous section, Figure 18 depicts the conceptual model used to scope our research and, ultimately, empirically evaluate the effectiveness of the STS-Tooltorial.

However, as we report in **Chapter 6**, the evaluation is conducted in a quasi-experimental setting, which means that the hypothesis represent more of a guidance than actual hypothesis testing. As we will see, it is the qualitative results that add the most value to this research.

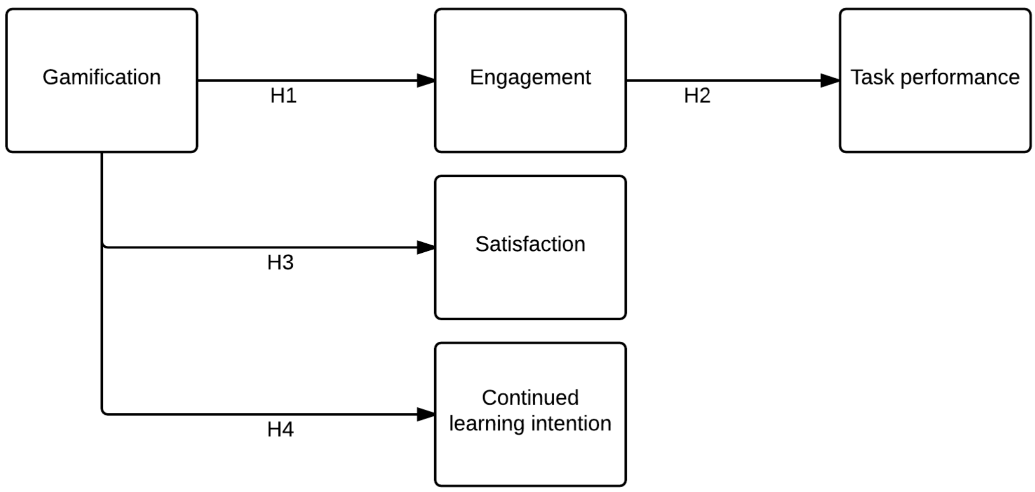


Figure 18 - Conceptual Model

III. Towards a Solution

After rising with greater understanding of the path forward, we attempt to make a dent in the world by constructing the STS-Tooltorial, a gamified interactive platform that helps novice learners absorb basic notions about Security Requirements Engineering, the STS Method, and its modelling language. We rely on the advice of SRE experts and our curated set of best practices to orchestrate a framework for creating this type of interactive tutorials. The creation of the platform is followed by a two-country, two-group evaluation in the hope of establishing its validity as an effective and innovative teaching method.

5. Design of the Artifact: STS-Tooltorial

To answer our research questions a gamified Web-based platform for learning about security requirements engineering was devised. The choice of opting for an online platform stems from the fact that it is both an innovative way of delivering education (e.g., MOOCs) and a scalable means of reaching a wide audience that would not be possible in a traditional classroom setting. While gamification can also be employed in an offline setting and we acknowledge the importance of instructors and classrooms, our review of literature showed that the use of digital technologies in the classroom warrants more investigation. Another main driver was that given the interactive tutorial would be embedded within the STS tool, we believed it would bring new ways of experiencing learning.

As mentioned in **Chapter 2**, the design STS-Tooltorial was inspired by a simplified version of the goal-directed design process (Cooper et al., 2014), a well-proven method for orchestrating the design of digital products. This method advocates a design philosophy based on a thorough understanding of the goals a person might have for using a product, and provides guidelines for devising a desirable solution. Table 9 shows a summary of the activities conducted and the main deliverables, which we elaborate upon in the following sections.

Table 9 - Overview of the artifact design methodology

Phase	Activity	Work products
Investigation	**Research**: As reported in **Chapter 2**, a combination of desktop research and semi-structured interviews with SMEs was used to inform the design of the interactive tutorial and the needs of potential users	**Section 5.1** reports the insights not introduced hitherto that informed the design of the tutorial
	Modeling: Interview results and characteristics of representative users were aggregated to create archetypical user models (i.e., personas for two primary users: Information Science/Computer Science students and IT/security professionals)	**Section 5.2** describes the proto-personas that guided the requirements definition process. Note that the accuracy of personas was not validated with representative users due to time restrictions
	Workflow: Models of the sequence through which users would potentially interact with such a platform were devised	Paper prototypes of major interaction patterns were created. They are not reproduced here.
	Requirements definition: Functional needs for each persona were captured in the form of user stories	**Section 5.2** presents the user stories (and prioritization method) that guided the development of the STS-Tooltorial

Framework	**Framework definition**: Data and functional elements for the interactive tutorial. Here we specifically focus on the design of the instructional material	**Section 5.1** describes the design of the instructional content following the best practices uncovered in our research
Design	**Architecture**: Technical specifications of the gamified platform are outlined	**Section 5.3.1** reports on the technical aspects of the platform embedded within the STS Tool. A specific section is devoted explaining the inner workings of the real-time interaction capabilities due to the novelty of the approach
	User interface: Once functional and educational elements were defined, the visual means of organizing and presenting them was defined	**Section 5.3.2** introduces the design of the user interface together with a functional walkthrough
	Game elements: We selected a subset of the game elements and guidelines for gamifying learning experiences to develop into the STS-Tooltorial	After introducing game elements in the functional walkthrough, **Section 5.4** presents how we operationalized the term "gamification", presenting an overview of the game elements that were finally adopted
Development	**Coding**: Following a highly adaptive and agile approach, development work began in parallel to the creation of the specifications	Source code is provided online as a by-product of the thesis for the benefit of Utrecht University / University of Trento but not reproduced here.

5.1 Curriculum and Instructional Design

This section elaborates upon instructional design of the STS-Tooltorial, that is, its educational component. The design follows a student-centered approach (Kennedy, Hyland, & Ryan, 2006) based on two pillars: intended learning outcomes and educational content.

Rather than only focusing on the topics to be covered by the tutorial, we first drew on learning outcome theory to define curriculum objectives. Intended learning outcomes (ILOs) are statements of what a learner is expected to know or be able to do at the end of a learning experience (Kennedy et al., 2006). ILOs help base curriculum design, content, and assessment on an analysis of the knowledge, skills, and values needed by students and society (McPhail, 2005, p. 72). The use of learning outcomes is compulsory in countries within the European Higher Education Area. Also, it has found wide applicability in the US (Kuh, Jankowski, Ikenberry, & Kinzie, 2014). Before defining the learning outcomes for the STS-Tooltorial, this section reviews relevant literature that aids in defining learning outcomes. We focus on learning taxonomies and, in particular, Bloom's taxonomy, which is the highest cited taxonomy.

After investigating how people come to learn, a group of researchers headed by Bloom constructed a taxonomy of intellectual behavior and learning based on three pillars: the cognitive, affective, and psychomotor domains (Bloom, Engelhart, Furst, Hill, & Krathwohl, 1956).

The cognitive domain deals with knowledge and the development of intellectual skills. It consists of six processes of varying complexity. Table 10 displays them in order of increasing complexity. This means that, for instance, recalling a fact (i.e., remember) is less complex than executing a process following a manual (i.e., apply), which in turn is less complex than writing an essay that compares Heidebberg's concept of oneness (i.e., evaluate) to that of Buddhism (i.e., create).

Table 10 - Processes in the cognitive domain. Adapted from Krathwohl (2002), which contains a revised version of Bloom's taxonomy.

Cognitive process	Description	Associated verb
Remember	Retrieving or recalling knowledge from memory	define, describe, list, enumerate, recognize, describe, reproduce
Understand	Determining the meaning of instructional message	interpret, exemplify, classify, summarize, infer, compare
Apply	Carrying out or using a procedure in a given situation	execute, implement, apply, compute, construct, discover, operate
Analyze	Breaking material into constituent parts and detecting how the parts relate to one another and to an overall structure	differentiate, organize, attribute, compare, contrast, discriminate
Evaluate	Making judgements based on criteria and standards	appraise, compare, conclude, check, critique
Create	Putting elements together to form a novel, coherent whole or make an original product	generate, plan, produce, revise, design, categorize

Secondly, the affective domain addresses how people deal with things emotionally, including feelings, values, motivation, and attitudes. Table 11 captures the list of affective processes.

Table 11 - Processes in the affective domain. Adapted from Krathwohl, Bloom, and Masia (1964)

Affective process	Description	Associated verb
Receiving	Being aware of or sensitive to the existence of certain ideas, material, or phenomena and being willing to tolerate them	differentiate, accept, listen, respond, ask, follow, accept
Responding	Exhibit new behavior as a result of experience. Active participation on the part of the learners	comply, follow, answer, aid, enjoy
Valuing	Willing to be perceived by others as valuing certain ideas. Show definite involvement or commitment	share, support, debate, value, appreciate, express
Organization	Refers to relating the value to those already held and bring it into a harmonious and internally consistent philosophy	compare, relate, examine, formulate, choose, consider, prefer
Characterization	Act consistently in accordance with the values the student has internalized.	revise, resist, manage, act on, exemplify

Finally, the psychomotor domain concerns the use of basic motor skills, physical movement, and coordination. Nowadays, psychomotor has been expanded to include a broad range of skills (e.g., social, problem-solving, communication, critical thinking). Although Bloom did not propose a taxonomy for this domain, other authors such as Dave (1970), Harrow (1972), and Simpson (1972), independently proposed their versions. Skills are learned through repeated practice, and its development is based on aspects such as precision, speed, technique, and distance (Clark, 2015). Owing to its wide applicability, Dave's (1970) taxonomy is the one reproduced in Table 12.

Table 12 - Processes in the psychomotor domain

Psychom. process	Description	Associated verb
Imitation	Observing and patterning behavior after someone else.	copy, follow, replicate, repeat
Manipulation	Being able to perform certain actions by memory or following instructions.	act, build, perform
Precision	Refining, becoming more exact. Performing a skill within a high degree of precision	master, perfect, calibrate
Articulation	Coordinating and adapting a series of actions to achieve harmony and internal consistency.	adapt, combine, construct, modify
Naturalization	Mastering a high level performance until it become second-nature, without needing to think much	create, design, invest

Overall, the taxonomy of learning behaviors can be regarded as "the goals of the learning process" (Bloom, 1956, p. 84). In other words, it helps in creating intended learning outcomes. Moreover, Bloom's taxonomy has been successfully for structuring exercises in computer-based instruction (e.g., Hernán, Lázaro, & Valázquez, 2004), of which the STS-Tooltorial is an application.

There are several guidelines for describing learning outcomes. Such guidelines can be readily found in higher education institution websites as well as academic literature (e.g., Kennedy et al., 2006). In general, there is consensus that ILOs are stated in a standardized manner, using a format similar to "By the end of this learning experience, students should/will be able to…". As a rule of thumb, a learning experience should typically contain less than six outcomes (MIT, 2016). Moreover, an ILO should be specific, attainable, and measurable (for assessment purposes). These were the criteria used to elicit possible ILOs for our interactive tutorial in our interviews with SRE experts.

5.1.1 Intended Learning Outcomes for the STS-Tooltorial

To extract educational objectives, we supplemented desktop research with semi-structured interviews with seven leading SRE experts with a teaching background to identify a list of intended learning outcomes for a hypothetical two-hour classroom based tutorial covering concepts of SRE and conceptual modeling (see **Section 2.1**). We specifically opted for this fictitious scenario because participants were familiar with such a setting and thus better able to provide us with insights that we could then extrapolate to an interactive tutorial. Participants answers were aggregated and clustered into mutually exclusive domains and subsequently refined until we arrived at the following list of learning outcomes:

After completing the tutorial, learners should be able to:

- **ILO 1**: Understand the different activities and deliverables of security requirements engineering
- **ILO 2**: Recognize the modeling rules and elements as well as their meaning (in an existing model)
- **ILO 3**: Choose the most appropriate set of elements for representing security-related aspects.
- **ILO 4**: Express interest in pursuing further learning in the field of (socio-technical) SRE.

Taken from a broad perspective, understanding the STS Method and language contributes to an overarching learning outcome of learners being able to effectively study organizations using modeling techniques and frameworks (Dalpiaz et al. 2016). Specifically, STS allows security practitioners and stakeholders to study the security aspects of socio-technical systems. On a related note, while we restrict our discussion to STS, it is also important for students to acquire a reflective understanding of the various modeling paradigms, their strengths and limitations, as well as those contexts in which they are most applicable (Babar, Nalchigar, Lessard, Horkoff, & Yu, 2015). This means that the tutorial should be able to transmit general concepts about (socio-technical) security requirements engineering and modeling, with STS being one of many approaches (cf. the different techniques and methods presented in **Section 3.1**).

We contend that, by successfully completing the gamified tutorial, students should be able to have a general understanding of the STS Method and a notion of what it takes to produce socio-technical security requirements models using the fundamental constructs of STS-ml (as measured by the performance construct in Figure 18). Apart from transmitting knowledge, the tutorial also aims at changing students' dispositions and attitudes toward learning more about STS and socio-technical systems design (**H4** in Figure 18). Therefore, the most relevant dimensions for the tutorial are the cognitive and affective domains.

Table 13 below matches each ILO with 1) the desirable competencies that requirements engineers should have as proposed by Maculay and Mylopoulos (1995) and 2) the cognitive/affective taxonomies proposed by Krathwohl (2002) and Krathwohl et al. (1964).

Table 13 - Mapping of ILOs with learning level taxonomies and core competencies of requirement engineers

ILO	Competency	Dimension/level
ILO 1	Method acumen	Understand
ILO 2	Modeling skills and abstraction skills	Remember
ILO 3	Modeling and analytical skills	Understand / Create (depending on the situation)
ILO 4	-	Affective (Receiving, Valuing, & Organization)

5.1.2 Best Practices and Heuristics for Designing Instructional Content about Conceptual Modeling Within (S)RE

The interviews also provided useful information that helped to complement the general guidelines on instructional design presented in **Section 3.3**. In particular, we distilled best practices and heuristics that academics use to teach SRE and modeling regardless of the delivery method. In brackets we trace the statements back to the interviewee codes as introduced in Table 2. The protocol in **Appendix II** reports how respondents were probed to answer these questions.

- **S1**: Modeling is an iterative process: models are constructed and iteratively refined [1, 2, 3, 5].
- **S2**: It is important to consider the attacker perspective [4, 5].
- **S3**: It is key to stress economic impact of security issues to raise awareness [2, 4, 5].
- **S4**: Distinguishing between document and information is important in socio-technical systems design [1].
- **S5**: Security is a multi-level concept (physical, network, and social). The social aspect is too often under-researched, under-focused, and under-prioritized [3, 6].
- **S6**: Teach through exercises, using a hands-on case-based approach [1, 2].
- **S7**: Include a hands-on task only if the learner has previous experience [3, 5].
- **S8**: Include a reflection phase after teaching [2].
- **S9**: Give feedback after completing a modeling task [2, 3].
- **S10**: Have students review each other's work through peer inspection [2, 5].
- **S11**: It is important to capture the rationale of models [2, 6, 7].
- **S12**: SRE should not be seen as a separate activity. The task of analyzing a system and creating a good set of security requirements should be done while conducting other software development activities [5, 7].

5.2 Persona Design and Platform Requirements

Based on the guidelines identified in the previous section we then defined the characteristics and behaviors of potential users through a technique called Personas (Cooper et al., 2014). Personas represent the "voice of the user" and can be used to guide the design of the interface and product requirements.

Given the novelty of the STS Method and the interestingness of (S)RE, we considered important to reach a varied audience comprising of both students and professionals. The characteristics, needs, and behaviors of both Computer Science (CS) and IS students and IT professionals were modeled into two proto-personas, underpinned by secondary research, the author's own knowledge, preconceptions and assumptions as member of both groups, and SME best practices reported in the previous section. The resulting persona models are depicted in Figure 19.

Figure 19 - Primary personas used to guide the design of the STS-Tooltorial

After defining the personas, we defined a set of requirements for the interactive tutorial from the perspective of the potential users of the platform to guide development work. First, persona goals and ideal experiences were translated into high-level features by posing questions such as: "*does feature X tap into the needs of the persona?; would feature X align with the persona goals?*". For completeness purposes, requirements were also devised for secondary users such as instructors and researchers, both roles assumed by the author.

Table 14 below captures the platform requirements in the form of user stories. It is worth mentioning that the user stories were written following the INVEST technique and checked against the quality guidelines set forth by Lucassen, Dalpiaz, van der Werf and Brinkkemper (2015).

The prioritization method used was the cost/value approach (Karlsson & Ryan, 1997), instantiated by combining effort/complexity as measured by story points, with business value following the MoSCoW technique (Cline, 2015). The product backlog to be consumed was then defined as the user stories that could be fitted within a 4.5-month development timeframe. There are two observations worth noting. First, known velocity estimates of the author were used for the calculation. Second, there were no requirements left out of development, which implies that the platform is as complete as it ideally was conceived.

Table 14 - Platform requirements for the STS-Tooltorial

As a	I want to be able to	so that I can	Cost/Value
learner (John/Silvia)	enroll in courses/tutorials	learn more about SRE and the STS Method	Value: Must Cost: 2 SP
learner (John/Silvia)	interact with the STS tool as I learn about the STS Method and modeling language	have a more engaging experience	Value: Must Cost: 40 SP
learner (John/Silvia)	enjoy bite-sized lessons featuring multimedia content	stay more engaged and digest content easily	Value: Must Cost: 100 SP
learner (John/Silvia)	pick up lessons where I left them	easily return to the course whenever I want	Value: Must Cost: 5 SP
learner (John/Silvia)	create and provide the rationale of models iteratively as I progress through the different lessons	understand that modeling is an iterative process	Value: Must Cost: 40 SP
learner (John/Silvia)	post comments in lectures	express my thoughts	Value: Must Cost: 2 SP
learner (John/Silvia)	chat with SRE experts	have my questions solved	Value: Must Cost: 13 SP
learner (John/Silvia)	check my progress	feel a sense of progress	Value: Must Cost: 3 SP

learner (John/Silvia)	compare myself against my colleagues	see how well I am performing	Value: Must Cost: 5 SP
learner (John/Silvia)	earn achievements	stay motivated and feel reinforced	Value: Must Cost: 40 SP
learner (John/Silvia)	solve quizzes regarding the lessons	reinforce my recently-acquired knowledge	Value: Must Cost: 40 SP
learner (John/Silvia)	receive rapid feedback to my answers	check if I am on the right track	Value: Must Cost: 100 SP
learner (John/Silvia)	retry quizzes	achieve maximum scores	Value: Must Cost: 13 SP
learner (John/Silvia)	receive points based on my performance	feel accomplished and incentivized	Value: Must Cost: 40 SP
learner (John/Silvia)	redeem points for rewards	earn boosters and thus feel more engaged	Value: Must Cost: 3 SP
learner (John/Silvia)	feel increasingly challenged	maintain a state of flow	Value: Must Cost: 13 SP
learner (John/Silvia)	get real time notifications	be informed about the activity of other members as well as recent events	Value: Should Cost: 40 SP
learner (John/Silvia)	take part in groups/communities	relate and communicate with like minded people	Value: Could Cost: 3 SP
instructor	see who enrolled in my course	perform Learner Management	Value: Must Cost: 2 SP
instructor	provide contextual feedback to learners when they answer quizzes	show students whether they failed and why	Value: Must Cost: 5 SP
instructor	provide learners with encouraging messages (e.g., after they answer a quiz correctly/incorrectly)	maintain students engaged and offer positive reinforcement	Value: Must Cost: 5 SP
instructor	view learning analytics	check how well learners perform	Value: Must Cost: 5 SP
instructor	grade learner answers	manually grade those quizzes that cannot be automatically graded	Value: Could Cost: 5 SP
instructor	chat with learners	interact with them in a more direct way	Value: Should Cost: 13 SP

researcher	track user behavior in real time (video analytics, page visits, heatmaps)	analyze it for research purposes	Value: Must Cost: 40 SP
researcher	learners and instructors to have 24x7 access to the platform	interact with the platform whenever they want	Value: Must Cost: 2 SP

5.3 Platform Architecture and User Interface Design & Functional Walkthrough

5.3.1 Platform Architecture behind the STS-Tooltorial

As mentioned in the previous subsection, the STS-Tooltorial runs on a Wordpress platform layered on top of PHP, Apache Server and a MySQL database. The Bitnami Wordpress Stack was used due to its easy installation and out-of-the-box compatibility with Google Cloud, our hosting provider (see below). To allow for maximum reusability and time-to-deliver, we opted for re-using existing Wordpress plugins and third-party solutions that met the aforementioned requirements. For instance, the core of the platform leverages Sensei, the top Learning Management System plugin for Wordpress, albeit significantly customized to accommodate specific functionality such as quizzes that require interacting with a STS-Tool diagram. Other interesting plugins include Talkus.io for learner-instructor communication, and Hotjar for gathering analytics regarding user behavior (e.g., heatmaps, screen recording, and so on).

A non-functional requirement of the platform was for it to reach as many users as possible. Thus, the Wordpress platform and associated database are hosted on a Google Cloud instance located in Europe West for faster access to European users. The instance also provides load balancing and auto-scaling capabilities to handle extreme peaks of demand.

Furthermore, DevOps practices such as Continuous Integration and Continuous Deployment were adopted to make more efficient the deployment process. When the source code is committed in our Git repository, a series of Jenkins jobs are launched to check the quality of our software and trigger automatic deployment processes from the local development machine to Google Cloud.

The STS-Tooltorial platform can be accessed as-is from a browser, but for our purposes the tutorial was meant to be used in conjunction with the STS-Tool, the robust modeling tool associated to the STS Method. The tool is an Eclipse Rich Client Platform application (RCP) written in Java and distributed for Win, Mac OS X, and Linux. RCP applications are modular, meaning they can be easily extended to support additional features. We created a self-contained Eclipse RCP plugin that embeds a browser window inside the tool using Eclipse's Standard Widget Toolkit (SWT), pointing directly to the STS-Tooltorial website.

However, as depicted in Figure 20, the browser window is more than a mere display of web content. Since it supports HTML5 and JavaScript, it is possible to establish a bi-directional communication bridge between STS-Tooltorial (JavaScript code) and STS-Tool (Java code). This basic communication mechanisms allows for rich forms of interaction and a seamless user experience which is at the core of an "embedded tutorial". For instance, when learners are asked to solve quizzes that require refining a pre-existing diagram, the embedded tutorial instructs STS-Tool to open a previously stored diagram through a JavaScript call that is seamlessly parsed to Java code. What the learner perceives is that the diagram automatically pop-ups on the canvas, ready to be edited. In addition, once they submit the answer, a series of custom-made algorithms in the plugin validate the correctness of the modeling task and feed back the results to the STS-Tooltorial through a JavaScript call. In this way, the learner is provided with contextual, rapid feedback.

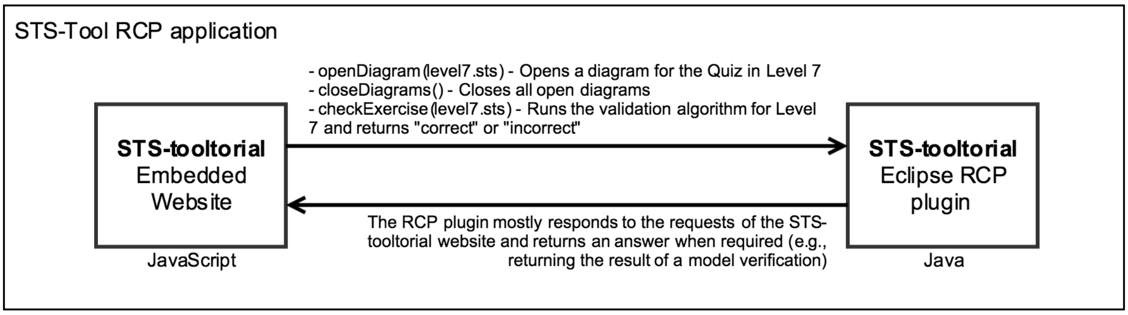


Figure 20 - Bi-directional communication architecture between the STS-Tooltorial and STS-Tool

5.3.2 User Interface Design and Functional Walkthrough

We already established in our literature review that it is important to present data and functionality in a simple and intuitive way. Minimalists and design experts such as Cooper both agree that well orchestrated user interfaces enable users' flow, that is, total concentration and loss of awareness of distractions and external stimuli (Csikszentmihalyi, 1990). Other purported benefits of a well-designed user interface include efficiency, ease of use, and improved usability to achieve goals or tasks (Cooper et al., 2014). To that end, we relied on the Social Learner theme by BuddyBoss (2017), one of the most highly acclaimed themes, touted for being aesthetically pleasing, usable, and with a clean responsive design.

The last point, responsiveness, was a key requirement to have. From the beginning of our research we acknowledged that real estate would be scant if we were to embed the tutorial within the STS-Tool. We estimated that the tutorial would have to occupy at most one-third (placed vertically) or one-half (placed horizontally) of the screen size to allow for a smooth user experience according to user's needs. In other words, the design had to follow best practices of small screen devices, thus resembling a mobile web app. Therefore, the number of elements present in the interface was kept to a minimum to help users focus on the task at hand. In addition, it helped to balance the amount of buttons and menu items present in the STS-Tool.

Based on a clean design, the STS-Tooltorial features a balance of blue and white calming hues aimed at evoking relaxation and contentment (see Figure 21). From left to right, the header prominently displayed the icon of the platform, a hamburger menu (to expand menu items), a search function and a photo of the learner linking to her user profile, a design pattern used by platforms such as Facebook. The header and left navigation are kept in a persistent mode so as to maintain context as users navigate through the different pages.

