BakeRE

Design and Development of a Serious Game for Teaching User Stories

MERLE DELEN

First supervisor: FABIANO DALPIAZ
Second supervisor: MARCELA RUIZ

Universiteit Utrecht

Department of Information and Computing Sciences
Faculty of Science
Utrecht University

June 2019 – version 12

ABSTRACT

Learning requirements notations is a tedious task for most people, because this learning activity is all but engaging. As an alternative to traditional educational methods, BakeRE is presented; a serious educational game for requirements engineering. The game focuses on specific learning objectives: the specification and analysis of requirements with user stories.

Throughout this thesis, the effectiveness of BakeRE as a part of a course on Requirements Engineering at the Utrecht University is examined. Although the results do not lead to visible effects concerning the gained knowledge, perhaps also due to the short session the students were exposed to, foundations have been set for future experiments, as well as a list of possible improvements for BakeRE itself.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Fabiano Dalpiaz for all the help he gave me during the 8 months of working on this thesis. Many hours of meetings and emails went into the making of this thesis and I could not have finished it without his help.
My thanks also go out to my second supervisor Marcela Ruiz for proofreading my thesis and helping with the initial design of both the experiment and BakeRE, as well as Kendra Cooper for her interest in BakeRE and her help with writing the paper for "RE@Next!".

Additionally, I would like to thank my friends and family, in particular my boyfriend Bram Jonkers who would proofread my texts, playtest my game and always support me, even when I was complaining or stressed; my parents and brother for always being there for me; and everyone in "El Familia" – you know who you are. Thank you for all your love and support!

Finally, my thanks go out to all students of the Requirements Engineering 2018/2019 course at Utrecht University for joining the workshop. There would quite literally be no thesis without your participation.

Special thanks go out to André Miede for the development of the `classicthesis` style and LaTeXpackage, which was used to typeset this thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

AE	Aesthetics
API	Application Programming Interface
BDD	Behaviour Driven Development
FA	Focused Attention
GDD	Game Design Document
IDE	Integrated Development Environment
PU	Perceived Usability
RE	Requirements Engineering
RW	Reward Factor
SRS	Software Requirements Specification
UES	User Engagement Scale
UES-SF	User Engagement Scale Short Form
UML	Unified Modeling Language
US	User Story
USs	User Stories

Part I

CONTEXT

1 INTRODUCTION

Software production is more important now than ever. Not only is the number of software companies ever growing, but also most companies that are not IT based have an increasingly large dependency on IT [Regnell, 2005]. In this day and age, it is almost unthinkable for a company to not depend on some form of computers and their programs. With this in mind, it comes as no surprise that software production is a growing industry with increasing demands in terms of production time and quality.

To assure efficiency and customer satisfaction, creating clear definitions before starting programming tasks is essential. In other words, requirements engineering – the process of defining, documenting and maintaining requirements (simply put; customer wants and needs) – has become an indispensable part of software production. After all, if clear plans and agreements on the inner workings of the product are made beforehand, there is a smaller chance of running into big issues later on in the development which could setback the project significantly, as well as a clear vision for the customer to expect [Beatty and Alexander, 2008; Lawrence et al., 2001]. Creating a functional Software Requirements Specifications document (or an SRS) is a multi-step process, and many different techniques exist to create a well-made set of specifications. This thesis discusses the development of a serious game to help beginning learners with their studies into requirements engineering by tackling one specific subject within this field of study: User Stories.

1.1 PROBLEM STATEMENT

As Axel van Lamsweerde states in his book *Requirements Engineering, from System Goals to UML Models to Software Specifications*:

> "To make sure that a software solution correctly solves a particular problem, we must first correctly understand and define what problem needs to be solved." [Lamsweerde, 2009]

Therefore, it is important that the right amount of thought is put into the specification of requirements. Too little work on a system's requirements can cause many issues later on in development, while too much can become a project bottleneck [Knauss et al., 2008]. User Stories are a very popular tool for specifying software requirements

in agile development projects [Lucassen et al., 2016]. Several different formats and models are used in practice [Wautelet et al., 2014], but for the purpose of this thesis, they are sentences that follow the following format:

Listing 1: User Story Examples

```
As a <Role> I want to <Action> so that <Benefit>

--for example--
As a <Teacher> I want to <have insight into the latest
    results of my students> so that <I can monitor their
    progress>
```

Creating good user stories is a skill in and of itself. They can easily be vague, encompass more than one point of focus or be trivial. The example above is already shaky on several of these grounds; for example, what "progress" does the teacher actually want to see? And what exactly are results? Do the results contain only (specific) grades or also the answers to whatever exercises were made by the students?

Practice is essential when starting out with learning anything new [Beckers and Pape, 2016; Brabeck et al., 2011; Rusu et al., 2011], therefore this is also true for user stories. This research aims to improve this process by introducing a new educational and practice tool concerning user stories in the form of a *Serious Game* [Michael and Chen, 2005]. This new tool aims to improve knowledge and skill retention by utilising the framework commonly known as Bloom's taxonomy, helping students not only 'remember' knowledge, but also 'understand' and 'apply' this knowledge in practice. For more information on Bloom's taxonomy, see Chapter 2, Section 2.2.2 in particular.

Several studies into using Bloom's taxonomy as a basis to create educational tools and improve (active) learning have been performed before, such as an exploratory research into the teaching of Domain Modeling using Bloom's taxonomy as a base [Bogdanova and Snoeck, 2017] and the creation of a student self-assessment tool used during a programming course [Alaoutinen and Smolander, 2010]. Bogdanova and Snoeck's study presents evidence that, when looking at existing material for software courses through the lens of this taxonomy, the level of learning goals that is eventually achieved can be predicted. Furthermore, Alaoutinen and Smolander's study suggests that using this taxonomy during the development of study material could even improve the knowledge gains.

Several serious games regarding RE already exist [Dalpiaz and Cooper, 2018]. Most of those games focus on the first phase of RE, the "Elicitation" of requirements, or gathering the necessary information about requirements from the customer. However, the total number of games is still rather small, around 25 games at the time of writ-

ing. The oldest of these games, UTS-RE [Zowghi and Paryani, 2003], was published in 2003, while the newest addition, Garuso [Huber Kolpondinos and Glinz, 2017], was published in 2017. For a period of over 15 years, that is a rather small number. Therefore, there is still a lot of ground to cover, both in the general field of RE serious games and especially regarding games for RE concerning other phases than eliciting.

1.2 RESEARCH QUESTIONS

The main objective of this thesis is to develop a serious game that will help learners begin their studies in requirements engineering to learn write better, higher quality user stories. To that end, this research is concerned with the question of whether or not a well-designed game, using known design patterns for serious games and educational material as its basis, will improve the quality of the user stories produced. In other words:

> Does the usage of a serious game *during* the learning of user stories improve the learning experience and lead to higher skill & knowledge gain? — *Research Question*

A focal point here is the word "during". The game is to be used as a part or support of Requirements Engineering teaching, not as a replacement thereof. Initially, it is to be tested as part of a course on Requirements Engineering at the Utrecht University. Normally there is one lecture on user stories combined with one workshop for a session of 3 hours. This lecture is not entirely passive, as it contains tutorial elements such as "write twenty user stories", "judge ten stories on INVEST criteria" etc. This game should be used to enhance learning such as this.

To answer the research question, three subquestions were created:

- Does the usage of a Serious Game during the learning of User Stories improve the Knowledge Gain regarding User Story Quality? — *Subquestion 1*
- Does the usage of a Serious Game during the learning of User Stories improve the Knowledge Retention regarding User Story Quality? — *Subquestion 2*
- Does the usage of a Serious Game during the learning of User Stories improve the Learning (User) Experience? — *Subquestion 3*

For further information on these subquestions and the experiment that was conducted to examine them, please see Chapter 4 and 5.

1.3 OUTLINE OF THESIS

This thesis is divided into three main sections; Context, Research Design and Results & Conclusion.

I **Context** – the current section – deals (as the name implies) with the context necessary to understand this thesis. Chapter 1 focuses on the problem statement and (a short description of) the research questions that are examined during this study, whereas Chapter 2 contains an extensive literature study.

II **Research Design** details the research method created to perform this study, as well as the process with which this came to be. Chapter 3 gives a description of the design and creation of the serious game itself, whereas Chapter 4 deals with the description of the research questions, the experimental setup and the execution of the BakeRE workshop.

III **Results & Conclusion** describes the course of the study, gathers the outcomes of the performed research and details additional findings in Chapter 5. In Chapter 6 it then takes these findings and draws conclusions concerning the research questions, puts these into perspective with the current state of affairs and describes possible future work.

2 LITERATURE STUDY

2.1 REQUIREMENTS ENGINEERING

Requirements Engineering (RE) can be described as "the process of eliciting, evaluating, specifying, analyzing and evolving (system) requirements to be met for a software solution which addresses a specific problem", or, as Axel van Lamsweerde states in his book, "Requirements engineering (RE) deals with the variety of prerequisites that must be met by a software system within an organization in order for that system to produce stellar results" [Lamsweerde, 2009]. In other words, RE is the process of gathering information on what the client wants from their software and turning it into documentation with requirements (that are usable and testable) for the developers to use.

2.1.1 *Software Requirements Specifications*

According to van Lamsweerde, five main phases exist within RE. These phases are called Eliciting, Evaluating, Specifying, Analyzing and Evolving and are described in further detail in Table 2.1.

When going through the process of RE, information about requirements is first elicited from the stakeholders, usually through interviews, brainstorms and user observations, after which these initial requirements are evaluated and analyzed within the second phase. When these initial requirements have been thoroughly evaluated and approved, it is time to move on towards the third and fourth phase; specification and analysis. For the purpose of this thesis, this is where the main focus lies.

Within these phases, the gathered (usually informal) requirements are combined, defined and specified into one of several types of formal requirements documentation. This formalized document is then further analyzed for desired qualities, after which the Software Requirements Specifications (SRS) are complete. This document should be kept up to date for the remainder of the project lifecycle, even after its initial completion.

The specification phase of RE focuses on two parts; validation and documentation. **Validation** is known as the formalization of informal documents, whereas **Documentation** focuses on increasing precision, defining conventions and providing further contextual information. Both of these focus heavily on the removal of **Ambiguity**, as most

PHASE	DESCRIPTION
Eliciting	Understanding the domain in which the software project takes place and gathering the right requirements for a new system – what should the system do? What problems should it solve? Who will be the future users?
Evaluating	The process of evaluating the elicited objectives, requirements and assumptions about the new system. As an example, an *objective* would be to bake a loaf of bread, a *requirement* would be that the bread does not come out burned, and an *assumption* would be that the baker knows the recipe required.
Specifying	Making the elicited and evaluated objectives, requirements and assumptions fully precise and organize them into some coherent structure. This structure could take the shape of a fully written document, some table(s), model(s) etc., depending on the target audience (who will read/use the specifications?).
Analyzing	Checking the Software Requirement Specification(s) (SRS) for desired qualities such as *completeness* (does implementing all requirements in the SRS result in a feature-complete application?), *consistency* (do all requirements in the SRS follow a similar template?), *adequacy* (are all requirements "useful", insofar that they describe an actual issue and do not contradict each other) or *measurability* (is every requirement written in such a way that a quantitative test can be performed to see whether or not a requirement is fulfilled?) of statements.
Evolving	Managing requirements evolution, so that objectives, requirements and assumptions stay relevant as the world and project change.

Table 2.1: The five phases of RE [Lamsweerde, 2009]

requirements are written in Natural Language, which is inherently ambiguous [Cohn, 2004].

The IEEE (Institute of Electrical and Electronics Engineers) created a set of recommended practices for a good SRS in 1998 [*IEEE recommended practice for software requirements specifications* 1998], which was a revised version of an earlier document created in 1993. These best practices state that a good SRS must answer the following questions:

- **Functionality**; what is the software supposed to do?
- **External Interfaces**; how does the software interact with people/hardware/other software?
- **Performance**; what is the speed, availability, response time and the recovery time of the features?
- **Attributes**; what are the software's portability, correctness, maintainability and security constraints?
- **Design Constraints Imposed upon an Implementation**; are there any required standards in effect (such as implementation language, policies for database integrity, resource limits, operation environments, etc.?

The SRS should not include design or project requirements!

The best practices also include quality frameworks for the requirements themselves; a good requirement must be

1. Correct
2. Unambiguous
3. Complete
4. Consistent
5. Ranked for importance and/or stability
6. Verifiable
7. Modifiable
8. Traceable

Gathering, specifying and analyzing all this information upfront will yield a very robust document, however it will be difficult to change as requirements change. Additionally, creating a document as thorough as this will take a lot of time, which reinforces the negative stereotype that RE takes a lot of time and does not produce "necessary" work such as code and designs (more on this in Section 2.1.5).

These kind of impressively thorough documents were also never truly meant for products. The IEEE has a background of electrical engineers, creating highly dependent systems such as railroad systems

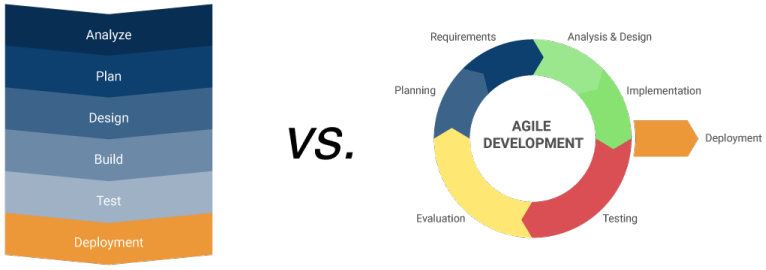

Figure 2.1: Waterfall model vs Agile model [1]

and power plants. In an industry like this, requirements are more like contracts and any missing or superfluous information before starting the development could prove fatal to the project, however, since coding standards have started to move from the traditional straightforward Waterfall model towards the cyclical Agile model [Beck et al., 2001, The Agile Manifesto], RE has to evolve with them, creating the more adaptive and iterative "Agile RE".

2.1.2 *Agile RE & User Stories*

Agile RE is defined as RE for an agile workflow. Due to the popularization of agile models within almost every branch of software development [Leffingwell, 2010], it has become increasingly important that RE evolves as well. Within this new stigma, extremely detailed and thorough SRS have become obsolete, instead giving rise to the new specification technique known as "User Stories".

User Stories (US) are sentences in the following format [Cohn, 2004]:

Listing 2: User Story Frameworks

```
1   As a <Role> I want to <Action> so that <Benefit>
    --or--
    As a <User> I want to <Means> so that <End>
```

Here, Role/User is a type of user, such as a Programmer or a Requirement Engineer, that states a requirement in the form of a US. This person has a want in the form of an Action/Means, such as "I want to learn about User Stories" or "I want to send a batch email". Finally we have an optional Benefit/End which describes the preferred outcome of the want or the reason for the want, for instance "[I want to send a batch email] so that I can reach many people at once". This Benefit/End is optional and does not always have to be included in every US. Some examples of USs in practice can be found in Listing 3.

Using USs makes it very easy to quickly represent "*who* it is for, *what* is expected from the system and (optionally) *why* it is import-

1 https://www.csgsolutions.com/blog/agile-vs-waterfall/

ant" [Lucassen et al., 2016]. Because of their simplicity and the fact that it is based on Natural Language, they are easy to understand and communicate to both client and programmers. Using USs has quickly become one of the most common practices within RE, with an adoption rate of 45% overall [Kassab, 2015] and an even higher adoption rate of 90% when focusing on Agile RE alone [Wang et al., 2014].

Listing 3: More User Story Examples

```
As a <Requirements Engineer> I want to <improve my skills
    concerning User Stories>.

As a <Baker> I want to <receive notifications when my oven
    reaches a certain temperature> so that <my baked goods do
    not burn>.

As a <Student> I want to <have access to my
    grades> so that <I can keep track of my academic
    progress>.
```

2.1.3 *User Story Quality*

The quality of a US can be tested using several Quality Frameworks. One of the first to be popularized by practitioners is Bill Wake's 2003 "INVEST" framework[2], an acronym or mnemonic that describes the six characteristics a quality US should have. According to INVEST, a good US must be:

- **I**ndependent; dependencies between USs shall be avoided (to the extent this is possible),
- **N**egotiable; details of the story can be discussed during the iteration planning meeting,
- **V**aluable (to the customer); no implementation-only USs should exist,
- **E**stimable; there are enough details within the USs to estimate the effort required for implementation,
- **S**mall/**S**calable (in effort); no "big" requirements,
- **T**estable (with certain acceptance criteria); see Section 2.1.4.

However, even when a US is found "correct" by the INVEST standards, there could still be numerous other types of errors within the US. For example, the INVEST framework does not test on ambiguity nor on length of USs (the "Small" in INVEST stands for the size of the effort, not the length of the US itself). Therefore, Lucassen et al. introduced another framework based on INVEST called the "QUS"

2 https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

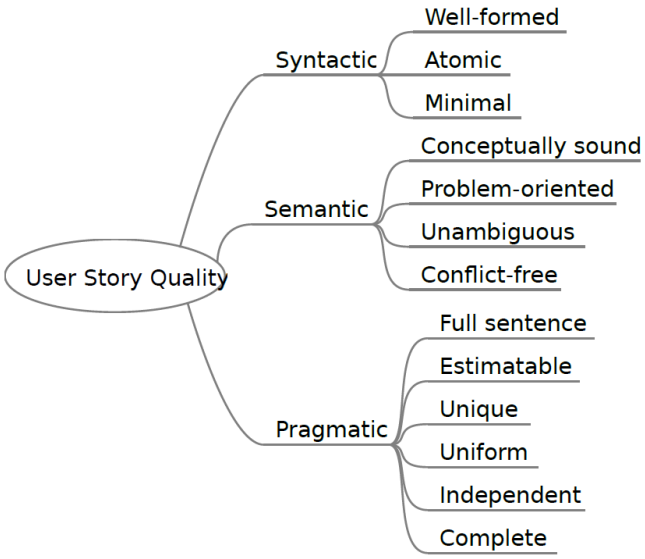

Figure 2.2: The categories of the QUS and their relationships [Lucassen et al., 2016]

or Quality User Story framework [Lucassen et al., 2016]. This framework categorizes quality indicators in using Lindland's categories; Syntactic, concerning the textual structure of the text without considering its meaning, Semantic, which does concern the meaning of the text, and Pragmatic, concerning the audiences subjective interpretation of a text [Lindland et al., 1994; Lucassen et al., 2016]. These can be seen in further detail in Figure 2.2.