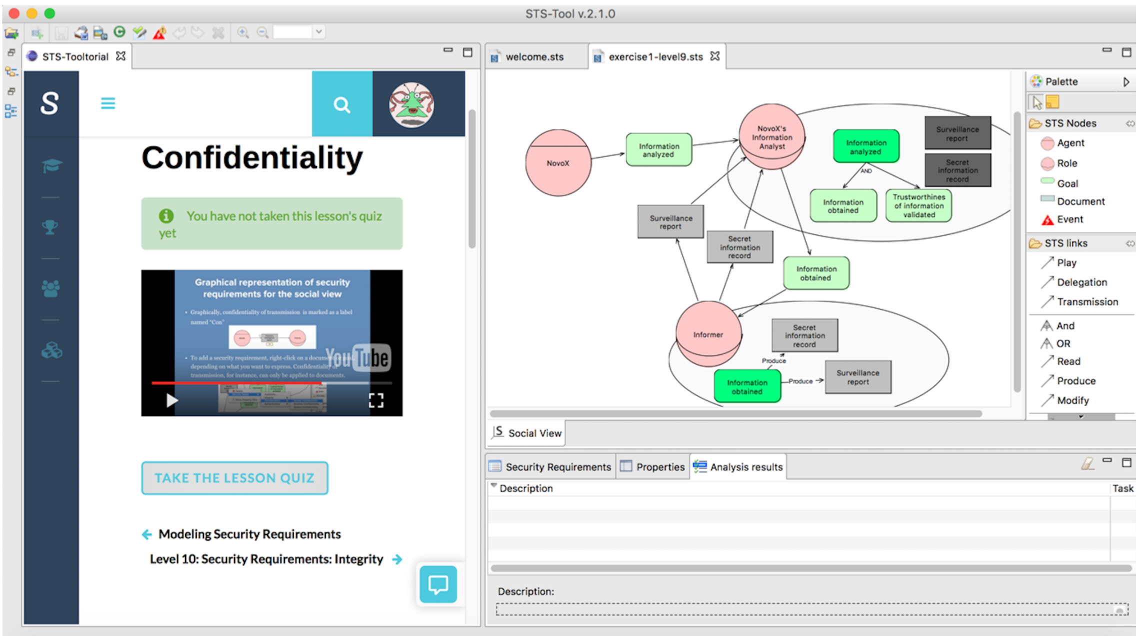


Figure 21 - Screenshot of the STS-Tooltorial (left) embedded in the STS-Tool

Rather than being at the top (due to space limitations), the primary navigation is located in the left-hand menu. The menu is by default collapsed unless the user desires to expand by mousing over them.

Courses and Lessons

The landing page for both visitors and registered learners contains an exhibit of the courses and tutorials available for enrollment. At the moment, learners can only enroll on the one we created as part of this research. After registering in the course, learners are presented with an overview of the different lessons, which they have to complete sequentially. For our purposes, each lesson (dubbed as level) follows the same pattern governed by three phases:

- **Learn phase**: the delivery method consists of bite-sized lessons (Levels) lasting between 1 and 5 minutes where learners are progressively introduced to SRE, socio-technical systems, and the STS Method and modeling language at increasing levels of difficulty to keep them in a state of flow. Currently, the tutorial contains three major modules: an introductory one about socio-technical systems and SRE in general, and two modules about the STS Method social view and security requirements. The educational material is presented in the form of animated videos featuring a slide deck supplemented by the instructor's voice/face and, optionally, supporting text (e.g., additional references). Observe that lesson content can be fully customized by instructors.
- **Play phase:** lessons contain Quizzes, which in turn may consist of multiple-choice, true/false, or what we refer to as "model interaction" questions. The latter feature provides learners with questions about the content they just learned that prompt them to interact with the modeling tool, drawing diagrams and getting quick feedback on their correctness (see Figure 22).
- **Reflect phase**: upon receiving contextual feedback, learners can investigate whether they completed the lesson successfully and uncover areas of improvement. They are allowed to retake quizzes to consolidate knowledge and achieve the maximum amount of points.

Observe that these phases are aligned with the pattern called "Time for action/Time for thought" proposed by Huynh-Kiml-Bang, Wisdom, and Labat (2010), which suggests separating practice moments from reflections phases when teaching high-level knowledge.

Figure 22 also shows that the bottom right hand side contains a blue rectangle widget based on the help desk technology Talkus.io (2017), which serves with a three-fold purpose:

- **Chat function**: upon clicking the rectangle, a right-hand half-screen pop-up appears and allows users to chat with the Game Master and instructors. The chat system is seamlessly integrated with Slack, from where the Game Master and instructors type their answers once learners have initiated a conversation.

- **Providing contextual feedback**, based on user actions (e.g., failing a quiz) or specific site locations. Instructors pre-define messages specifying what, when, and where should be sent. We used this feature to provide normative and positive reinforcement feedback to boost self-efficacy and intrinsic motivation. For instance, when a user passed a quiz with 100% accuracy we would send applicable message like "*Congratulations! You're on fire… you receive 100 extra points, 80% of other students failed to complete this quiz, you should feel proud*".
- **Browsing through the Rewards Marketplace**: At the top of the pop-up there is a section featuring the Rewards Marketplace, where learners can claim various rewards (see next section).

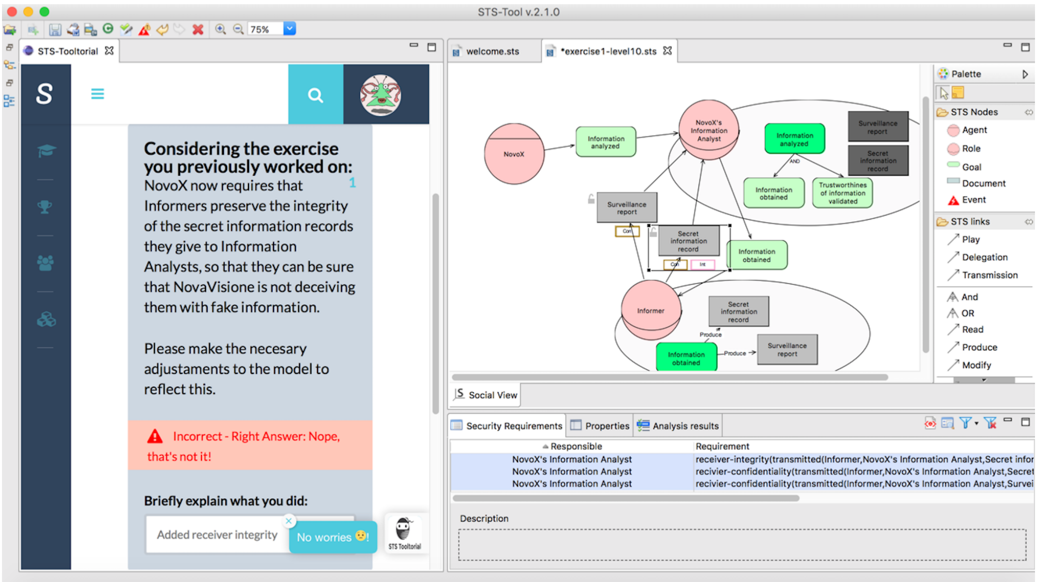


Figure 22 - Example of how contextual, instant feedback is given to the learner

5.4 Overview of Game Elements

Aside from **Levels** and **Quizzes**, the use of other game-like elements is pervasive throughout the STS-Tooltorial, as they support the different learning activities and tap into both intrinsic and extrinsic motivators to keep learners engaged.

5.4.1 Narrative grounded on Science Fiction

The tutorial makes use of science fiction to present the educational material (**Story/ Narrative**). In particular, the learner begins as a Rookie security requirements analyst on the payroll of NovoX, an Intelligence Agency, tasked with protecting the Empire's interests across the Galaxy through designing a secure socio-technical system (i.e., the running example used throughout **Section 3.1**).

After completing the first module on basics of socio-technical systems, the learner is on-boarded into the story in a dedicated lesson called "Scenario Introduction", which contains a Star Wars-esque video explanation. The educational content (i.e., slide decks, examples) and exercises thereafter are grounded on this fictitious story in an attempt to increase learner engagement.

At the risk of sounding stereotypical, in thinking of what narrative elements to include we referred to our proto-personas supposed characteristics as well as the author's preferences, which was at the time truly fascinated by Asimov's novels. As will be seen in the following chapters, the choice of narrative appears to be adequate.

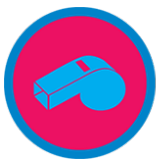
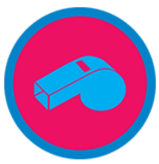
5.4.2 Achievements and Leaderboard

In designing achievements we opted for a non-cluttered, simple set of elements that could be used to communicate learner status and progression within the tutorial, classified across three achievement types: Levels (already explained), Points, Badges, and Ranks.

Simply stated, by successfully completing the educational content, the learner is awarded Points, which in turn determine their Rank and position in the Leaderboard. The learner can achieve up to five Ranks in their quest to become Protector of the Galaxy, a position awarded to a few distinguished officials. Additionally, the platform contains a couple of Badges to reward learners for basic actions such as visiting the course page for the first time or greeting the Rookie Analyst after completing the Scenario Introduction lesson. A summary of Badges and Ranks is shown in Table 15.

Table 15 - Overview of achievements

Type	Title	Appearance	Description	Congratulatory message
Rank	Rookie Analyst		Askoni has begun to take notice of your deeds, but more accomplished Heroes still call you a Rookie Analyst. You get this badge by earning your first 47 points	Congratulations [username]: you have been promoted to Rookie Analyst. The secret organization NovoX places great trust in your ability to serve the interests of the Galaxy. If you do not know what this means yet, you will know soon enough! We trust you will serve the interests of the Galactic Council in the best possible way.
Rank	Case Officer		If you become a Case Officer, it means that NovoX values your contributions to the secret war against NovaVisione. Case Officers work with Informers and Information Analysts to ensure the safety of the Empire. To achieve this, they use the STS Method to model and analyze our great socio-technical system.	Congratulations Rookie, you have been promoted to Case Officer. Now that you are more acquainted with security requirements and socio-technical systems, it is time to get serious. As a Case Officer, you will need to apply the formal method the Galaxy uses to wage a secret war against NovaVisione: the STS Method!
Rank	Lieutenant Information Analyst		A Lieutenant Information Analyst rank is a significant milestone that awards you for your accomplishments. It is said to be one of the fastest paths to the top management of NovoX.	Congratulations, you have been promoted to Ltd. Information Analyst. You have reached a significant milestone in your career. To further improve your skills, the Galactic Council has appointed you as Lieutenant within NovoX Information Analysts unit. There, you will keep learning and defending the Empire by analyzing social interactions that might prove to be potentially devastating.

Type	Title	Appearance	Description	Congratulatory message
Rank	CSCO		Becoming the Chief SecComs Officer is an outstanding and one-of-a-kind accomplishment within NovoX. To earn this badge, the Galactic Council must appoint you as a result of your continued dedication to designing a bullet-proof, secure, socio-technical system to help fight the war against NovaVisione.	Congratulations Ltd., effective immediately you are now the Chief Secure Communications Officer of NovoX. This is proof of your constant dedication to serving the people of the Empire. The job carries the heavy burden of ensuring secure communications for the sensitive information of the Empire and Galactic Council. Your new mantra from now on: ensuring confidentiality, integrity, availability and non-repudiation.
Rank	Protector of the Galaxy		Protectors of the Galaxy are influential advisers to the Galactic Council tasked with ensuring the success of the Empire. Note that since there are only seven protectors for the entire Empire, earning this rank is very difficult. You must be a C-level executive within one of the divisions of the Empire (e.g., NovoX) and have an excellent track record.	Congratulations [currentuser_username], your remarkable and heroic achievements have led us to promote you to Protector of the Galaxy for all intelligence and security matters. Under your command, NovoX has designed a socio-technical system worthy of fighting NovaVisione and ensuring peace in the Galaxy. As you know, the Galactic Council only appoints seven protectors. We trust that, under your protectorship, your allies will help us defend the interests of the Empire in the best possible way.
Badge	Welcome, Fellow!		A badge says as much about the person or group who issued it as it does about the recipient. To earn this "Welcome" badge, you have to register and login for the first time	You have successfully created an account and logged in for the first time. Your journey has just started: WELCOME ABOARD! We hope you have an amazing time and learn insightful things about security requirements engineering!
Badge	Galactic Rookie		Congratulations on reaching this point! You are now familiar with the scenario that we will use throughout the tutorial. We hope you serve the interests of the Galactic Council in the best possible way	Same as description

The last subsection predominantly introduced the concept of **Levels** as a means to orchestrate the learning experience. They represent expected achievements and help learners to establish goals for themselves and create an overall schema of course progression. Their completion is highly visible at all times thanks to the presence of the course progress bar and the lesson title, which displays the level the learner is currently at. In stark contrast, **Ranks** follow under the category of unexpected achievements: although an ambiguous description can be consulted, there is no clear path to achievement. This is done to encourage creative play (Chou, 2015). However, achievement notification occurs immediately after a **Rank** or **Badge** is awarded. In all instances, a dialog pop-up appears with a brief congratulatory message and an overview of the earned achievement. The notification appears before or after the Reflect phase, thus occurring in a natural break of action that is not prone to generate disruption of flow. The timing of the notification combined with the unexpectedness represents a form of immediate feedback tied to performance aimed at boosting performance.

All the achievements can be consulted at the learners' user profile section as well as the Leaderboard, the latter being accessible from the primary navigation. Learners can go over their earned achievements as well as those of their peers, a feature intended to generate competition and recognition. In addition, these elements contribute to a vicarious experience, which is believed to improve self-efficacy levels (Kapp, 2012) and, thus, task performance (Stajkovic & Luthans, 1998).

The next subsection examines the inner workings of the point and rank systems.

5.4.3 Point and Rank Systems

The **Point System** awards points based on lesson completion and quiz correctness, as a function of the estimated time they take to be completed (i.e., in a ratio of ten points for each minute). However, a correction factor is used for quiz completion, as it requires learners to be more actively engaged and presents higher difficulty. In practice this implies that accurate quiz completion allows learners to receive twice as much points as viewing lesson content. In other words, completing the (estimated) five-minute lecture present in Level 6 yields 50 points, whereas successful completion of its (estimated) three-minute quiz would yield 60 points. In addition, boosters in the form of bonus points are awarded for difficult levels (accompanied by positive reinforcement messages through the chat widget), provided that the quiz grade is 100%. As Figure 23 shows, the current version of the tutorial awards up to 1170 points.

On another note, **Rank** is determined using an algorithm that makes it harder to rank up as learners obtain more points (and thus become more skilled). We opted for function based on a linearly rising level gap, since it was found to be a commonly reported way of designing leveling systems when points are awarded linearly, as in our case (Kapp, 2012; Chou, 2015). The inner workings of both the **Points** and **Rank System** are shown in Figure 23. Note that these systems are hidden to learners.

As previously mentioned, the tutorial also features the so-called Rewards Marketplace, where learners can exchange points in favor of 1) chatting with the Game Master and Security Requirements Experts or 2) claiming other Rewards such as: deduct points to another player, get personalized help in case of problems, purchase contextually-relevant hints, etc.

Level	Learn phase (Passive)	Play phase (Active)	Bonus points
Tutorial Overview	30		
First Thing to Do	10	40	
Why SRE is important	30		
Level 1	50	20	
Level 2	20		
Level 3	25	50	
Level 4	30	20	
Level 5	40	40	
Level 6	18	64	100
Level 7	20	60	80
Level 8	10		
Level 9	20	60	
Level 10	12	56	
Level 11	30		
Level 12	13	80	60
Level 13	15		
Level 14	13	54	
Total points	386	544	240
Grand total points			1170

Rank System. Point thresholds	
For the 5 available Ranks, the polynomial function is defined by: (Rank/0.1462)^2	
Rookie Analyst	46,78485144
Case Officer	187,6524676
Lieutenant Information Analyst	421,063663
Chief Secure Communications Officer	748,557623
Protector of the Galaxy	1169,621286

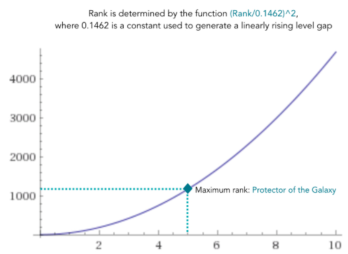


Figure 23 - Overview of Point and Rank systems

For traceability purposes with regard to our previous literature reviews, Table 16 and 17 report a comprehensive list of the final game elements and gamification guidelines included in the interactive tutorial.

Table 16 - Summary of implemented game elements aligned with our literature reviews

Game element	How the element was used
Point system	Points are granted to reward users upon completing Levels (i.e., lessons) and correctly passing quizzes (with 100% accuracy/pass grade). A specific Point System was designed following de-facto standards
Badge / Rank	Ranks and Badges are awarded upon completing several actions (these can be customized according to the instructor's requirements). They represent visualization of individual achievement to give a sense of progression. A specific Rank System was designed following de-facto standards (linearly rising level gap)
Leaderboard	Learners are ranked based on points and rank achieved to foster competition and recognition
Level	Used to keep game space manageable and give a sense of progressoin. Completion is rewarded with points and rank ups. Quizzes are included here
Progress bar	Used to track overall goal progression with respect to the different modules
Story/ Narrative	A science fiction story was used in an attempt to add meaning, provide context, and guide action in a novel way
Avatar	Users can access their user profile and customize it, including their avatar. Avatars are randomly generated and represent the learner's virtual characterization within the platform
Rewards	A Rewards Marketplace was devised using a chat plugin where learner can redeem rewards in exchange of points
Time restriction	Countdown functionality was not developed per se. However, we instilled offline pressure during the evaluation. This will be examined in the next chapter
Verbal/visual/ sound effects	Used to tell users about the state of the system as a result of interacting with the tutorial and modeling tool
Animated feedback	Animated visual aids are used to help users locate 1) contextual messages, 2) specific points of interest in a slide
Conformity anchor	The platform informs the user how close a user is to the social norm (i.e., real-time performance reporting using the chat widget)
Social prod	Users can "Like" events (e.g., earned achievements, completed lessons, comments, and so on) in the Activity feed present in their personal profile as well as deduct points to other players (by purchasing the reward through the Marketplace)
Game master	Observes and orchestrates the learning process. Assists or influences the behavior of learners. This figure interacts with learners through the chat widget using Slack. He may also send private messages or post comments within the platform
Chat room	Live chat where users can interact with the Game Master or SRE experts, but not with each other
Activity feed	Stream of recent events in the course displaying what activities and achievements learners enrolled in the same course have completed/earned

Table 17 - Summary of considered gamification guidelines aligned with our literature reviews

Guideline	How the guideline was considered
Freedom of choice	Learners are giving the feeling of freedom as they can freely navigate the different pages and sites in a non-linear way. However, this guideline was not respected considering the current educational content, as learners can only progress through the tutorial progressively. The platform however is extensible enough to allow for multiple paths depending on learner's preferences
Freedom to fail	Learners are able to retake quizzes, and fails do not result in negative feedback or loss of points (note that learners do not even know how points are awarded). In addition, positive reinforcement messages are delivered upon quiz failures.
Baby steps	The current tutorial features, concise, bite-sized lectures that do not go for more than five minutes
I know this!	Nearly 70% of the Levels (lessons) contain a Play phase where learners are asked to complete a quiz or a modeling exercise
Together is better	Learners are automatically assigned into course groups where they can exchange comments and like each other's posts. A leaderboard was included to motivate learners through peer pressure and social comparison
Boost it!	Learners are awarded bonus points after completing a difficult quiz. Difficulty was judged on the basis of the subjective opinion and expertise of the author
The good Samaritan	Compensate students not only for academic achievement but also prosocial behavior
Crystal clear	Offer immediate feedback and inform learners of their progression within the tutorial (e.g., progression bars. Frequent and immediate feedback leads to greater learning effectiveness and engagement
Ethical designer	The platform does not intend to cause harm Learners can customize what is visible to others and whether they want to appear in the leaderboard Although the Rank and Point system are intentionally not explained to the learners, we do not believe this constitutes an ethical dilemma
Know your users	Proto-personas were created based on observed characteristics of the target populations as well as interviews with SRE experts with a teaching background The author of this thesis is considered to be a member of both groups and thus (arguably) has high relatedness
The road ain't easy	We reinforced this message in the introductory and final notes to the tutorial with a spoken speech communicated through video, where the author (in the role of instructor) directly engages the learners
Location, location, location	We attempted to control location by conducting on-site user evaluations. However, this seemed to be partly irrelevant, as we will explore next.

In summary, the first version of our interactive platform, called STS-Tooltorial, introduces learners to basic concepts of socio-technical systems and security requirements engineering to then delve into the STS Method and modeling language. The platform is embedded in STS-Tool, thereby offering a first-of-a-kind experience for learners as they can learn at their own pace following a hands-on and highly interactive approach with short feedback cycles. At the time of this writing, the tool covers a subset of the STS Method that can be completed at once within 90 minutes. However, we are planning to release further learning modules to cover the entire spectrum of STS.

Rather than including a detailed overview, screenshots, and flow of the platform's different sections and features, the reader with access to Internet is highly encouraged to download a copy of the STS-Tooltorial and feel the experience by herself[3].

[3] Mac: http://bit.ly/ststoolmac. Windows: http://bit.ly/ststooltorialwin. Linux: http://bit.ly/ststoollin64 (64 bits) or http://bit.ly/ststoollin32 (32 bits). Terms and conditions from the STS-Tool apply.

6. Evaluation design of the STS-Tooltorial

In this chapter, we report the conduct of the evaluation of the STS-Tooltorial within a quasi-experimental setting. The scope and design of the evaluation is reported on **Section 6.1**, followed by the instruments and measures used other than the STS-Tooltorial itself (**Section 6.2**). The chapter concludes with a brief discussion of validity.

6.1 Scope and Design

Initially, we intended to perform an experimental evaluation on the effectiveness of gamified interactive tutorials, exposing participants to a gamified vs. non-gamified version of the platform. However, we pivoted our research along the way for two reasons. The first was time constraints: the personal and professional situation of the author changed, the scope had to be reduced, and the evaluation was a clear candidate. Second, we deemed more important to conduct a qualitative evaluation geared towards understanding the true benefits of a gamified interactive platform, exposing every participant we could obtain to the platform at its fullest extension. In other words, rather than creating two versions of the platform for research purposes (with and without gamified elements), we decided to allocate more time to consuming the entire backlog of requirements, thereby creating an extensible, comprehensive, and engaging platform that can be used in a real setting for the benefit of students and professionals. Also, more emphasis was put on gathering and processing qualitative results rather than comparing the two approaches quantitatively.

On a related note, in our experience, evaluations conducted within a postgraduate setting always find it challenging to recruit study participants, with the "law of small numbers" problem this generates when attempting to derive findings (such as causality) and test hypotheses that hold in the real population (Kahneman, 2011). We thus found it more valuable to study how participants subjectively experienced the interaction with the STS-Tooltorial.

To that end, we devised a study consisting of two separate empirical evaluations featuring eight postgraduate information science students and five IT professionals with three to twelve years of experience, respectively. As Table 18 shows, each evaluation falls under the category of "posttest-only non/quasi-experimental design", as participants were exposed to the gamified interactive platform and then requested to complete a post-evaluation questionnaire[4]. The inclusion criteria for students was to have 1) little to zero experience about SRE or socio-technical systems modeling 2) no knowledge of the STS Method and 3) giving consent to participate in the study. For professionals, only the last two criteria were used. In addition, all participants were informed that they could withdraw at any time from the evaluation if they so desired.

Table 18 - Design of the quasi-experimental evaluation

	Assignment	Pretest	STS-Tooltorial	Post-Test
Students	NA	NA	X	O_1
IT Prof.	NA	NA	X	O_2

6.1.1 Pilot Study

Prior to the evaluations, a pilot was conducted to refine a beta version of the platform to determine the adequacy of the timeframe, participant initial impressions, good functioning of the platform (i.e., game elements, overall flow, and learning analytics) and make changes to the platform where required. We selected a convenience sample of three representative users from the first group (students) and requested them to interact with the tool for 90 minutes. After the pilot, participants were asked to provide honest feedback about what they liked the most and what could be improved. The insights, reproduced in Figure 24, led to top 9 actions, which were implemented to be readily available during the evaluation. Note that the data collected here was *not* used as input for **Chapter 7** (Results).

6.1.2 Student Evaluation

The student evaluation was conducted on-site at Utrecht University in two separate sessions following the same procedure and setting. In particular, students were grouped in a room and requested to complete as much of the tutorial as possible within a 90-minute timeframe, under the supervision of the authors who were not to intervene unless, e.g., a participant redeemed a reward for hints or engaged with the Game Master, both through the chat function. By having participants co-located, we hoped to induce certain (social) pressure by the combination of 1) the time restriction,

[4] http://bit.ly/ststooltorialposttest

2) projecting the real-time leaderboard status in a big screen, and 3) having the Game Master physically present indirectly judging on the correctness of the modeling activities.

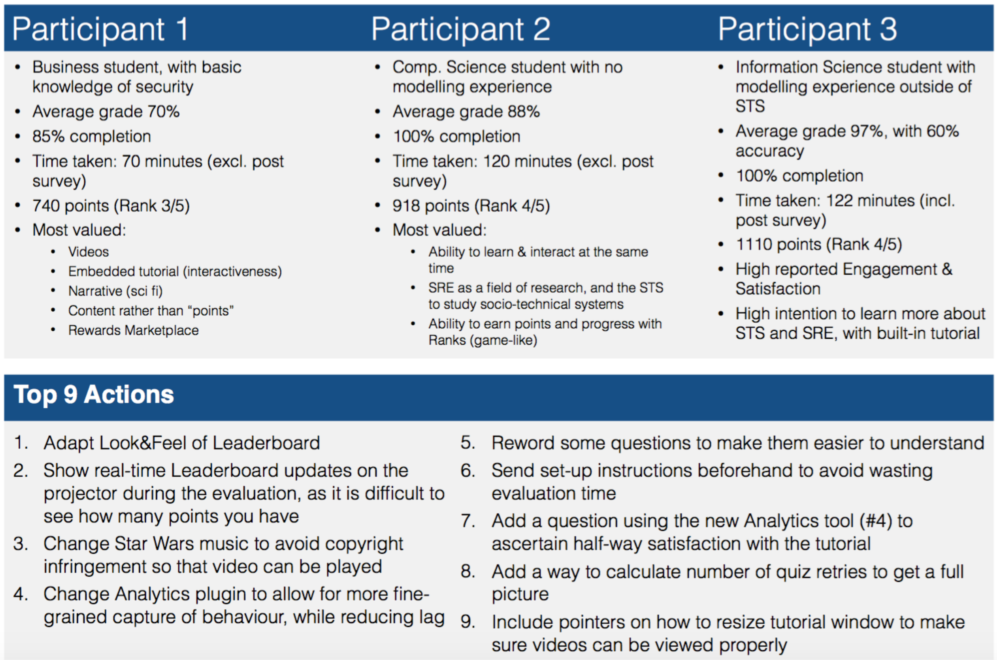


Figure 24 - Insights derived from the pilot study and change actions

6.1.3 IT Professional Evaluation

Since results of the student evaluation suggested that physical co-location of learners was not a key factor to have them complete the tutorial, we decided to organize the evaluation with professionals in an online, remote fashion, with participants setting their pace independently. To allow for some degree of comparison with the student group, participants were asked to abide by the same procedure (i.e., completing the tutorial at once within a 90-minute timeframe), to which professionals kindly complied. These participants were selected based on a convenience sample of IBM employees, provided that they met the inclusion criteria.