When judging and improving the quality of US, it is common practice to first consider the Structural and Syntactic categories to "clean" the set of US, after which the Pragmatic checks will only be performed on the USs that remain. The 13 quality criteria themselves are described as follows by Lucassen et al. [Lucassen et al., 2016], or a quality US should be:

- **Syntactic**
  1. Well-formed; includes at least a Role and an Action,
  2. Atomic; expresses a requirement for exactly one feature,
  3. Minimal; contains nothing more than Role, Action and Benefit,
- **Semantic**
  4. Conceptually Sound; the Action expresses a feature and the Benefit a rationale,
  5. Problem-oriented; only specifies the problem, not the solution to it,
  6. Unambiguous; avoids terms that lead to multiple interpretations,

7. Full-sentence; is a well-formed full sentence,
8. Estimable; does not denote an unrefined requirement that is difficult to plan and prioritize,
9. Conflict-free; there should not be >2 inconsistent USs,

- **Pragmatic**
  10. Unique; duplicate USs shall be avoided,
  11. Uniform; all USs in a specification follow the same template,
  12. Independent; a US is self-contained and has no inherent dependencies on other USs,
  13. Complete; implementing a set of USs creates a feature-complete application.

2.1.4 *Acceptance Criteria*

After having ascertained that all USs within a set are of good quality, criteria should be created with which can be tested whether or not a US has been implemented and "completed". This is known as Behaviour Driven Development (BDD), as popularized by Dan North [North, 2006]. The tests that are created to complement each US are called "Acceptance Criteria". Generally speaking, an acceptance test has the following formula:

Listing 4: Acceptance test framework

```
Given   <some context>
When    <some action is carried out>
Then    <a set of observable consequences occurs>
```

BDD advocates writing acceptance tests that promote complementing the who, what and why parts of a US and determine when a US is fulfilled. Acceptance tests are, as the name implies, written to be testable and executable; therefore each acceptance test needs a clear set of conditions that can be met, with a clear pass/fail indicator. For instance, an ambiguous statement such as "[..] the webpage should load as fast as possible" is not measurable, while "[..] the webpage should load within 2 seconds in 95% of all cases" is. There are several tools available to automatically execute acceptance tests, the most famous of which is the open source toolset Cucumber[1] with it's natural language processor Gherkin[2].

1 https://docs.cucumber.io/
2 https://docs.cucumber.io/gherkin/

2.1.5 *State of Affairs*

The main issue concerning RE right now is that, although it has been shown that about 50% of defects are introduced within the RE phase of software development [Beatty and Alexander, 2008; Lawrence et al., 2001], the value added by RE is often hard to see for both developers and clients [Knauss et al., 2008]. This is mainly because RE does not immediately produce code nor designs, therefore giving the illusion that it is "wasted time", especially when most software projects already have very limited time on their hands. However, mistakes and defects that occur during the RE phase of a project are usually very difficult to tackle later on in the project, with (defect) requirement reforming amounting to 50-80% of total project effort when discovered very late in the product development lifecycle [Lawrence et al., 2001].

Additionally, since RE consists of many different phases, it combines analytical tasks with creativity [Pinto-Albuquerque and Rashid, 2014] and combines widely varying skillsets, it is difficult to attain the complex set of skills necessary to become a true RE specialist [Beatty and Alexander, 2008]. Practitioners, both beginners and more advanced users, need help to attain this skillset. Not everyone can receive private guidance however; therefore it would be more practical and reach more potential practitioners to create a tool that is available to use at all times.

However, as of now, not many tools exist to further this goal. Additionally, the tools that do exist focus mainly on the "Eliciting" phase of RE, most likely due to this part being what most people think about when they think about RE (more on this in Section 2.2). Therefore, the aim of this research is to add to this ongoing field of study by creating a tool to practice creating and analyzing User Stories for Agile RE in the form of a Serious Game.

2.2 SERIOUS GAMES

Many definitions of "serious games" exist. These definitions vary wildly from person to person, but the part where nearly everyone agrees is that "Serious Games are (digital) games where the primary purpose is something other than entertainment" [Michael and Chen, 2005]. Here, a game is defined as "a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome" [Salen and Zimmerman, 2003], while this primary purpose can be any subject matter, such as medicine, teaching languages and the military [Hainey et al., 2011]. Of course, this also includes software engineering and RE (more on this subject in Section 2.3).

For the purposes of this paper, a clear distinction has to be made between the sometimes interchanged terms "Serious Games" and "Gamification". Gamification is commonly described as "the use of game (design) elements in non-game contexts" [Deterding et al., 2011]. In other words, serious games concern true games or tools, whereas gamification only concerns game elements within none-game surroundings. This research is concerned with serious games and, therefore, not with gamification.

2.2.1 *Games for Learning*

Serious games are currently popular in the context of education. These games have "learning" something, such as a skill, a tool or knowledge, as its primary purpose apart from entertainment. These types of "Games for Learning" are sometimes called "edutainment", a combination of education and entertainment. This combination of terms, "learning" and "games", may seem contradictory at first, or as Huynh-Kim-Bang et al. state:

> "Games are associated with pleasure and freedom to play when and where one wants to; while learning is more readily associated with constraints and difficulties. Video games are associated with interaction, practice, and immersion in the game environment; on the other hand, learning can require a break in the activity so as to take the time to think about what is going on and what one is learning" [Huynh-Kim-Bang and Labat, 2008].

However, nothing could be further from the truth, as it is exactly this interactive and exploratory nature of games that make them so useful for learning. This contradiction does need to be taken into account while designing serious game applications however, as a game can become too constrictive to still be entertaining, as well as too loosely focused to still be educational.

SUBJECT	STUDIES & FINDINGS
Knowledge	"[..] instruction with serious games yields a higher level of retention than training with conventional instructional methods. Indeed, the results show that the superiority of serious games over conventional instructional methods is maintained in a delayed test" [Wouters et al., 2013]. "Recent meta-analyses consistently report improved cognitive learning outcomes (up to 17% better)" [Sitzmann, 2011; Vogel et al., 2006]. "Results from media comparisons indicated that digital games significantly enhanced student learning relative to nongame condition" [Douglas et al., 2016].
Motivation	"[..] serious games are more motivating than conventional instructional methods. Although the summary effect size is [..] in favor of serious games, [..] the difference in motivation is not statistically significant" [Wouters et al., 2013]. "Findings from the present meta-analysis do diverge slightly from the [Wouters et al., 2013] finding in that game conditions and nongame instructional conditions did not differ in terms of motivation outcomes [so there *is* a significant different in terms of motivation]. In the current study, the intrapersonal learning outcome domain not only included motivation but also included intellectual openness, work ethic and conscientiousness, and positive core self-evaluation" [Douglas et al., 2016]. "Entertainment can increase the students emotional involvement with the material and thereby increase content retention" [Millbower, 2003]
Supplemental Methods	"serious games supplemented with other instructional methods will yield higher learning gains than serious games without supplemental instructional methods" [Wouters et al., 2013].
Duration	"With regard to duration of game play, Sitzman found that media comparisons in which trainees had unlimited access to the game demonstrated significantly better learning outcomes than media comparisons in which the trainee had limited access to the game" [Sitzmann, 2011].
Other Variables	"Numerous moderator variables (e.g., demographics, domain, game concept, game design features) have little effect on the learning outcomes" [Dalpiaz and Cooper, 2018; Wouters et al., 2013].

Table 2.2: A short overview of studies into the effectiveness of serious games

It is said that games improve the learning process by a) influencing motivation and b) changing cognitive processes [Wouters et al., 2013] (both of these dimensions will be discussed in further detail in Section 2.2.2 and 2.2.3). There have been several studies that ascertain whether or not serious games improve skill/knowledge retention as well as motivation and affective learning; an overview of these findings relevant to this research can be found in Table 2.2. From these findings it can be concluded that games for learning (if designed well) should improve learning gains as well as knowledge/skill retention, as well as (somewhat) improving motivation and affect.

2.2.2 *Cognitive Processes*

The basis of a good educational game lies in educational theory. After all, even when a game is highly addictive and very fun to play, if the educational theory behind it does not hold up to scrutiny, no lessons will be learned and no learning goals will be met [Greitzer et al., 2007; Marne et al., 2012].

First, a distinction needs to be made between "Passive" and "Active" learning.

- **Passive learning**; learning by listening to lectures, reading a predetermined text, watching a prerecorded video etc.
- **Active learning**; replacing or complementing traditional lectures with active or autonomous discovery learning experiences, as well as types of open-ended problems requiring critical or creative thinking.

Implementing multimedia elements (such as hyperlink navigation or video players) may give the illusion of active learning, however, it is still passive learning in practice [Greitzer et al., 2007]. However, games for learning obviously fall into the second (active) category. It has been shown that students are more likely to retain information taught through a variety of methods than through only passive lectures [Beatty and Alexander, 2008].

In 1956, Bloom et al. created a highly influential piece for educational theory; Bloom's taxonomy [Bloom et al., 1956]. This framework is meant to be used by educators of all kinds to categorise educational goals. The original taxonomy consisted of six categories (see Figure 2.3) which described educational goals. Each goal builds upon the layer before it, hence when one could "Create" something from scratch, they had mastered this particular field. In 2001 this framework was revised [Anderson et al., 2001], resulting in a set of "action words" or verbs to accompany goals within the framework. For instance, "Comprehension" got the verbs "restate, translate & describe".

3 https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy

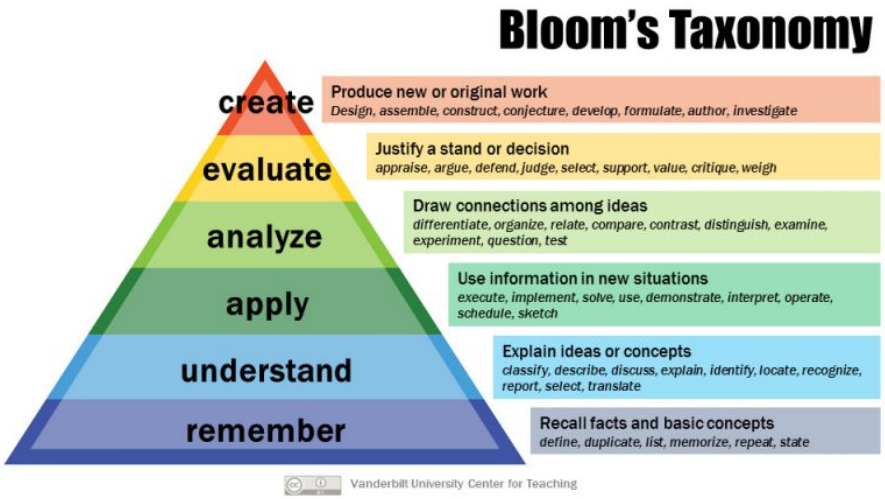

Figure 2.3: Bloom's taxonomy, revised (2001) [Anderson et al., 2001] [3]

The 6 layers, their action words and some examples are detailed below:

1. **Knowledge** *"Remember"*: memorise, state, name, recognise. For example; being able to state the name of the capital of your country.

2. **Comprehension** *"Understand"*: restate, translate, describe. For example; being able to explain photosynthesis and restate it in other words.

3. **Application** *"Apply"*: calculate, write, solve. For example; being able to use the Pythagorean theorem to calculate the length of the hypotenuse of a triangle.

4. **Analysis** *"Analyze"*: categorise, differentiate, discriminate. For example; being able to discern whether a computer program is correct or not.

5. **Evaluation** *"Evaluate"*: assess, evaluate, judge. For example; being able to judge and comment on other peoples work.

6. **Synthesis** *"Create"*: create, design, plan. For example; being able to design, plan and create a computer program from scratch, without relying on outside help.

The authors of the revised taxonomy also created a subdivision of the bottom level of the pyramid, "Knowledge" [Anderson et al., 2001]. These levels are also hierarchical, and therefore build upon the levels that preceded them. According to the authors, knowledge consists of four different levels:

1. **Factual knowledge**: the basics of studied discipline, such as basic terminology.

2. **Conceptual knowledge**: implies understanding of the connections and interrelationships between the factual level 1 elements.

3. **Procedural knowledge**: refers to the subject specific methods, procedures and rules.

4. **Meta-cognitive knowledge**: implies strategic knowledge and the student's awareness of his/her own knowledge.

When looking at this framework, it becomes obvious that just providing a learner with information through passive learning will not lead to true mastery of the skillset. Therefore, a gradual increase of the difficulty of the questions and tasks coupled with the addition of new types of activities higher up in the hierarchy will amount to the highest knowledge gain and retention.

This information can be coupled with cognitive theory. This theory states that human working memory is very limited, but people do have a nearly unlimited long-term memory [Atkinson and Shiffrin, 1968; Miller, 1956]. However, when performing learning tasks, humans have to mainly use their short-term memory, especially in the beginning of their learning process. To help with this, humans have many "schemas" representing relationships among facts in their knowledge structures, allowing for many elements of knowledge to be treated as one piece of knowledge in working memory [Kotovsky et al., 1985; Schneider and Shiffrin, 1977; Shiffrin and Schneider, 1977]. This reduces cognitive load and allows for faster and less demanding processes.

To properly construct schemas in ones memory, it is important that these schemas are traced many times. In other words; practice. When practicing something often, the schemas get solidified in ones mind, which causes them to be treated like one piece of knowledge in cognitive memory, which leads to higher levels of automaticity and better knowledge retention.

Finally, when designing a serious game for learning, it is important that the learner has time to reflect in between the tasks he or she is performing [Mayer, 2004]. This too has to do with knowledge schemas; when reviewing and reflecting information, people make sure that they "save" the data in the right locations and create relations between knowledge structures in their head. Or, as Huynh-Kim-Bang et al. say in their Serious Game Design Patterns: "(there is a) Time for Action (and a) Time for Thought" [Huynh-Kim-Bang and Labat, 2008].

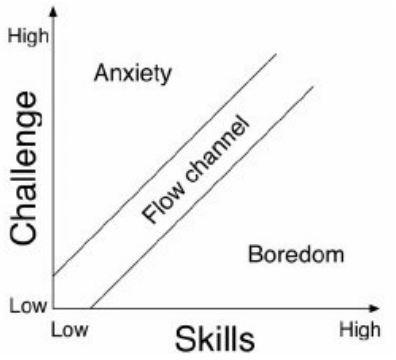

Figure 2.4: Csikszentmihalyis's model of Optimal Flow [Csikszentmihalyi, 1990] [4]

2.2.3 *Motivation*

When most people consider the benefits of serious games, one of their first thoughts is improved motivation. After all, games are supposed to be fun, whereas for learning this is not always the case. Whether or not serious games truly improve motivation among learners has not been definitely proven, as many studies both accepting and denying the fact exist [Wouters et al., 2013].

(User) Engagement can be defined as "the emotional, cognitive and behavioural connection that exists at any point in time, and possibly over time, between a user and a resource" [Lombriser et al., 2016]. Motivation is concerned with keeping the user (or in this case player) engaged with the material, for either Intrinsic or Extrinsic reasons [Ryan and Deci, 2000].

- Intrinsic motivation: when a person does something simply because they enjoy this activity
- Extrinsic motivation: when a person does something for either external rewards (such as money or a good grade) or to avoid negative consequences

Preferably, games should always be intrinsically motivating. However, due to the nature of serious games, it is often impossible to create experiences without any extrinsic motivation, as people who play these games usually have a clear, real-world goal to obtain by playing the game (therefore becoming unable to truly immerse themselves in the experience) and are often told to by teachers or managers.

Another important theory to consider when discussing serious game effectiveness is Csikszentmihalyis's "theory of Flow" [Csikszentmihalyi, 1990]. This theory states that, in order to become truly engaged with an activity, the perfect middle ground between "Challenge" and

4 https://www.researchgate.net/publication/251772214_Engaging_the_Learner_How_Can_the_Flow_Experience_Support_E-learning/figures?lo=1

"Skill" needs to be found. In other words, in the case of a game, the gameplay should never be so difficult that it causes frustration or anxiety, but should also not be so easy that the player will become bored with the game (see Figure 2.4).

This optimal flow can be achieved by having the game grow with the player; as the player progresses within the game, their mastery of (a part of) the skill increases. When the player has mastered this, the next part of the challenge will become more difficult, combine formerly learned skills in new ways or introduce a new skill altogether to prevent boredom. On the other hand, a game should also provide continuous feedback so players do not get "stuck" and become frustrated [Lombriser et al., 2016]. When both of these tasks are performed correctly, it should be possible to create an activity that adheres to this optimal flow, therefore maximizing engagement for the player.

2.2.4 *State of Affairs*

As of now, there are many schools of thought as to the "correct" or "best" way to create a serious game. Among the most famous examples are Peters's "The Art of Game Design; a Book of Lenses" [Peters, 2014] and Marne et al's Six Facets of Serious Game Design" [Marne et al., 2012]. In this study, the design of the user story serious game will mainly take inspiration and guidance from Huynh-Kim-Bang and Labat's "Design Patterns in Serious Games: A Blue Print for Combining Fun and Learning" [Huynh-Kim-Bang and Labat, 2008].

The preference for this framework stems from the fact that it is a) concise (as opposed to the highly complete but somewhat convoluted Book of Lenses) and b) focused, detailing several design patterns for creating games for learning, split among six clear-cut categories. It is also very pragmatic, making it easily applicable to real-world scenarios, as opposed to the very high level Six Facets.

The six categories detailed in the Design Patterns are as follows:

1. When to combine entertainment & learning?
2. How to make interaction interactive?
3. How to initiate the reflective process?
4. How to convey information without disturbing game immersion?
5. How to motivate users?
6. How to help users advance in the game?

For every category there are multiple subcategories, complete with context and examples. These patterns take on the form of statements

like "Questions and Answers", "Informative Loading Screens" and the aforementioned "Time for Action/Time for Thought". These patterns offer a solid foundation to design the game upon, as well as offer inspiration when dealing with related issues.

Another support and inspiration for the design of the game are the "Instructional design principles for Active Learning" [Greitzer et al., 2007]:

- Stimulate semantic knowledge;
- Manage the learners cognitive load;
- Immerse the learner in problem-centered activities;
- Emphasize interactive experiences;
- Engage the learner.

Although more vague than the design patterns mentioned above, these five lessons are very important to keep in mind when creating an active learning experience, to ensure that the learner gains and retains as much knowledge as possible.

2.3 GAMES FOR RE

The field of games for - learning or practicing - RE is still in its infancy. A recent analysis of the current state of affairs by Dalpiaz and Cooper [Dalpiaz and Cooper, 2018] reveals that there are currently 21 games for RE (backed up by published research papers), ranging from physical games (such as RE-O-Poly based on the famous board game Monopoly [Smith and Gotel, 2008] and card game HATCH [Beckers and Pape, 2016]) to digital games (such as SW-Quantum [Knauss et al., 2008] and REfine [Snijders et al., 2015]). Most of these games focus on the Elicitation phase of RE however; 11 out of 21 that were analyzed within this study concern this part of the RE practice. There are also slightly more games for learning about the RE discipline than there are for existing practitioners.