In both evaluations, participants were contacted in a separate day to be briefed on their results (Figure 25 depicts the template filled for one of the participants) as well as to conduct personalized semi-structured interviews through to gather more information about how participants experienced the tutorial. The interviews were performed via Skype (for students residing in the Netherlands) and face-to-face (for IT professionals). A brief protocol, included in **Appendix III**, was created to guide the interviews, less strict than what was used for the interviews with SMEs.

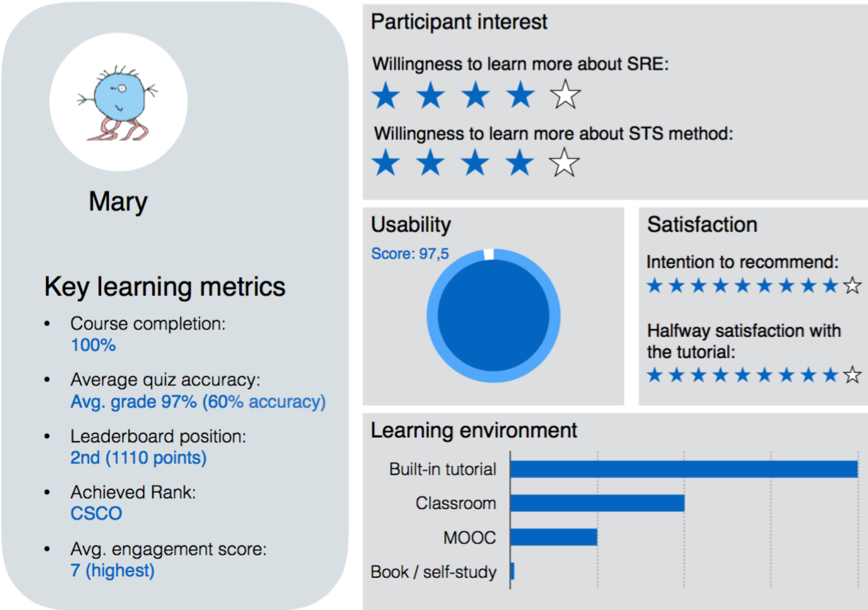


Figure 25 - Example of filled template for briefing participants during the semi-structured interviews

6.2 Instrumentation

The objective of the evaluation was to gather learner's perspective on the use of the interactive platform, as well as learning analytics (including behavior metrics, flow, and performance). To derive helpful insights, both quantitative and qualitative measures had to be gathered. While some of these are computer-harvested, others are supplied by the learner at the end of the intervention through the post-evaluation questionnaire.

In designing the questionnaire, the aim was to create a relatively short scale instrument that could be completed within ~10 minutes. The concepts under study were: learner engagement, usability and overall satisfaction, performance, intention to learn, and preferred learning environment.

Engagement and Behavior

The concept of engagement was underpinned by the nine-item Short Flow Scale (Jackson, Martin, & Eklund, 2008) ranging from 1 (minimum flow) to 7 (maximum flow). Although simple, this instrument exhibits acceptable model fit, internal reliability (with a Cronbach's alpha between .81 and .92), and distributions, making it a pragmatic means to capture participants' flow after an intervention (Jackson et al., 2008; Rheinberg, Vollmeyer, & Engeser, 2003).

In addition, we configured the platform to gather various analytics on how learners behaved within it:

- **Dwell time** (Muntean, 2011; Dupret & Lalmas, 2013): time spent on the website per session
- **Video play time** (Dobrian et. al, 2011): video starts, pauses and finish events were configured to be sent automatically to Google Analytics with the help of Segment analytics API (Segment.com, 2017)
- **Page views and average time spent per page** (Muntean, 2011; Dupret & Lalmas, 2013)
- **Social interactions**, as evidenced by number of interactions with the chat function present in the Rewards Marketplace (i.e., as a means to engage in a conversation with the Game Master) and use of social features (e.g., Likes, Comments)

Furthermore, the Hotjar analytics software (Hotjar.com, 2017) was used to track mouse clicks, record screens of visitor sessions, and derive heatmaps to fully understand learner interactions (see Figure 26 for an example). Participants were provided with randomly generated usernames and passwords to be able to trace all this information without compromising confidentiality.

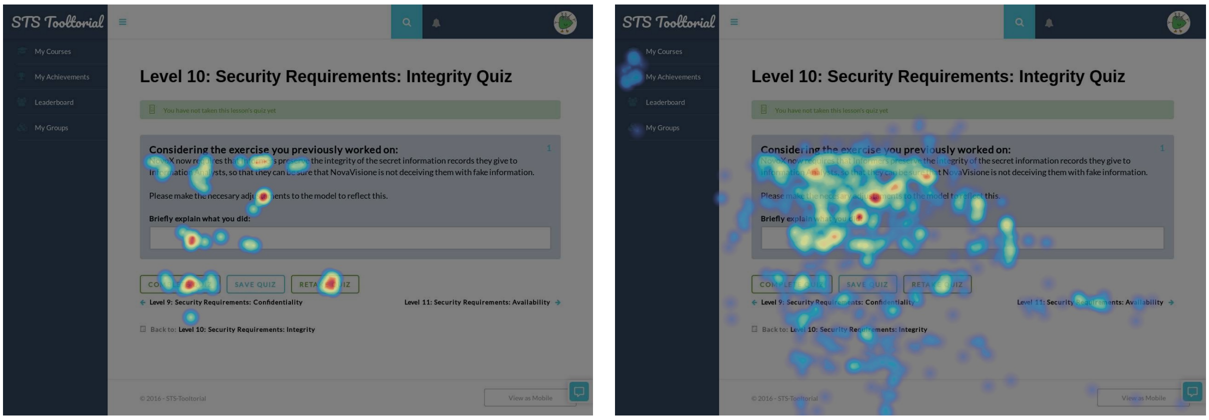


Figure 26 - Heatmaps allow fine-grained analysis of learner behavior by studying clicks (left) and movements (right)

Usability and overall satisfaction

Usability was measured using the System Usability Scale (SUS): a valid, reliable instrument widely used in industry and academia. With a reported alpha score of 0.92, the SUS scale possesses high reliability, similar to other scales of a commercial nature. Its wide usage and publication record makes it easy to establish benchmarks of what "average usability means" (i.e., a score of 68). Furthermore, the SUS scores are believed to predict (or at least correlate with) customer loyalty and thus, satisfaction (Lewis & Sauro, 2009).

To measure satisfaction, half-way through the tutorial learners were asked to rate on an 11-item Likert scale how satisfied they were with the tutorial through the Hotjar plugin. Subsequently, in the post-evaluation survey, the Net Promoter Score (NPS) ranging from 0 to 10 was used as a proxy measure to satisfaction (as evidenced by intention to recommend the tutorial to a colleague).

In conjunction, these three questions provided good coverage of the interrelated concepts of usability, satisfaction, and intention to recommend.

Performance

Performance was operationalized using the following metrics, all of them automatically gathered by the platform:

- **Number of Levels completed**
- **Badges and Rank earned**
- **Number of Points obtained**
- **Quality of task execution**, as measured by the number of correct quizzes vs. total attempts

Intention to Learn

We examined the extent to which the platform influenced the learner's intention towards learning more about SRE in general and the STS Method in particular, which as we argued, has been shown to be a strong precursor of behavior (Ajzen, 1991). Particularly, we employed five-point Likert scale where we posed the statement that "I am interested in studying more about..." ranging from 1 (strongly disagree) to 5 (strongly agree).

Preferred Learning Environment

Given that the STS-Tooltorial represents an innovative teaching method, we attempted to ascertain the extent to which participants believed the platform was conducive to learning. To that end, participants were finally asked to rank their preferred choice of learning environment among four choices: MOOC, Book/Self-study, Built-in tutorial (e.g., our platform), and Classroom setting, from 1 (most preferred) to 4 (least preferred). Such a ranking question allowed us to position participants' preferences about the platform against other traditional and modern approaches.

6.3 Validity Discussion

Borrowing from case study and software engineering research, a validity defense must at least cover construct, internal, and external validity as well as repeatability and applicability (Yin, 2013; Wohlin et al., 2012).

With regard to construct validity, this section has shown that, whenever possible, the concepts of interest have been operationalized drawing on internally consistent and valid measures such as SUS, Short Flow Scale, and NPS. However, since we heavily relied on a non-probabilistic sampling method (i.e., based on convenience and proximity), it is possible that some degree of bias was introduced in our observations. Here we took two precautions: 1) the instructor/researcher was instructed to act in a neutral way, without consciously leading participants and being involved in the evaluation only when necessary, 2) participants were requested to provide honest feedback and explained that answers would remain anonymous. Observe that while we operationalized gamification with a comprehensive list of game elements (see Table 16), it would have been difficult to compare the platform in an experimental and non-experimental setting, as 1) there is no single valid operational measure of gamification that can be isolated and 2) the game elements interact with other factors (e.g., user interface design, video lectures), thus potentially causing confounding effects (Wohlin et al., 2012)

Nonetheless, as this investigation was of a quasi-experimental nature, internal validity is not a significant concern: we were not trying to establish casual relationships, nor was there a control group exposed to a non-gamified version of the platform. History, maturation, testing, and instrument decay are all valid threats that can cause issues when attempting to derive casual relationships on the effectiveness of the platform. At this point it is worth mentioning that we did not conduct "formal" hypothesis testing on the hypothesis put forth in our conceptual model in Figure 18, but rather served as a guidance during the study. On the other hand, the evaluation and the post-test questionnaire happened sequentially within a short time frame, so it is safe to assume that self-reported measures of engagement and satisfaction are closely influenced by prior interaction with the gamified tutorial. In fact, the influence could even be problematic. Our last point of attention in this matter refers to the results that can be inferred from comparing students and professionals. Since the two groups experimented the tutorial in different modes of interaction (co-located vs. remote), the reader should interpret the results reported in the following chapter with caution.

We were more concerned with the applicability of the platform in a real context and how learners perceived and experimented the STS-Tooltorial. To that end, we favored external validity of the study by sampling both students and IT professionals from two countries. In addition, conducting the evaluation using the two modes of interaction helped us to evaluate how learners experimented the tutorial in two different settings, thus increasing external validity. A third way by which this form of validity is increased is through the extensibility of the platform, given that it can be modified both to integrate new tutorials/courses into the current platform as well as work in other contexts outside the realm of (S)RE

Finally, repeatability is achieved by reporting the procedures and techniques by which the artifact has been designed and implemented. Furthermore, the report can be used to inspire and inform researchers on the framework needed to create innovative teaching methods such as the one reported here. Finally, repeatability is mildly compromised in that the source code of the platform cannot be distributed freely due to copyright issues with both the STS-Tool and the Sensei Wordpress plugin.

7. Results

In this chapter we report findings from investigations that focused on a subjective evaluation of the effectiveness of the STS-Tooltorial on learner engagement, usability and overall satisfaction, performance, intention to learn, and preferred learning environment within the quasi-experimental setting discussed in the previous chapter. Where relevant, we make comparisons between the groups, but note that establishing group differences was not the focus.

Overall, the evaluations ran smoothly without major threats or hindrances, except for a severe issue with the quality of the videos. In particular, two of the participants in one of the sessions complained about constant video freezes. The root cause was that the external hosting provider where video lectures were posted (YouTube) suffered an outage that impacted on the quality of the videos. In fact, these participants reported lower-than-average flow scores. However, the usability and satisfaction scores of the STS-Tooltorial were not impacted, which at least is indicative of robust construct validity of the instruments.

During the on-site student evaluation, participants only interacted with each other on a few occasions. Nearly 80% of participants ignored or paid little attention to the **Leaderboard**, in line with the limitations reported in literature of merely using PBL (e.g., Deterding, 2014). Instead, they were entirely focused on progressing through the tutorial. This is precisely what led us to conduct the IT professional evaluation remotely, when/wherever participants wanted to complete it (notifying in advance the author, as he had to offer online support by acting as Game Master and "fictitious" SRE expert).

7.1 Engagement and Behavior

In terms of engagement, the student group self-reported a higher than average flow during the completion of the tutorial (*M*=4.69, *SD*=1.03), with no significant difference with respect to the professional's flow (*M*=4.5, *SD*=0.90). Cronbach's alpha for the Short Flow Scale was .87, indicative of high reliability as other studies have reported (Rheinberg, Vollmeyer, & Engeser, 2003; Jackson et al., 2008).

With regard to behavior, we examined the dwell and video play time, page views, and social interactions. Each of these is reported in turn.

Dwell and Video Play time

In line with our pilot study, participants spent on average just above 80 minutes interacting with the platform (*M*=83, *SD*=12.81, range: 64-100), and five to ten additional minutes completing the online interview.

An analysis of logs and Google Analytics events indicating when learners started, paused, and finished videos highlighted that most participants (77%) consumed the video lectures in their entirety, with a few minor exceptions. For instance, one participant skipped four video lectures and went straight to the quizzes in an attempt to guess the answers. It is worth noting that correct answers were shown for incorrect multiple-choice and true/false questions. Thus, should this type of behavior become common practice, future versions of the tutorial will have to be more restrictive when revealing correct answers. Another person from the student group stopped at Level 10 without completing the remaining levels. He was the lowest performing participant in terms of flow, points obtained, and quiz accuracy.

Page Views

According to literature, page views represent a form of engagement, so it is not surprising to see that, in both groups, engagement and page views were strongly related, although the relationship fell slight short of statistical significance (r=.65, n=13, p=.08). We contend that the relationship is not significant because all of the participants completed the entire tutorial sequentially and thus there was little room and incentive for some learners to visit more pages than others. In addition, access logs showed that learners progressed through the pages linearly without backtracking to the video lectures in case of quiz failures.

Figure 27 shows the most visited lessons (left) and quizzes (right). Of relevance is that the "Scenario Introduction" lesson containing the introduction of our fictitious narrative/story is ranked among the top-viewed sites. Regarding quizzes, it can be seen that visits were equally distributed except for those in Level 10 and Level 6, which learners retook multiple times in an attempt to answer correctly (as we will see these represent the hardest quizzes for our sample). The fact that learners persisted in 1) re-attempting quizzes, and 2) seeing it through the end of the tutorial is suggestive of high engagement.

However, note that top lesson results (shown in the left) may be confounded. In particular, careful analysis of access logs revealed that learners sometimes skipped to the next lesson without clicking on the "Complete Lesson" button. Since lesson prerequisites exist, the learner would then be forced to go to the previous button and click on the button. This erratic behavior on the learner's side (arguably due to a usability problem) may explain why some levels that only required watching a video appear at the top (e.g., such as Levels 6 or Modeling Security Requirements, which do not contain quizzes).

On a final note, it is worth mentioning that we did not put in place specific game elements to encourage page views, as other researchers have pointed out adverse effects such as reducing time spent on page (Kapp, 2012) and thus the quality of learning.

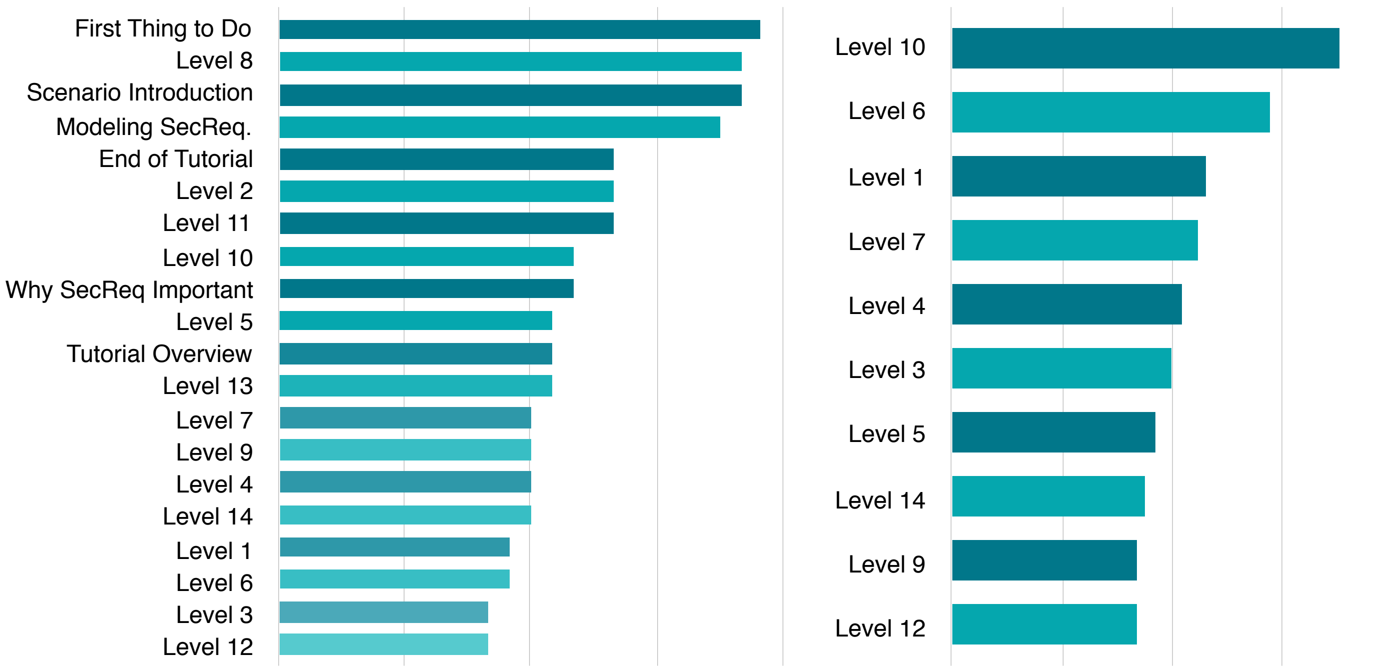


Figure 27 - Pages views for lessons (left) and quizzes contained within levels (right)

Social Interactions

Several elements were included to afford social interactions, namely: 1) a specific lesson that requested learners to post a comment, 2) the leaderboard, 3) the chat widget featuring the Rewards Marketplace and 3) groups. We review them in turn.

In the second lesson of the tutorial, learners were asked to introduce themselves to their colleagues by posting a comment (unbeknownst to them, this activity allowed learners to earn 40 extra points). We observed that five out of eight students and half of IT professionals posted a comment. However, the social interaction stopped there. In fact, students in the on-site evaluation only shared off-topic comments in a few occasions, and in no occasion did they exchange information or hints about the quizzes.

In addition, we found that learners did not visit the leaderboard as much as we would have expected, with 20% of participants visiting the page at least once. Leaderboard visits (a sign of social comparison) was positively associated with performance in terms of points obtained and page views (r=.76, n=13, p=.05): top performers were more likely to check their Leaderboard status, albeit sparingly.

The Rewards Marketplace was also a direct social channel with the Game Master and SRE experts. Here, we also found the interaction with this innovative feature lacking. Only 37.5 of participants interacted wit the feature (and only once), to ask about: 1) how to switch between lessons, 2) questions about a certain quiz. Moreover, an analysis of screen recording showed that some participants started typing in the chat box, but did not end up sending the message. In another occasion, one participant, upon asking for help and being informed that the assistance would cost 20 points, negotiated a discount of 10 points. In the follow-up interviews learners indicated they were deterred from asking help for two reasons, to wit: 1) participants wanted to retake quizzes many times before asking for help (70%), and 2) they did not want to lose points (40%). All these insights are suggestive of participants' interest in earning and not losing points, a phenomenon left for discussion in the following chapter. They also revealed that the best use case for the widget was the sending of contextual/ positive reinforcement feedback.

Finally, groups were not used at all, since we did not find a suitable use case and thus did not promote the use of them for these evaluations. As we explore in our last chapter, we believe there is room for improving the learner experience by exploiting the use of this feature in bigger groups.

Overall, we may conclude that the social component was lacking in this first version of the platform. However, this did not seem to be a deterrent. First, we saw that participants reported above-average engagement, superior to other similar studies (e.g., Lombriser et al., 2016), as well as high overall satisfaction and preferred choice of learning environment, as we explore in the remaining sections.

7.2 Usability and Overall Satisfaction

The aggregated SUS scores of both samples indicate that the interactive platform has high usability and is well integrated within the modeling tool (M=80, SD=8.07), with no significant difference among students and professionals; $t(11)=.16$, $p=.88$. Surprisingly, Cronbach's alpha for the 10-item SUS scale was .71, which indicates a medium reliability. In stark contrast, literature (e.g., Lewis & Sauro, 2009) suggests that the SUS instrument is highly reliable (i.e., with $\alpha >= .85$). Note that this could be attributable to either our small sample size or the fact that the STS-Tooltorial was embedded within the STS-Tool, thereby confounding participants' opinion.

With regard to satisfaction, the groups reported high degrees of half-way satisfaction (M=7.38, SD=1.19). This measure was found to be positively correlated to engagement (r=.71, n=13, p=.05), indicating that deep focus on the task at hand brings higher levels of satisfaction, and vice versa. In addition, participants indicated a high intention to recommend (i.e., NPS score) the STS-Tooltorial (M=7.88, SD=1.46), with again no significant difference between professionals and students. Furthermore, the satisfaction and NPS scales were found to be strongly correlated across groups (r=.86, n=13, p=.007), indicating that a person's own satisfaction with the tool is a strong correlate of word-of-mouth recommendation. However, in contrast to common belief, intention to recommend was highly uncorrelated to usability (r=.05, n=13, p=.89).

7.3 Performance

Most of the participants completed between 80% and 100% of the content with success (but not upon first attempt as we will show). In particular, 75% of students successfully completed 80% or more of video lectures and quizzes as measured by the number of **Points** obtained (M=950, SD=203, range: 579-1130). IT professionals were awarded slightly more points on average (M=1066, SD=103, range: 920-1170). Results show that the latter exhibited a better, although not significant performance ($t(11)=-1.17$, p=0.27, d=0.72). While the non-significant result may be attributable to the small sample size, the large effect size may be indicative of some of the IT professional's prior experience in conceptual modeling.

The highest **Rank** (Protector of the Galaxy) was only awarded to one learner belonging to the professionals group, although the second highest ranking learner was very motivated to reach it as well; he even engaged the Game Master via the Rewards Marketplace asking for hints (he was quickly discouraged when he was informed that hints came with a loss of points in exchange for answers).

Quality of Task Execution

The quiz accuracy of both students (48%, *SD*=13%) and IT professionals (53%, *SD*=9%) professionals was moderate, which suggests that task difficulty was appropriate.

Quiz accuracy also provided a means to evaluate the inherent difficulty of **Levels**. The average level quiz accuracy was 48%, with Level 5 (80%) and Level 3 (70%) being the easiest, and Level 10 (18%) and Level (37%) being the hardest (see Figure 28). As could be expected, this finding strongly correlates with top viewed pages reproduced in Figure 27 above.

We also found that quiz accuracy was positive related to the number of points obtained (*r*=.74, n=13, *p*=.035). Although participants were shown feedback in case they failed and allowed to retake quizzes (without penalty), this suggests that top performers did not fail as many quizzes as other participants did. Or, from another lens, that participants with a low accuracy rate either 1) did not fully exploit the possibility of retaking quizzes, or 2) did not manage to answer quizzes correctly.

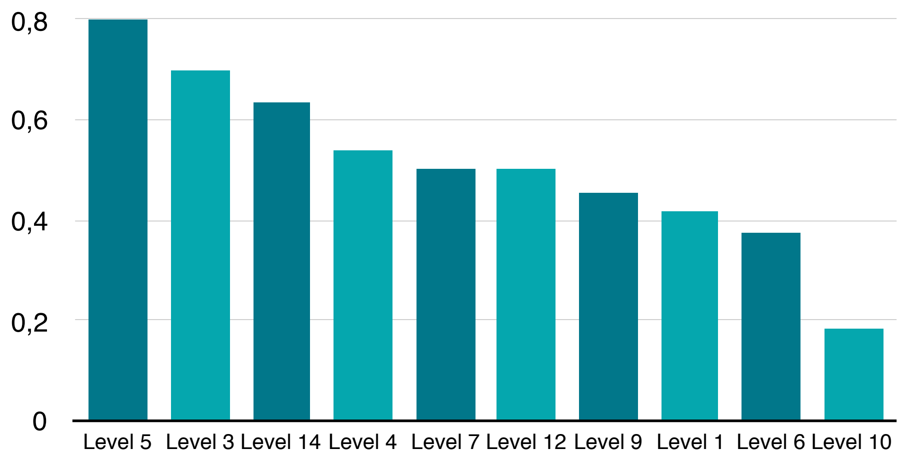


Figure 28 - Level difficulty as measured by quiz accuracy, from left (easiest) to right (hardest)

Finally, results showed a consistent, positive relationship between the number of points and 1) half-way satisfaction (*r*=.65, n=13, *p*=.08) and 2) engagement (*r*=.64, n=13, *p*=.09), albeit in both cases this fell short of statistical significance.

7.4 Intention to Learn

Findings suggest that students had a significant higher intention to learn more about both SRE (*t*(11)=3.12, *p*=0.01, *d*=1.88) and the STS Method (*t*(11)=2.15, *p*=0.05, *d*=1.29) than professionals did. In addition, a consistent pattern we observed across groups was that learners had more intention to learn about SRE (*M*=3.15, *SD*=1.14) than the STS Method in particular (*M*=2.77, *SD*=1.01), although the latter was the main focus of the tutorial; *t*(11)=-2.13, *p*=0.054, *d*=1.28. Nonetheless, both measures were highly correlated (r=.825, n=13, p=.001), which implies that the tutorial was effective in raising awareness and interest in both the STS Method and (through it) the broader field of SRE.

Moreover, the number of times people (re)attempted quizzes was strongly related to their intention to learn more about SRE (*r*=.74, n=13, *p*=.04), but not significantly compared to the STS Method (*r*=.43, n=13, *p*=.29). On the one hand, this shows good signs of engagement: consistent, repeated practice brings higher levels of interest in the broader topic. But this result is nonetheless surprising given the fact that the tutorial mostly focused on the STS Method and not SRE.

A last finding worth exploring is the relationship between engagement and intention to learn more about STS (arguably one of the major objectives of the tutorial). We found that the more flow (engagement) participants reported after interacting with the STS-Tooltorial, the more the intention to learn (*r*=.72, n=13, *p*=.04).

Overall, learners expressed at least an average-to-positive interest in pursuing further learning, with students being significantly more positive than IT professionals.

7.5 Preferred Learning Environment

While books were consistently ranked as the least preferred option by 62.5% of the participants (M=3.25, SD=1.16), the platform (M=2.13, SD=1.13) ranked first 37.5% of the time, closely followed by traditional classroom (M=2.25, SD=1.04) and MOOCs (M=2.38, SD=1.06). The results suggest that this innovative teaching method stands as an effective competitor to both traditional and modern approaches. Also, observe that participants still regard traditional classroom setting as an environment conducive to learning.

IV. The Road Ahead

In this final block, we succinctly explain our new-found understanding and insights that emerged from this research. We ponder the results of our study and describe the significance of our findings in relation to the broader fields of inquiry, while constantly acknowledging our limitations. Finally, we cannot help but look beyond the present, at what is to come. Our work is a small contributing drop in the ocean of future research studies. But, for now, we truly believe the STS-Tooltorial represents the inception of an innovative teaching method that can revolutionize the education of the next generation of requirements engineers and conceptual modelers.

8. Discussion

In this chapter we present a collection of our main findings (**Section 8.1**), provide a succinct answer to our research questions (**Section 8.2**), discuss the limitations of the research (**Section 8.3**), and present a conceptual model to guide a possible future experimental evaluation of the STS-Tooltorial (**Section 8.4**).

8.1 Summary of Findings

In the paragraphs that follow we draw our main conclusions in subjective order of importance and put them in context supported by literature. Note that these statements are tentative by nature given that 1) they have not been validated in an experimental setting and 2) sample size limitations exist.

The STS-Tooltorial may stand as an innovative teaching method that is as effective and preferred as others. The tutorial was successful in raising awareness on the importance of security requirements engineering, as evidenced by participants' average-to-positive interest in learning more about the field in general and the STS Method in particular, with students being more intent than professionals.

Lastly, the aggregated sample of students and IT professionals ranked the interactive platform as the most preferred option followed by traditional classroom and MOOCs, indicating that they would be happy to use this type of method in other settings. Of course, we may be falling victim to the same mere-exposure effect (Zajonc, 2001) we criticized in **Section 3.2.3**, which suggests that the mere, repeated exposure to an stimuli increases people's preference toward it. Consequently, more research is needed to determine whether this type of teaching method lives up to its promise in terms of standing as an effective competitor to both traditional (e.g., books or classroom) and modern (e.g., MOOCs) approaches.

The interactive platform represents the first successful attempt to use in-tool tutorials in (S)RE. When asked, participants reported good satisfaction (7.33 out of 0-10) and high intention to recommend (7.88 out of 0-10), and above-average usability scores (i.e., a score of 80 corresponds to an "A" system at the 90th percentile). As per Dutch standards, a grade close to 8 or more would represent an "excellent" grade (footnote), a fact worth observing given that nearly half of our sample consisted of Dutch students, and thus prone to follow their culture[5].

[5] https://students.uu.nl/sites/default/files/geo-grading-systems-holland-vs-us-uk.pdf

Points, badges, and leaderboard are not good enough to create an engaging learning experience. While PBL may work for short-term, repetitive, dull tasks that participants conduct for extrinsic purposes (e.g., earn money), learning environments must also be underpinned by intrinsic factors such as an enticing narrative. In our case, follow-up interviews highlighted the real-time interaction with the modeling tool, the narrative/storyline and the delivery format as key success factors. Extrinsic traditional elements such as points, badges, and leaderboards were not found to be interesting to participants. For instance, although we included a non-intrusive link to the Leaderboard (displaying Ranks and Points) within the navigation menu, an overwhelming majority of participants ignored it. In contrast, result from our evaluation suggest that participants enjoyed going through the learning content *per se* because of the way it was presented, i.e., interactive exercises, supported with video lectures, and framed around a story. Observe that all of these elements are of an intrinsic nature, which is said to improve feeling of autonomy, mastery, and relatedness and make the task inherently enjoyable (Deci & Ryan, 2011). This is very much aligned with the recent trend on rethinking gamification from individual game elements to delivering a holistic, immersive game experience (Deterding, 2014).

But, do the elements that, apparently, contributed most to the success of the platform fall within the realm of *gamification*? What is the line between gamification and good educational instructional design coupled by proper application of user experience design? Gamification very much depends on how the concept is operationalized. Philosophical debates aside, we cannot state that gamification was successful, but rather that the combination of the elements included in Table 16 together contributed to the overall success of the platform, as evidenced by the results reported here.

To conclude this reasoning, at a minimum, we argue that gamification must be operationalized through a balanced set of both intrinsic and extrinsic motivational affordances, with a bias toward intrinsic motivators for tasks that require intellectual ability or hint at some form of behavior change. In line with our observations, it would be interesting to see the effect of removing entirely points and leaderboards in future investigations.

Participants cared to some extent about points, and not losing what was already theirs. This finding is consistent with the concept of "loss aversion" in prospect theory (i.e., losses loom larger than corresponding gains; Kahneman & Tversky, 1979) and, particularly the so-called endowment effect (Kahneman, Knetsch, & Thaler, 1991) whereby people tend to overestimate the price of their (believed) possessions. In relation to the underutilization of the Rewards Marketplace, we contend that learners would have interacted more with it had we removed the warning that it resulted in a loss of points.

Contextual feedback is an important way of increasing engagement and sense of self-efficacy. Participants felt "happy" and "excited" when the Rewards Marketplace widget announced a booster (e.g., 100 extra points for answer correctly) and offered positive reinforcement feedback. However, the different features present in the widget were not properly understood by all participants. To improve its usability and drive further adoption, we recommend creating a specific lecture that explains why and how it can be used.

Flow appears to positively influence attitude towards learning. Participants with higher self-reported levels of flow showed a higher interest in continued learning. Now, this relationship may be mediated by other factors such as self-efficacy, which we deliberately attempted to influence through the introduction of rapid, contextual feedback, notification messages, and boosters in the form of extra points in case of good performance. To further study this point and others, the next section establishes a conceptual framework that ought to be tested in an experimental setting.

We contend that this platform is specifically helpful in raising awareness of the STS Method and the field of SRE and reaching broad audiences at scale within a limited time span (i.e., the time it takes to complete the tutorial). However, to add on our previous statement, it must be seen as complimentary to traditional teaching, part of the broader curriculum (Kleman, 2013), and not as a replacement.

Social is not a prerequisite for improved learning and engagement. Results showed that social features were underutilized and that social interactions were lacking as evidenced by e.g., students not interacting with each other during the on-site evaluation, despite being colleagues. Some authors have argued that gamification encourages anti-social behavior and lack of face-to-face time with peers (Marquis, 2013). Although our evidence is not conclusive to support or refute this view, it is worth mentioning that learners had to interact with headphones to hear the video lectures, thus effectively isolating themselves. We are inclined to say that social may not be necessary when absorbing conceptual or procedural knowledge in a relatively short timeframe (e.g., less than 3 hours).

A new version of the platform that fully leverages social components in the form of e.g., teamwork, collaborative play, or group-based competition and requires social interaction to be successful would be needed to validate these statements.

The STS-Tooltorial is more likely to convince students than IT professionals about the benefits of SRE and the STS Method. Interviews with professionals revealed that they found the method and modeling language too complex to use in a real context, where diagrams would tend to grow to the point of being unmanageable[6]. This explains why we noticed a significant difference in the intention to learn about SRE and the STS Method compared to students. It is worth noting that IT professionals worked for a large technology consulting services firm where projects have a bias for action over formal methods. Nonetheless, we should investigate further whether this perception is shared by other practitioners to better understand if this was due to the language itself or to some features of our tutorial[7]. This can be done by 1) broadening the sampling method and replicating the evaluation conducted here, and 2) conducting action research on the real-world applicability of the STS Method and STS-ml itself (Rosemann & Vessey, 2008).

8.2 Answers to Research Questions

At the beginning of our inquiry, the objective of this research was two-fold:

G1	**Improve** the **learning experience** and **increase** learner **engagement** while learning socio-technical security requirements engineering, compared to other traditional approaches

G2	Devise an assessment framework and guidelines for **determining** and **improving** the **quality** of security requirements models

To tackle **G1**, the following research question and sub-questions were formulated:

RQ1	How can gamification be best applied in the context of SRE to improve the learning experience and engagement of learners?

[6] It would be interesting to investigate further whether this common belief held by practitioners corresponds to reality. In our experience there is little evidence that diagrams grow dramatically in practice.

[7] Note that the first supervisor delivers full-day trainings with STS-ml that rely on an interplay of traditional teaching and hands-on sessions, while the in-tool tutorial has a more limited time span.

To effectively answer the the main **RQ1**, several sub-questions were posed. In conjunction, their answers provide researchers and society as a whole with a robust framework and insights that can 1) guide the design of future versions of the STS-Tooltorial, 2) inform the research agenda of academics in the fields of education, gamification and (S)RE, and 3) help practitioners create real-world artifacts. It is worth noting that our framework and findings with regard to this research question have been accepted for publication in a major international conference (see **Appendix V**).

- SQ1 What are the intended learning outcomes (ILOs) of (socio-technical) security requirements engineering for first-time learners?

We conducted a literature study and semi-structured interviews with 7 renowned experts in the field of (S)RE to extract intended learning outcomes for novice modelers interested in secure socio-technical systems design Although our final list of four ILOs were specifically particularized for STS Method, they can be used to inform the instructional design of other methods or modeling languages. Furthermore, we complement the ILOs with a set of best practices and heuristics that experts leverage to teach SRE and conceptual modeling, irrespective of the delivery method.

- SQ2 What aspects can be effectively gamified to improve the effectiveness of an interactive tutorial?

We explored this research question with a broad literature review on gamification in general and gamification of learning in particular. The investigation led to a 1) comprehensive overview of game elements and motivational affordances from both industry and academia (supported by social science and psychology theories such as SDT or CET), and 2) best practices for gamifying learning experiences. Most of these guidelines have greatly informed the design of the STS-Tooltorial, and positive evaluation results hint at their aggregated empirical validity.

- SQ3 How can an interactive tutorial for teaching a socio-technical security requirements method (viz., STS) be designed?

By studying the literature in innovative teaching methods (with a focus on computer-based tools), we provide the first account of its current state of the art within the (S)RE field. Moreover, the literature review on interactive tutorials helped us to distill high-level guidelines for creating such artifacts. The best practices are instantiated and complemented with the process for conceptualizing and developing the artifact reported in **Chapter 5**.

We thus argue that these deliverables, combined with the insights derived from answering **SQ1** and **SQ2**, come together into a holistic framework for building interactive tutorials for conceptual modeling languages. This framework can be readily applied in the field of conceptual modeling, and can potentially be re-used across other fields.

- SQ4 Do gamified tutorials show any promise in the case of security requirements engineering?

Results from our preliminary evaluation featuring a two-country (Spain, Netherlands), two-population (postgraduate students, IT professionals) sample suggest that the use of gamified interactive tutorials for teaching (S)RE conceptual modeling has the potential to increase learner engagement and intention to learn (see **Section 7.1** and **7.4**).

Our interactive platform represents the first successful attempt to use in-tool tutorials in (S)RE. In addition, this type of platform represents an innovative teaching method that stands as an effective competitor to both traditional and modern approaches.

For traceability purposes, below we include a brief summary of our initial hypothesis and their empirical evidence based on our evaluation. Again, note that hypothesis testing was not a major objective of our research (see **Section 4.2**).

H1: *Learners exposed to a gamified interactive tutorial will report high levels of engagement.*

The student group self-reported a higher than average flow during the completion of the tutorial (*M*=4.69, *SD*=1.03), with no significant difference with respect to the professional's flow (*M*=4.5, *SD*=0.90). We may conclude that participants reported above-average engagement, superior to other similar studies (Lombriser et al., 2016).

Therefore the hypothesis may be tentatively accepted.

H2: *The higher the engagement of learning during their participation in the gamified tutorial, the higher their performance will be.*

Most of the participants completed between 80% and 100% of the content with success. On average, participants were awarded 950 points out of 1170 (*SD*=203, range: 579-1130), but not upon first attempt: quiz accuracy of both students (48%, *SD*=13%) and IT professionals (53%, *SD*=9%) professionals was moderate.

Since we have no baseline to compare these results, the hypothesis cannot be tentatively accepted nor rejected at this moment.

H3: *Learners exposed to a gamified interactive tutorial will report a positive satisfaction towards the use of the system.*

The aggregated SUS scores of both samples indicate that the interactive platform has high usability and is well integrated within the modeling tool (M=80, SD=8.07), effectively making the STS-Tooltorial an "A" system at the 90th percentile.

The groups groups reported high degrees of half-way satisfaction (M=7.38, SD=1.19). This measure was found to be positively correlated to engagement (r=.71, n=13, p=.05). In addition, participants indicated a high intention to recommend (i.e., NPS score) the STS-Tooltorial (M=7.88, SD=1.46), with again no significant difference between professionals and students.

Consequently, the hypothesis may be tentatively accepted.

H4: *Deploying a gamified interactive tutorial positively influences intention to learn more about SRE and the STS Method.*

After the intervention, participants reported an neutral-to-low intention to learn more about SRE and the STS. Specifically, students had a significant higher intention to learn more about both SRE ($t(11)$=3.12, p=0.01, d=1.88) and the STS Method ($t(11)$=2.15, p=0.05, d=1.29) than professionals did. Lastly, engagement was found to be strongly correlated to intention to learn (r=.72, n=13, p=.04).

Therefore, the hypothesis cannot be accepted at this moment, and a more varied sample is necessary to determine the suitability of the platform for experienced professionals.

To tackle **G2**, the following research question and sub-questions were formulated:

RQ2 How can the quality of security requirements models be appraised and improved?

- **SQ5** What does quality of security requirements models mean?
- **SQ6** How can the quality of security requirements models be appraised?
- **SQ7** What improvement strategies can be executed to improve the quality of security requirements models?

While it is our contention that we have advanced the current body on knowledge regarding our first research question, **RQ2** remains largely unanswered. The artifact attempting to shed light on this question, namely the QuaSMOD Framework, represents a de-prioritized work product that warrants further validation and research. It is only included as a self-contained deliverable in **Appendix IV**.

8.3 Limitations

Regarding the main artifact of this thesis (i.e., STS-Tooltorial), the main limitation of this research from a pure scientific point of view is the lack of a rigorous experimental evaluation featuring both a treatment and control group. However, we believe we have successfully defended our case in **Section 6.3**, in that we opted to focus on a qualitative evaluation and on creating a platform that can be applied outside the laboratory. Further, we also address this limitation by proposing an approach to conduct a more formal, experimental inquiry in the next section. We also fell prey to both some of the limitations we exposed in **Section 3.2.3** when reviewing research on innovative teaching methods as well as some of those outlined by Hamari et al. (2014) in their review of empirical studies on gamification, namely: 1) sample size was small, 2) the evaluation lacked a experimental control and was solely based on user evaluation, and 4) evaluation timeframe was relatively short. However, this was a conscious decision based on time constraints, participant availability, and lack of resources.

On a related note, our scientific contribution with respect to the effectiveness of gamification in teaching is rather weak, and should be interpreted with caution. On the contrary, we have successfully established in-tool interactive tutorials as an alternative, innovative teaching method. The social relevance of these results should be encouraging for scholars and practitioners in the fields of instruction, learning, and conceptual modeling. However, more research is needed to both 1) validate the applicability of in-app tutorials in the broad field of conceptual modeling and 2) understand their external validity in other contexts.

As far as the STS-Tooltorial itself, some limitations are of note. First, since participants on average completed the tutorial within 83 minutes, we cannot derive conclusions from the long-term effects of (in-tool) gamified interactive tutorials. Second, with regard to quiz accuracy, we decided not to enforce a required pass grade because we wanted learners to keep going even when hampered by wrong quiz answers. It is hard to conclude whether this was the right approach. On the one hand, quiz accuracy results (48% on average) suggest this was a good decision, since otherwise learners would have had severe problems in progressing through the levels, thus impacting flow and engagement (i.e., learners who failed too many quizzes compared to their peers tended to report lower engagement scores). On the other hand, quiz failures may have contributed to an increase in the use of the Rewards Marketplace (i.e., to talk with the Game Master / SRE experts and to purchase hints), a feature that was rarely used.

Finally, it is worth mentioning that by the nature of the evaluation, participants were not representative users in that they were extrinsically motivated to finish the experiment on account of their voluntary participation in the study. Thus, the very nature of the evaluation may have confounded some of the results (e.g., completion rate, quiz retries) either positively or negatively.

8.4 Suggested Experimental Evaluation

Most of the studies reviewed did not perform any form of experimental evaluation (including ours), thus thwarting the possibility of generalizing results (Wohlin et al., 2012)..

To further advance our current understanding of the effectiveness of interactive tutorials for teaching SRE and conceptual modeling, the remainder of this chapter presents a revised conceptual model underpinned by literature and the relationships uncovered from our own statistical tests. In particular, **Section 8.4.1** introduces the concepts of interest and elaborates upon the relationships and hypotheses among them. Subsequently, **Section 8.4.2** introduces the improved version of the conceptual model, which can then be used to inform future experimental evaluations of the STS-Tooltorial. The discussion ends with an analysis of sample size requirements (**Section 8.4.3**).

8.4.1 New Relationships and Hypotheses

The Gamified Teaching Trifecta: Gamification, Engagement, and Task Performance

Using the same rationale as introduced in **Section 4.1.1**, H1 and H2 would now be stated as:

H1: *Learners exposed to a gamified interactive tutorial will report higher levels of engagement compared to those exposed to a non-gamified version.*

> **H2**: *If learners are more engaged while completing the gamified interactive tutorial, their performance will also increase.*

Observe that H1 and H2 can be combined into the following hypothesis to highlight the mediating role of engagement:

> **H0**: *Engagement mediates the relationship between gamification and task performance.*

A partial answer to this hypothesis has recently been reported in a controlled experiment by Lombriser, Dalpiaz, Lucassen, and Brinkkemper (2016). The authors found that, in the context of scenario-based RE, participants exposed to a gamification platform produced more creative and higher-quality requirements than those exposed to a non-gamified platform.

Gamification, Self-efficacy, and Task Performance

Albert Bandura's social cognitive theory (1986) revolves around the concept of self-efficacy, defined as the extent or strength of one's belief in one's own ability to complete tasks and reach goals. People enter an activities with varying grades of self-efficacy (which depend on e.g., prior experience, personal qualities) and, as they work on tasks, receive information on their performance. This information in turn affects their self-efficacy (Schunk, 1995). In this sense, games present many of the incentives that stimulate self-efficacy, such as goal setting, feedback, and rewards (Ryan, Rigby, & Przybylski, 2006). In fact, playing games has been known to increase a learner's sense of self-efficacy (Barab, Thomas, Dodge, & Carteaux, 2005). Given that gamification uses game design elements, it seems plausible to expect that the use of gamification may be positively related to higher self-efficacy levels. This leads to a new third hypothesis:

> **H3**: *Learners exposed to a gamified interactive tutorial will report higher self-efficacy levels compared to those exposed to a non-gamified version.*

Another question that can arise is which group, high or low in self-efficacy, could benefit more from the STS-Tooltorial? In other words, how does self-efficacy moderate the relationship between the deployment (or not) of gamification and engagement?

To begin with, Schunk (1991) notes that self-efficacy can be seen as a motivator, which indicates a positive influence on activities such as gamified training interventions. People who feel confident about their abilities visualize scenarios that provide positive feelings and support performance (Bandura, 1993). In addition, learners with high self-efficacy levels can show more effort and persistence in the task at hand (Bandura, 1997). Taken together, these two aspects may lead to higher motivation to complete the gamified training. Secondly, given that people with high self-efficacy levels are more committed when it comes to achieving goals (Bandura, 1997), this may be beneficial for the relationship between partaking in a gamified tutorial and engagement. In other words, self-efficacy may exert a moderating influence in such a way that those individuals with high self-efficacy will benefit more from the gamified tutorial with respect to their reported engagement. Authors such as Schunk and Zimmerman (1994) observed a positive relation between engagement levels and students' confidence and self-efficacy for achieving learning outcomes. Consequently, this argument leads to the fourth hypothesis:

> **H4**: *Self-efficacy moderates the relationship between the deployment of a gamified interactive tutorial and engagement, such that learners that score high in self-efficacy will report higher levels of engagement than those who score low in self-efficacy.*

Finally, the positive relationship between self-efficacy and task performance has been consistently reported in literature. Locke, Frederick, Lee, and Bokbo (1984) observed that self-efficacy was more strongly related to past performance than to future performance. Schunk and Hanson (1985) found a positive relationship between self-efficacy levels and student effort and rate of performance. More generally, meta-analytic evidence suggests that self-efficacy is strongly related to performance (Stajkovic & Luthans, 1998). Based on this theoretical evidence, the last hypothesis concerning self-efficacy that should be expected is:

> **H5**: *Self-efficacy is positively associated with task performance.*

Acceptance

While our previous model only explored intention to use, the revised model also considers attitude toward system use and rephrases satisfaction in terms of intention to use and intention to recommend.

For the purposes of this experiment, attitude toward using the system shall refer to the overall evaluation of the system's usage, either it being favorable or unfavorable (Fishbein & Ajzen, 1975; Ajzen, 1991). Several studies have shown the strong relationship that exists between attitudes and behavioural intention (e.g., Baker &

White, 2010; Lin & Bhattacherjee, 2010), the latter being the most proximate predictor of actual behavior (Ajzen, 1991).

Also, word-of-mouth (WOM) refers to an individual's willingness to recommend a product or service to others. It acts as a reflection of the person's satisfaction with the artifact in question (here, the STS-Tooltorial) and the trust that the service will continue to fulfil her expectations and those of her relatable peers (Kim & Son, 2009).

Therefore, we hypothesize the following:

- **H6**: *Deploying a gamified interactive tutorial positively influences the attitude towards use of the system.*
- **H7**: *Attitude positively influences intention to use the system.*
- **H8**: *Attitude positively influences intention to learn more about SRE.*
- **H9**: *Attitude positively influences intention to recommend the system.*

Control Variables

Several studies have shown that work or task experience are positively related to work performance (Quiñones, Ford, & Teachout, 1995; Littlepage, Robison, & Reddington, 1997). Furthermore, if we wish to influence attitude towards learning, current interest in SRE and socio-technical systems must also be accounted for. Therefore, these concepts are included as control variables when testing for H2 and H5.

8.4.2 Proposed Conceptual Model

Based on the concepts, relationships, and hypotheses set forth in the previous section, Figure 29 depicts the new conceptual model.

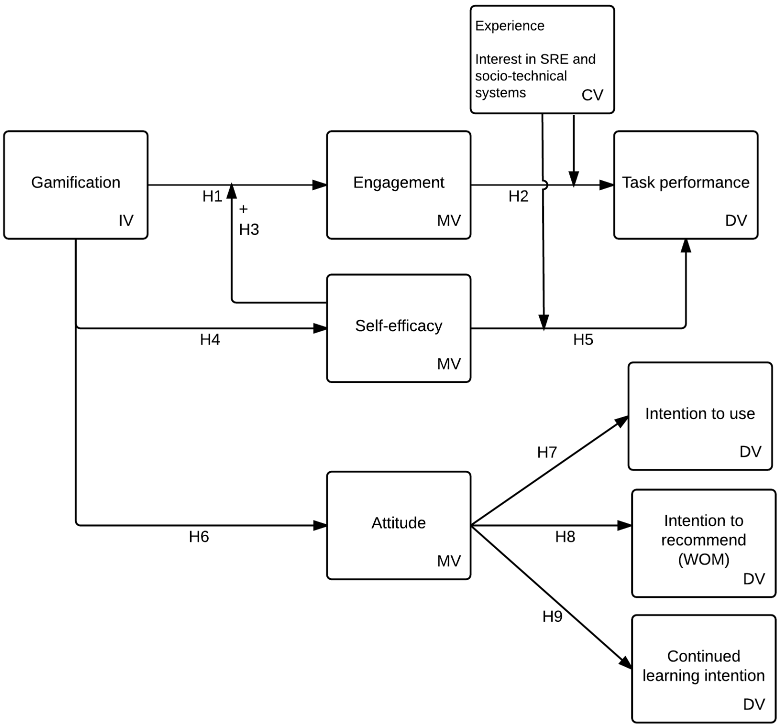


Figure 29 - Proposed Conceptual Model

8.4.3 Sample Size Analysis

The follow-up experiment outlined in this section must be wary of sample size limitations for it to be scientifically sound. To that end, below we report an analysis of sample size requirements.

An essential part of ensuring an adequate sample size is performing a power analysis. Power is mostly influenced by sample size and effect size (Cohen, 1992). It is wildly accepted that the determination of minimum sample size should occur before conducting an investigation (Cohen 1992). We relied on Cohen's approach for calculating sample size, which is based on three parameters, namely:

- **Effect size**: This parameter depends on each specific test, and can be categorized as small, medium, or large (Cohen, 1992). Literature review can inform the selection of the expected effect e.g., similar experiments or standard cut-offs. For instance, behavioral scientists use a medium effect (Cohen, 1992).
- **Power**: Defined as the probability that a given test will find an effect assuming that one exists in the population (Field, 2005, p. 58). Cohen (1992) recommends a minimum power of .8.
- **Significance criterion**: For all the tests, this criterion was set at the .05.