Most of these games use some form of metaphor to get their point across, introduce some form of narrative and make the game more game-like. For instance, the Jigsaw Puzzle [Pinto-Albuquerque and Rashid, 2014] shows conflicts within an RE process as puzzle pieces which seem to fit at first glance but upon closer inspection are just too big or too small, while the Earth Defense game [Rusu et al., 2011] shows the impact of (wrongly) elicited requirements as a defense mechanism against aliens that will or will not defend the earth depending on whether the player has gathered enough information within the time limit to build the system correctly.

There have been numerous studies into the effectiveness of metaphors for learning. Some remain neutral in their findings, finding little to no (additional) effect on learning gains when using (in this case visual) metaphors [Rieber and Noah, 2008], while others do see an increase in (foundational) knowledge gain [Mouraz et al., 2013].

Furthermore, according to linguists George Lakoff and Mark Johnsen, metaphors are the mind's natural mode of mental operation, as humans necessarily conceptualize their world in terms of bodily functions (such as vision, hearing etc) [Lakoff and Johnsen, 2003]. Therefore they claim that using (appropriate) metaphors will help with the mental processes of learning, as it conceptualizes the knowledge in a way the human mind is equipped to take in.

The results of the analyzed games seem to corroborate this, as they all see some growth in the knowledge gained, although none of these papers specifically addressed (the usefulness of) the metaphor as part of their study.

Part II

RESEARCH DESIGN

3 GAME DESIGN

This chapter describes in chronological order the process through which the game was designed. A short timeline is as follows: first, several game ideas were created. These were then assessed using an ad-hoc framework made specifically for this research. The two game ideas that were considered most promising were then expanded upon in two short Game Design Documents (GDD). These two GDDs have been compared and discussed thoroughly, after which one GDD remained.

Using this short GDD as a template, the final game design was created. As part of this, the appropriate programming tools for the task had to be decided upon. Therefore a comparison of available techniques was performed to find the programming tools best suited to the task, which are further described in Section 3.7.

BakeRE can be found, and is free to download, within the Google Play store for Android[1]. The full sourcecode is freely available on GitHub[2].

3.1 GAME DESIGN SELECTION

To start out the game design process, a list of user story activities was created. These served as the foundation of the types of metaphors that could be created to transfer the appropriate skills and knowledge. This list was based on the material on user stories taught during the course on RE at Utrecht University, the course where the workshop using BakeRE was conducted. A short version of this list (without sub elements) is as follows:

- Writing User Stories
- Analyzing User Stories - Ambiguity/Natural Language
- Analyzing User Stories - INVEST
- Analyzing User Stories - QUS
- Creating User Story Models
- Creating Acceptance Tests
- Prioritizing User Stories

1 https://play.google.com/store/apps/details?id=tech.bakere.bakere
2 https://github.com/YinUkume/BakeRE

3.1.1 *Game Ideas*

Using the list above, several concepts were created that could be made into a serious game explaining and practicing these skills. The following is a short description of the original six game ideas that were considered "best" and made it to the scoring phase. These texts are the (slightly altered) original notes concerning the conceptual game design phase, as it was unnecessary to create an entire game design document for every idea before initial scoring had taken place.

1. **Building**: A building game, for example making sandwiches/-cooking, building houses or puzzles etc. where the building blocks are US components. The available blocks will have more "wrong" parts the further the player progresses, players can judge existing structures etc.
2. **Card**: A "trading card game" or TCG where different categories/mistakes can be used like resources/attacks to battle opponents (which are for example faulty US).
3. **Detective**: A mystery-esque game where people need to be interrogated to iron out inconsistencies/paradoxes between testimonies. Includes finding "evidence" (for example INVEST levels) to be used on faulty testimony to get the "truth" out.
4. **Machine**: Making a machine based on USs collected and strung together, then watch what kind of monstrosity it creates if mistakes were made during the process.
5. **Puzzle**: A game where an image needs to be completed, either something jigsaw like (all pieces only fit together in one way) or more free-form (like a tangram where there might be multiple ways to finish the same predetermined image).
6. **Turn-based**: Turn-based strategy/role playing game where team members can be gathered with special skills to fulfill quests/-missions and eventually defeat a boss monster. The special skills correspond (for example) to categories/criteria/writing rules etc.

3.1.2 *Scoring the concepts*

To see which of the game ideas described above would be extended into a Game Design Document (GDD), the author of this thesis and one additional student of the Department of Information and Computing Science performed a scoring exercise on the concepts (the two sets of scores are shown side by side, divided by a "|"). This scoring was performed using a scale of one to five (where 1 = worst and 5 = best) based on four categories:

GAME IDEA	FUN	USEFULNESS	EFFORT	EXTENSIBILITY	SCORE	TOTAL
Building	3 | 3	5 | 5	3 | 3	5 | 5	16 | 16	32
Card	3 | 3	3 | 4	2 | 1	2 | 4	10 | 12	22
Detective	4 | 4	4 | 5	2 | 3	3 | 2	13 | 14	27
Machine	3 | 3	4 | 4	4 | 2	4 | 2	15 | 11	26
Puzzle	2 | 1	2 | 2	5 | 5	2 | 3	11 | 11	22
Turn-based	5 | 4	2 | 3	2 | 2	3 | 4	12 | 13	25

Table 3.1: Scoring of the initial game concepts

- Fun: does a game like this seem fun to play? Would it not become boring or repetitive?
- Usefulness: would a game like this be a suitable metaphor for learning the desired skillset?
- Effort: would it be feasible to create a game like this within the desired timeframe, e.g. around three months?
- Extensibility: is the game easily extensible to include more concepts, both during this study and in possible future work?

The results of this can be found in Table 3.1. When looking at this table there is one clear "winner" which is *Building* with 32 points, however it scored only average on "Fun" which is one of the major points for a good serious game. Therefore the runner-up *Detective* (with 27 points) was also chosen as a potential concept. These two ideas were then further expanded upon in short GDD's.

3.1.3 *Short Game Design Documents*

The two short GDD's are both attached to this thesis, in the Appendices A and B respectively. After discussing the pros and cons of both concepts with the thesis supervisor, it was decided that both concepts showed clear promise for a game for teaching RE/User Stories. The "Building" concept however was better suited to both the scope and timeframe of this thesis. Therefore this is the concept that was taken as the basis for the actual game design as described in Section 3.2.

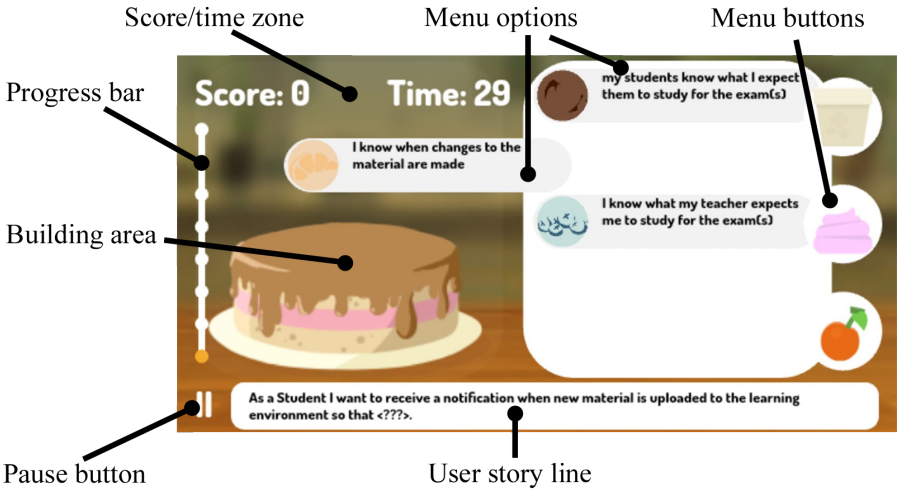

Figure 3.1: Explanatory screenshot of the play state

Figure 3.2: Correct answer

Figure 3.3: Level complete

Figure 3.4: Menu overview (actions)

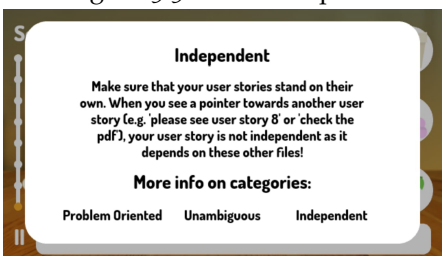

Figure 3.5: The pause screen

3.2 DESIGN OF "BAKERE"

3.2.1 *Introduction*

BakeRE is an educational game to assist in the learning and practicing of user stories. The game is an association puzzle game, where a player has to "build" a user story based on a description using predetermined building blocks. Thematically, the game takes place in a bakery, where building blocks are varying ingredients. The goal is to create the correct user stories (or "cakes") as fast as possible to achieve the highest score. Players compete against each other to get the highscore through an online leaderboard (see Figure 3.6), which can be found at `http://www.bakere.tech`.

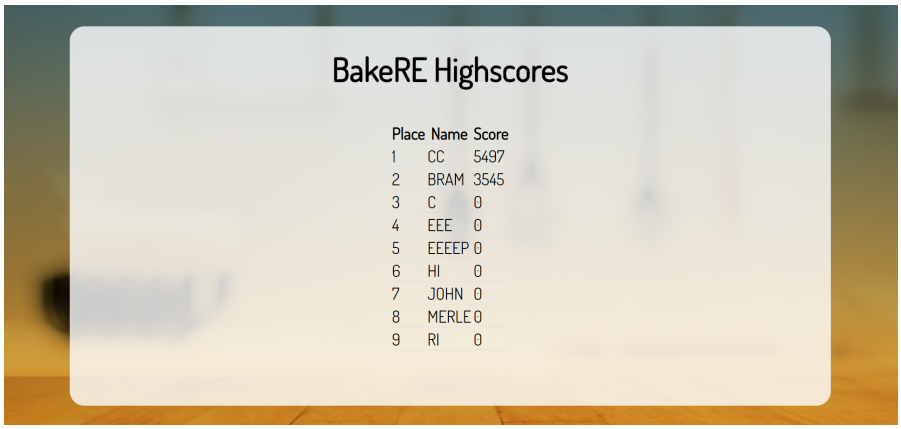

Figure 3.6: The online leaderboard of BakeRE

Figure 3.1 through 3.5 show the current version of the game. To understand the descriptions of the levels within this document, the following terms (as shown in Figure 3.1) are important:

- **Context**: a textbox that pops up at the start of a new batch (that can be reviewed again if necessary) describing the context of this batch of user stories in one or two short sentences (as shown in Figure 3.5).
- **Menu**: where the building blocks are stored.
  - *Menu Buttons*: each button opens one of the "dropdown" menu options for whichever category is available within this level.
  - *Menu Options*: the (at most 5) options for building blocks within this category that the player can choose from.
- **Building area**: where the building blocks (or ingredients) from the menu need to be dragged to build the user story structure.
- **User story line**: the user story that is created as a cake in the Building area is shown in text along the bottom of the screen.
- **Score/time area**: where the points earned by the player during the level is shown, as well as the time that is left to finish this particular user story.

The scene setup of BakeRE can be seen in Figure 3.7. Every level starts with an introduction phase, where the focal concepts within it are explained using short pieces of text, after which the gameplay begins.

At the end of a level the player is shown a "Debriefing" of their performance during the level. The debriefing contains an overview of level performance. It also contains some additional information concerning the areas where many mistakes were made, a general state-

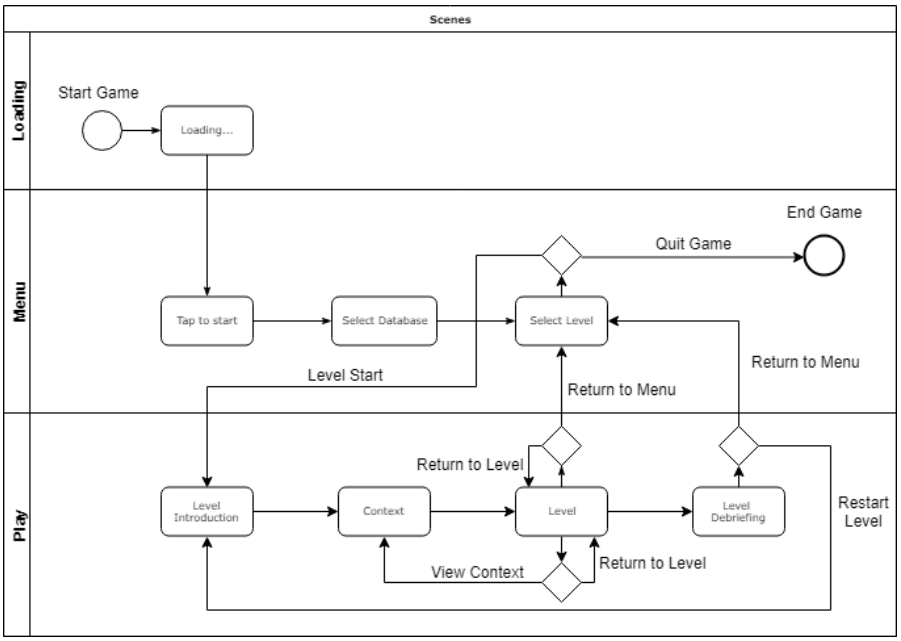

Figure 3.7: Scene setup of BakeRE

ment about the level, and the time and score with which the player has completed the level.

3.2.2 *Dataset*

The database is created using an Excel template made specifically for BakeRE. An Excel plugin reads this database and turns it into usable JSON files, which are then read within the game. The database used during the creation of- and initial workshop with BakeRE is attached to this thesis in Appendix C. The JSON format for a database is shown in Appendix D.

A dataset for BakeRE should contain the following:

- A title (the title of the Excel file is used for this)
- 8 "Epics" (Context)
  - 5 user stories per epic, for a total of 40 user stories, containing a role, an action and a benefit
  - Faulty user stories per epic, 4 per QUS category focused on, for a total of 24. These mistakes are "wrong" versions of existing user stories already in the set (see Table 3.2 for an overview).
    - for epic 1; 2 atomic mistakes & 2 minimal mistakes
    - for epic 2; 2 problem oriented mistakes & 2 full-sentence mistakes
    - for epic 3; 2 unambiguous mistakes & 2 independent mistakes

    * for epic 4 & 6; 1 mistake each in atomic, minimum & independent
    * for epic 5 & 7; 1 mistake each in problem oriented, full-sentence & unambiguous
  - For epic 8, 3 acceptance tests per user story belonging to this epic, for a total of 5 * 3 = 15 acceptance tests.

EPIC	1	2	3	4	5	6	7	8
Atomic	2	-	-	1	-	1	-	-
Minimal	2	-	-	1	-	1	-	-
Problem Oriented	-	2	-	1	-	1	-	-
Full Sentence	-	2	-	-	1	-	1	-
Unambiguous	-	-	2	-	1	-	1	-
Independent	-	-	2	-	1	-	1	-

Table 3.2: Mistakes per user story in database

3.3 LEVELS

The next section gives an overview of the five levels within BakeRE; four levels on user stories and one "tutorial" level on the basics of the game itself. Every level has allotted learning goals, structured by Bloom's revised taxonomy [Bloom et al., 1956]. For more information on the levels of learning, see Section 2.2.2.

3.3.1 *Tutorial*

The tutorial introduces the basic game mechanics, as well as a very short introduction into what a user story is. It takes the player through the gameplay of a level by highlighting areas of the screen and only allowing the player to do exactly as the tutorial says, guiding them on the path of creating a user story within the confines of this game.

- **Learning goals:** The basics of the game
- **Expected playtime:** 3:00 – 4:00
- **Introduction:** "Welcome to BakeRE! In this game, you author user stories by making cakes. Every cake consists of three ingredients: cake bottom, icing and decorations. This mirrors the makeup of a user story with their Role-Action-Benefit structure. A structurally correct user story has the following format:

BATCH	US SHOWN
1	3
2	4
3	4
4	5
5	5

Table 3.3: Level 1 - structure - 5 batches, 21 correct user stories

'As a <Role> I want to <Action> so that <Benefit>'

To complete a batch of cakes, you have to complete all correct user stories in that batch, and to complete a level, you have to complete all batches in the level." *(from here on out, the actual tutorial starts and parts of the screen are pointed out in order until the tutorial ends, see description below)*

First, a short introduction is given. Then the player is shown how to create a user story with only one building block to chose from per category. Finally, the player is given free reign to finish a first batch of 3 user stories. Completing this batch leads to a short outro, after which the player is brought back to the level select and can instantly start Level 1.

No points can be earned within the tutorial.

3.3.2 *Level 1 - Structure*

In this level, the player is educated on the basic structure and usage of a user story.

- **Learning goals:**
  - Remember: the structural make-up of a user story: role, action, and benefit.
  - Understand: the connection between an epic (context) and a user story.
  - Apply: the knowledge from the two points above by combining building blocks into correct user stories given a context.
- **Expected playtime:** 6:00 – 10:00 min.
- **Introduction:** "Welcome back! To win this level, complete again all user stories across the different batches. You can always review the context by pausing BakeRE. Good luck!"

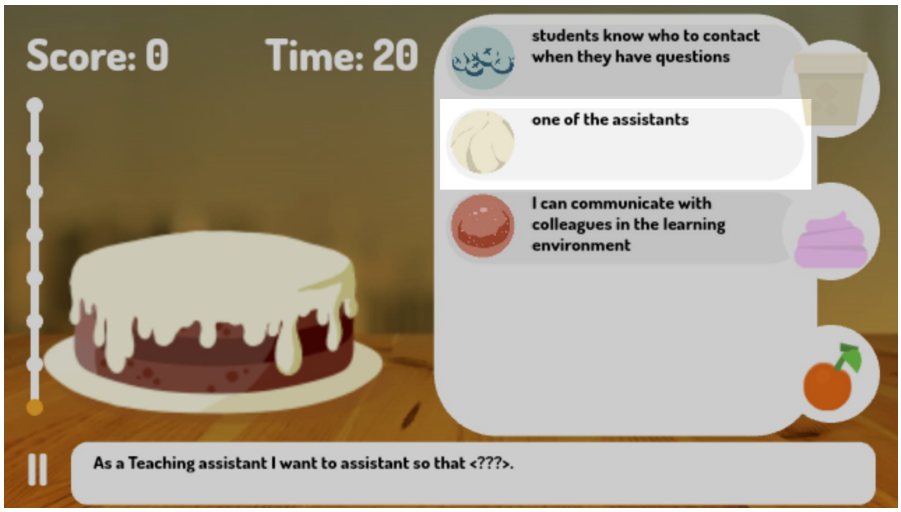

Figure 3.8: An example of a "faulty" (non-full sentence) user story element

BATCH	EPIC KEY	US SHOWN	MISTAKES	CRITERIA
1	6	3	1	1 (full sentence)
2	1	3	1	1 (minimal)
3	4	3	1	1 (atomic)
4	2	4	2	2 (full sentence)
5	1	4	2	2 (atomic/minimal)
6	6	5	3	3 (at./min./f. sent.)
7	4	5	3	3 (at./min./f. sent.)