Conducting a power analysis for the conceptual model outlined above is rather complex. First, there is no literature we can draw on to estimate effect sizes or power, as research in this area is not based on solid experimental grounds. Second, the conceptual model calls for the execution of diverse statistical tests based on different methods and multiple independent and dependent variables, which further complicates the ability to select a worst-case scenario.

Nonetheless, to better inform future studies, two power analyses were conducted. The first was done prior to the research study to estimate sufficient sample size. Next, a post-hoc analysis was performed after the experiment to ascertain the actual power of the study. In the latter case we assumed that the effect size was the same in the sample as the one in the population.

The a priori study estimated minimum sample size based on significance, power, and effect size. The significance and power values were set at .05 and .8, respectively. To determine the expected effect size we broadened our scope to the field of software engineering. In their literature review Kampenes, Dybå, Hannay, and Sjøberg (2007) observed that effect sizes in SE are equal to those found psychology and slightly higher than those found in the behavioral sciences. Specifically, they reported standardized mean values for medium and large effect sizes of .6 and 1.40, respectively. Therefore, 1.0 was chosen as an optimistic value between these two, reflecting our confidence in detecting a medium-to-large effect as per the results reported in **Chapter 7**. Given this input parameters, the required sample size was 34 for a two-group t-test according to the software G*Power (Erdfelder, Faul, & Buchner, 1996). This value must be kept in mind if one is to derive solid scientific insights on future studies.

9. Conclusions and Future Research

9.1 Conclusions

Overall, we find that the platform shows promising results across different key areas. All participants completed between 80% and 100% of the content with success, thereby sufficiently reaching intended learning outcomes within the time-frame that we set for them[8]. Self-reported measures of usability, engagement, satisfaction, and intention to learn and recommend were positive and robust for both students and professionals, which entails that the STS-Tooltorial caters to diverse audiences. However, improvements are certainly possible and necessary: our effort should be seen a first-of-a-kind attempt to use in-tool tutorials in (security) requirements engineering.

This research has shown evidence that gamification can also improve a person's ability to comprehend digital learning material and understand certain areas of study. Specifically, our work sheds light on the applicability of gamified interactive tutorials for teaching conceptual modeling, an avenue that the (S)RE and gamification community had not explored hitherto. Specifically, the early findings reported as part of this thesis, although preliminary and suggestive at best, indicate that incorporating a gamification-powered platform into the teaching process leads to good results and possibly better reception of classes by the learners. Interestingly, we can conclude that our novel approach is perceived at least competitive with and as effective as other established methods for teaching conceptual modeling such classroom settings, MOOCs, or books/self-study..

9.2 Future Research

In the near future, we are planning on conducting a large-scale evaluation with graduate students, which we hope to report in an academic venue. The effectiveness of the platform can be further evaluated in other academic contexts (e.g., different universities, longer research timeframe, refined versions). Another possibility includes the implementation of the learning platform in an organizational setting, a form of action research that would arguably yield helpful insights about the applicability of in-app tutorials as well as the STS Method (Whyte, 1991).

---

[8] As we discuss in our limitations (**Section 8.3**) it is possible that not all learning outcomes were reached as a result of showing the correct answers of multiple-choice questions upon failure. A post-test should have been conducted to better ascertain this situation.

With regard to the STS-Tooltorial, it would be interesting to study the effects of social components such as group competition, teamwork, or collaborative play in interactive tutorials. Our results suggested that social is not a prerequisite, and possibly unnecessary when absorbing conceptual or procedural knowledge in a short timeframe.

Further, the current understanding regarding the effectiveness of gamification could be better understood by isolating and evaluating single game elements in an experimental context, rather than studying the overall compound effect. This, however, would be detrimental to the philosophy of our research, which is aimed at understanding the effect of innovative teaching methods that leverage many game elements and motivational affordances. The synergy of game elements is what often characterizes a good experience, not the individual elements.

Our long-term roadmap includes covering the entire STS Method with additional courses and extending the platform with other (S)RE-relevant modules. For example, a similar tutorial could be created for teaching about the recently released iStar 2.0 language for goal-oriented early RE (Dalpiaz, Franch, & Horkoff, 2016). In this matter, an important research question to address is to determine whether certain modeling languages are better adequate for learning via interactive tutorials.

We call for the joint support of the research community in helping advance innovative teaching methods such as this one, in the hope of more effectively engaging and training the next generation of (security) requirements engineers and conceptual modelers.

References

Adam, S. (2004). Using Learning Outcomes, Report for United Kingdom Bologna Seminar 1-2 July 2004, Heriot-Watt University (Edinburgh Conference Centre) Edinburgh.

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50*(2), 179–211. http://doi.org/10.1016/0749-5978(91)90020-T

Alexander, I. (2002). Initial industrial experience of misuse cases in trade-off analysis. Proceedings of the IEEE International Conference on Requirements Engineering (Vol. 2002-January, pp. 61–68). http://doi.org/10.1109/ICRE.2002.1048506

Anderson, J., Boyle, C., Farrell, R., & Reiser, B. (1987). Cognitive principles in the design of computer tutors. In Modelling Cognition (pp. 93–133). Retrieved from http://act-r.psy.cmu.edu/papers/121/CogPrinciples.pdf

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, *4*(2), 167–207. http://doi.org/10.1207/s15327809jls0402_2

Anderson, J. & Rainie, L. (2012). *The Future of Gamification. Pew Research Center: Internet, Science & Tech*. Retrieved 21 April 2016, from http://www.pewinternet.org/2012/05/18/the-future-of-gamification/

Andrews, T. (2012). What is social constructionism. *Grounded theory review*, *11*(1), 39-46.

Antin, J., & Churchill, E. (2011). Badges in social media: A social psychological perspective. *Chi 2011*, 1–4. Retrieved from http://uxscientist.com/public/docs/uxsci_2.pdf

Anton, A. I. (1996). Goal-based requirements analysis. In Proceedings of the Second International Conference on Requirements Engineering (pp. 136–144). http://doi.org/10.1109/ICRE.1996.491438

Anton, A. I., & Earp, J. B. (2000). Strategies for developing policies and requirements for secure electronic commerce systems. *E-Commerce Security and Privacy*, *2*, 29–46.

Babar, Z., Nalchigar, S., Lessard, L., Horkoff, J., & Eric, S. K. (2015). Instructional Experiences with Modeling and Analysis using the i* Framework. In *iStarT@ CAiSE* (pp. 31-36).

Baker, R., & White, K. (2010). Predicting adolescents' use of social networking sites from an extended theory of planned behaviour perspective. *Computers in Human Behavior*, *26*, 1591-1597. doi:10.1016/j.chb.2010.06.006

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Upper Saddle River, NJ: Prentice Hall.

Bandura, A. (1993). Perceived Self-Efficacy in Cognitive Development and Functioning. *Educational Psychologist*, *28*, 117-148. http://doi.org/10.1207/s15326985ep2802_3

Bandura, A. (1997). *Self-Efficacy: The Exercise of Control*. New York, NY: Freeman.

Barab, S., Thomas, M., Dodge, T., Carteaux, R., & Tuzun, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development*, *53*(1), 86–107. http://doi.org/10.1007/BF02504859

Barjis, J., Gupta, A., Sharda, R., Bouzdine-Chameeva, T., Lee, P. D., & Verbraeck, A. (2012). Innovative teaching using simulation and virtual environments. *Interdisciplinary Journal of Information, Knowledge, and Management*, *7*, 237–255.

Basin, D., Clavel, M., & Egea, M. (2011). A Decade of Model-driven Security. Proceedings of the 16th ACM Symposium on Access Control Models and Technologies (pp. 1–10). New York, NY: ACM. http://doi.org/10.1145/1998441.1998443

Baxter, G., & Sommerville, I. (2011). Socio-technical Systems: From Design Methods to Systems Engineering. *Interacting with Computers*, *23*(1), 4–17. http://doi.org/10.1016/j.intcom.2010.07.003

Berdichevsky, D., & Neuenschwander, E. (1999). Toward an ethics of persuasive technology. *Communications of the ACM*, *42*(5), 51–58. http://doi.org/10.1145/301353.301410

Beus-Dukic, L. (2011). Final year project: A test case for requirements engineering skills. *6th International Workshop on Requirements Engineering Education and Training*, 5–8. http://doi.org/10.1109/REET.2011.6046272

Bezivin, J., France, R., Gogolla, M., Haugen, O., Taentzer, G., & Varro, D. (2010). Teaching Modeling: Why, when, What?. Proceedings of the 2009 International Conference on Models in Software Engineering (pp. 55–62). Berlin, Heidelberg: Springer-Verlag. http://doi.org/10.1007/978-3-642-12261-3_6

Bjork, S., & Holopainen, J. (2004). *Patterns in Game Design (Game Development Series)*. Rockland, MA: Charles River Media, Inc.

Bloom, B. S. (Ed.). (1956). *Taxonomy of Educational Objectives: The Classificaton of Educational Goals*. Longmans, Green.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). Taxonomy of educational objectives, handbook I: The cognitive domain.

Bogost, I. (2011). *Gamification is Bullshit.* Retrieved 21 April 2016, from http://bogost.com/writing/blog/gamification_is_bullshit/

Börstler, J., Kuzniarz, L., Alphonce, C., Sanders, W. B., & Smialek, M. (2012). Teaching Software Modeling in Computing Curricula. Proceedings of the Final Reports on Innovation and Technology in Computer Science Education 2012 Working Groups (pp. 39–50). New York, NY: ACM. http://doi.org/10.1145/2426636.2426640

Braber, F., Hogganvik, I., Lund, M. S., Stølen, K., & Vraalsen, F. (2007). Model-based security analysis in seven steps - A guided tour to the CORAS method. *BT Technology Journal*, *25*(1), 101–117. http://doi.org/10.1007/s10550-007-0013-9

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203–236. http://doi.org/10.1023/B:AGNT.0000018806.20944.ef

Browne, G. J., & Ramesh, V. (2002). Improving information requirements determination: a cognitive perspective. *Information & Management*, *39*(8), 625-645.

Brucker, A. D., Hang, I., Lückemeyer, G., & Ruparel, R. (2012). SecureBPMN: modeling and enforcing access control requirements in business processes. *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*, 123–126. http://doi.org/10.1145/2295136.2295160

BuddyBoss. (2017). *BuddyBoss Themes*. Retrieved 3 April 2017 from https://www.buddyboss.com/

Burr, V. (2003). *Social constructivism*. New York, NY: Routledge.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*. Hoboken, NJ: Wiley Publishing. http://doi.org/978-0-471-95869-7

Buschmann, F., Henney, K., & Schmidt, D. C. (2007). *Pattern Oriented Software Architecture Volume 5: On Patterns and Pattern Languages*. http://doi.org/10.1093/intimm/dxu027

Carini, R. M., Kuh, G. D., & Klein, S. P. (2006). Student engagement and student learning: Testing the linkages. *Research in Higher Education*, *47*(1), 1–32. http://doi.org/10.1007/s11162-005-8150-9

Carroll, J. M. (1990). *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Cambridge, MA: MIT Press.

Carroll, J. M., & Carrithers, C. (1984). Training Wheels in a User Interface. Commun. *ACM*, *27*(8), 800–806. http://doi.org/10.1145/358198.358218

Carroll, J. M., & Van Der Meij, H. (1996). Ten misconceptions about minimalism. *IEEE Transactions on Professional Communication*, *39*(2), 72–86. http://doi.org/10.1109/47.503271

Charette, R. N. (2005). Why software fails. *IEEE Spectrum, 42*(9), 36–43. http://doi.org/10.1109/MSPEC.2005.1502528

Chou, Y. (2015). *Actionable gamification : beyond points, badges, and leaderboards.* Octalysis.

Cialdini, R. B. (2007). *Influence: The psychology of persuasion*. New York, NY: Collins.

Clark, D. (2015). *Bloom's Taxonomy: The Psychomotor Domain*. Retrieved 2 April 2017, from http://www.nwlink.com/~donclark/hrd/Bloom/psychomotor_domain.html

Cline, A. (2015). *Agile development in the real world*. New York, NY: Apress.

Cohen, J. (1992). A power primer. *Psychological Bulletin*, *112*(1), 155-159. doi: 10.1037/0033-2909.112.1.155

Connolly, T. M., Stansfield, M., & Hainey, T. (2007). An application of games-based learning within software engineering. *British Journal of Educational Technology*, *38*(3), 416–428. http://doi.org/10.1111/j.1467-8535.2007.00706.x

Consalvo, M. (2009). There is No Magic Circle. *Games and Culture*, *4*(4), 408–417. http://doi.org/10.1177/1555412009343575

Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014). *About Face: The Essentials of Interaction Design* (4th ed.). Hoboken, NJ: Wiley Publishing.

Corbetta, P. (2003). *Social research: Theory, methods and techniques*. Thousand Oaks, CA: Sage.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York, NY: Harper & Row.

Dalpiaz, F., Franch, X., & Horkoff, J. (2016). iStar 2.0 language guide. *arXiv preprint arXiv:1605.07767*.

Dalpiaz, F., Paja, E., & Giorgini, P. (2016). *Security Requirements Engineering: Designing Secure Socio-Technical Systems*. Cambridge, MA: MIT Press.

Damian, D., Hadwin, A., & Al-Ani, B. (2006). Instructional design and assessment strategies for teaching global software development: A framework. In Proceedings - International Conference on Software Engineering (Vol. 2006, pp. 685–690).

Dardenne, A., Van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of computer programming*, *20*(1), 3-50.

Dave, R. H. (1970). Psychomotor levels. *Developing and writing educational objectives*, 33-34.

De Schutter, B., & Abeele, V. (2014). Gradequest: Evaluating the impact of using game design techniques in an undergraduate course. In T. Barnes, & I. Bogost (Eds.), *Foundations of Digital Games (FDG 2014)*. Fort Lauderdale, FL. Retrieved April 22, 2016, from http://www.fdg2014.org/papers/fdg2014_paper_07.pdf

Deci, E. L., Koestner, R., & Ryan, R. M. (1999). A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, *125*, 627–668.

Deci, E., & Ryan, R. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. Berlin, Heidelberg: Springer.

Deci, E., & Ryan. R. (2011). Self-determination theory. Handbook of theories of social psychology, 1, 416-433.

Deci, E., & Ryan, R. (2012). Motivation, Personality, and Development Within Embedded Social Contexts: An Overview of Self-Determination Theory.

The Oxford Handbook of Human Motivation. http://doi.org/10.1093/oxfordhb/9780195399820.013.0006

Deterding, S. (2011). Situated motivational affordances of game elements: A conceptual model. Workshop at CHI.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness : Defining "Gamification". In Envisioning Future Media Environments (pp. 9–11). http://doi.org/10.1145/2181037.2181040

Deterding, S. (2014). Eudaimonic design, or: Six invitations to rethink gamification.

Dhillon, M., & Dasgupta, S. (2011). Individual Differences and Conceptual Modeling Task Performance: Examining the Effects of Cognitive Style, Self-efficacy, and Application Domain Knowledge. *Enterprise, Business-Process and Information Systems Modeling*, *81*, 483–496. http://doi.org/10.1007/978-3-642-21759-3_35

Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in Education: A Systematic Mapping Study. *Journal of Educational Technology & Society*, *18*(3), 75–88.

Dobrian, F., Sekar, V., Awan, A., Stoica, I., Joseph, D., Ganjam, A., ... & Zhang, H. (2011). Understanding the impact of video quality on user engagement. In *ACM SIGCOMM Computer Communication Review* (Vol. 41, No. 4, pp. 362-373). ACM.

Dubois, E., & Mouratidis, H. (2010). Guest editorial: security requirements engineering: past, present and future. *Requirements Engineering, 15*(1), 1–5. http://doi.org/10.1007/s00766-009-0094-8

Dupret, G., & Lalmas, M. (2013, February). Absence time and user engagement: evaluating ranking functions. In *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 173-182). ACM.

Ehret, B. D. (2002). Learning where to look: Location learning in Graphical User Interfaces. Proceedings of the SIGCHI Conference on Human Factors in

Computing Systems Changing Our World, Changing Ourselves - CHI ’02, (4), 211. http://doi.org/10.1145/503376.503414

Eisenhardt, K. M.. (1989). Building Theories from Case Study Research. *The Academy of Management Review*, *14*(4), 532–550. Retrieved from http://www.jstor.org/stable/258557

Erdfelder, E., Faul, F., & Buchner, A. (1996). GPOWER: A general power analysis program. *Behavior Research Methods, Instruments, & Computers*, *28*, 1-11.

Fernquist, J., Grossman, T., & Fitzmaurice, G. W. (2011). Sketch-sketch revolution: an engaging tutorial system for guided sketching and application learning. In J. S. Pierce, M. Agrawala, & S. R. Klemmer (Eds.), UIST (pp. 373–382). ACM.

Field, A. (2005). *Discovering statistics using SPSS* (2nd ed.). London: Sage Publications.

Firesmith, D. (2003). Security Use Cases. *Journal of Object Technology*, *2*(3), 53–64. Retrieved from http://dblp.uni-trier.de/db/journals/jot/jot2.html#Firesmith03b

Fishbein, M., & Ajzen, I. (1975). Belief, attitude, intention, and behavior: an introduction to theory and research. Reading, MA: Addison-Wesley.

Fogg, B. J. (2003). Persuasive Technology: Using Computers to Change What We Think and Do. Persuasive Technology Using Computers to Change What We Think and Do (Vol. 5). http://doi.org/10.4017/gt.2006.05.01.009.00

Fullerton, T. (2008). *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Amsterdam: Morgan Kaufman.

Gabrysiak, G., Giese, H., Seibel, A., & Neumann, S. (2010). Teaching requirements engineering with virtual stakeholders without software engineering knowledge. In Requirements Engineering Education and Training (REET), 2010 5th International Workshop on (pp. 36–45). http://doi.org/10.1109/REET.2010.5633109

Gabrysiak, G., Giese, H., & Seibel, A. (2011). Why should I help you to teach requirements engineering? In Requirements Engineering Education and Training (REET), 2011 6th International Workshop on (pp. 9–13). http://doi.org/10.1109/REET.2011.6046271

Galletta, A. & Cross, W. E.(2013). *Mastering the Semi-Structured Interview and Beyond: From Research Design to Analysis and Publication*. New York: NYU Press.

Gibson J J. (1977). *The theory of affordances*. Hoboken, NJ: John Wiley & Sons, Inc.

Giorgini, P., Massacci, F., Mylopoulos, J., & Zannone, N. (2004). Requirements Engineering Meets Trust Management, Model, Methodology, and Reasoning. Lecture Notes in Computer Science (Vol. 2995, pp. 176–190).

Giorgini, P., Massacci, F., Mylopoulos, J., & Zannone, N. (2005). Modeling security requirements through ownership, permission and delegation. Proceedings of the 13th IEEE International Conference on Requirements Engineering. (pp. 167–176). http://doi.org/10.1109/RE.2005.43

Goffman, E. (1986). *Frame Analysis: An Essay on the Organization of Experience*. Boston, MA: Northeastern University Press.

González, C., & Area, M. (2013). Breaking the rules: Gamification of learning and educational materials. In Proceedings of the 2nd international workshop on interaction design in educational environments (pp. 7-53).

Grossman, T., & Fitzmaurice, G. (2010). ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1515–1524. http://doi.org/10.1145/1753326.1753552

Haley, C. B., Laney, R., Moffett, J. D., & Nuseibeh, B. (2008). Security Requirements Engineering: A Framework for Representation and Analysis. *IEEE Transactions on Software Engineering*, *34*(1), 133–153. http://doi.org/10.1109/TSE.2007.70754

Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work?--a literature review of empirical studies on gamification. 47th Hawaii International Conference on System Sciences, 3025–3034. http://doi.org/10.1109/HICSS.2014.377

Hambrick, D. Z., & Engle, R. W. (2002). Effects of domain knowledge, working memory capacity, and age on cognitive performance: an investigation of the knowledge-is-power hypothesis. *Cognitive Psychology*, *44*(4), 339–387. http://doi.org/10.1006/cogp.2001.0769

Harrell, M. C., & Bradley, M. A. (2009). *Data collection methods. Semi-structured interviews and focus groups*. Santa Monica, CA: RAND.

Harrow, A. J. (1972). *A taxonomy of the psychomotor domain*. D. McKay Company.

Harter, J. K., Schmidt, F. L., & Hayes, T. L. (2002). Business-unit-level relationship between employee satisfaction, employee engagement, and business outcomes: a meta-analysis. *Journal of applied psychology*, *87*(2), 268.

Harter, J. K., Schmidt, F. L., Killham, E. A., & Agrawal, S. (2013). *The relationship between engagement at work and organizational outcomes*. Washington: Gallup Poll Consulting University Press.

Hatebur, D., Heisel, M., & Schmidt, H. (2007). A Security Engineering Process based on Patterns. *18th International Conference on Database and Expert Systems Applications*, *1*, 734–738. http://doi.org/10.1109/DEXA.2007.36

Heeter, Carrie, Lee, Magerko, & Medler. (2011). Impacts of Forced Serious Game Play on Vulnerable Subgroups. *International Journal of Gaming and Computer-Mediated Simulations*, *3*(3), 34–53.

Hernan, S., Lambert, S., Ostwald, T., & Shostack, A. (2006). Uncover security design flaws using the STRIDE approach. *MSDN Magazine*. Retrieved from http://msdn.microsoft.com/en-us/magazine/cc163519.aspx

Hernán, I., Lázaro, C. A., & Velázquez, J. A. (2004). On the use of Bloom's taxonomy as a basis to design educational software on programming. In

Proceedings of World Conference on Engineering and Technology Education, WCETE 2004 (pp. 351-355).

Herrmann, A. (2013). Requirements engineering in practice: There is no requirements engineer position. In Lecture Notes in Computer Science (Vol. 7830 LNCS, pp. 347–361). http://doi.org/10.1007/978-3-642-37422-7_25

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. http://doi.org/10.2307/25148625

Hoffmann, A., & Weißbach, R. (2014). ImProject: Improving RE training by combining it with improvisation theatre sessions. In CEUR Workshop Proceedings (Vol. 1217, pp. 11–19).

Hoo, K. S., Sudbury, A. W., & Jaquith, A. R. (2001). Tangible ROI through secure software engineering. *Secure Business Quarterly*, *1*(2), 1-4.

Horn, R. E. (1999). *Horn's Review of the Nurnberg Funnel*. Stanford. Retrieved from http://web.stanford.edu/~rhorn/a/topic/stwrtng_infomap/RvwOfNurnbrgFnnl.html

Huotari, K., & Hamari, J. (2012). Defining Gamification: A Service Marketing Perspective. In Proceeding of the 16th International Academic MindTrek Conference (pp. 17–22). New York, NY: ACM. http://doi.org/10.1145/2393132.2393137

Huynh-Kim-Bang, B., Wisdom, J., & Labat, J. M. (2010). Design patterns in serious games: A blue print for combining fun and learning. *Project SE-SG*.

Jackson, M. (2001). *Problem Frames: Analyzing and Structuring Software Development Problems*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Jackson, S. A., Martin, A. J., & Eklund, R. C. (2008). Long and short measures of flow: The construct validity of the FSS-2, DFS-2, and new brief counterparts. *Journal of Sport and Exercise Psychology*, *30*(5), 561-587.

Jacobson, I. (1992). *Object-oriented Software Engineering*. New York, NY: ACM.

Jalali, S., & Wohlin, C. (2012). Systematic literature studies: database searches vs. backward snowballing. *Proceedings of the 2012 6th ACM_IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 29–38. http://doi.org/10.1145/2372251.2372257

Jick, T. D.. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Administrative Science Quarterly*, *24*(4), 602–611. http://doi.org/10.2307/2392366

Johnson, D. G. (2006). Computer systems: Moral entities but not moral agents. In Ethics and Information Technology (Vol. 8, pp. 195–204). http://doi.org/10.1007/s10676-006-9111-5

Johnstone, K. (2012). Impro: *Improvisation and the Theatre*. London: Taylor & Francis.

Jones, C. (1986). *Programming Productivity*. New York: McGraw-Hill.

Jung, J. H., Schneider, C., & Valacich, J. (2010). Enhancing the Motivational Affordance of Information Systems: The Effects of Real-Time Performance Feedback and Goal Setting in Group Collaboration Environments. *Management Science*, *56*(4), 724–742. http://doi.org/10.1287/mnsc.1090.1129

Jürjens, J. (2002). UMLsec: Extending UML for secure systems development. Proceedings of the 5th International Conference on The Unified Modeling Language, 412–425. http://doi.org/10.1007/3-540-45800-X_32

Juul, J. (2011). *Gamification Backlash Roundup.* Retrieved 21 April 2016, from https://www.jesperjuul.net/ludologist/gamification-backlash-roundup

Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.

Kahneman, D., Knetsch, J. L., & Thaler, R. H. (1991). Anomalies: The endowment effect, loss aversion, and status quo bias. *The journal of economic perspectives*, *5*(1), 193-206.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the econometric society*, 263-291.

Kampenes, V. B., Dybå, T., Hannay, J. E., & Sjøberg, D. I. K. (2007). Systematic Review: A Systematic Review of Effect Size in Software Engineering Experiments. *Information and Software Technology*, *49*(11-12), 1073–1086. http://doi.org/10.1016/j.infsof.2007.02.015

Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. Hoboken, NJ: John Wiley & Sons.

Karlsson, J., & Ryan, K. (1997). A cost-value approach for prioritizing requirements. *IEEE software*, *14*(5), 67-74.

Kennedy, D., Hyland, A., & Ryan, N. (2006). Writing and using learning outcomes, Bologna handbook, implementing Bologna in your institution.

Kenrick, D. T., Neuberg, S. L., & Cialdini, R. B. (2009). *Social Psychology: Unraveling the mystery*. Boston: Allyn and Bacon.