Table 3.4: Level 2 - QUS - 7 batches, 14 correct US

The player is shown an epic as the context, as well as given the role/action/benefits as menu options (Roles = Cake bases, Actions = Icing and Benefits = Decorations).

To score points, the player has to create suitable user stories for the shown epic. Negative points are only given for handing in wrong user stories. The make-up of the level can be seen in Table 3.3. In this level, the epic numbers do not point to specific epics, but epics can be randomized in any order to get a subset of 5 epics and their matching user stories.

3.3.3 *Level 2 - QUS*

In this level, the player is educated on existing quality frameworks for user stories, specifically the Quality User Story framework or QUS.

- **Learning goals:**

  - Remember: that different quality criteria exist for userstories, specifically the quality user story (QUS) frame-work and the criteria it entails.
  - Understand: that high-quality user stories fulfill all these criteria and how to meet these.
  - Evaluate: which user stories in a set are correct and which ones contain defects.
  - Apply: the knowledge from the two points above by combining building blocks into correct user stories given a context.

- **Expected playtime:** 7:00 – 11:00 min.
- **Introduction:** "Welcome back! From this level forward, we introduce faulty user stories into the mix. These user stories can be structurally correct, but do not comply with some of the QUS quality criteria. To complete the level, associate all correct user stories and leave out the faulty ones. The QUS categories to focus on are:
  - Full Sentence
  - Atomic
  - Minimal

  For a refresher on the categories, open the pause menu and click the corresponding category name. Good luck!"

The first of these two QUS levels focuses on whether user stories are:

- Full Sentence (are all user stories well-formed (english) sentences?)
- Atomic (do user stories only tackle one problem at a time?)
- Minimal (do user stories contain nothing more than a role, action and optional benefit?)

The player creates user stories in the usual manner, using the building blocks, the menu (still with Role = Cakebases, Actions = Icing and Benefits = Decorations) and the order, that specifically fit one or more of the criteria. The player is given specific "wrong" options, not because they do not fit the context/epic, but because they do not conform to the criteria – see Figure 3.8 for an example of this. The amount of criteria to keep track of at once increases as the level progresses. See Table 3.4 for an overview of the content in this level.

BATCH	EPIC KEY	US SHOWN	MISTAKES	CRITERIA
1	7	3	1	1 (independent)
2	3	3	1	1 (unambiguous)
3	5	3	1	1 (problem oriented)
4	3	4	2	2 (unam./ind.)
5	2	4	2	2 (problem oriented)
6	7	4	2	3 (prob. or./unam./ind.)
7	5	4	2	3 (prob. or./unam./ind.)

Table 3.5: Level 3 - QUS (continued) - 7 batches, 14 correct user stories

3.3.4 *Level 3 - QUS (continued)*

In this level, the player continues to be educated on existing quality frameworks and QUS.

- **Learning goals:**
  - Remember: that different quality criteria exist for userstories, specifically the quality user story (QUS) frame-work and the criteria it entails.
  - Understand: that high-quality user stories fulfill all these criteria and how to meet these.
  - Evaluate: which user stories in a set are correct and which ones contain defects.
  - Apply: the knowledge from the two points above by combining building blocks into correct user stories given a context.
- **Expected playtime:** 7:00 – 11:00 min.
- **Introduction:** "Welcome back!" To win this level, complete all user stories across the batches without associating the faulty ones. The QUS categories to focus on are:
  - Independent
  - Unambiguous
  - Problem Oriented

  You can always review the context by opening the pause menu, which also shows a refresher of those categories. Good luck!"

The second of these two QUS levels focuses on whether user stories are:

- Unambiguous (do user stories have one meaning – and no more than one meaning?)

US key	AT shown
1	1
2	2
3	2
4	3
5	3

Table 3.6: Level 4 - Acceptance Tests - 6 batches, 14 correct acceptance tests

- Independent (do user stories work on their own and do they not point to other user stories in the set?)
- Problem Oriented (do user stories only specify the problem, not the solution to it?)

The gameplay within this section works the same as it does in Level 2. See Table 3.5 for an overview of the content in this level. The final batch can be either batch 6 or batch 7, this is randomly assigned at the start of the level.

3.3.5 *Level 4 - Acceptance Tests*

In this level, the player is educated on acceptance tests, their purpose and their structure. The following structure is used for the acceptance tests: "*Given* some context, *When* some action is carried out, *Then* a set of observable consequences occurs."

- **Learning goals:**
  - Remember: what acceptance criteria are and why they are used.
  - Understand: the relationship between acceptance criteria and user stories.
  - Apply: the knowledge from the two items above to build acceptance criteria by combining existing blocks.
- **Expected playtime:** 3:00 – 7:00
- **Introduction:** "Welcome back! In this final level, we look at the back of a user story card: acceptance tests. We follow Dan North's syntax:

  'Given <some context>, When <some action is carried out>, Then <some observable consequence occurs>'

  "Create correct acceptance tests given the context (a user story, in this case). Just like before, complete all batches to complete

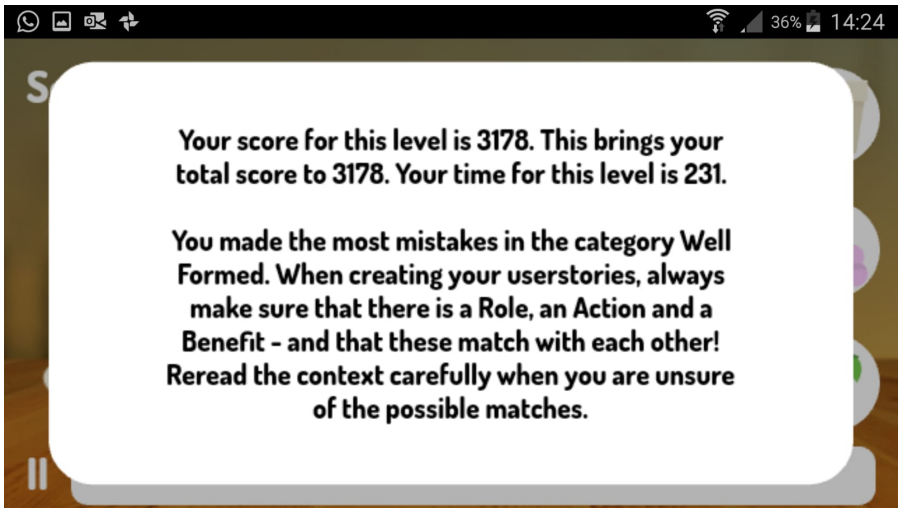

Figure 3.9: A screenshot of a debriefing session

the level. There are no mistakes in this level: every acceptance test has to be completed. Good luck!"

In this level, the "Order" or context part will contain the user stories, one at a time, while the menu options contain the different parts of an acceptance test (Context = Cakebases, Action = Icing and Consequences = Decorations).

Scoring works virtually the same as it did in Level 1; quick answers give more points than slow answers, however wrong answers deduct points. Level make-up is once again shown in Table 3.6.

3.4 DEBRIEFING

After finishing every level, the player is confronted with a debriefing of their performances, to allow for some time to reflect. The debriefing shows the following information:

- The score achieved in this level
- Total score within the game up until this point
- The time it took the player to complete the level
- Their most frequent type of mistake and some additional information/tips for the future (for example; when a player made a lot of "Full Sentence" mistakes, a possible tip could be "Always read the sentence while making it and check if it makes grammatical sense. Does this sound like a natural English sentence?"

  When no mistakes were made, the player is shown the congratulatory text "You have not made a single mistake in this level! Congratulations!".

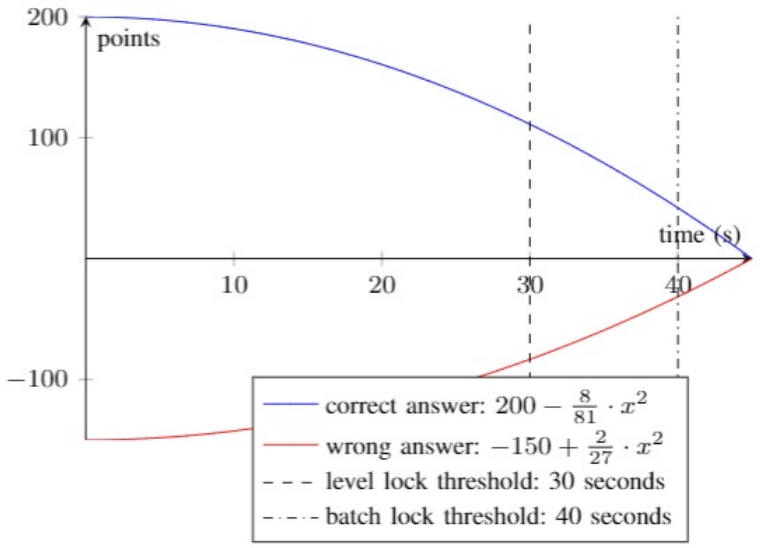

Figure 3.10: Scoring system BakeRE

3.5 SCORING

Points are scored as defined in Figure 3.10. The scoring of each user story depends on the time it takes for the player to associate the elements of that user story (in seconds, see x-axis). The faster a user story is completed correctly, the more points are assigned (from 200 points for nearly instantaneous correct answers to 0 for correct answers in more than 45 seconds).

For incorrect answers, the scoring system is inverted: many negative points for a quick incorrect answer (to avoid guessing), and smaller penalties for incorrect answers that take longer. The scoring scheme employs an exponential function where the number of points scored decreases only slightly in the first few seconds, but the reward becomes much smaller as we approach the 45 seconds. A converse scheme is used for wrong answers, with a lower upper bound.

The constants in the functions were decided on by testing. It is the value that is found when substituting for x = the time it would take to finish a batch when taking the slowest measured time * 1.5, divided by the number of user stories in a batch. So zero points would be awarded for **x = 150 * 1.5 / 5 = 45 seconds**, therefore **200 / 45^2 = 8 / 81**. Negative points are given with the same parabole as the formula, but start at -150 points instead of 200. This makes the constant **150 / 45^2 = 2 / 27.** Even when awarded negative points, the point total will never drop below 0 points to avoid unnecessary frustration.

Players are expected to earn a certain number of points within each level. This to prevent a "brute force"-tactic of just trying every other option in a batch until the batch is finished. See Figure 3.11 and

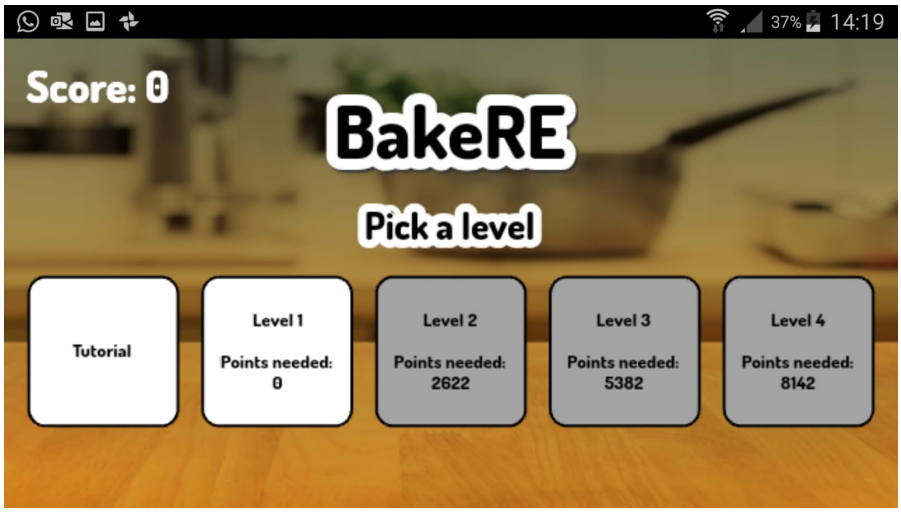

Figure 3.11: The phases of the game where the amount of points you have could lock/open new areas of the game

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Score	0	2109	3663	5217
Penalty	0	332	249	249
Level-lock	0	1777	3082	4387

Table 3.7: Level-locking scores per level

Table 3.7 for an example within the game and an overview of the level-locking scores respectively. The score-locks are implemented in two different manners; within a level itself and between levels.

- Batch lock: to complete a batch, a certain number of points must be scored. Roughly, we require a player to take less than 40 seconds per story on average. If an insufficient number of points is scored, an additional batch of user stories is given to the learner, who has to associate one of those user stories correctly in one try, shown as the "flawless user story" in Figure 3.12.
- Level lock: there exists a level lock for total score up until this point. When the player has not yet gathered enough points within the levels up to this moment, the next level will remain locked and the player has to complete another level again to gain more points. The points that were already earned in that level before retrying will be deducted, to minimize the chances of the players only repeating the first level over and over again for a highscore. The level-locks are based on an average time of 30 seconds per user story minus the penalty of making a mistake in 25% of the user stories – see Table 3.7.

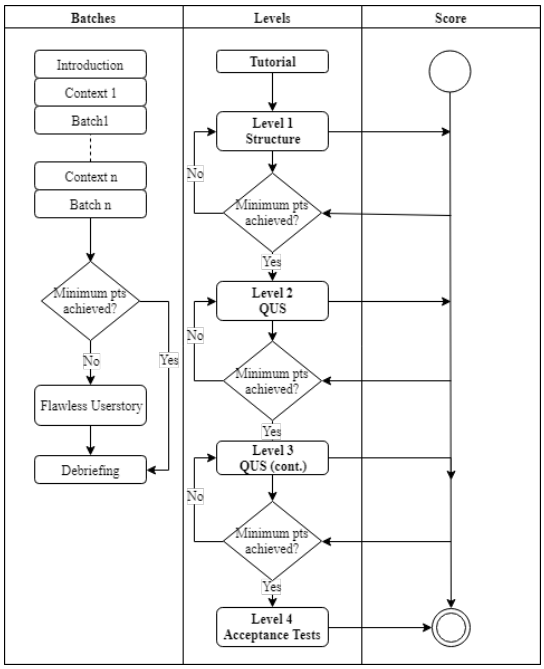

Figure 3.12: The level-locking systems. The Batch lock is shown to the left of the diagram, and the Level lock is positioned in the middle. While playing the game, the player goes through all phases in the left diagram per level in the middle diagram, sending the score at the moments shown in the right diagram.

3.6 VISUAL AND AUDIO DESIGN

As can be seen in Figure 3.2 – 3.5, the visual design gives the player the idea that they are creating cakes, as described in the introduction. The colours are based on publicly available bakery- and pastry themed colour palettes to emphasize the theme and atmosphere as seen in Figure 3.13 – 3.16.

When creating the cakes, animations are played to give feedback and create a feeling of satisfaction when dropping an item. The animations are accompanied by an audio cue, both to improve the feel of the game and to generate feedback. Finally, when correctly finishing a user story, a celebratory audio cue is given as well as a colourful particle explosion (as seen in Figure 3.2). When finishing a batch or a level, a set of fireworks is shown accompanied by another, longer celebratory sound. When a user story is made incorrectly, a buzzing sound is played and the parts of the cake that are on the screen return to the menu with little other fanfare.

3.7 PROGRAMMING TOOLS

When choosing the programming tools best suited for creating the game in question, first a list of desired functionalities and other pre-

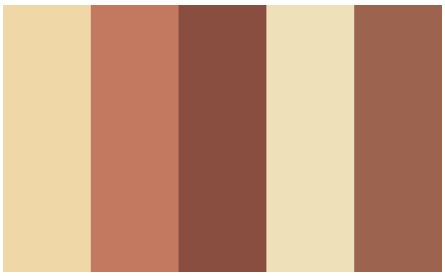

Figure 3.13: Cake colour palette

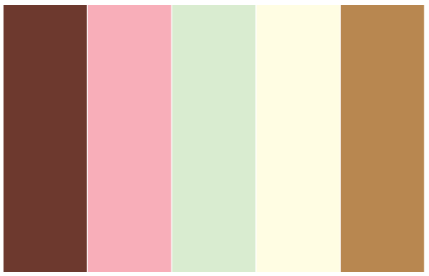

Figure 3.14: Icing colour palette

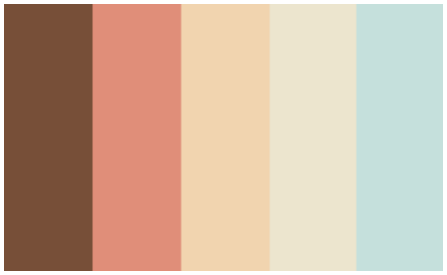

Figure 3.15: Decorations colour p.

Figure 3.16: Background colour p.

requisites was made. Due to the technical simplicity of the concept, this list can be summed up as follows:

- **The game should be on smartphone.** Almost everybody has a phone and most people carry this phone around wherever they go. Therefore a mobile game would have a bigger reach compared to regular computer-based games, which of course requires either a PC or a laptop to be available.
- **The game should be able to support drag-and-drop functionality.** This is because it is the main mechanic of the chosen concept, therefore the game would be significantly divergent from the chosen GDD to alter this mechanic, or altering this mechanic would change the entire game.
- **The game should be cross-platform.** This so that the game can have an audience that is as large as possible.
- **The game should be a mobile app instead of a web-based game.** This mainly follows from the former three constraints, as webpages on mobile have difficulty with accepting drag-and-drop functionality, as many build-in phone functionalities (such as zooming or scrolling) can not easily be overwritten for a specific webpage.
- **The game should have Internet access and an online database to create and update high scores.** This is rather self explanatory.

Using these as the basis, an in-depth Internet search was performed to find as many highly regarded tools as possible. After taking out the

tools that do not support some (or all) of the constraints, the list looks like it does in Table 3.8.

The two best performing tools were "Corona SDK + Lua" and "Visual Studio + Apache Cordova + Phaser (JavaScript)". In the end, the choice between these two came down to experience, as there was no prior experience with Lua and many years of experience with JavaScript, the second option was chosen.

In the following sections, a more in-depth description of the chosen technologies will be given, starting with the IDE Visual Studio and followed by the frameworks Apache Cordova and Phaser.

3.7.1 *Visual Studio (2017)*

Visual Studio is an IDE produced by Microsoft. The first version of VS as we know it was released in april 1995 (as versions earlier than this one were more loose components and not the package deal that VS has become). The version used during the development of this game is **VS 2017**, released in march 2017, as it is (as of the time of writing) the most recent version of the program.

VS contains many different programming tools, not only standard code editing and integrated debugging, but also IntelliSense (automatic code completion) as well as the ability to send code directly to (connected) smartphones, without the need to install applications on the phones for every single debug.

Another useful functionality is the ease with which additional libraries and toolsets can be loaded into VS. Using the Node.js package manager "npm", which can be downloaded and installed on installing VS itself or added later on, packages and tools (or other reusable code), that were put on npm by other developers, can be easily added to the program without manual rewriting of code.