Kim, A. J. (2008). *Putting the fun in functional: applying game mechanics to functional software*. Retrieved from http://slideshare.net/amyjokim/putting-the-fun-in-functiona/

Kim, S. S., & Son, J. Y. (2009). Out of dedication or constraint? A dual model of post-adoption phenomena and its empirical test in the context of online services. *MIS Quarterly*, 49-70.

Kleman, M. (2013). *The Debate: Gamification and Education*. Retrieved April 2, 2017, from http://technologyadvice.com/blog/information-technology/debate-gamification-education/

Kohn, A. (1999). Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes. Boston, MA: Houghton Mifflin Harcourt.

Koven, B. (2013). *The Well-Played Game: A Player's Philosophy*. Cambridge, MA: MIT Press.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, *41*(4), 212-218.

Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1964). *Taxonomy of educational objectives, handbook II: affective domain*. New York: David McKay Company. Inc.

Krogstie, J., Lindland, O. I., & Sindre, G. (1995). Defining quality aspects for conceptual models. In *Information System Concepts* (pp. 216-231). Springer US.

Kuh, G. D., Jankowski, N., Ikenberry, S. O., & Kinzie, J. (2014). *Knowing What Students Know and Can Do: The Current State of Student Learning Outcomes Assessment in US Colleges and Universities*. Urbana, IL: University of Illinois and Indiana University.

Lamsweerde, A. (2004). Elaborating Security Requirements by Construction of Intentional Anti-Models. In ICSE '04: Proceedings of the 26th International Conference on Software Engineering (pp. 148–157). Washington, DC: IEEE Computer Society.

Layton, E. T. (1974). Technology as Knowledge. *Technology and Culture*, *15*(1), 31. http://doi.org/10.2307/3102759

Lewis, J., & Sauro, J. (2009). The factor structure of the system usability scale. *Human centered design*, 94-103.

Li, W., Grossman, T., & Fitzmaurice, G. (2012). GamiCAD: A Gamified Tutorial System for First Time Autocad Users. In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (pp. 103–112). New York, NY: ACM. http://doi.org/10.1145/2380116.2380131

Lin, C. P., & Bhattacherjee, A. (2010). Extending technology usage models to interactive hedonic technologies: A theoretical model and empirical test. *Information Systems Journal*, *20*(2), 163–181. http://doi.org/10.1111/j.1365-2575.2007.00265.x

Lin, L., Nuseibeh, B., Ince, D., Jackson, M., & Moffett, J. (2003). Introducing abuse frames for analysing security requirements. Proceedings of the IEEE International Conference on Requirements Engineering (Vol. 2003-Janua, pp. 371–372). http://doi.org/10.1109/ICRE.2003.1232791

Littlepage, G., W. Robison and K. Reddington (1997). Effects of task experience and group experience on group performance, member ability, and recognition of expertise. *Organizational Behavior and Human Decision Processes*, *69*(2), 133-147.

Liu, L., Yu, E., & Mylopoulos, J. (2003). Security and privacy requirements analysis within a social setting. Proceedings 11th IEEE International Requirements Engineering Conference, 3, 151–161. http://doi.org/10.1109/ICRE.2003.1232746

Locke, E. A., Frederick, E., Lee, C., & Bobko, P. (1984). Effect of self-efficacy, goals, and task strategies on task performance. *Journal of Applied Psychology*, *69*(2), 241–251. http://doi.org/10.1037/0021-9010.69.2.241

Lodderstedt, T., Basin, D., & Doser, J. (2002, September). SecureUML: A UML-based modeling language for model-driven security. In *International Conference on the Unified Modeling Language* (pp. 426-441). http://doi.org/10.1007/3-540-45800-X_33

Lombriser, P., Dalpiaz, F., Lucassen, G., & Brinkkemper, S. (2016). Gamified requirements engineering: model and experimentation. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 171-187). Springer International Publishing.

Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2015, August). Forging high-quality user stories: towards a discipline for agile requirements. In *Requirements Engineering Conference (RE), 2015 IEEE 23rd International* (pp. 126-135). IEEE.

Macaulay, L., & Mylopoulos, J. (1995). Requirements Engineering: an educational dilemma. *Automated Software Engineering*, *2*(4), 343-351.

Mahaux, M., & Maiden, N. (2008). Theater improvisers know the requirements game. *IEEE Software*, *25*(5), 68–69. http://doi.org/10.1109/MS.2008.128

Malone, T. W. (1980). What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games. In Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems (pp. 162–169). New York, NY: ACM. http://doi.org/10.1145/800088.802839

Marquis, J. (2013) *Debates about Gamification and Game-Based Learning (#GBL) in Education*. Retrieved April 2, 2017, from http://classroom-aid.com/2013/04/07/debates-about-gamification-and-game-based-learninggbl-in-education/

Mayer, N., Rifaut, A., Dubois, E., Recherche, C., & Tudor, P. H. (2005). Towards a Risk-Based Security Requirements Engineering Framework. In In Proc. of REFSQ'05.

McConnell, S. (2001). *An Ounce of Prevention. IEEE Software*, *18*(3), 5–7. http://doi.org/10.1109/MS.2001.922718

McDermid, D. (2006). *Pragmatism*. Retrieved from http://www.iep.utm.edu/pragmati/

McDermott, J., & Fox, C. (1999). Using abuse case models for security requirements analysis. Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99), 55–64. http://doi.org/10.1109/CSAC.1999.816013

McGraw, G. (2006). *Software Security: Building Security In*. Boston: Addison-Wesley Professional.

McPhail, C. J. (Ed.). (2005). *Establishing and sustaining learning-centered community colleges*. Washington, DC: American Association of Community Colleges.

Mead, N., Hough, E., & Stehney, T. (2005). *Security Quality Requirements Engineering* (CMU/SEI-2005-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Mead, N. R., Shoemaker, D., & Ingalsbe, J. (2009). Teaching Security Requirements Engineering Using SQUARE. Requirements Engineering Education and Training (REET), 2009 Fourth International Workshop on (pp. 20–27). http://doi.org/10.1109/REET.2009.12

Mellado, D., Fernández-medina, E., & Piattini, M. (2006). Applying a Security Requirements Engineering Process. In Computer Security – ESORICS 2006 (Vol. 4189, pp. 192–206). http://doi.org/10.1109/ARES.2006.2

Mellado, D., Fernández-Medina, E., & Piattini, M. (2007). A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards & Interfaces*, *29*(2), 244–253. http://doi.org/10.1016/j.csi.2006.04.002

Merten, T., Schäfer, T., & Bürsner, S. (2012). Using RE knowledge to assist automatically during requirement specification. IEEE 7th International Workshop on Requirements Engineering Education and Training (REET) (pp. 9–13). http://doi.org/10.1109/REET.2012.6360068

Mich, L. (2014). Teaching requirements analysis: A student project framework to bridge the gap between business analysis and software engineering. In 8th International Workshop on Requirements Engineering Education and Training, REET 2014 (Vol. 1217, pp. 20–25).

Microsoft. (2005). *The STRIDE Threat Model*. Retrieved from https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

Microsoft. (2007). *Microsoft Threat Analysis & Modeling v2.1.2*. Retrieved from https://www.microsoft.com/en-us/download/details.aspx?id=14719

Microsoft.com. (2011). *Download Ribbon Hero 2*. Retrieved 21 April 2016, from https://www.microsoft.com/en-us/download/details.aspx?id=26531

Microsoft.com. (2016). *Benefits of the SDL*. Retrieved 21 April 2016, from https://www.microsoft.com/en-us/SDL/about/benefits.aspx

MIT. (2016). *Intended Learning Outcomes*. Retrieved 21 April 2016, from http://tll.mit.edu/help/intended-learning-outcomes

Moffett, J. D., Haley, C. B., & Nuseibeh, B. (2004). *Core Security Requirements Artefacts*. Technical Report. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.3926&rep=rep1&type=pdf

Mollick, E. R., & Rothbard, N. (2014). Mandatory fun: consent, gamification and the impact of games at work. *The Wharton School research paper series*.

Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, *55*(3), 243-276.

Moore, A. P., Ellison, R. J., & Linger, R. C. (2001). Attack modeling for information security and survivability. *Technical Note CMUSEI2001TN001*, *17*, 15–33. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.9283&rep=rep1&type=pdf

Mouratidis, H., & Giorgini, P. (2007). Secure Tropos: a Security-Oriented Extension of the Tropos Methodology. *International Journal of Software Engineering and Knowledge Engineering*, *17*(02), 285–309. http://doi.org/10.1142/S0218194007003240

Muntean, C. C. I. (2011). Raising engagement in e-learning through gamification. *The 6th International Conference on Virtual Learning ICVL 2011*, 1, 323–329. Retrieved from http://icvl.eu/2011/disc/icvl/documente/pdf/met/ICVL_ModelsAndMethodologies_paper42.pdf

Nah, F. F. H., Zeng, Q., Telaprolu, V. R., Ayyappa, A. P., & Eschenbrenner, B. (2014). Gamification of education: A review of literature. In Lecture Notes in Computer Science (Vol. 8527 LNCS, pp. 401–409). http://doi.org/10.1007/978-3-319-07293-7_39

Navarro, E. O., & Van Der Hoek, A. (2004). SimSE: An interactive simulation game for software engineering education. Proceedings of the Seventh IASTED International Conference on Computers and Advanced Technology in Education (pp. 12–17).

Navarro, E., & Hoek, A. Van Der. (2009). Multi-site evaluation of SimSE. *Technical Symposium on Computer Science Education*, *41*(1), 4. http://doi.org/10.1145/1539024.1508981

Neimeyer, R. A. (1998). Social constructionism in the counselling context. *Counselling Psychology Quarterly*, *11*(2), 135-149.

Nelson, H. J., Poels, G., Genero, M., & Piattini, M. (2012). A conceptual modeling quality framework. *Software Quality Journal*, *20*(1), 201–228. http://doi.org/10.1007/s11219-011-9136-9

O'Donovan, S., Gain, J., & Marais, P. (2013, October). A case study in the gamification of a university-level games development course. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 242-251). ACM.

Oliveira, A., Werneck, V. M. B., Leite, J. C. S., & Cysneiros, L. M. (2015). The Monopoly Game to Teach ERi*c - Intentional Requirements Engineering. In J. Horkoff, J. Lockerbie, X. Franch, E. S. K. Yu, & J. Mylopoulos (Eds.), *Proceedings of the 1st International iStar Teaching Workshop co-located with the 27th International Conference on Advanced Information Systems Engineering (CAiSE 2015)*, Stockholm, Sweden, (Vol. 1370, pp. 49–54).

Omg.org. (2015). *BPMN 2.0*. Retrieved 21 April 2016, from http://www.omg.org/spec/BPMN/2.0/

OWASP. (2016). *OWASP CLASP Project - OWASP*. Retrieved from http://www.owasp.org/index.php/Category:OWASP_CLASP_Project

Park, I., & Hannafin, M. J. (1993). Empirically Based Guidelines for the Design of Interactive Multimedia. *Educational Technology Research and Development*, *41*(3), 61–85. http://doi.org/10.1007/BF02297358

Patsula, P. J. (1999). *Applying learning theories to online instructional design.* University of Seoul. Retrieved from http://www.patsula.com/usefo/webbasedlearning/

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research.

Journal of Management Information Systems, *24*(3), 45–77. http://doi.org/10.2307/40398896

Peng, L., & de Graaf, O. (2010). Experiences of using role playing andwiki in requirements engineering course projects. In Requirements Engineering Education and Training (REET), 2010 5th International Workshop on (pp. 1–6).

Penzenstadler, B., Mahaux, M., & Heymans, P. (2013). University meets industry: Calling in real stakeholders. IEEE 26th Conference on Software Engineering Education and Training (CSEET) (pp. 1–10). http://doi.org/10.1109/CSEET.2013.6595231

Pink, D. H. (2011). *Drive: The surprising truth about what motivates us*. Penguin.

Plaisant, C., & Shneiderman, B. (2005). Show Me! Guidelines for producing recorded demonstrations. In 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05) (pp. 171–178). http://doi.org/10.1109/VLHCC.2005.57

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques* (1st ed.). Berlin, Heidelberg: Springer Publishing.

Ponemon. (2015). *2015 Cost of Data Breach Study: Global Analysis*. Www-01.ibm.com. Retrieved 21 April 2016, from http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=SEW03053WWEN

Quiñones, M. A., Ford, J. K., & Teachout, M. S. (1995). The relationship between work experience and job performance: A conceptual and meta-analytic review. *Personnel Psychology*, *48*(4), 887–910.

Rey, P. (2015). *Gamification and post-fordist capitalism*. Retrieved 21 April 2016, from http://www.pjreyresearch.com/documents/Rey-Gamification_and_Post-Fordist-Capitalism.pdf

Rheinberg, F., Vollmeyer, R., & Engeser, S. (2003). *Die erfassung des flow-erlebens*. na.

Riegler, A. (2001). Towards a radical constructivist understanding of science. *Foundations of science*, *6*(1), 1-30.

Rosemann, M., & Vessey, I. (2008). Toward improving the relevance of information systems research to practice: the role of applicability checks. *MIS Quarterly*, 1-22.

Ryan, R. M., Rigby, C. S., & Przybylski, A. (2006). The motivational pull of video games: A self-determination theory approach. *Motivation and Emotion*, *30*(4), 347–363. http://doi.org/10.1007/s11031-006-9051-8

Salnitri, M., Dalpiaz, F., & Giorgini, P. (2014). Modeling and Verifying Security Policies in Business Processes. In Proceedings of the 15th International Working Conference on Business Process Modeling, Development, and Support (BPMDS'14) (Vol. 175, pp. 200–214). Berlin, Heidelberg: Springer Publishing.

Salnitri, M., Dalpiaz, F., & Giorgini, P. (forthcoming). Designing Secure Business Processes with SecBPMN. *Software and Systems Modeling*.

Schaufeli, W. B., Salanova, M., Bakker, A. B., & Gonzales-Roma, V. (2002). The measurement of engagement and burnout : A two sample confirmatory factor analytic approach. *Journal of Happiness Studies*, *3*, 71–92. http://doi.org/10.1023/A:1015630930326

Schell, J. (2009). *The Art of Game Design: A Book of Lenses*. Amsterdam: Morgan Kaufman.

Schneier, B. (1999). Attack Trees. *Dr Dobbs Journal*, *24*(12), 21–29. Retrieved from http://www.schneier.com/paper-attacktrees-ddj-ft.html

Schumacher, M., Fernandez, E., Hybertson, D., & Buschmann, F. (2005). *Security Patterns: Integrating Security and Systems Engineering*. Hoboken, NJ: John Wiley & Sons.

Schunk, D. (1991). Self-Efficacy and Academic Motivation. *Educational Psychologist*, *26*, 207-231. http://doi.org/10.1207/s15326985ep2603&4_2

Schunk, D. H. (1995). Self-efficacy, motivation, and performance. *Journal of Applied Sport Psychology*, *7*(2), 112–137. http://doi.org/10.1080/10413209508406961

Schunk, D. H., & Hanson, A. R. (1985). Peer models: Influence on children's self-efficacy and achievement behaviors. *Journal of Educational Psychology*, *77*, 313–322.

Schunk, D. H., & Zimmerman, B. J. (1994). *Self-regulation of Learning and Performance: Issues and Educational Applications*. Hillsdale, NJ: Erlbaum.

Schwartz, S. H. (2006). A Theory of Cultural Value Orientations : Explication and Applications. *Comparative Sociology*, *5*(2-3), 137–182. http://doi.org/10.1163/156913306778667357

Sedrakyan, G., Snoeck, M., & Poelmans, S. (2014). Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. *Computers and Education*, *78*, 367–382. http://doi.org/10.1016/j.compedu.2014.06.014

Sedrakyan, G., & Snoeck, M. (2015). Advances in Conceptual Modeling. In A. M. Jeusfeld & K. Karlapalem (Eds.) *ER 2015 Workshops* (pp. 327–336). Cham: Springer Publishing. http://doi.org/10.1007/978-3-319-25747-1_32

Shahri, A., Hosseini, M., Phalp, K., Taylor, J., & Ali, R. (2014). Towards a code of ethics for gamification at enterprise. In The Practice of Enterprise Modeling (pp. 235-245). Berlin, Heidelberg: Springer.

Shanks, G., & Corbitt, B. (1999, December). Understanding data quality: Social and cultural aspects. In *Proceedings of the 10th Australasian Conference on Information Systems* (Vol. 785). Victoria University of Wellington, New Zealand.

Shanks, G., Tansley, E., & Weber, R. (2003). Using Ontology to Validate Conceptual Models. *Communications of the ACM*, *46*(10), 85–89. http://doi.org/10.1145/944217.944244

Siau, K., & Loo, P.-P. (2006). Identifying Difficulties in Learning UML. *Information Systems Management*, *23*(3), 43–51. http://doi.org/10.1201/1078.10580530/46108.23.3.20060601/93706.5

Sikkel, K., & Daneva, M. (2011). Getting the client into the loop in information system modelling courses. 6th International Workshop on Requirements Engineering Education and Training (REET) (pp. 1–4). http://doi.org/10.1109/REET.2011.6046270

Simpson, E. (1972). *The psychomotor domain*. Washington DC: Gryphon House.

Sindre, G., Firesmith, D. G., & Opdahl, A. L. (2003). A Reuse-Based Approach to Determining Security Requirements. Proceedings of the 9th International Workshop on Requirements Engineering, Foundation for Software Quality, 127–136.

Sindre, G., & Opdahl, A. L. (2005). Eliciting security requirements with misuse cases. *Requirements Engineering*, *10*(1), 34–44. http://doi.org/10.1007/s00766-004-0194-4

Smith, R. and O. Gotel. (2007). RE-O-POLY: A Game to Introduce Lightweight Requirements Engineering Good Practices. 2007 *International Workshop on Requirements Engineering Education and Training (REET).*

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., … Paige, R. (2011). Large-scale Complex IT Systems. *Communications of the ACM*, *55*(7), 12. http://doi.org/10.1145/2209249.2209268

Spolin, V. (1999). *Improvisation for the theater: a handbook of teaching and directing techniques*. Northwestern University Press.

Stajkovic, A. D., & Luthans, F. (1998). Self-efficacy and work-related performance: A meta-analysis. *Psychological Bulletin*, *124*(2), 240–261. http://doi.org/10.1037/0033-2909.124.2.240

Stallings, W., & Brown, L. (2012). *Computer Security: Principles and Practice*. New York: Pearson.

Swartout, W., & van Lent, M. (2003). Making a Game of System Design. Commun. *ACM*, *46*(7), 32–39. http://doi.org/10.1145/792704.792727

Takeda, H., Veerkamp, P., Tomiyama, T., & Yoshikawa, H. (1990). Modeling Design Processes. *AI Magazine*, *11*, 37–48. http://doi.org/10.1609/aimag.v11i4.855

Talkus.io. (2017). *Talkus. Your help desk. In Slack*. Retrieved 3 April 2017, from https://www.talkus.io/

Trist, E. (1981). The Evolution of socio-technical systems. *Conference on Organizational Design and Performance*. http://doi.org/0-7743-6286-3

Tychsen, A., Hitchens, M., Brolund, T., & Kavakli, M. (2005, November). The game master. In Proceedings of the second Australasian conference on Interactive entertainment (pp. 215-222). Creativity & Cognition Studios Press.

Vaishnavi, V. K., & Kuechler, B. (2004). Design Research in Information Systems. *MIS Quarterly*, *28*(1), 75–105. http://doi.org/10.1007/978-1-4419-5653-8

van der Meij, H., & Carroll, J. M. (1995). *Principles and heuristics for designing minimalist instruction*. Business Insights: Essentials.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: four longitudinal field studies. *Management Science*, *46*(2), 186–204. http://doi.org/10.1287/mnsc.46.2.186.11926

Verbeek, P.-P. (2006). Materializing Morality: Design Ethics and Technological Mediation. *Science, Technology & Human Values*, *31*(3), 361–380. http://doi.org/10.1177/0162243905285847

Von Glasersfeld, E. (1991). An exposition of constructivism: Why some like it radical. In *Facets of systems science* (pp. 229-238). Springer US.

Wang, W., & Brooks, R. J. (2007). Empirical investigations of conceptual modeling and the modeling process. In Proceedings - Winter Simulation Conference (pp. 762–770). http://doi.org/10.1109/WSC.2007.4419671

Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems, 15*(4), 285–304. http://doi.org/10.1016/0167-9236(94)00043-6

Wand, Y., & Weber, R. (2002). Research commentary: information systems and conceptual modeling—a research agenda. *Information Systems Research*, *13*(4), 363-376.

Wang, A. I., Øfsdahl, T., & Mørch-Storstein, O. K. (2007). Lecture Quiz: A Mobile Game Concept for Lectures. In Proceedings of the 11th IASTED International Conference on Software Engineering and Applications (pp. 305–310). Anaheim, CA: ACTA Press.

Wang, A. I., Øfsdahl, T., & Mørch-Storstein, O. K. (2008). An evaluation of a mobile game concept for lectures. Proceedings of the Software Engineering Education Conference (pp. 197–204). http://doi.org/10.1109/CSEET.2008.15

Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, *12*(4), 5-33.

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, *26*(2), 13-23. http://doi.org/10.1.1.104.6570

Wendel, V., Göbel, S., & Steinmetz, R. (2012). Game mastering in collaborative multiplayer serious games. In E-learning and games for training, education, health and sports (pp. 23-34). Springer Berlin Heidelberg.

Whyte, W. F. E. (1991). *Participatory action research.* Sage Publications, Inc.

Whitman, M. E., & Mattord, H. J. (2007). *Principles of Information Security* (3rd ed.). Boston, MA: Course Technology Press.

Wiegers, K. (2001). *Inspecting Requirements. StickyMinds.* Retrieved 21 April 2016, from https://www.stickyminds.com/article/inspecting-requirements

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering.* London: Springer Verlag.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimentation in Software Engineering: An Introduction*. Norwell, MA: Kluwer Academic Publishers.

Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

Zajonc, R. B. (2001). Mere exposure: A gateway to the subliminal. *Current directions in psychological science*, *10*(6), 224-228.

Zhang, P. (2008). Technical Opinion: Motivational Affordances: Reasons for ICT Design and Use. Commun. *ACM*, *51*(11), 145–147. http://doi.org/10.1145/1400214.1400244

Zowghi, D., & Paryani, S. (2003). Teaching requirements engineering through role playing: Lessons learnt. In Proceedings of the IEEE International Conference on Requirements Engineering (Vol. 2003-Janua, pp. 233–241). http://doi.org/10.1109/ICRE.2003.1232754

Appendixes

I. Research Method PDD

This appendix contains the PDD of the research method (Figure 30). Note that, although not reflected in it, the *Write long proposal* activity was conducted concurrently with the *Literature study* and *Survey research* activities).

Again, please observe that due to time restrictions the QuaSMOD Framework represents a de-prioritized work product of this thesis, included as an appendix that warrants further research.

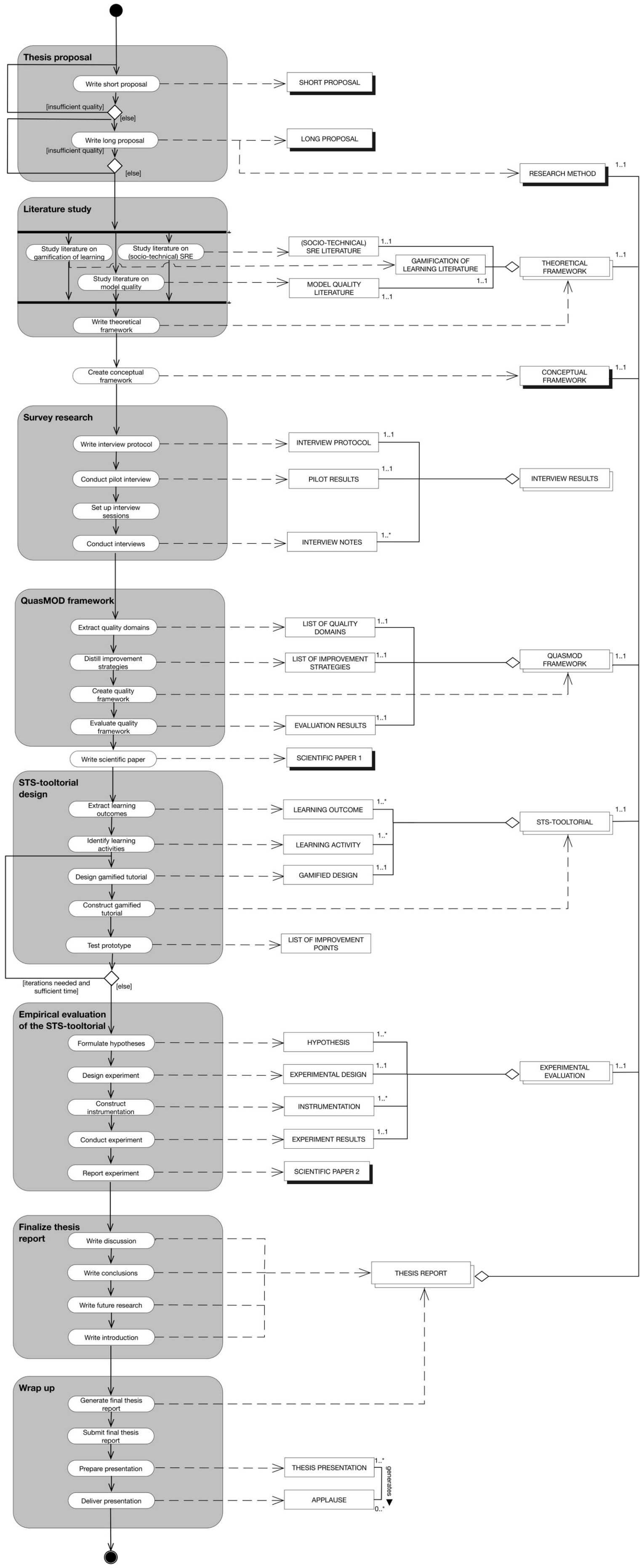


Figure 30 - PDD of the activities performed during this thesis

Table 1 - Research plan

ID	Name	Duration	Start	End	Predecessor	Actual	Research question	Research method	Chapter/Section
First phase									
1	Write short proposal	2 weeks	January (Week 2)	January (Week 3)	-	~January (Week 3)	-	-	-
2	Tentative scope thesis (problem, research goals, and research questions)	3 weeks	January (Week 2)	January (Week 4)	-	~January (Week 4)	-	-	1
3	Define research method	1 week	January (Week 3)	January (Week 3)	-	~January (Week 3)	-	-	-
4	Study literature on innovative teaching methods	2 weeks	January (Week 4)	February (Week 1)	3 with 1 week float	~February (Week 1)	SQ2	Literature review	3.2
5	Study literature on gamification of learning	1 month	February (Week 1)	February (Week 4)	4	~February (Week 4)	SQ2	Literature review	3.4
6	Study literature on interactive tutorials	3 weeks	February (Week 2)	February (Week 4)	4 with 1 week float	~February (Week 4)	SQ3	Literature review	3.3
7	Study literature on (socio-technical) SRE	3 weeks	March (Week 1)	March (Week 3)	-	~March (Week 3)	SQ1	Literature review	3.1
8	Study literature on model quality	3 weeks	March (Week 4)	April (Week 4)	7 with 1 week float and 2 weeks reserve	~April (Week 4)	SQ5, SQ6	Literature review	XX
9	Write long proposal	3 months	January (Week 4)	April (Week 3)	-	~April (Week 4)	-	-	-
10	Prepare 1st presentation MBI colloquium	1 week	April (Week 3)	April (Week 3)	-	~April (Week 3)	-	-	-

ID	Name	Duration	Start	End	Predecessor	Actual	Research question	Research method	Chapter/Section
M1	Milestone: 1st presentation MBI colloquium	-	-	April (Week 3)		18th April		-	
Second phase									
11	Create interview protocol and conduct pilot	3 weeks	March (Week 1)	March (Week 3)				Survey research	2.1
12	Conduct interviews	6 weeks	March (Week 3)	April (Week 4)	11	Start: 30/Mar End: 25/Apr	SQ1, SQ6	Survey research	XX
13	Create QuaSMOD framework: quality domains and improvement strategies	1 month	April (Week 4)	May (Week 3)	12	10th August, de-prioritized & included as Appendix	SQ5, SQ6, SQ7	Design science	XX
14	Evaluate QuasMOD	2 weeks	May (Week 3)	June (Week 1)	13	Not completed	-	Survey research; Meta-model comparison	XX
15	Write scientific paper #1 on QuaSMOD	6 weeks	May (Week 1)	June (Week 2)		Not completed	RQ2	-	-
16	Build background design-based process and gamification frameworks	1 week	April (Week 3)	April (Week 4)	M1 (specif. 7 and 8)	~May (Week 1)	SQ3	Literature review	XX
17	Identify learning outcomes and learning activities	2 weeks	April (Week 3)	April (Week 4)		ILOs: 31/Mar Activities: in parallel to 18	SQ1	Survey research; Literature review; Ideation	XX
18	Design & Build STS-tooltorial †	11 weeks	April (Week 3)	July (Week 1)	16	Start: 10/Aug End: 20/Dec	SQ3	Design-based process; best practices & guidelines identified	XX

ID	Name	Duration	Start	End	Predecessor	Actual	Research question	Research method	Chapter/Section
19	Empirical validation of STS-tooltorial	1 month	July (Week 2)	August (Week 1)	18 with 1 week float	Start: 20/Nov End: 15/Feb	SQ4	Experimental design	XX
20	Write scientific paper #2 on STS-tooltorial	4 weeks	July (Week 4)	August (Week 3)		Start: 24/Feb End: 04/Mar	RQ1	-	-
M2	Milestone: review another colleague's proposal	-	July	July	-	Pending	-	-	-
M3	Milestone: 2nd presentation MBI colloquium	-	August	August	-	Pending	-	-	-
Final thesis report and wrap-up									
23	Finalize thesis report	2 weeks	August (Week 4)	September (Week 2)	15, 20 with 1 week float and 1 week reserve	9th April (draft)	-	-	-
24	Prepare thesis defense	1 week	September (Week 2)	September (Week 2)	23	Start: 9/Apr End: -			
M4	Submit final thesis report	-	September	September	-	Pending	-	-	-
M5	Milestone: thesis defense	-	September	September	-	Pending	-	-	-

† Due to changes in the personal and professional situation of the first author, the first author stopped thesis works for a period of 15 weeks (between June and August). This situation led to a change of thesis scope and delays in delivery dates. The table above transparently reports deviations from a plan that was optimistic from its inception.

II. Interview Protocol for SRE Experts

Note. *The guidelines by Harrell and Bradley (2009) and Galletta and Cross (2013) were followed during the creation of the protocol.*

The following interview protocol was used to gather ILOs for SRE, teaching experiences, and quality statements for appraising the quality of socio-technical security requirements models.

The protocol follows a combination of inverted funnel and tunnel methods, whereby an interview begins with closed (e.g., background) questions, and gradually builds to more open-ended questions. Inverted funnels give the participant time to become comfortable. The tunnel method is the most effective strategy when time is limited, as was in this case (Stewart & Shamdasani, 1990). The combination of both methods was therefore regarded as a fruitful way to structure the interview process.

Questions were checked so that they were not regarded as "problematic questions". Specifically, each question was analyzed so that it could not be identified as one of the following: double-barreled, leading, stated in double negative, use of unfamiliar jargon, vague, use of emotional language, and beyond the respondent's capability to answer (Fowler, 1995; Neuman, 2010). These types of questions obscure the semi-structured interview and therefore should not be used.

Introduction (**5 min**)
Brief the participant.

General information about the researcher, the research project, and interviewee. Identify interviewee's areas of expertise and establish rapport with the participant.

Introduce myself:

- About me
- Topic and research goals
- Structure of the interview (and amount of time allocated to it)

Interviewee background (**5 min**)
Ask the interviewee to tell me about his background and experience with security requirements engineering.

Warm-up and introductory questions destined to engage the participant:

- Background, industry or academia?
- For academics: Experience in teaching security requirements engineering and modeling
- For industry professionals: Experience in using and learning security requirements engineering and modeling

Unguided part (35 min)

<u>About teaching experiences</u> (for interviewees with a teaching background) (**10 min**)
Ask the following descriptive questions:

"I'm very interested in your experiences in teaching conceptual modeling and modeling languages (*if applicable: in the context of security requirements*). Can you give me some examples of how you teach your students to model security requirements?"

Probe the interviewee whenever something is unclear or follow-up questions are needed to elicit additional information. Whenever relevant, also probe to inquire about the following topics:

1. **techniques** used when teaching,
2. "do you use a different **approach** than for other courses?"
3. frequent **modeling mistakes** made by students, and
4. use of **tools** in the classroom (balance between lectures and workshops) .

Prompt the interviewee to give more examples (e.g., "can you give me any other examples/mistakes/techniques that you use?"): aim for a broad overview.

<u>About learning experiences</u> (for interviewees with a professional/industry background)
Ask the following combination of descriptive/structural questions, with the objective of getting a glimpse at the students' perspective.

"How did you learn to create security requirements (models)?" "What aspects did you find helpful/difficult?"

About learning outcomes (**15 min**)

If respondent is familiar with the concept of learning outcomes, introduce that the objective of this question is to identify ILOs. Give examples of ILOs to get the interviewee familiar with the concept.

Learning outcomes are usually stated in a standardized manner, using a format like "By the end of this tutorial, students should be able to…".

Also, they consist of:

- *A verb*
- *A learning statement*
- *(Optional) a criterion or standard for acceptable performance*

Some examples:
By the end of this course, students should be able to:

- ***calculate** the probability that two sample means will differ by more than 5%*
- ***appreciate** the ways in which online marketing can be used to influence consumer behavior*
- ***design** an experiment to determine the effect of a gamified, interactive tutorial on student engagement*
- ***describe** the major ethical issues one must consider when planning an animal study.*
- ***demonstrate** active listening skills when interviewing clients*
- ***recognize** the rules of the BPMN modeling language*

"Imagine now that you are given the chance to teach a tutorial for 5-6 hours. What would you expect your students to learn?"

Probe about what would (s)he expect students to:

- know (cognitive)
- do (skills)
- value (affective)

About the QuaSMOD framework (**10 min**)

Ask the following structural questions:

"How do you determine when a security requirements model is of good quality? Specific probes: What other factors have an impact on the quality of such a model?

In general, follow-up whenever it is required. Some suggested non-leading and non-directive probes to be used for each section are:

- "Tell me more about…"
- "Could you explain a bit more what you meant by…?"
- "Could you describe…?"
- "How did students respond to that?" (e.g., use of tools in the classroom)
- "Why is that important to you?" (e.g., a specific learning outcome)
- "What motivated this change?" (e.g., new teaching approach)
- "It sounds like you are saying "…". Is that a fair summary?" (i.e., to clarify vague statements)
- "I'd like to understand more about how this relates to the earlier topic we were talking about." (e.g., to avoid digressions)

[*THIS SECTION WAS REMOVED AFTER THE PILOT STUDY*]
Validation (introduce the list of learning outcomes and statements underlying the quality domains identified so far) (**15 min**)

Quality domains (represented by a list of statements):

"Do you agree with these statements?"
"Is there anything that you **miss**? What, why is that important?"
"Can you think of any other statements that should be considered?"

Learning outcomes

"Do you agree with these learning outcomes?"
"Is there anything that you **miss**? What, why is that important?"
"Can you think of any other learning outcomes that could be included?"

Conclusion (**5 min**)
Thank you for your time. Mention that (s)he will be kept updated (e.g., findings) if desired. If applicable, recommend contacts from industry to interview.

III. Protocol for Post-Evaluation Interviews

The objective of the post-evaluation interviews was to contact participants and ask, when relevant, follow up questions some days (i.e., between 4 and 6 days) after the intervention. According to social psychology (Kenrick, Neuberg, & Cialdini, 2009), the rationale for waiting a few days was that participants would have had more time to reflect and they would remember, ex-post, only what struck them the most, thus better ascertaining their cognitive responses, thoughts, and overall impressions.

While the evaluation and survey helped us to gather self-reported and automatically collect qualitative metrics for statistical analysis, these interviews aimed at deepening our qualitative understanding on participant's responses that warranted further elaboration.

Since the interview demanded flexibility and free discovery and elaboration of information, the semi-structured interview format was preferred. Sessions focused on experiential questions related to how participants experienced the intervention, with the goal of understanding:

- **Objective 1**: what positive aspects of the tutorial the participant remembered the most
- **Objective 2**: what areas of improvement exist for future versions
- **Objective 3**: what they like or dislike most about the instrument (i.e., STS tooltorial) and setting (classroom-style environment)

Introduction (5 min)

Introduce myself and explain objectives of the interview.

Brief the participant about her results by using the template below, to help her remind what her performance and self-reported measures were, indicating that results will remain anonymous.

Depending on the performance and answers provided during the post-evaluation survey, the interview can take any path. To that end, the following questions should serve as a non-comprehensive list of areas of interest that can be leveraged to keep the interview on-topic.

[STS-Tooltorial specific]

- Have you ever used come across something similar to the STStooltorial?
- What did you think about the tutorial? *(Print the list of Game Elements, Guidelines for designing interactive tutorials, and Platform Requirements and probe about the different features of the platform)*
- How did the tutorial help you better understand security requirements for socio-technical systems?

[Learning Environment]

[this question depends on the choice of learning environment selected in the survey]

- What did you think about the learning environment? *(Here we have three considerations: the tutorial itself, the presence of an instructor, and the presence of other people)*
- Would it be interesting for you to find this teaching format in this or other topics?
- Which aspects did you like/dislike the most?

[Overall satisfaction]

- What are the top three things you would improve?
- What is your overall impression of the experience?

IV. QuaSMOD Framework

This appendix contains a preliminary framework for appraising the quality of security requirements models. As mentioned throughout this report, it has been structured according to semiotic theory. The goals and dimensions have been synthesized from interviews with subject-matter experts from academia.

It should be noted that during the execution of the thesis, the QuaSMOD framework was defocused due to time constraints in favor of the STS-Tooltorial, which represents the major artifact of our research.

1. Epistemological Foundations of Quality of Models

Evaluating conceptual models is more of an art than a science: subjective and based on common sense or experience (Moody, 2005). Moreover, it is widely established in literature that, the higher the quality of a conceptual model, the better the quality of the actual system (Browne & Ramesh, 2002; Wand & Weber, 2002; ISO 9126).

However, the term quality (in particular, in the context of models) has been repeatedly used throughout this study. A precise definition and discussion of its definition has hitherto been neglected. Incidentally, despite the ordinary use of the word, researchers have struggled for years on how to appraise the quality of conceptual models. Several frameworks have been proposed, none of which seem to have attracted the interest of practitioners.

In his research, Moody (2005) identifies common problems associated with quality frameworks:

- ***Non-compliance and lack of consistency with related field standards***
- ***Lack of guidelines for improvement***
- ***Lack of definition of quality***
- ***Lack of metrics for quality criteria***

Here, we propose a framework for appraising the quality of secure socio-technical models that supports the first three points. While **Section 3** elaborates upon guidelines for improvement, here we introduce our working definition of quality drawing on the ISO standard:

"The degree to which a set of characteristics of a security requirements model fulfills a need or expectation that is stated, generally implied, or obligatory." - Proposed **security requirements model quality definition**. Adapted from the definitions of *quality* and *requirement* of ISO 9000:2005.

In addition, we take a viewpoint rooted on social constructivism and pragmatism, departing from the traditional positivist approach that dominates the Information Science sphere. We discuss these viewpoints in the next section.

2. Discourse on the Need of Constructivist Approaches to Information Science

Constructivism is the idea that we construct our own world rather than it being determined by an outside reality. Reality and knowledge are thus formed by a solipsistic mental construction of individuals (Riegler, 2001). This anti-positivist epistemology implies that, since knowledge is subjective, so are models and symbols. Von Glasersfeld observes that (1991, p. 27) "neither problems nor solutions are ontological entities, but arise out of particular ways of constructing". Thus, words, symbols, and models can no longer be regarded as carriers of information, but merely a result of mutual interpretation and communication processes. The consequence is clear: we, as individuals, cannot transcend our experiences.

Viewing the theoretical underpinnings of IS from this perspective is powerful, given that most literature has so far adopted a positivist approach. For one, it offers variety to the possible ways of conducting scientific research in the discipline of IS. It also opens up debate on the impact and changes that such an approach can bring about, as it has already had in other traditional areas.

As enticing as it sounds, the promise of constructivism falls short in accounting for interactions among individuals. Owing to the communication means that models afford, their discussion of quality must be heightened to the sphere of social reality. Having a social focus, social constructionism proposes the redefinition of social realities as constituted through discourse (Neimeyer, 1998). It places great emphasis on everyday interactions between people, and how they use language to participate in the creation of their (perceived) social reality, which is dynamic (Andrews, 2012).

Since social interactions shape the perceptions and interactions of reality (which are in turn shaped and mediated by language), multiple views of reality may arise (Burr, 2003). Note that this relative stance implies that all perceptions are equally valid or acceptable, and that there is no such thing as preferred reality.

Consider the implications that this relativistic approach has on e.g. the modeling of security requirements for socio-technical systems. Imagine that a group of system analysts and designers ponder the security needs of a new casino system. While business analysts might create a model that considers the social and organizational aspects in terms of actors and their interactions, technical designers might create an overview of the informational assets that need protection. (Note that, for example, the STS-ml supports this multi-view perspective by integrating three different views, namely: social, information, and authorization). Although both models speak of the same universe of discourse (i.e. the security needs of a casino system), it begs the question of which one would better fit the intended purpose of e.g. communicating those needs to the client based on his own mental model. Detached from the context of the situation at hand, discussions as to the validity, quality, and superiority of these models (which present partial truths of the perceived reality) are deemed impossible.

Pragmatism offers a way to tackle the relativistic nature of social constructivism. The central precept behind this tradition, which emerged in the US in late 1870 with the works by Peirce, James, and Dewey is that an "ideology or proposition is true if it works satisfactorily; that the meaning of a proposition is to be found in the practical consequences of accepting it; and that unpractical ideas are to be rejected" (McDermid, 2006, p. 1). In other words, a proposition is held true in terms of its usefulness and practical relevance in the situational and historical context in which it occurs. In a way, this tenet is in broad agreement with social constructivism, while does not promote strong forms of relativism (Proctor, 1998).

3. The QuaSMOD Framework

This section finally presents our framework for appraising the quality of security requirements models. It is articulated on the basis of semiotic theory. The quality dimensions in Table 19 have been identified based on past works on conceptual model quality (Wang & Strong, 1996; Krogstie, Lindland, & Sindre, 1995; Shanks & Corbitt, 1999) as well as synthesized from interviews with SRE experts 1 to 3 (as per Table 2). Next to the dimensions, we also suggest improvement strategies modelers can use to improve models. Again, this work should be considered preliminary, to be refined and validated in future research.

Table 19 - Overview of QuaSMOD Framework's semiotic levels and improvement strategies

Semiotic Level	Goal	Dimension	Improvement Strategies
Syntactic	Consistent model	• Well-defined syntax	Syntax checking Training for security requirements analysts (modeling language)
Semantic	Complete and accurate representation of the problem domain	• Justified placement of security requirements • Appropriate coverage • Textual explanation • Unambiguity	• Iterative refinement • Training for security requirements analysts (domain knowledge)
Pragmatic	Useful and usable model	• Well-scoped model • Timeliness of the model	• Iterative refinement
Social	Shared understanding of meaning of the model	• Ease of understanding by analysts • Ease of understanding by domain stakeholders • Stakeholder agreement on the knowledge captured in the model • Stakeholder agreement on the interpretation of the model	• Viewpoint analysis • Conflict analysis (e.g., silent critique) • Model walkthrough Model merging • Training for security requirements analysts (public speaking)

The remaining sections go into greater detail about the different semiotic levels.

3.1 Syntactic Level

Syntactic quality concerns the structure of the security requirements model, without considering its meaning. The hallmark of this semiotic level is consistency of the model with respect to the modeling language. To that purpose, the following dimension is defined:

- Well-defined syntax. To make the model as Unambiguous as possible, the model should abide to the syntactical rules afforded by the language. It is assumed that the language does indeed provide precise rules and syntactical elements.

3.2 Semantic Level

Semantics deals with the meaning of symbols. In the context of QuaSMOD, the semantic level concerns the meaning attached to the model. The overarching goal is to ensure a complete and accurate representation of the problem domain. The following dimensions are of relevance:

- **Justified placement of security requirements**. Security requirements should not be randomly placed. In other words, when a security requirement is added to the model, it should be based on a conscious and justified decision. It is best if this decision is recorded (see next dimension).
- **Textual explanation**. The model should be accompanied with a textual description that resolves any unclarity and justifies the design rationale. Having a document that analysts and other stakeholders can resort to will be helpful when revisiting the diagram and making decisions down the road.
- **Appropriate coverage**. A model ought to have an appropriate coverage of relevant security needs and requirements. Such coverage should be measured against the domain under study. For instance, if the analyst is creating a security requirements model for a safety-critical system, failing to consider redundancy requirements would not be appropriate.
- **Unambiguity**. Whenever possible, the model should avoid terms or elements that may lead to multiple (unnecessary) interpretations. The use of a Textual explanation and Well-defined syntax can help reduce ambiguity.

3.2 Pragmatic Level

Pragmatics predominantly involves the usage of the security requirements model. The goals to pursue at this level are usability and usefulness. We define usability as the extent to which stakeholders are able to access and use the model. Next to that, usefulness refers to the degree to which the model effectively helps stakeholders to accomplish their tasks.

- **Well-scoped model**. The contents of the model should be contextually appropriate to serve the purpose they were created for. Every (security) requirements analysis focuses on a given scope (e.g., an information system for a given organization or part of it). For that reason it is important that the model is a focused, to-the-point representation of the given scope.
- **Timeliness of the mode**l. The currency of the security requirements model should be appropriate to the task at hand. An updated representation of the domain under study is a must for making decisions. The required "age of the

model” will depend on the specific circumstances but, in general, a model that is iteratively refined is sign of a timely model.

3.4 Social Level

The social level concerns the shared understanding that emerges from interpreting a security requirements model. The goal is to create a socially constructed model.

The following dimensions are relevant in this level:

- **Ease of understanding by analysts**. This dimension refers to the degree to which the model is clear and easily comprehended by analysts well-versed in the jargon, rules, and formalisms used to construct the model.
- **Ease of understanding by domain stakeholders**. This dimension deals with stakeholders outside of the community of practice. Contrary to the previous dimension, the way in which the model is communicated to “outsiders” needs to be tailored to the specific audience. For instance, managers and developers will probably be interested in different aspects of a security requirements model. While a manager might only be interested in knowing what assets are being protected to comply with regulatory laws, developers may be interested in the specific technical mechanisms that need to be enforced to guarantee such protection.
- **Stakeholder agreement on the knowledge captured in the model**. This dimension and the one that follows deal with the degree of agreement. Since many stakeholders participate in the SRE process, each of these is bound to have her personal views of the requirements and knowledge to be captured in the model. The goal is to reach a suitable agreement about what knowledge should (and should not) be stated.
- **Stakeholder agreement on the interpretation of the model**. Similar to the previous dimension, stakeholders should agree about what they think is stated in the model. Observe that the type of agreement should prevail is a contested topic. See Krogstie, Lindland, and Sindre (1995) for a discussion on this issue. In any case, agreement in this dimension will be easier to achieve than in the previous one.

When constructing shared understanding, it is important to consider two aspects. First, stakeholders are bound to have different viewpoints. This situation calls for an inclusive approach where viewpoints are freely expressed and dealt with. Second, social construction of meaning is heavily determined by cultural practices. The interpretation of and meaning attached to a security requirements model will inevitably introduce bias in the process, based on the interpretation of different social groups. Since we see security requirements modeling as a social activity, countering the effects of biases and including multiple perspectives becomes increasingly important. This is even more so in the context of distributed software development, where analysts have to juggle various cultures.

3.5 Other Contributing Dimensions

Aside from the ones already identified, our interviews and domain literature surfaced some other aspects that are seen as key contributors to quality. These aspects are not about the conceptual model being constructed, but rather refer to the quality of the modeling language itself and the tools used to support it. Note that the following dimensions could be positioned at the pragmatic level, as they aid stakeholders in completing their tasks:

- **Tool support**. Ability to create a model and possibly reason about it using a computer-aided tool. Reasoning is most applicable under semi-formal or formal model representations.
- **Asset generation**. Ability to generate products based on the model. Examples include security requirements specifications, code, or test cases.
- **Integration**. Ability to generate a representation of the model for integration purposes. For instance, a high-level social model that can be integrated with other security modeling tools and languages (e.g., to define specific security mechanisms) would be of higher quality than one that does not provide such possibility.
- **Reputation of data source and creator**. The reputation of the data source used to create the model (e.g., results from interviews, domain knowledge of the stakeholders) plays an important role in the quality of the resulting model. High-quality sources should be preferred. Whenever known, the reputation of the model’s creator may also be an important dimension.

V. List of Publications

This appendix contains the list of contributions to scholarly venues that have been produced as part of this research endeavor.

Publication 1: RE@Next!

Alami, D., & Dalpiaz, F., (2017). A Gamified Tutorial for Learning about Security Requirements Engineering. Accepted at the *25th IEEE International Requirements Engineering Conference, RE@Next! track*.

* Please refer to the conference proceedings for the most updated version of the publication.

A Gamified Tutorial for Learning about Security Requirements Engineering

Daniel Alami
International Business Machines S.A.
Madrid, Spain
daniel.alami.cabezas@es.ibm.com

Fabiano Dalpiaz
Dept. of Information Science, Utrecht University
Utrecht, The Netherlands
f.dalpiaz@uu.nl

***Abstract*—Thanks to the advent of interactive technologies, education institutions are looking for innovative teaching methods to increase the engagement and reach of students. Besides the uprise of MOOCs, *gamification* has been shown to produce positive results when it comes to increasing people's engagement and interest in conducting tasks. Unfortunately, the application and benefits of these technologies in teaching requirements engineering (RE) remain largely unexplored. In this paper we introduce the *STS-Tooltorial*, an interactive gamified platform that executes within a security requirements modeling tool and helps learners apprehend the STS-ml language and basic notions about security requirements. We present the design principles of our functional prototype: its educational content as well as the embedded game elements. Furthermore, we report on an early evaluation with IT professionals and postgraduate information science students focused on the platform's effectiveness and usability.**

***Index Terms*—gamification, security requirements, requirements engineering, interactive tutorial.**

I. Introduction

Requirements Engineering (RE) is a crucial activity for creating high-quality software and a vital component of successful projects [1]. Among others, security is a key quality requirement of software systems. Failing to consider security requirements early in the development process increases the chances of introducing serious security breaches; security is often considered as an afterthought in software engineering [2], [3].

Several studies have highlighted the importance of including security since the early stages of the systems development process [4], [5]. These efforts fall under the domain of Security RE (SRE), which emerged as a response to the monetary expenditures associated with (bad) security [6].

The quality of the SRE process is key to minimize the likelihood of introducing vulnerabilities. Many SRE approaches advocate the use of modeling languages that, based on conceptual modeling foundations [7], create models of the system to precisely document and analyze security requirements together with design requirements [8], [9]. The challenge for modelers is to *adequately* capture security requirements into the models.

The problem we tackle in this paper is that of "How to effectively teach model-driven security RE?" Mastering model-driven SRE requires the skills of developing, manipulating, and understanding models. Exercising these skills depends on both analytical capabilities and, in line with modern educational theories [10], an effective educational process.

Teaching conceptual modeling is not easy [11]. The quality of a conceptual model depends on both the modeler's understanding of the modeling language (rules, semantics and constructs) and her knowledge about the modeled domain [12].