3.7.2 *Apache Cordova*

Cordova is an (open source) mobile development framework created by Apache. A Cordova application acts like a mobile app, but behaves like a website behind the scenes. Therefore, many of the advantages of creating a website, such as the usage of npm, JavaScript

1 https://coronalabs.com/product/
2 https://visualstudio.microsoft.com/vs/features/mobile-app-development/
3 https://visualstudio.microsoft.com/xamarin/
4 https://visualstudio.microsoft.com/vs/features/cordova/
5 https://phaser.io/download
6 https://www.buildbox.com/
7 https://www.eclipse.org/ide/
8 https://unity3d.com/
9 https://cordova.apache.org/docs/en/latest/guide/overview/

Table 3.8: Comparison of programming tools

IDE	LANGUAGE	PROS	CONS
Corona SDK [1]	Lua	Many API's/Plugins available, Real-time simulation (updates all versions of apps on network)	No prior experience with Lua
Visual Studio (2017)[2]	.NET/Xamarin [3] + C#	Native UI and API access, Reusable codebase across platforms (cross-platform), Cloud services, Emulators for testing available	Convoluted, Difficult to step into without experience, No prior experience with Xamarin
Visual Studio (2017)[2] + Apache Cordova [4]	JavaScript + HTML + Phaser [5]	Creates Webapp (cross-platform), JavaScript frameworks can be added, Emulators for testing available, Native UI and API access	No prior experience with Cordova and Phaser
Visual Studio (2017)[2]	C#	Many libraries available	No baked-in functionality without libraries, so build from scrap
Buildbox[6]	Buildbox editor	Simple editor, Fast workflow	Limited/restrictive set of features
Eclipse [7]	Java	Works with many different languages, Plugins available	Works best with Java which is not ideal for this project, Difficult to step into without prior experience
Unity [8]	C#	Prior experience, Many useful features, It does a lot of the work for you	Limited/restrictive set of features, Not great for 2D, No easy cross-platform

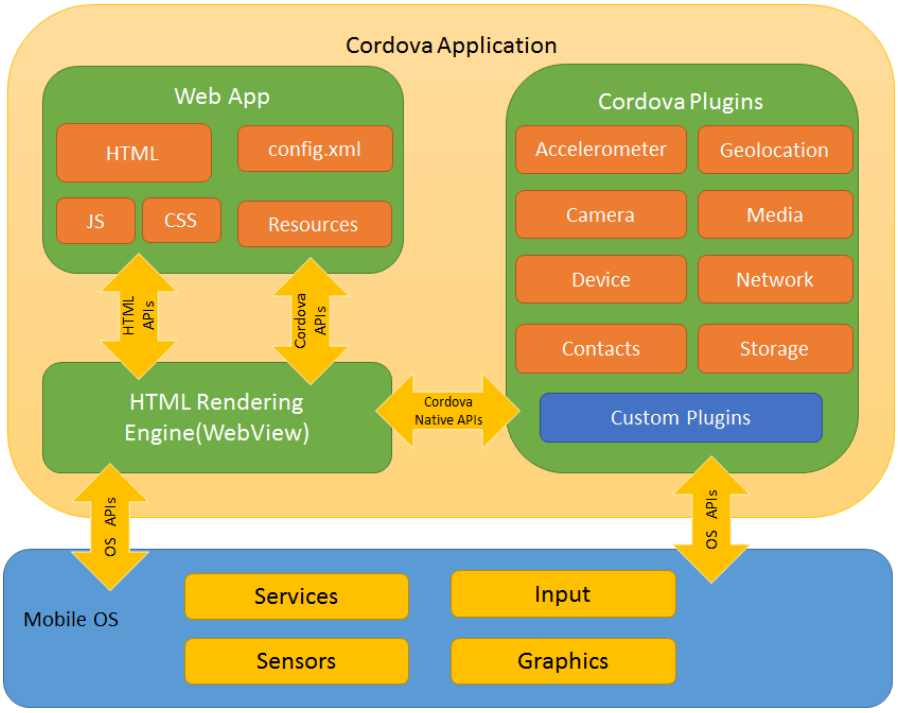

Figure 3.17: The architecture of an Apache Cordova application[9]

and its accompanying libraries, combined with the advantages of a "regular" mobile application – such as native mobile functions and cross-platform functionality. Cordova then uses an API to render the platform appropriate version of the website as an application on the mobile it is currently rendered on. An overview of the architecture of a Cordova application can be seen in Figure 3.17.

3.7.3 *Phaser*

Phaser is a free open-source platform for the creation of web-based, canvas-based games. It is made to do much of the game management for you, such as game object creation and management, animation and updating/terminating game loops. As of now many community made tutorials exist and as it is HTML/CSS/JavaScript based, it can very easily be combined with Apache Cordova.

4

RESEARCH METHOD

4.1 SETUP

The setup of this research was formed around the "Engineering Cycle" of Wieringa's Design Science [Wieringa, 2014]. This cycle consists of the following steps:

1. Problem investigation: What phenomena must be improved? Why?
2. Treatment design: Design one or more artifacts that could treat the problem.
3. Treatment validation: Would these designs treat the problem?
4. Treatment implementation: Treat the problem with one of the designed artifacts.
5. Implementation evaluation: How successful has the treatment been?

Here, an artifact could be anything related and suitable to the research at hand, such as a framework, a tutorial or a piece of software. In the case of this thesis, the artifact consists of the serious game "BakeRE".

As can be seen in Figure 4.1, this thesis follows the setup of the Engineering cycle, but contains three subcycles in the "Treatment Design" and "Treatment Validation" phases. This is due to the nature

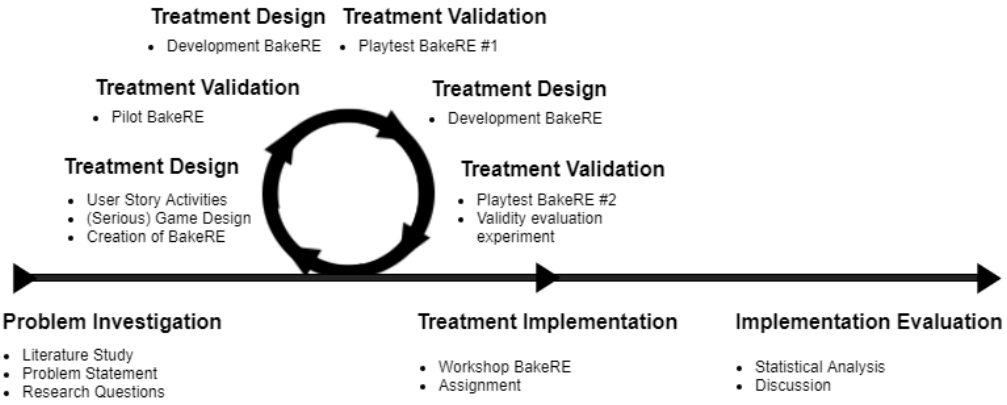

Figure 4.1: The Wieringa Engineering Cycle as used in this thesis

of the artifact; since both games and educational material need a lot of back-and-forth between evaluation and development to assure that a treatment has been created that is both usable and addresses the correct learning goals respectively.

In the "Problem Investigation" phase, the preparatory work was done. This includes gathering information on the field as part of the literature study, creating and refining a problem statement and generating the appropriate research questions. The findings of this step are detailed in Chapter 1 & 2.

Next, the "Treatment Design" and "Treatment Validation" phases are iterated several times as a sub-cycle, where first the initial Game Design was created, which was then evaluated with a pilot session. The results of this pilot session were then taken to the next phase of development etc. This setup was repeated several times, with the game being at increasing levels of maturity. The game was tested and discussed between the author of this thesis and the supervisor on a weekly basis. Additionally, two test sessions were held over a period of two months with three and nine participants respectively. The conceptualization and creation of the game is discussed in further detail in Chapter 3.

The next phase of "Treatment Implementation" contains the experiment itself in the form of the 60 minute workshop, as well as the assignment the students hand in during their course in RE. This implementation was then evaluated during the "Implementation Evaluation" phase in the form of the statistical analysis, conclusion and discussion explained in Chapter 5 & 6. Possible further iterations of this cycle fall out of the scope of this thesis; they will be discussed in the "Future Work" section in Chapter 6.

4.2 TIMELINE

An overview of the timeline of this thesis can be found in Figure 4.2. The Engineering cycle of Figure 4.1 is implemented within this timeline in the form of the coloured bars at the bottom of the diagram. Additionally, the major milestones are shown as diamonds in between the bars.

4.3 RESEARCH QUESTIONS

The research questions will be answered by performing an experiment on students learning Requirements Engineering for the first time. The participants will be given the serious game "BakeRE" during a special workshop as part of their course on RE. The accompanying lecture on user stories will be passive and not contain the

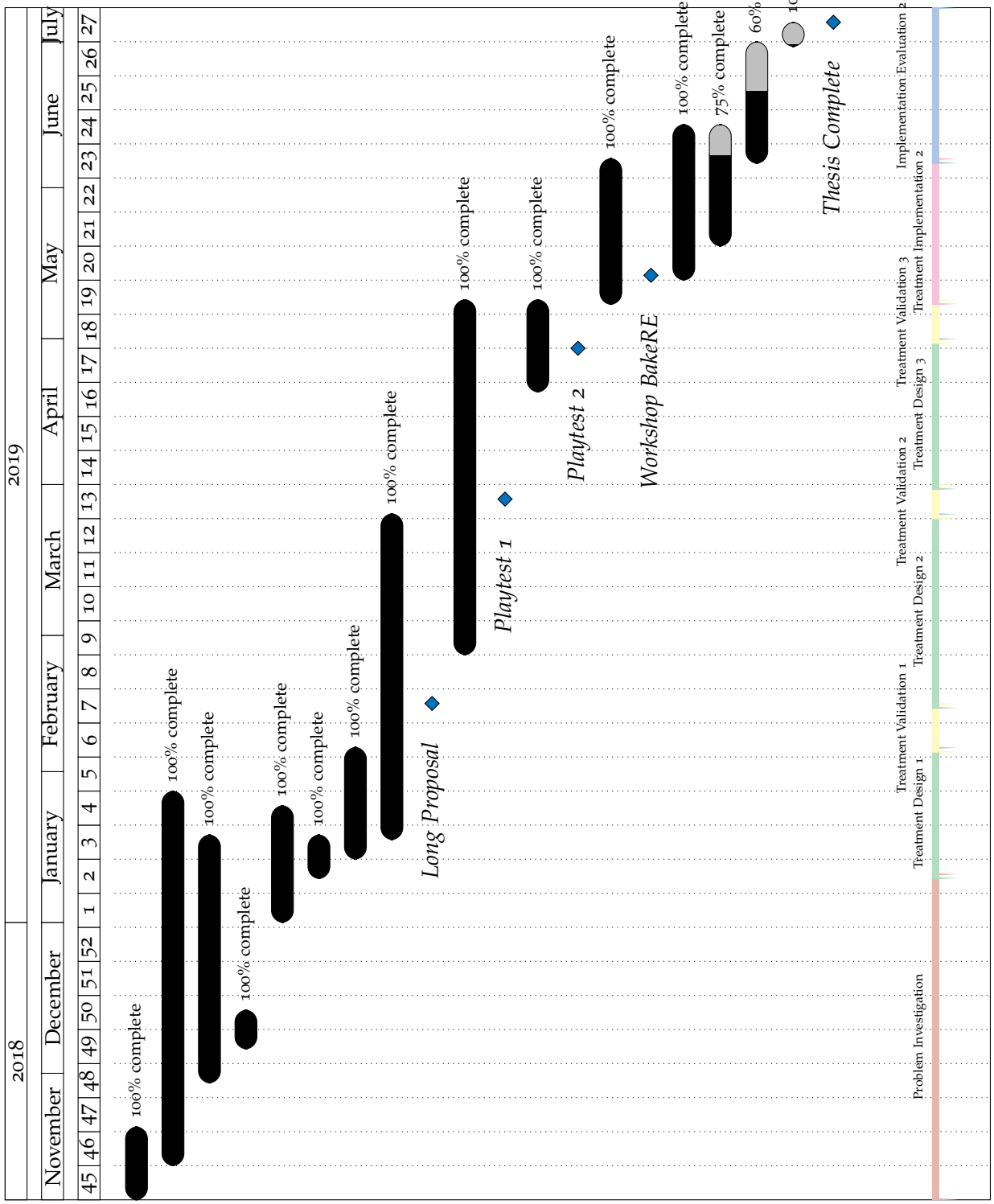

Figure 4.2: Timeline of the project

RESEARCH QUESTION	OBJECT OF INTEREST	TEST METHOD
Subquestion 1	Knowledge Gain	Pre- and Post-test (Workshop)
Subquestion 2	Knowledge Retention	Assignment on User Stories
Subquestion 3	Learning Experience	UES-SF [O'Brien et al., 2018]

Table 4.1: A summary of the research (sub)questions

tutorial elements it normally does, as these tutorial elements will be provided during the workshop in the form of the game. For the sake of comparison, students are tested on their knowledge on user stories before and after this workshop. There is no group that does not play the game; this is because it would be unfair from an educational perspective to give some students a possible advantage in case the game does/does not improve results.

Ideally, a "long-term" check would be performed repeatedly over a longer period of time to assure that knowledge retention remains the same (or at least an improvement) even when not actively engaged with user stories or requirements engineering. However, the time limit for this thesis does not allow for such an extended period of examination, but an assignment on user stories is part of the RE curriculum. Therefore these results will also be examined to consider knowledge retention. These constraints have mainly been born out of necessity, as the research subjects will be fellow students taking the Requirements Engineering course, here on the Utrecht University.

Finally, there is also a question regarding the "learning experience", which for the purpose of this thesis is considered as a combination of motivation and affect. In other words, whether or not learners enjoy playing the game and would like to continue playing. This subquestion is to test whether or not the game heightens learner motivation and/or learner entertainment. For a short summary of the research subquestions and their methods of answering, see Table 4.1.

4.3.1 *Knowledge Gain*

To see whether there is an educational impact to using this serious game during the learning of user stories, the first thing to look at is the potential of knowledge gain. Therefore, the following subquestion was created:

Subquestion 1

> Does the usage of a Serious Game during the learning of User Stories improve the Knowledge Gain regarding User Story Quality?

This subquestion is answered by taking two tests during the workshop; one before playing BakeRE and one after working with it for 40 minutes. The results of the pre- and post-test are compared to each other to see whether or not the scores improve after playing the game. It is assumed that people can not "unlearn" knowledge on user stories in that time, therefore we sort participants on having made progress and participants who did not. This data is also compared with other information gathered during the session, such as whether or not subjects attended the lecture, or the score they achieved within the game. The results can be found in Chapter 5.

4.3.2 *Knowledge Retention*

When considering the educational merit of the game, it is also important to look at knowledge retention – or the knowledge and skills that stay with the learner over a longer period of time. This leads to the following subquestion:

> Does the usage of a Serious Game during the learning of User Stories improve the Knowledge Retention regarding User Story Quality? *Subquestion 2*

As mentioned before, ideally this would be tested several times over a longer period of time, but the timeframe of this thesis does not allow for that. However, the students do work on an assignment on user stories in the form of a database for BakeRE in the two weeks after the workshop. Information on the quality of the databases handed in can be used to examine which parts of the (quality) user story creation process students have mastered and which parts they still struggle with.

As students work on this assignment in teams over a longer period of time, not all of the progress in the area of user story quality can be attested to BakeRE, but it does give some interesting insight into the knowledge gain over a longer period of time – the knowledge retention.

4.3.3 *Learning Experience*

Last but not least is the issue of learning experience. For the purpose of this study, learning engagement is divided into:

- Motivation; is the player (intrinsically or externally) motivated to (continue to) play the game?
- **Affect;** does the player like the game and does he/she enjoy playing it?

Ideally, the game should be entertaining as well as educational to increase the learning experience. If the game is motivating, this can increase the students emotional involvement with the content and increase knowledge gain (and retention) [Millbower, 2003]. This leads to the following subquestion:

Subquestion 3

> Does the usage of a Serious Game during the learning of User Stories improve the Learning (User) Experience?

This subquestion is answered by using the User Engagement Scale (short form) [O'Brien et al., 2018]. The UES-SF is a 12 question long survey measuring user engagement divided over 4 subcategories:

- Focused Attention (FA)
- Perceived Usability (PU)
- Aesthetics (AE)
- Reward Factor (RW)

Every subcategory contains three questions on a Likert scale of 1–5. Grading the UES-SF means summing the scores of each of the 12 questions (or 5 minus the score in the case of the "negative" category PU) and dividing it by 12 to create the average user engagement. This survey is given as a part of the post-test and is described in further detail in Section 4.4.

4.4 EXPERIMENTAL DESIGN

The effectiveness of BakeRE as an educational tool is tested by performing an experiment. During this experiment, a group of students enrolled in the Requirements Engineering course at the Utrecht University participate in a 2-hour workshop after having been a part of the introductory lecture. This workshop consists of a 10 minute pre-test to test the initial or baseline knowledge after the introductory lecture, a 40 minute play session with BakeRE and then another 10 minute post-test to see if playing the game has lead to quality improvements.

Afterwards, the students work on an assignment on user stories for the course. This assignment consists of creating a database for BakeRE (see Section 3.2.2 for more information on the contents of a database). These databases were then checked for mistakes and compilation errors, after which some of the datasets can be used to create new levels for BakeRE. An overview of the timeline of the experiment can be seen in Figure 4.3.

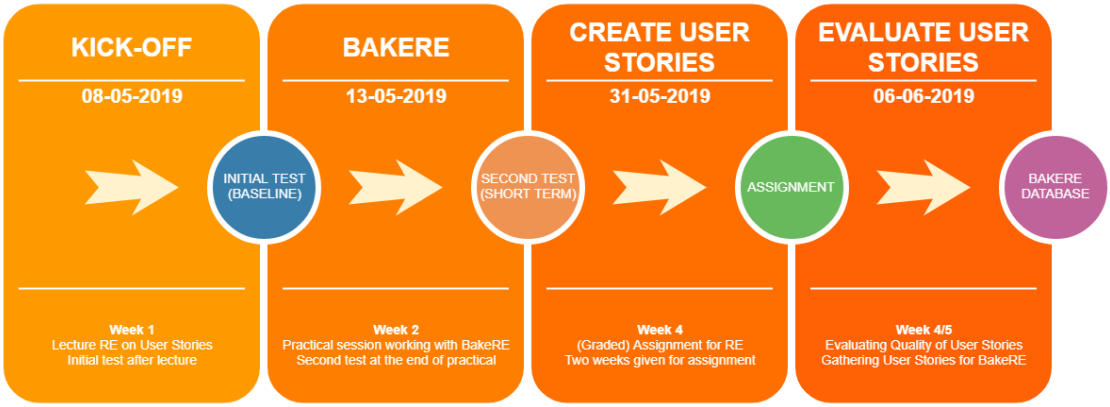

Figure 4.3: Overview experimental design

4.4.1 *Testing*

The pre-test consists of four demographic questions:

- Gender; male, female or other. This variable was gathered to compare whether things like theming were looked upon more favourably by one gender than by the other.
- Experience; these questions were asked to get a good overview of the knowledge baseline of the participants.
  - Attended lecture; whether or not a participant has attended the lecture on user stories on the 8th of May.
  - Reviewed material; whether or not a participant has reviewed the additional material on user stories given as a part of the course on RE before the lecture.
  - Prior experience user stories; the depth of the (possible) prior experience with user stories a participant has. This can be either:
    - No experience
    - Used it as part of a course more than 6 months ago
    - Used it as part of a course less than 6 months ago
    - Used it as part of a project
    - Used it as part of a course and in projects

Following this, the participants are given ten tasks where (potential) defects in existing user stories should be pointed out. The tasks are performed in a multiple choice fashion, where every user story has four potential defects associated with it and the participants can choose 0–4 of these as their answer. These tasks were based on a similar exercise performed during exams for the RE course at Utrecht University in previous years to test students on user story quality knowledge.