In this paper, we build on modern education theories and we propose the use of an interactive gamified tutorial that is embedded within a security requirements modeling tool. We do so with the aim to foster self-learning of the basic concepts of SRE without relying on traditional (passive) lecturing methods that are proven to be ineffective for today's learners [13]. In particular, the paper makes the following contributions:

- We define a framework for building interactive tutorials for conceptual modeling languages. A key component of our framework is gamification [14], which we include to foster learner engagement by raising engagement and active participation (Sec. II).
- We describe the design of the curriculum and gamified experienced of our STS-Tooltorial, which instantiates our framework for SRE and for the STS-ml goal-oriented language [15] in particular (Sec. III).
- We report on preliminary results on the effectiveness of our approach for learning the basics of STS-ml and SRE, based on two studies with small groups of students and industry professionals (Sec. IV).

We conclude the paper with a discussion of our approach and by sketching future directions (Sec. V).

II. A Framework for Building Interactive Tutorials for Conceptual Modeling Languages

We studied the literature in innovative teaching methods, interactive tutorial design, and gamification of learning. Based on our findings, we distill guidelines for building interactive tutorials for conceptual modeling languages.

A. Innovative Teaching Methods

We explored venues where software and requirements engineer researchers gather to share perspectives on education. Venues like the Conference on Software Engineering Education and Training (CSEE&T), the ICSE Software Engineering

Education and Training Track (SEET), and the workshop on Requirements Engineering Education and Training (REET) have called for the cooperation among multidisciplinary fields.

Our inquiry showed that the last decade has seen a growing interest in the application of innovative learning methods, techniques, and tools for RE and SE curricula. Innovative methods for enriching classroom training include improvisation theater [16], role playing [17], [18], project simulations [19], and case-based courses with either virtual [20] or real stakeholders [21].

Interestingly, we found that in almost all cases, the material taught with these approaches is predominantly about the teaching of high-level, soft skills. Only a few approaches (e.g., [17]) dive deeper into teaching specific methods, techniques or tools.

Therefore, we observe that room for improvement exists for introducing a novel teaching approaches that go deep into teaching procedural knowledge about a method or technique.

B. Design of Interactive Tutorials

The design of interactive tutorials calls for a different approach than the design of traditional learning. Our investigation revealed a solid baseline, including the principles and implications for the design of learning systems of Park and Hannafin [22], the principles and heuristics for minimalist instruction design [23], and the rich literature on computer tutoring.

Recent research points to 1) the use of video tutorials as a prevalent source of information for users, for which comprehensive guidelines exist [24], and 2) the use of contextual assistance for enhancing understanding, quality, and flow [25].

We assembled a collection of 33 guidelines for designing an interactive tutorial, which we use in Sec. III to inform the design of our STS-Tooltorial. An excerpt of these guidelines is reproduced in Table 1 (the complete list is available in an online appendix[1]).

TABLE 1: EXCERPT OF GUIDELINES FOR DESIGNING INTERACTIVE TUTORIALS

ID	*Guideline*
1	Provide instructions in the problem-solving context, anchoring the tool in the task domain.
2	Promote an abstract understanding of the problem-solving context.
3	Minimize working memory load and complement cognitive processes by structuring presentations and interactions.
4	Provide on-the-spot, immediate error information that supports detection, diagnosis, and recovery.
5	Facilitate successive approximations to the target skills.
6	Provide an immediate opportunity to act.
7	Organize lesson segments into internally consistent, meaningful, and self-contained units.
8	Differentiate important information through cosmetic amplification, repetition, and recasting to direct learners' attention.
9	Use videos that: 1) provide procedural or instructional information rather than conceptual information, 2) keep segments short, 3) ensure tasks clarity, and 4) coordinate demonstrations with text.

[1] www.staff.science.uu.nl/~dalpi001/appendix-sts-tooltorial.pdf

C. Gamification of Learning

Effectively designing a gamified tutorial requires profound knowledge review of the state of the art on gamification elements and best practices for gamifying learning experiences.

We studied both white and grey literature about game elements. The rationale for covering both was to missing key success factors; most academic work seemed to mainly rely on motivational affordances of an extrinsic nature. In contrast, industry gamification experts such as Chou [26] highlight the importance of intrinsic motivation (this is also confirmed by recent studies by academics such as Deterding [14]).

We found 17 relevant game elements, including point system, badge, leaderboard, level/mission, progress bar, story/narrative, avatar, easter egg, time restriction, onboarding, game master, etc. After studying those game elements and their associated motivational affordances, we distilled a list of best practices for gamifying learning experiences (see Table 2).

TABLE 2: BEST PRACTICES FOR GAMIFYING LEARNING EXPERIENCES

Guideline	*Description*	
Freedom of choice	Allow for freedom of choice. E.g., learners should be able to choose the order or speed of the challenges to be completed, or what goals to pursue. At a minimum, give the feeling of freedom.	[26]–[29]
Freedom to fail	Adopt the freedom to fail principle: poor task performance should not incur in penalties. E.g., learners should be able to retake quizzes.	[27], [29]
Baby steps	Divide and present the educational content in small pieces of coherent information.	[30]
I know this!	Conduct an evaluation step (e.g., exercises, quizzes) after presenting educational content.	
Together is better	Make the system social to motivate students via peer pressure or comparison with other students.	[27], [30]
Boost it!	Give bonuses after the completion of hard tasks.	[28], [30]
The good Samaritan	Compensate students not only for academic achievement but also prosocial behavior	[27], [30]
Crystal clear	Offer immediate feedback and inform learners of their progression within the tutorial (e.g., progression bars. Frequent and immediate feedback leads to greater learning effectiveness and engagement.	[29]–[31]
Ethical designer	Design with ethics in mind: e.g., full transparency and opt-in principles.	[14], [26]
Know your users	Target population must be studied, regardless of whether gamification is considered or not	[28]
The road ain't easy	Communicate that the training will be challenging	[28]
Location, location	The location in which users engage with the application matters	

Taken together, an understanding of 1) curricula design, 2) guidelines for designing interactive tutorials, 3) game elements and 4) principles for gamifying learning experiences constitute the framework that we used for designing the STS-Tooltorial.

III. STS-TOOLTORIAL: A GAMIFIED TUTORIAL FOR TEACHING SECURITY REQUIREMENTS ENGINEERING

We explored how to effectively teach a particular SRE method through an interactive tutorial: the Socio-Technical Security (STS) method, which is a goal-oriented and model-driven approach [15]. Our choice is due to two main reasons: 1) the socio-technical approach that considers security issues both from a technical and organizational/social perspective; and 2) the availability of a robust modeling tool: STS-Tool[2].

Our interactive platform, called STS-Tooltorial, introduces learners to basic concepts of socio-technical systems and security requirements engineering to then delve into the STS method and modeling language. The platform is embedded in STS-Tool, thereby offering a first-of-a-kind experience for learners as they can learn at their own pace following a hands-on and highly interactive approach with short feedback cycles.

At the time of this writing, the tool covers a subset of the STS method that can be completed at once within 90 minutes. However, we are planning to release further learning modules to cover the entire spectrum of STS.

A. Curriculum design

To extract educational objectives, we supplemented desktop research with semi-structured interviews with seven leading SRE experts with a teaching background to identify a list of intended learning outcomes for a hypothetical two-hour classroom-based tutorial covering concepts of SRE and conceptual modeling. The refined list of learning outcomes finally was:

- *ILO1*: Understand the different activities and deliverables of security requirements engineering.
- *ILO2*: Recognize the modeling rules and elements as well as their meaning (in an existing model)
- *ILO3*: Choose the most appropriate set of elements for representing security-related aspects.
- *ILO4***:** Express interest in pursuing further learning in the field of (socio-technical) SRE.

The interviews also provided useful information that helps complement the general guidelines on instructional design presented in the previous section, based on best practices and heuristics that academics use to teach SRE and modeling, regardless of the delivery method:

- *S1***:** Modeling is an iterative process: models are constructed and iteratively refined.
- *S2***:** It is important to consider the attacker perspective.
- *S3***:** It is key to stress economic impact of security issues to raise awareness.
- *S4***:** Distinguishing between *document* and *information* is important in socio-technical systems design.
- *S5***:** Security is a multi-level concept (physical, network, and social). The social aspect is too often under-researched, under-focused, and under-prioritized.
- *S6***:** Teach through exercises, using a hands-on case-based approach.
- *S7***:** Include a hands-on task only if the learner has previous experience.
- *S8***:** Include a reflection phase after teaching.
- *S9***:** Give feedback after completing a modeling task.
- *S10***:** Have students review each other's work through peer inspection.
- *S11***:** It is important to capture the rationale of models.
- *S12***:** SRE should not be seen as a separate activity. The task of analyzing a system and creating a good set of security requirements should be done while conducting other software development activities.

B. Tutorial design

The general guidelines and the insight of SRE experts set the foundation for the design of the interactive platform. The STS-Tooltorial is a Web-based learning management platform based on Wordpress that can be embedded in the STS modeling tool. The delivery method consists of bite-sized lessons (*Levels*) lasting between 1 and 5 minutes where learners are progressively introduced to SRE, socio-technical systems, and the STS method and modeling language at increasing levels of difficulty to keep them in a *state of flow* [32]. The educational material is presented in the form of videos and text. In addition, the lessons contain *Quizzes*, which in turn may consist of multiple-choice, true/false, or what we refer to as "model interaction" questions. The latter feature provides learners with questions about the content they just learnt that prompt them to interact with the modeling tool, drawing diagrams and getting quick feedback on their correctness (see Fig. 1).

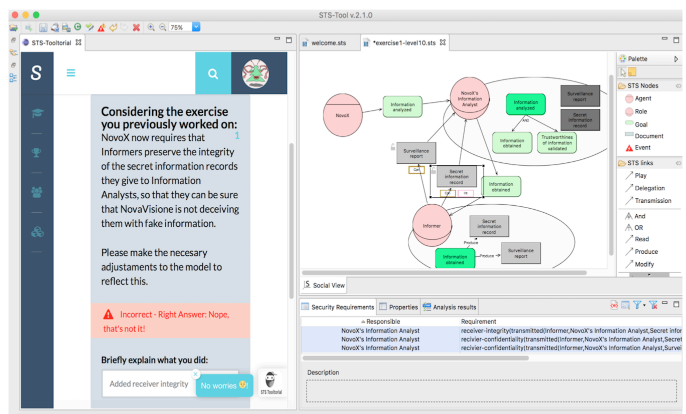


Fig. 1 Screenshot of the STS-Tooltorial (left) embedded in the STS-Tool

C. Game elements

Aside from *Levels* and *Quizzes*, the use of other game-like elements is pervasive throughout the STS-Tooltorial, as they support the different learning activities and tap into both intrinsic and extrinsic motivators.

The tutorial makes use of science fiction to present the educational material (*Story/Narrative*). In particular, the learner begins as a Rookie security requirements analyst hired by NovoX, an Intelligence Agency, to protect the Empire's interests across the Galaxy through designing a secure socio-technical system. By successfully completing the training material, the learner is awarded *Points*, which in turn determine their *Rank* and position in the *Leaderboard*. The learner can achieve up to 5 Ranks in their quest to become "Protector of the Galaxy", a position awarded to a few distinguished officials. The *Point System* awards points based on the estimated time the lessons and exercises take, whereas the *Rank* is determined using an algorithm that makes it harder to rank up as they obtain more

[2] http://www.sts-tool.eu/

points (and thus become more skilled). Note that the inner workings of the Points and Rank system are hidden to learners.

The tutorial also features the so-called *Rewards Marketplace*, where learners can exchange points in favor of 1) chatting with the *Game Master* and Security Requirements Experts or 2) *claiming* other *Rewards* such as: deduct points to another player, get personalized help in case of problems, purchase contextually-relevant hints, etc.

Table 3 shows a comprehensive list of the game elements used in the interactive tutorial (the theoretical underpinnings and references *are* available in our online appendix).

TABLE 3: SUMMARY OF IMPLEMENTED GAME ELEMENTS

Element	***Description***	***Expected motivational affordances***
Points	Points are granted to reward users upon task completion	Feedback, competition, sense of achievement, and positive emotions
Rank	Visualization of achievements to give a sense of progression	Progression, instruction, reputation, and group identification
Leaderboard	Ranking of users based on points and rank achieved	Competition, recognition
Level	Used to keep game space manageable and give a sense of progression. Completion rewarded with points and rank ups. Includes quizzes.	Reward, status, competition, achievements
Progress bar	Used to track overall goal progression with the different modules	Feedback, achievement
Story/Narrative	Used to add meaning, provide context, and guide action. Comprises the elements of characters, plot, tension, and resolution	Foster learning, immersion, attitude change
Rewards Marketplace	Chat where users can redeem rewards in exchange of points	Reward, feedback, competition
Time restriction	Used to instill pressure	Sense of urgency
Social prod	Action of minimal effort to create a social interaction (e.g., "Like", "Deduct points")	Social
Activity feed	Stream of recent events in the course	Foster a sense of community and lead to feelings of recognition
Conformity anchor	The system informs the user how close a user is to the social norm (i.e., tutorial real-time performance)	Feedback, social
Animated feedback	Animated visual aid used to help users locate something in a system	Feedback
Verbal/visual/sound effects	Used to tell users about the state of the system as a result of interacting with the tutorial and modeling tool	Feedback
Avatar	Graphical representation of a user profile that users can customize to their preferences	Encourage self-expression, ownership
Game master	Observes and orchestrates the learning process. Assists or influences the behavior of learners	Feedback, social

IV. PRELIMINARY RESULTS

We report preliminary findings from investigations that focused on a subjective evaluation of the effectiveness of the STS-Tooltorial on learner *engagement*, *performance*, *intention to learn*, and *preferred learning environment* within a non-experimental setting. The artifact was also evaluated in terms of overall *usability and satisfaction*.

We have conducted two separate empirical evaluations featuring eight postgraduate information science students and five IT professionals with three to 12 years of experience. The student evaluation was conducted at Utrecht University in two separate sessions following the same procedure and setting. In particular, students were grouped in a room and requested to complete as much of the tutorial as possible within a 90-minute timeframe, under the supervision of the authors who were not to intervene unless, e.g., a participant redeemed a reward for hints. We hoped to induce certain (social) pressure by the combination of the time restriction, showing the leaderboard in real-time, and having a Game Master physically present indirectly judging on the correctness of the modeling activities.

Since results of this evaluation suggested that physical co-location of the learners was not a key factor to have learners complete the tutorial, we decided to organize the evaluation with professionals in an online, remote fashion, with participants setting their pace independently.

In both evaluations data was gathered via pre-post survey and through the analytics module of the interactive platform. In addition, we followed-up after a week with some of the participants to brief them on their results and gather more qualitative feedback on their responses and behavior.

Next, we examine the early findings across the different constructs of interest: engagement, usability and overall satisfaction, intention to learn, and preferred learning environment. To do so, we used the questionnaire available online[3]. No comprehensive statistics are reported here due to the inherent noise present in our small-sized sample.

Below, we use the following abbreviations taken from the APA guidelines for reporting statistical results: M for mean, SD for standard deviation, r for Pearson's correlation, n for sample size, p for the p-value, and t for the statistical t-test.

A. Engagement

The measure of engagement was mainly underpinned by the Short Flow Scale [33] ranging from 1 (minimum flow) to 7 (maximum flow). The student group self-reported a higher-than-average flow during the completion of the tutorial (M=4.69, SD=1.03), with no significant difference with respect to the professional's flow (M=4.5, SD=0.90).

In both groups, the relationship between engagement and the number of points awarded (r=.64, n=13, p=.09) and page views (r=.65, n=13, p=.08) fell slightly short of statistical significance, given that the p value is above 0.05.

[3] http://bit.ly/ststooltorialposttest

B. Usability and overall satisfaction

Usability was measured using the System Usability Scale (SUS): a valid, reliable instrument widely used in industry and academia. The aggregated SUS scores of both samples indicate that the interactive platform has *high usability* and is well integrated within the modeling tool (M=80, SD=8.07), with no significant difference among students and professionals; $t(11)=.16$, $p=.88$.

To measure satisfaction, half-way through the tutorial learners were asked to rate on an 11-item Likert scale how satisfied they were with the tutorial. Subsequently, in the post-evaluation survey, the Net Promoter Score (NPS) ranging from 0 to 10 was used as a proxy measure to satisfaction (as evidenced by intention to recommend the tutorial to a colleague). The groups reported high degrees of half-way satisfaction (M=7.38, SD=1.19) and intention to recommend (M=7.88, SD=1.46), with again no significant difference between professionals and students. In addition, both measures were found to be strongly correlated across groups (r=.856, n=13, p=.007).

C. Intention to learn

Lastly, we examined was the extent to which the platform influenced the learner's intention towards learning more about SRE in general and the STS method in particular, which has been shown to be a precursor of behavior [34]. We have used a 5-point Likert-type scale where we posed a statement saying that "I am interested in studying more about…" ranging from 1 (disagree) to 5 (agree).

Findings suggest that students had a significant higher intention to learn more about both SRE ($t(11)=3.12$, $p=0.01$) and the STS method ($t(11)=2.15$, $p=0.05$) than professionals did. In addition, a consistent pattern we observed across groups was that learners had more intention to learn about SRE (M=3.15, SD=1.14) than the STS method in particular (M=2.77, SD=1.01), albeit the latter was the main focus of the tutorial; $t(12)=-2.13$, $p=0.054$.

Overall, learners expressed *at least* an average-to-positive interest in pursuing further learning, with students being significantly more positive than IT professionals.

D. Preferred learning environment

Participants were finally asked to rank their preferred choice of learning environment among four choices: MOOC, Book/Self-study, Built-in tutorial (e.g., our platform), and Classroom setting.

While books were consistently ranked as the least preferred option by 50% of the participants (M=3.14, SD=1.21), the platform (M=2.14, SD=1.21) ranked first 37.5% of the time, closely followed by MOOCs (M=2.28, SD=1.11) and traditional classroom (M=2.43, SD=0.97). The results suggest that this innovative teaching method stands as an effective competitor to both traditional and modern approaches.

V. Discussion and Future Directions

Overall, we find that the platform shows promising results across different key areas. All participants completed between 80% and 100% of the content with success, thereby reaching the intended learning outcomes within the time-frame that we set for them. Self-reported measures of usability, engagement, satisfaction, and intention to learn and recommend were positive and robust for both students and professionals, which entails that the platform can cater to diverse audiences. However, improvements are certainly possible and necessary: our effort should be seen a first-of-a-kind attempt to use in-tool tutorials in (security) requirements engineering.

As for game elements, follow-up interviews highlighted the real-time interaction with the modeling tool, the storyline and the delivery format as key success factors. Extrinsic traditional elements such as points, badges, and leaderboards were not found to be highly interesting to participants, as the narrative and exercises (i.e., intrinsic factors) were engaging per se [35]. This is very much aligned with the recent trend on rethinking gamification from individual game elements to delivering a holistic, immersive game experience [14].

Interviews with professionals revealed that they found the method and modeling language too complex to use in a real context, where diagrams would tend to grow to the point of being unmanageable. This explains why we noticed a significant difference in the intention to learn about SRE and the STS method compared to students. We should investigate further whether this perception is shared by other practitioners to better understand if this was due to the language itself or to some features of our tutorial. It is worth mentioning that the second author delivers full-day trainings with STS-ml that rely on an interplay of traditional teaching and hands-on sessions, while the in-tool tutorial has a more limited time span.

These early findings, although preliminary and suggestive at best, indicate that incorporating a gamification-powered platform into the teaching process leads to good results and possibly better reception of the classes by the learners. Interestingly, we can conclude that this novel approach is perceived at least competitive with and as effective as other established teaching methods such as classroom settings, MOOCs or books/self-study for teaching conceptual modeling.

In the near future, we are planning on conducting a large-scale evaluation with graduate students, which we hope to report in the upcoming RE conference.

Our long-term roadmap includes covering the entire STS method with additional courses and extending the platform with other (S)RE-relevant modules. For example, we plan to use a similar tutorial for teaching about the recently released iStar 2.0 language for goal-oriented early RE [36]. An important research question to address is to assess whether certain modeling languages are better adequate for learning via interactive tutorials.

We call for the joint support of the research community in helping advance innovative teaching methods such as this one, in the hope of more effectively engaging and training the next generation of requirements engineers and conceptual modelers.

References

[1] K. Pohl, Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated, 2010.

[2] P. Devanbu and S. Stubblebine, "Software Engineering for Security: a Roadmap," Proceedings of the Conference on the Future of Software Engineering, pp. 227–239, 2000.

[3] W. Stallings and L. V Brown, Computer Security. Prentice Hall, 2008.

[4] N. R. Mead, E. D. Hough, I. I. Stehney, and R. Theodore, "Security Quality Requirements Engineering (SQUARE) Methodology," 2005.

[5] G. McGraw, "Software Security: Building Security in," in Proceedings - International Symposium on Software Reliability Engineering, ISSRE, 2006, p. 6.

[6] E. Dubois and H. Mouratidis, "Guest Editorial: Security Requirements Engineering: Past, Present and Future," Requirements Engineering, vol. 15, no. 1, pp. 1–5, 2010.

[7] Y. Wand, D. E. Monarchi, J. Parsons, and C. C. Woo, "Theoretical foundations for conceptual modelling in information systems development," Decision Support Systems, vol. 15, no. 4, pp. 285–304, 1995.

[8] D. Basin, M. Clavel, J. Doser, and M. Egea, "Automated analysis of security-design models," Information and Software Technology, vol. 51, no. 5, pp. 815–831, 2009.

[9] F. Dalpiaz, E. Paja, and P. Giorgini, Security Requirements Engineering: Designing Secure Socio-Technical Systems, 1st ed. MIT Press, 2016.

[10] J. B. Biggs and C. Tang, Teaching For Quality Learning At University. 2011.

[11] G. Sedrakyan, M. Snoeck, and S. Poelmans, "Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling," Computers & Education, vol. 78, pp. 367–382, 2014.

[12] H. J. Nelson, G. Poels, M. Genero, and M. Piattini, "Quality in conceptual modeling: Five examples of the state of the art," in Data and Knowledge Engineering, 2005, vol. 55, no. 3, pp. 237–242.

[13] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," Proceedings of the National Academy of Sciences, vol. 111, no. 23, pp. 8410–8415, 2014.

[14] S. Deterding, "Eudaimonic Design, or: Six Invitations to Rethink Gamification," in Rethinking Gamification, 2014, pp. 305–323.

[15] F. Dalpiaz, E. Paja, and P. Giorgini, Security Requirements Engineering: Designing Secure Socio-Technical Systems, 1st ed. MIT Press, 2016.

[16] A. Hoffmann and R. Weißbach, "ImProject: Improving RE training by combining it with improvisation theatre sessions," in CEUR Workshop Proceedings, 2014, vol. 1217, pp. 11–19.

[17] D. Zowghi and S. Paryani, "Teaching requirements engineering through role playing: Lessons learnt," in Proceedings of the IEEE International Conference on Requirements Engineering, 2003, vol. 2003–Janua, pp. 233–241.

[18] P. Liang and O. De Graaf, "Experiences of using role playing and wiki in requirements engineering course projects," in 2010 5th International Workshop on Requirements Engineering Education and Training, REET 2010, 2010, pp. 1–6.

[19] D. Damian and B. Al-ani, "Instructional Design and Assessment Strategies for Teaching Global Software Development : A Framework," in ICSE '06 Proceedings of the 28th international conference on Software engineering, 2006, pp. 685–690.

[20] L. Beus-Dukic, "Final year project: A test case for requirements engineering skills," in Requirements Engineering Education and Training (REET), 2011 6th International Workshop on, 2011, pp. 5–8.

[21] L. Mich, "Teaching requirements analysis: A student project framework to bridge the gap between business analysis and software engineering," 8th International Workshop on Requirements Engineering Education and Training, REET 2014, in Conjunction with the 22nd International Requirements Engineering Conference, RE 2014, vol. 1217, pp. 20–25, 2014.

[22] I. Park and M. J. Hannafin, "Empirically-Based Guidelines for the Design of Interactive Multimedia," Educational Technology, Research and Development, vol. 41, no. 3, pp. 63–85, 1993.

[23] J. M. Carroll and H. Van der Meij, "Principles and heuristics for designing minimalist instruction," Business Insights: Essentials, no. May. pp. 1–28, 1995.

[24] C. Plaisant and B. Shneiderman, "Show me! guidelines for producing recorded demonstrations," in Proceedings - 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, 2005, vol. 2005, pp. 171–178.

[25] T. Grossman and G. Fitzmaurice, "ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1515–1524, 2010.

[26] Y.-K. Chou, Actionable gamification: Beyond points, badges, and leaderboards. 2016.

[27] S. O'Donovan, J. Gain, P. Marais, S. O. Donovan, and P. Marais, "A Case Study in the Gamification of a University-level Games Development Course," in ACM International Conference Proceeding Series, 2013, pp. 242–251.

[28] B. De Schutter and V. Vanden Abeele, "Gradequest — Evaluating the impact of using game design techniques in an undergraduate course," in Foundations of Digital Games, 2014.

[29] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, "Gamification in education: A systematic mapping study," Educational Technology and Society, vol. 18, no. 3, pp. 75–88, 2015.

[30] C. C. I. Muntean, "Raising engagement in e-learning through gamification," The 6th International Conference on Virtual Learning ICVL 2011, no. 1, pp. 323–329, 2011.

[31] K. Kapp, "The Gamification of Learning and Instruction, Pfeiffer," San Francisco, p. 480, 2012.

[32] K. Werbach and D. Hunter, For the win : how game thinking can revolutionize your business. Wharton, 2012.

[33] S. Jackson, A. J. Martin, and R. C. Eklund, "Long and short measures of flow: the construct validity of the FSS-2, DFS-2, and new brief counterparts," Journal of sport & exercise psychology, vol. 30, no. 5, pp. 561–587, 2008.

[34] I. Ajzen, "The theory of planned behavior," Organizational Behavior and Human Decision Processes, vol. 50, no. 2, pp. 179–211, 1991.

[35] R. Ryan and E. Deci, "Self-determination theory and the facilitation of intrinsic motivation," American Psychologist, vol. 55, no. 1, pp. 68–78, 2000.

[36] F. Dalpiaz, X. Franch, and J. Horkoff, "iStar 2.0 language guide," CoRR abs/1605.07767, 2016.