The post-test consists of a similar task with a similar set of ten user stories, as well as the User Engagement Scale (short-form) [O'Brien et al., 2018]. The UES is a tool developed to measure user engagement. For this experiment, the UES-SF or short-form is used to minimize the number of questions participants have to answer and keep the questionnaires of reasonable length.

The pre- and post-test user stories and their corresponding answers, as well as the 10 UES-SF questions, can be found in Appendix E. As recommended by the authors of the UES-SF, the UES-SF questions were shown to the participants in a random order, as opposed to the categorically grouped list shown in Appendix E. This too minimize the threat of subjects catching on to the different categories within the UES-SF and answering the questions in such a way that a specific category ends up with very high or very low score, even though some of the separate subcategories might have been scored higher by themselves. The randomization happened before the test was given to the students, and the order was the same for all students.

4.4.2 *Validity Threats*

- **Student motivation:**
  - ***Mono-method bias/No control group:*** there was no control group of students that did not participate in the workshop at all. This was mainly necessary for educational reasons – it would be unfair for some of the students to have more/better information on this subject than those that did not attend the workshop, regardless of which of the treatments would have been "better" than the other.
  - ***Mandatory attendance BakeRE:*** all students partaking in the course on RE at the Utrecht University were supposed to attend the workshop on BakeRE. As the course included an assignment on user stories in the form of a database on BakeRE, it would be difficult for students that did not experience the game to be as well prepared for this assignment as those that did not.
  - ***Timing of the experiment:*** the workshop using BakeRE was planned on a Monday from 15:15 to 17:00, right after a two hour lecture. Therefore, students had probably already used quite a lot of their concentration.

  All of these threats are the result of decisions that had to be made (educational value versus experimental value and the planned timeslots of the course) and could therefore not be mitigated further, but this might have lead to students whose motivation was lower than if it had been an optional/voluntary workshop.

- **Knowledge gain/knowledge retention:**
  - ***Baseline/treatment time difference:*** the time difference between the baseline data and the data after treatment is only 40 minutes, and the baseline already included the knowledge gained during the lecture. This might not be enough time to have truly improved user story quality knowledge. To properly test this notion, future versions of this experiment might want to compare with a different baseline, for example by testing before the lecture as well.
  - ***No long-term test:*** to properly test for knowledge retention, ideally there would be multiple tests over a longer time into participants user story skills. However, due to the thesis time constraint of this project, it does not allow for these extra long term tests. Instead of these tests, the knowledge retention is examined by looking at the assignments filled in three weeks later. Additionally, because students work on this project over a longer period of time and in teams, it can no longer be said that the knowledge retention that was observed was truly the result of this workshop. Future works based on this study should examine the true knowledge retention using a method utilizing multiple tests, as described above, to truly mitigate this threat.

- **Hypothesis guessing:** due to the experiment being performed in a classroom setting, it might have been that students tried to "guess" the purpose of the study, especially while filling in the UES ("does this game motivate me more than a normal workshop would?" etc). This threat was mitigated as much as possible by randomizing the order of the UES, so that categories were split up and it became harder to "steer" the answers.

- **Focus on QUS quality criteria:** during the pre- and post-test, the questions focus on the participants' knowledge on QUS. However, the game does more than only teach about quality criteria; it also helps students focus on user story structure and familiarize themselves with the way good user stories flow. This progress is not easily tested though, and therefore is not accounted for in the results.

- **Instrumentation:** due to unforeseen circumstances regarding the technology, not all devices were able to play BakeRE. For instance, devices running iOS were unable to access the game on the appstore and some Huawei devices crashed during the experiment. Therefore, the group had to be split up in two sessions and not all players were able to use the phone they were accustomed to.

- ***Two sessions:*** due to the lack of back up phones, not all participants could play in the workshop at the same time. Therefore, the group was split in two sessions, as this was deemed better/more fair than letting the few that did not have a phone play later than those that did. However, this most likely influenced motivation, especially in the second group that had to wait for the first group to finish before they could start the workshop.
- ***Borrowed devices:*** due to the lack of additional devices, some participants were unable to play BakeRE on their own phone. This might have influenced their performance with BakeRE, as different screen-sizes, -shapes and -sensitivity take some getting used to.

Both of these threats were also introduced out of necessity and could not be mitigated further, apart from noting that they existed and memorizing this for future works.

4.4.3 *Performing the experiment*

The initial lecture on user stories was held on the 8th of May (2019) from 09:00 until 10:45. The experiment itself was carried out on the 13th of May (2019), from 15:15 to 17:00. 58 students participated in the experiment, out of which 39 had also attended the lecture. Every participant was given a unique ID, so that their performance in both the tests and the game could be coupled together while still assuring their anonymity.

Due to unforeseen issues with the Apple appstore, BakeRE was not playable on iOS devices during this workshop, and not enough additional Android devices could be found to account for all students with an iOS device. Therefore the group had to be split in two sessions, which will be called "Session 1" and "Session 2" from now on.

Both groups started with the 10 minute pre-test at the same time, but upon its completion the participants of Session 2 left the room while Session 1 started playing BakeRE. After 40 minutes the first group was told to stop playing and start filling in the post-test. During this time the participants of Session 2 came back into the room and started playing the game as well. When they too had played BakeRE for 40 minutes, they were told to stop playing as well and filled in the post-test.

Several issues arose while conducting the experiment. Firstly, there was confusion about the goal of the 10 user story quality assessment tasks. Some of the students thought that they were supposed to point out what mistakes *were not* made in the faulty user stories, as opposed

to which mistakes *were* made as was the intention. When this came to light, many students had already sent their questionnaire and could no longer change their answers. To mitigate this threat, a paper was passed around the room were all students that made this mistake could fill in their participant ID so that these results could be found when correcting them. These results were then "flipped", as we can assume that the mistakes that *were* made and the mistakes that *were not* made are mutually exclusive.

Secondly, four of the devices crashed during the workshop due to unknown reasons. It occurred at the same point in the game for all four participants (at the end of level two upon finishing the final user story) but at different points in time. The common denominator between these four cases is that all four have Huawei devices, which have had several (mostly visual) issues during both of the test sessions before. As the game could only been restarted by completely rebooting the app, all of their data up to that point was lost. These four datasets were therefore discarded from the results while performing the statistical analysis.

Finally, there were two participants that participated in both sessions. It is unsure how or why this happened, but their results have been disregarded from the total number of results.

The rest of the experiment went on as planned. During the playing of the game, the subjects score per level and the time per batch were logged, for which the participants gave permission at the start of the workshop. The results of this experiment are attached to this thesis in Appendix F and the accompanying analysis is described in Chapter 5.

Part III

RESULTS

5 RESULTS

This chapter describes the data that was gathered during the experiment performed as a part of the Requirements Engineering course 2018/2019 given at Utrecht University. After a description of the population, descriptive analytics are presented. Then, a section on correlations and other relationships is introduced, followed by an analysis on the impact of prior experience. The complete dataset can be found in Appendix F.

5.1 DESCRIPTIVE ANALYTICS

As described in Section 4.4.3, the experiment was conducted on Monday the 13th of May (2019), from 15:15 to 17:00. **58 participants** attended the workshop.

Out of these 58 participants, six datasets were discarded for further analysis:

- **4 students** experienced a game crash during the execution of the experiment; thereby invalidating their data.
- **2 students** participated in both sessions, therefore creating half a dataset in each session and taking longer than the other students, which invalidates their data as well.

This means that the population for our intents and purposes consists of the **52 participants** who have generated a full dataset. Out of these 52 participants, **37** were male and **15** were female. Initially, 28 people attended Session 1 and 30 attended Session 2. After discarding the faulty data, the split became **27 to 25**. This can be seen in Table 5.1.

	SESSION 1	SESSION 2	TOTAL
Male	21	16	37
Female	6	9	15
Total	27	25	52

Table 5.1: Participant distribution – gender

According to the answers given by the participants in the demographic part of the pre-test questionnaire, the prior experience with user stories differed within the group (see Figure 5.1). Out of the 52

participants, **13** said they had no experience with user stories whatsoever and another **11** said that the last time they had used user stories (as part of a course) was more than six months ago.

Nine students said that they had used user stories more often, both in a course and in practice, **eleven** had used these user stories in a project before and **eight** had used them in a recent course (less than six months ago).

It is assumed that those that have worked with user stories within projects and those that have used them in a recent course have more experience than those that have last used them more than six months ago. Therefore, we split these five categories in two groups based on the amount of experience these students have; the "Experience" and the "No experience" group. This could give the 28 students that have more experience an advantage when playing BakeRE and/or when performing the pre- and post-test tasks; this is further expanded upon in Section 5.3.

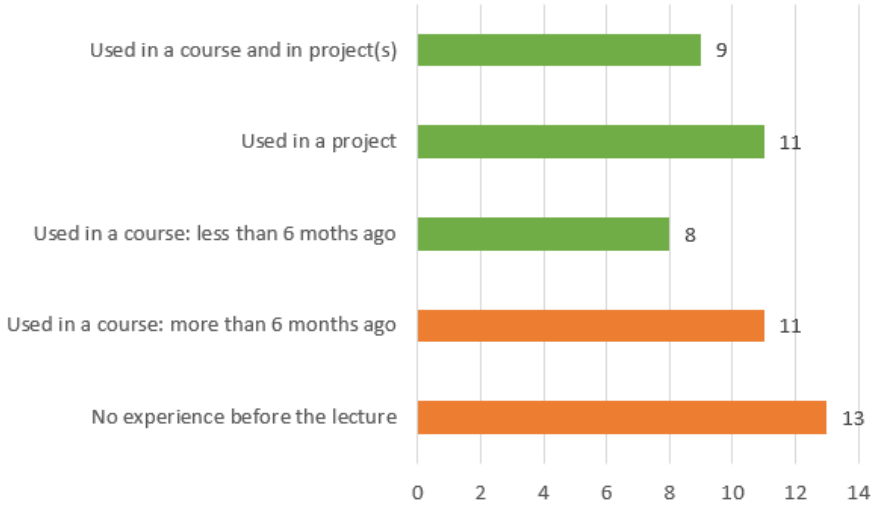

Figure 5.1: Prior experience with user stories within the testgroup

As shown in Table 5.2, **36 participants** attended the lecture on user stories on Thursday the 10th of May. Out of these 36, **7 participants** reviewed the material on user stories as well before coming to the workshop. **9 participants** reviewed the material instead of attending the lecture and **7 participants** neither attended the workshop nor reviewed the material. Interestingly, there is no special correlation between students not attending the lecture/not reviewing the material and students having more experience with user stories; the different experience levels are evenly divided among each of the four categories described above.

		ATTENDED LECTURE		
		Yes	No	Total
REVIEWED MATERIAL	Yes	7	9	16
	No	29	7	36
	Total	36	16	52

Table 5.2: Participant distribution – attended lecture/reviewed material

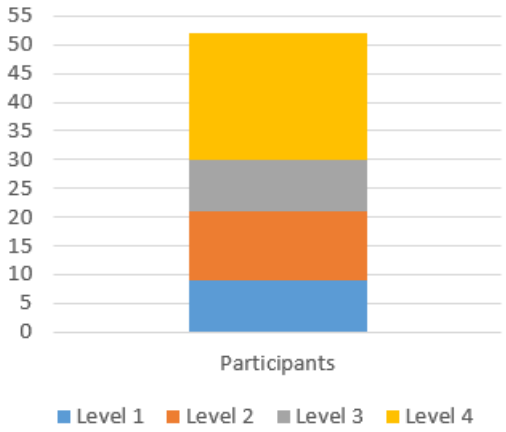

Figure 5.2: Highest level finished by participants

Lastly, out of 52 participants, **22** completed the game in its entirety within the time limit of 40 minutes. **Nine** students were still playing the final level when the time limit was reached, another **twelve** were still busy with level 3 and the final **nine** only finished the first level and were therefore still in level 2 when the time limit was reached. This is illustrated in Figure 5.2.

5.1.1 *Scores, Times and UES*

To start off, the total score distribution can be found in Table 5.3 and Figure 5.3, as well as the distribution of scores per level in Figure 5.4. For both the score and the time data in this section, only the data of those that have completed the game is taken into account to ensure that the number of data points remains consistent across all levels.

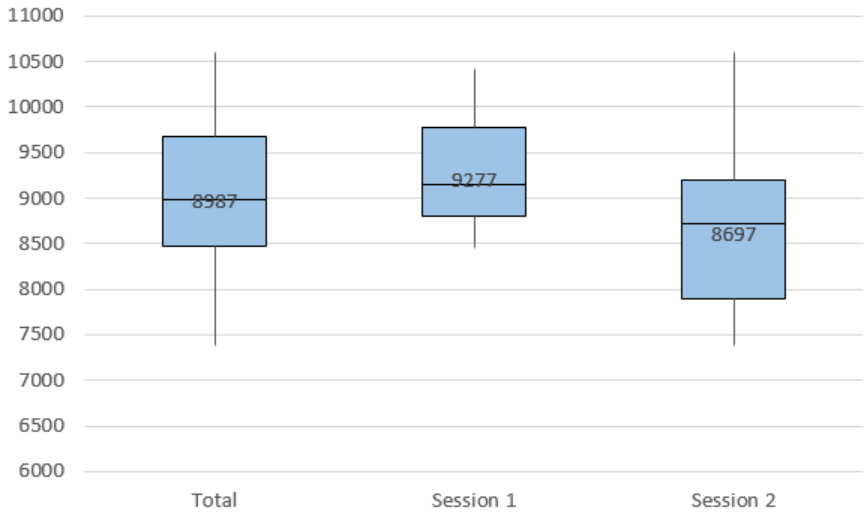

Figure 5.3: Score distributions per level and average scores (normalized)

	TOTAL	SESSION 1	SESSION 2
Average Score (normalized)	8687	9277	8697
St. Dev.	856	959	856
Average Time (normalized)	1268	1154	1382
St. Dev.	193	116	188

Table 5.3: Average scores/times and standard deviations

The first thing that stands out from this data is that the average score in Session 1 (9277) is higher than the average score in Session 2 (8687). After calculating the variances, a two-tailed T-Test (with equal variances) was performed. However, there is no significant difference, with a p-value of ≈ 0.12, which is higher than the α of 0.05.

Additionally, there is a clear (and significant; p-value $\approx 0.000 < \alpha$ 0.05) difference in scores between the levels 1 & 4 and the levels 2 & 3 – or the levels with faulty user stories and the levels without. This could mean two things; either the participants made more mistakes in the levels where not all faulty user story parts should be used, or they take more time completing these user stories than in levels where all associations should be completed.

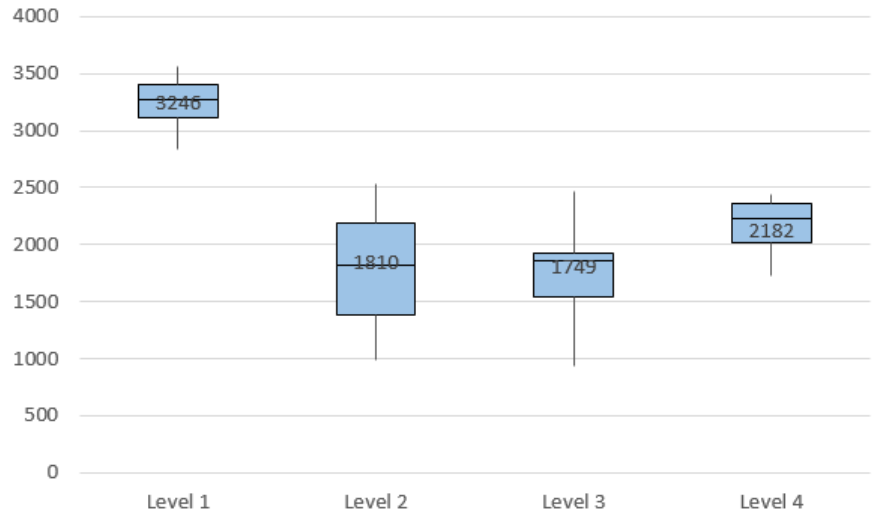

Figure 5.4: Total score distribution and average scores (normalized)

Combining this information with the data in Figure 5.5 & 5.6, the question of whether this difference in score is because of taking more time or making more mistakes is immediately answered, as there is a notable and significant (p-value $\approx 0.000 < \alpha\ 0.05$) difference in time when comparing the levels with "faulty" user stories to the levels without. Therefore, it can be assumed that players take a lot more time in levels with faulty user stories than in levels without them.

Figure 5.3 and 5.6 also show that, although the score is lower in Session 2, time is also significantly (p-value $0.004 < \alpha\ 0.05$) higher in this session. Therefore, there is no obvious increase in mistakes made between both sessions and the participants of Session 2 took more time without making more mistakes.

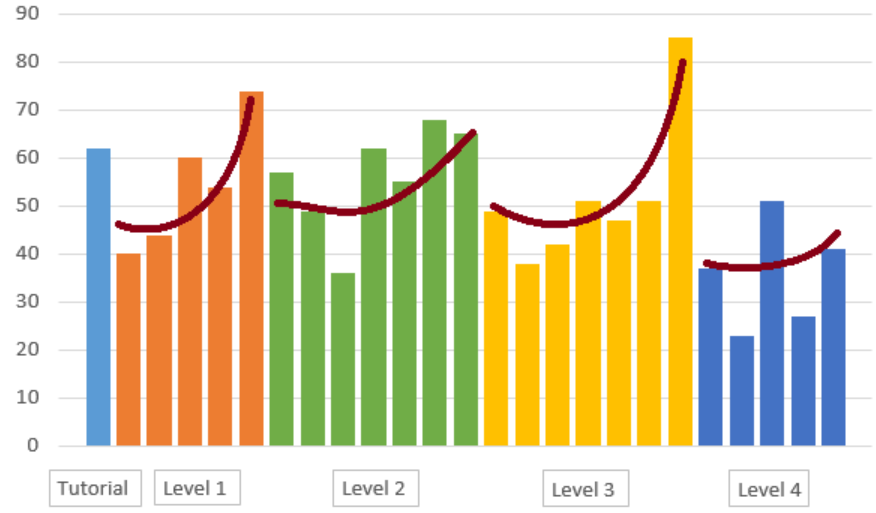

Figure 5.5: Average time per level per batch (normalized)

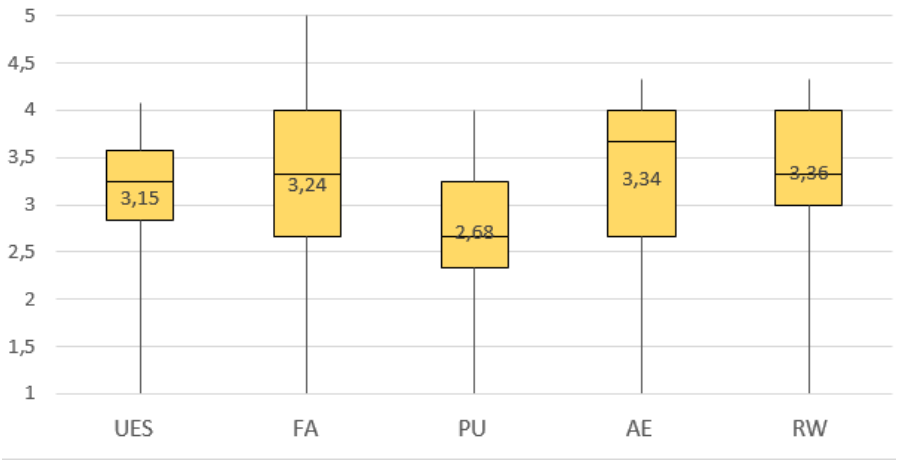

Figure 5.7: User engagement score per category

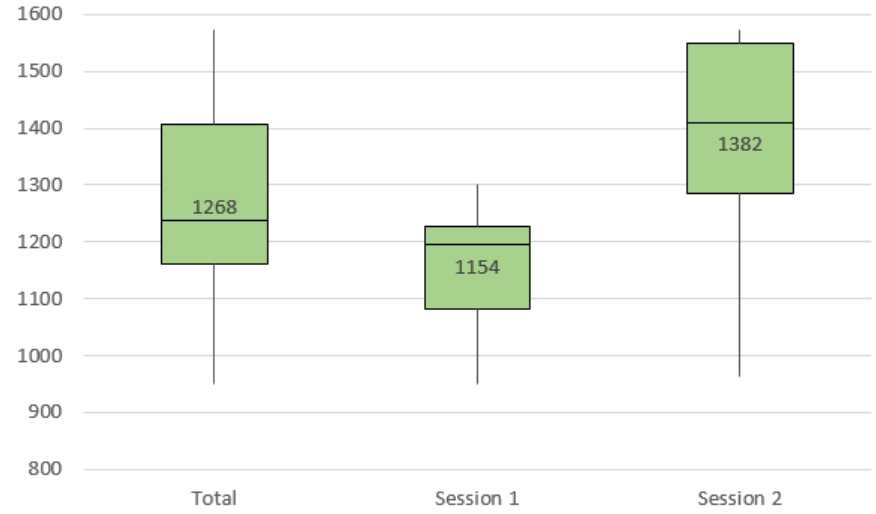

Figure 5.6: Average total time per session (normalized)

Finally, Figure 5.5 shows the average time per batch, colour coded per level. This graph shows that the time in most levels is laid out in a sort of "U"-shape, where the first batch in a level takes a relatively long time due to getting used to the new system. Following this, the next batches take less time but progressively more as the difficulty curve goes up, with the final batch taking the longest.

Exceptions to this are level 1, where the tutorial takes the place of the first batch in the level in terms of time, and level 4, which has an above average third batch. It is unclear why this is the case, as this batch is not significantly more difficult than the other batches in this level (see Table 3.6).

The UES-SF data was gathered through the post-test questionnaire. The test consists of 12 questions divided over 4 categories; Focused

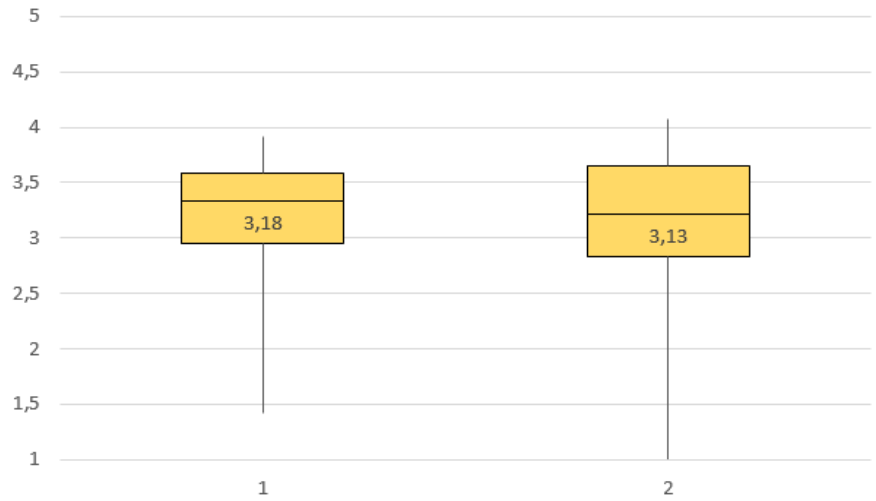

Figure 5.8: Average user engagement score per session

	FOCUSED ATTENTION	PERCEIVED USABILITY	AESTHETICS	REWARD FACTOR	USER ENGAGEMENT
Session 1	3.27	2.52	3.42	3.51	3.13
Session 2	3.11	2.88	3.21	3.21	3.18
Total Score	3.24	2.68	3.34	3.36	3.15

Table 5.4: User engagement scores

Attention (FA), Perceived Usability (PU), Aesthetics (AE) and Reward Factor (RW). Every category contains 3 questions. Participants give each category a score from 1 (completely disagree) to 5 (completely agree). The scores for each category are then calculated by summing each of these categories and dividing them by the number of items in the category, or all twelve for the entire UES. The exception to this is the PU category, which works with an inverse score scheme. Therefore, the PU points are calculated by summing (6 - points given) instead of just adding the given points.

After calculating these scores for each category and the UES as a whole, the results are shown in Figure 5.7 and 5.8, as well as in Table 5.4.

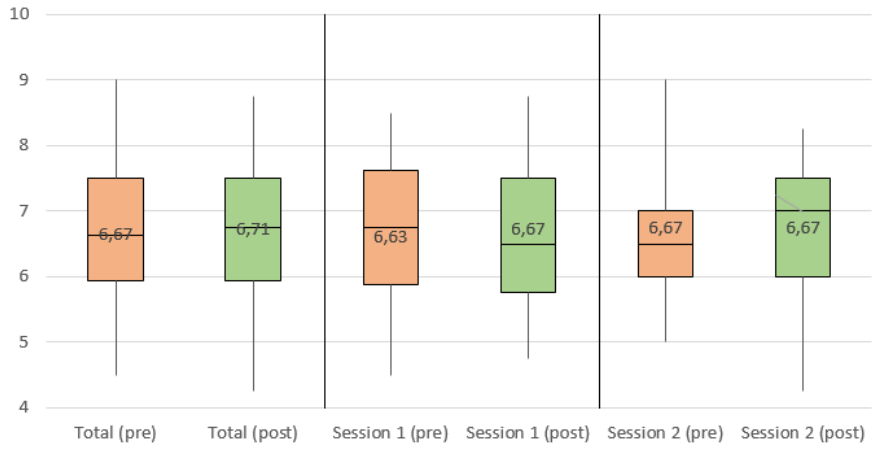

Figure 5.9: Grades of the pre- and post-test, total and per session

	SESSION 1		SESSION 2		TOTAL	
	PRE	POST	PRE	POST	PRE	POST
Average	6.67	6.71	6.63	6.67	6.67	6.67
St. Dev.	1.0197	1.0222	1.0152	1.1263	1.1406	1.1107

Table 5.5: Average grades of the pre- and post-test

As mentioned by [O'Brien et al., 2018], a UES of 3 is an average/neutral engagement score. Anything above this counts as "above average", everything below this as "below average". Most of the scores for BakeRE are around this neutral/"average" mark, with a total UES of 3.15. The highest scored category is "Reward Factor" with a score of 3.36, whereas the lowest scored category is "Perceived Usability" with a score of 2.68. This is coincidentally also the only category scored below the "passing grade" of 3.00. Therefore, usability would be one of the main focus points for possible future work (more on this in Chapter 6).

There is no statistically significant difference in user engagement scores between the two sessions (p-value 0.71 > α 0.05), nor is there a significant difference in score per category between the sessions.

Finally, the grades of the pre- and post-test are shown in Figure 5.9 and Table 5.5. These grades were calculated as follows:

1. Every participant can get a grade of 0–10, were 0 is nothing correct at all and 10 is everything perfect.
2. There are ten questions, therefore with every question worth 1 point.
3. Since every question contains four options (see Appendix E), every option is worth .25 points. A participant can either be

correct about an option and receive the .25 points or incorrect and not receive the points.

4. All of these points are added together for the overall grade of the participant.

5. An example; say that two out of four options are supposed to be picked. Then a student would get the full point for picking both of these options and not picking the other two. They would lose .25 points for picking an option that was not supposed to be picked, but they would also lose .25 points when not picking an option that was supposed to be picked.

There is no significant difference between the pre- and post-test grades, neither in the total score (p-value 0.98 > α 0.05) nor for any of the sessions (p-value 0.88 & 0.90 respectively). This also shows in the difference in averages; there is a slight increase, but almost nothing (see Table 5.5).

5.2 CORRELATIONS

Several Pearson tests have been run on the dataset to test for possible correlations. Correlations between variables could be interesting to find, especially when they are unexpected. For example; we can assume that there will be a correlation between time and score, as the calculation of the score is largely based on the time it took to complete a user story. However, if an unexpected relation between for instance UES and progress is found, this could shed a different light on the results as a whole, and new insights can be gained from trying to figure out why this relation exists.

The results of these tests can be found in Table 5.6.

5.2.1 *Variables*

The terms used here are (in order):

- **Session:** whether a participant was a part of Session 1 or Session 2 *(binary)*;
- **Score:** the total BakeRE score a participant achieved;
- **Time:** the time it took a participant to complete the tutorial and all 4 levels of BakeRE (22 entries);
- **Levels Completed:** the highest level completed by a participant (value from 1–4)
- **Attended Lecture:** whether or not a participant attended the lecture on user stories *(binary)*;

- **Reviewed Material:** whether or not the participant reviewed the material on user stories before the workshop *(binary)*;
- **Prior Experience:** whether or not the participant has prior experience with user stories *(binary)*;
- **Flipped:** whether or not the participant has made the mistake of "flipping" their answers during the pre-test (see Section 4.4.3) *(binary)*;
- **Progress:** whether or not the participant has made progress between the pre- and post-test. It is assumed that students cannot unlearn knowledge on user stories; therefore, a student counts as having made progress when the difference between the pre- and post-test is higher than zero *(binary)*;
- **UES:** the User Engagement Score a participant has given to BakeRE.

5.2.2 *Correlations*

The following correlations between these variables were found:

1. **Session – Time;** the participants in Session 2 took more time than those in Session 1. This was already apparent when looking at the data in Figure 5.6 and confirms the statistical significance.
2. **Session – Attended Lecture & Session – Reviewed Material;** the participants that attended the lecture were more likely to be in Session 1 than in Session 2. Inversely, those that reviewed the material were more likely to be in session 2.
3. **Attended Lecture – Reviewed Material;** additionally, the participants that attended the lecture mostly did not review the material. This is of course one of the prerequisites for having a split between those that attended the lecture/those that reviewed the material among the two sessions.
4. **Attended Lecture – Progress;** those who attended the lecture made less progress between the pre- and post-test than those who did not. This seems sensible, because those who attended the lecture already gained knowledge on the quality criteria on user stories during this lecture, which gives these students an advantage on the pre-test.
5. **Score – Time;** finishing the game using less time means a higher score. Since the height of the score is decided based on time spent per user story, this makes perfect sense.

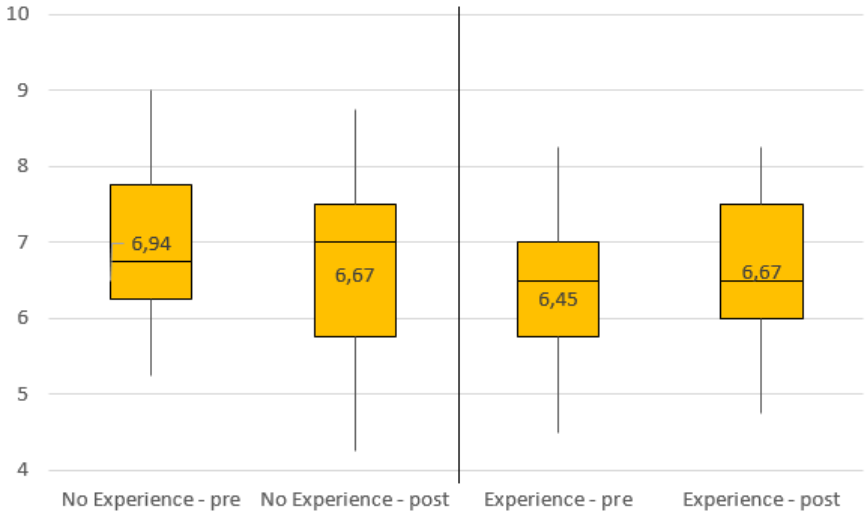

Figure 5.10: Effect of experience/no experience on progress

6. **Time – Attended Lecture;** the participants that attended the lecture were faster when playing the game than those who did not.

7. ***Time – Levels Completed & Score – Levels Completed;*** *only the times and scores of those that completed all levels is included in this section, therefore this makes every entry for "Levels" level 4, therefore a constant, therefore no meaningful correlation can be calculated.*

5.3 EFFECT OF EXPERIENCE ON RESULTS

To test whether or not there is a difference in results between groups with and without prior (user story) experience, the answers from the demographic pre-test quesionnaire were used to divide the participants into two groups; those that had little to no (recent) experience and those that did have (recent) experience (see Figure 4.3). The group labeled "No experience" contains **24 participants** ("No experience before the lecture" and "used more than six months ago"), while the group labeled "Experience" contains **28 particpants** ("Used in a recent course", "used in a project" and "used in a course and in project(s)").

The first point of interest is whether or not prior experience influences the possible progress between the pre- and post-tests (see Figure 5.10 and Table 5.7).

When looking at Table 5.7, it seems as though (interestingly enough) the group that already had prior experience made more progress by playing the game than those that did not. This could mean that the game helped them jump-start their memory and bring up (almost) forgotten knowledge on user stories.

		Session	Score (normalized)	Time (normalized)	Levels Completed	Attended Lecture	Reviewed Material	Prior Experience	Flipped	Progress	UES
Session	Cor.	1	-.339	**.590****	-.005	**-.526****	**.443****	-.036	-.086	.079	-.054
	Sig.		.123	**.004**	.971	**.000**	**.001**	.802	.544	.580	.702
	N	52	22	**22**	52	**52**	**52**	52	52	52	52
Score (normalized)	Cor.	-.339	1	**-.648****	.bbb	.181	-.072	.060	.240	.043	-.064
	Sig.	.123		**.001**	.bbb	.420	.749	.791	.282	.850	.779
	N	22	22	**22**	.bbb	22	22	22	22	22	22
Time (normalized)	Cor.	.590**	-.648**	1	.bbb	**-.531***	.183	.047	-.122	.341	.257
	Sig.	.004	.001		.bbb	**.011**	.415	.834	.590	.121	.248
	N	22	22	22	.bbb	**22**	22	22	22	22	22
Levels Completed	Cor.	-.005	.bbb	.bbb	1	.056	.053	.077	.042	.072	.170
	Sig.	.971	.bbb	.bbb		.695	.709	.585	.768	.612	.227
	N	52	.bbb	.bbb	52	52	52	52	52	52	52
Attended Lecture	Cor.	-.526**	.181	-.531*	.056	1	**-.368****	.135	.039	**-.308***	-.199
	Sig.	.000	.420	.011	.695		**.007**	.340	.783	**.026**	.158
	N	52	22	22	52	52	**52**	52	52	**52**	52
Reviewed Material	Cor.	.443**	-.072	.183	.053	-.368**	1	.032	-.039	.141	.179
	Sig.	.001	.749	.415	.709	.007		.821	.783	.318	.203
	N	52	22	22	52	52	52	52	52	52	52
Prior Experience	Cor.	-.036	.060	.047	.077	.135	.032	1	.212	.190	-.139
	Sig.	.802	.791	.834	.585	.340	.821		.132	.177	.325
	N	52	22	22	52	52	52	52	52	52	52
Flipped	Cor.	-.086	.240	-.122	.042	.039	-.039	.212	1	.164	.050
	Sig.	.544	.282	.590	.768	.783	.783	.132		.244	.725
	N	52	22	22	52	52	52	52	52	52	52
Progress	Cor.	.079	.043	.341	.072	-.308*	.141	.190	.164	1	.108
	Sig.	.580	.850	.121	.612	.026	.318	.177	.244		.445
	N	52	22	22	52	52	52	52	52	52	52

Table 5.6: * (Pearson) Correlation is significant at the 0.05 level (2-tailed).
** (Pearson) Correlation is significant at the 0.01 level (2-tailed).
bbb Cannot be computed because at least one of the variables is constant.

	EXPERIENCE	NO EXPERIENCE
PROGRESS	14 (58.23%)	10 (41.67%)
NO PROGRESS	11 (41.39%)	17 (58.62%)

Table 5.7: Progress/no progress per experience level

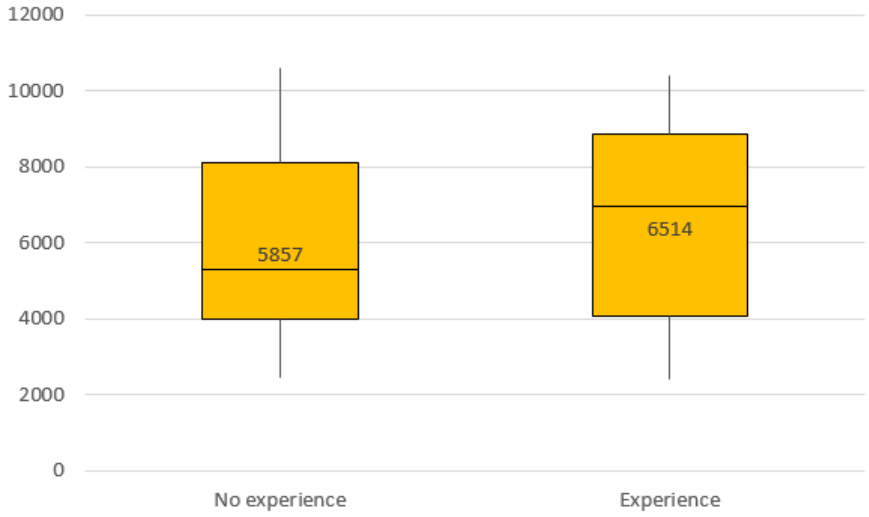

Figure 5.11: Effect of experience/no experience on score

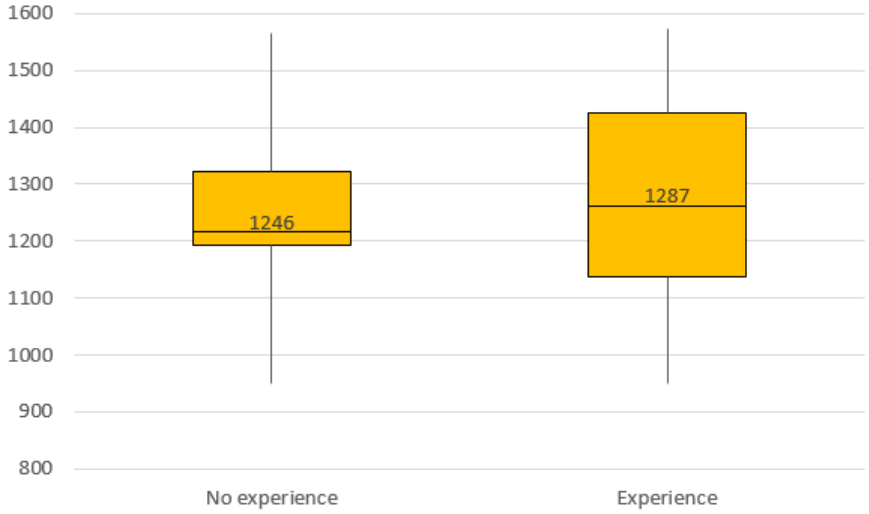

Figure 5.12: Effect of experience/no experience on time

However, no significant differences were found; neither between "No Experience" and "Experience" (-2.010 < -1.307 < 2.010) nor between each of the pre- and post-tests (-2.013 < 0.818 < 2.013 and -2.005 < -0.817 < 2.005 for "No Experience" and "Experience" respectively).

Secondly, Figure 5.11 and 5.12 show the effect of experience on the distribution of participant score and time.

5.4 ASSIGNMENT

Approximately three weeks after the workshop, students worked in groups of 2–3 people to participate on an assignment on user stories as a part of the course on RE. The assignment consisted of creating

	ATOMIC	MINIMAL	FULL SENTENCE	PROBLEM ORIENTED	UNAMBIGUOUS	INDEPENDENT	OTHER
Average	0.1379	0.6209	0.1034	0.4138	0.6207	2.5862	2.2758
St. Dev.	0.3448	1.1269	0.3045	0.8517	0.7617	1.6716	4.4790

Table 5.8: Average number of Mistakes (quality) made during the assignment on user stories, with a total of 6 answers per category.

a database for BakeRE (see Section 3.2.2 for more information on a BakeRE database).

29 groups of students handed in the assignment. For the purpose of this thesis, the mistakes made by students on the "Mistakes"-sheet were counted, tallied and divided over the categories used within BakeRE (Atomic – Minimal – Full Sentence – Problem Oriented – Unambiguous – Independent). This is shown in Figure 5.13. Additionally, the average number of mistakes per handed-in dataset is shown in Table 5.8. Every category has 6 entries, therefore the average number of mistakes is on a scale of 0–6. "Other" however deals with all mistakes made that do not belong to any of the categories in particular (for example making something that uses illegal characters or leaving necessary parts of the file blank) and therefore has no set limit of entries.

When looking at these, it becomes apparent that the main type of mistakes is in the "Independent" category, with an average of more than 2 per dataset. When looking at the assignments, this is mostly because many students mixed up this category and the "Minimal" category. This does make sense when looking at the descriptions of both categories:

- Minimal: a user story contains nothing more than a Role, an Action and a Benefit
- Independent: a user story is self-contained and has no inherent dependencies on other user stories

They do have some overlap, for instance the sentence

> "As a student, I want to learn to make high quality user stories (using the tools described in user story 12 concerning QUS criteria), so that I can improve my RE grades"

is neither Minimal (it contains additional information between brackets) nor Independent (it points toward another user story, without which this user story can not be implemented). However, the user story

> "As a student, I want to make high quality user stories (as per my professor's instructions), so that I can improve my RE grades"

is not Minimal (it includes additional information between brackets) but it *is* Independent (these additional instructions have nothing to do with implementations nor are they instructions for the developers). Future versions of BakeRE could put more emphasis on this subtle difference; more on this in Chapter 6.

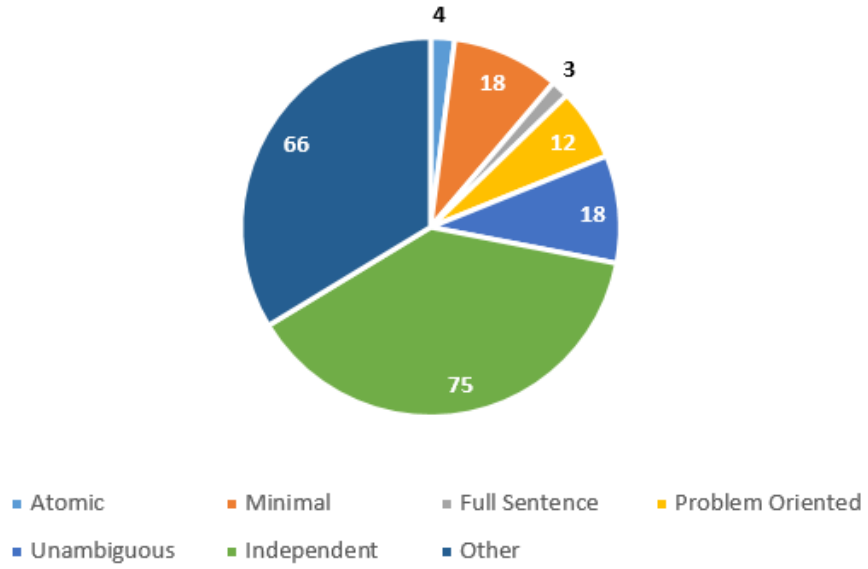

Figure 5.13: Mistakes (quality) made during the assignment on user stories

Finally, students did very well on the "Atomic" and "Full Sentence" categories. These are also the categories which are the "easiest" to spot when playing BakeRE, as a building block containing only one or two words (in the case of Full Sentence) or a lot of text, usually containing the words "and/or" stand out from all the regular building blocks. Future work could focus on gathering information whether or not there is a correlation between playing BakeRE and the high performance regarding these categories.

6 DISCUSSION

6.1 CONCLUSION

Using the data gathered and examined in Chapter 5, the following conclusions can be drawn regarding the research (sub)questions:

6.1.1 *Knowledge Gain*

> Does the usage of a serious game during the learning of user stories improve the knowledge gain regarding user story quality? *Subquestion 1*

To see if this question was answered, the results of the pre- and post-test, specifically the answers regarding the user story quality tasks, were compared. As mentioned in Section 5.1.1, there is no significant difference (or "progress") between the pre- and post-tests, although there is a minor increase in average (see Figure 5.9 and Table 5.5). Interestingly, an inverse correlation was found (see Section 5.2 and Table 5.6) between lecture attendance and progress made. In other words; those that attended the lecture were less likely to make (additional) progress by playing BakeRE.

Possibly, the game could help more with solidifying knowledge gained during the lecture, but this would not show up in the difference between the short-term pre- and post-test, instead becoming more apparent when testing with participants (with and without the treatment) across multiple sessions. Alternatively, as no clear immediate feedback was shown to students when they made a mistake, participants may not have learned from their mistakes enough, therefore having made little progress.

At the end of the day, this experiment *did not* succeed in proving that there was an improvement on knowledge gained regarding user story quality by using a serious game, *nor* does it show the game to have detrimental effects.

6.1.2 *Knowledge Retention*

> Does the usage of a Serious Game during the learning of User Stories improve the Knowledge Retention regarding User Story Quality? *Subquestion 2*

Because the assignment on user stories for the Requirements Engineering course was performed in teams instead of by single participants, and because the participant IDs were not used when handing in these assignments, it is difficult to draw conclusions regarding each participant's growth and progress in between the workshop and the assignment.

However, conclusions can be drawn regarding which user story quality categories came across particularly well and which need more attention. As shown in Figure 5.13 and Table 5.8, as well as explained in Section 5.4, participants performed particularly well on the categories of "Atomic" and "Full Sentence", whereas they performed less well on the "Independent" category.

Future work regarding BakeRE should definitely focus on whether or not this high performance regarding Atomic and Full Sentence categories correlates to playing BakeRE, or if these are just the "easy" categories regardless of the game. Additionally, new iterations of BakeRE could focus on improving the low performance on Independent categories. More on these points in Section 6.3.

6.1.3 *Learning Experience*

Subquestion 3

> Does the usage of a Serious Game during the learning of User Stories improve the Learning (User) Experience?

As there was no comparison between the learning experience of playing with BakeRE and another form of treatment, it cannot be said that the learning experience has increased or decreased. However, BakeRE did score (slightly) above/on average regarding user experience. It would be interesting to compare this outcome to the outcome of another treatment when doing future experiments to see if this is a (significant) increase, but more on this in Section 6.3.

As mentioned in Section 5.1.1, there was no significant difference between the UES of Session 1 and Session 2. However, when looking at the scores in Table 5.4, Session 1 does seem to be more positive overall than Session 2. This might have something to do with participants in Session 1 performing "better" on average within the game itself, with a significantly higher average score than participants in Session 2. Usually, when people feel that they are good at something, they will rate that something higher than if they feel that they are not [Baric et al., 2014].

The only exception on this trend is the "Perceived Usability", which was rated much lower by the participants of Session 1 than by those in Session 2. Possibly, these students were more competitive than those in Session 2 and therefore more annoyed by what they perceived as usability issues, but this is conjecture; there is no way to gain this information for certain from the data that was gathered.

6.1.4 *Main Research Question – Conclusion*

After combining all this information, the following can be concluded regarding the main research question:

"Does the usage of a serious game *during* the learning of user stories improve the learning experience and lead to higher skill & knowledge gains?" *Research Question*

Sadly, none of the subquestions could be definitively proven. Therefore, the answer to the main research question is that this experiment *did not* succeed in proving that using a serious game during the learning of user stories improves the learning experience, *nor* that it leads to higher knowledge gains. However, there is also no detrimental effect, and there is still much possible future work to be done; see Section 6.3.

6.2 DISCUSSION

The following observations were made during and after the workshop with BakeRE:

6.2.1 *BakeRE*

In the version of BakeRE that was used during the workshop, there was no immediate feedback on mistakes, therefore participants did not understand what they did wrong (and some made the same mistake multiple times "just to be sure"). This most likely influenced the (lack of) progress between the pre- and the post-test, as this has not given the students enough time to reflect.

6.2.2 *Workshop*

Due to the lack of mobile devices, the group had to be split into two sessions; something that was not part of the original experimental design. This probably had some influence on motivation, especially for the second group that had to wait for the first group to finish before being allowed to start the game.

Additionally, even though all results are anonymous, some participants most likely still exaggerated how much effort they put into the course (for instance, almost everyone that didn't attend the lecture said they had spent considerable time preparing the material at home, which is unlikely – so much so that this showed up in the correlations). This means that the demographic data is most likely somewhat skewed, although there is no way to check this.

6.2.3 *Testing – Knowledge Gain*

Due to the setup of Blackboard[1], the tool used to perform the pre- and post-test, all instructional text was all the way at the start of the exercise. In other words, the explanation of the demographic questionnaire and the information on the multiple choice tasks on user story quality were both at the top of the page.

As a result of this, participants read everything at the beginning and did not understand the assignment correctly, which caused the issues as described in Section 4.4.3 where the answers were "flipped".

6.2.4 *Testing – Knowledge Retention*

Due to a mistake on the part of the researchers, participants did not keep their participant ID after exiting the workshop. Therefore future tests could no longer be coupled to the results of the workshop. This made it much more difficult to gain useful information concerning knowledge retention and other follow up tests.

6.2.5 *Instrumentation*

As described in Section 5.1, several participants experienced a game crash. All of these devices are Huawei devices, which apparently have an issue with the game library Phaser that was used for BakeRE. Additionally, two participants were part of both sessions and thereby invalidated their data.

6.3 FUTURE WORK

The future works section has been split into two parts; a section regarding possible expansions/improvements to the game BakeRE and a section on recommended (additional) tests that could be run to better test the merit of BakeRE, both regarding knowledge gain/retention as well as learning experience.

6.3.1 *Expansions and/or Improvements to BakeRE*

The biggest possible improvement to BakeRE will probably be the introduction of immediate feedback after making a mistake when matching a user story. Right now, as mentioned in Chapter 3, BakeRE plays an auditory cue when a mistake is made and then gives a pop-up message at the end of a level with a debriefing on the category in which the most mistakes were made. This category can either be one

1 https://www.blackboard.com/blackboard-learn/index.html

of the 6 categories focused on in level 2 and 3, or "Well-formed" if the user story does not make sense grammatically.

However, as this specific feedback is only given at the end of a level, the time to reflect is likely too late for the students to truly benefit from this debriefing. A solution for this would be to implement a similar debriefing system, but show the message immediately upon making the mistake instead of at the end of the level. This might increase the knowledge gain after playing with BakeRE, as this immediate feedback is a learning opportunity which can be used to further improve one's skills.

Additionally, as was shown by the results of the assignment that students handed in for the Requirements Engineering course (see Figure 5.13), students performed poorly on the quality category of "Independent" user stories. There was a clarification message sent concerning what was and what was not allowed within this particular category after students asked questions for more clarifications, but it is quite possible not all students read this.

Whether or not these results are then due to the not reading of the clarification email, BakeRE or because it is "just a difficult subject", additional information or different examples on this particular subject could improve the knowledge gain associated with this quality criterion.

Finally, as the UES has shown, the perceived usability is below average. Possible usability fixes that were offered as comments by participants are:

- to increase the "touchable" areas for buttons and the draggable objects;
- to lower the threshold on when something is/is not a touch;
- to add the option of clicking menu items instead of dragging them to the center.

Additional fixes and other extensions could be discovered by performing a usability examination with a (new) focus-group of participants.

6.3.2 *Experimentation*

When performing future experiments regarding the usefulness of BakeRE during the learning of user stories, the first advice would be to perform an experiment with a control group that does not use the game, but does perform similar exercises (for instance, an exercise sheet on matching user story parts in a similar way to BakeRE, but on paper). Both groups should then perform the same pre- and post-test, after which their results and progress can be compared. This way the

(possible) differences in using and not using a serious game can be shown and examined.

A second addition would be to perform multiple of these tests over a longer period of time to properly test for knowledge retention. Depending on the focus of these future experiments, this could be over the course of weeks or months. For proper comparison, the tests of the initial knowledge gain tests and the knowledge retention tests should be comparable.

Thirdly when gathering more data on user experience using the UES (SF), it might be interesting to compare the calculated user engagement scores with the UES these same participants would give another, "known" piece of software. This way a participants bias can be shown – for instance, if someone is normally a very optimistic person, a low score would be much worse than if an already pessimistic person gave that same score.

Finally, it might be interesting to examine the effect of BakeRE on the specific categories. When taking this thesis as an example, in Figure 5.13 it is clearly shown that participants perform particularly well on Full Sentence and Atomic criteria, and poorly when working with Independent criteria. This could either be because BakeRE gave a highly efficient introduction on the first two categories and was lacking in the latter, or the lecture did, or it is a coincidence and these are just easy and difficult criteria respectively.

Future experiments could gather data to figure this out; for instance, by tracking which specific mistakes a player makes within BakeRE and combining this knowledge with future versions of the assignment. This information can then be used to improve BakeRE even further, until it is the most optimal educational tool for learning user stories it can be.

Part IV

APPENDICES

A

APPENDIX: GDD – BUILDING GAME

A.1 GLOBAL DESCRIPTION

A game in which structures have to be created out of existing building blocks. These building blocks are User Story components, which can vary from Role-Action-Benefit to (for example) INVEST categories. The available blocks will have more "wrong" parts the further the player progresses. Players can create structures based on given descriptions, as well as judge existing structures as they progress through the levels. The main activity the player will perform is associating a description or task with a set of building blocks, either existing structures or a structure to be.

A.1.1 *Audience*

This game is meant for students learning the basics of User Stories within the greater field of Requirements Engineering. No prior knowledge of User Stories is required, however the basics of Requirements Engineering's goals and benefits are assumed as prior knowledge.

A.1.2 *Environment*

The game will (ideally) be played in combination with a practical session during a course on Requirements Engineering. The game should be playable on mobile devices, therefore it could be played anywhere and anytime, as long as an internet connection is available.

A.1.3 *Learning Goals*

The following table gives an overview of the knowledge goals and skills that should be attained in (and are at the center of) each level:

	KNOWLEDGE	SKILLS
Level 0 – Tutorial	General (conceptual) knowledge of US structure.	Writing (structurally correct) US. Seeing whether or not a US is structurally correct. Seeing whether or not a US is correct compared to the original text it was based on & pointing out where in the US the mistake is made.
Level 1 – User Stories	General (conceptual) knowledge of US structure.	Seeing whether or not a US is correct compared to the original text it was based on & pointing out where in the US the mistake is made.
Level 2 – INVEST	General knowledge about quality frameworks. The 6 categories of INVEST & the overall meaning of these categories.	Writing US that are correct according to INVEST. Seeing whether or not a US is correct according to INVEST and why.
Level 3 – QUS	The QUS criteria & the overall meaning of these criteria. The difference in importance between these criteria/which criteria to focus on first and why. Lindland's criteria in relation to QUS.	Writing US that are correct according to QUS. Seeing whether or not a US is correct according to (each criteria included in) QUS and why (focusing on first 7 criteria).
Level 4 – QUS, cont.	The QUS criteria & the overall meaning of these criteria. The difference in importance between these criteria/which criteria to focus on first and why.	Writing US that are correct according to QUS. Seeing whether or not a US is correct according to (each criteria included in) QUS and why (focusing on the other 6 criteria).
Level 5 – Acceptance Tests	General knowledge about acceptance tests. General (conceptual) knowledge of a (BDD) acceptance test. Knowing which criteria are/aren't testable.	Creating (structurally correct) acceptance tests. Determine whether or not passing the acceptance test would mean the User Story is fulfilled.

Table A.1: Knowledge goals and skills per level

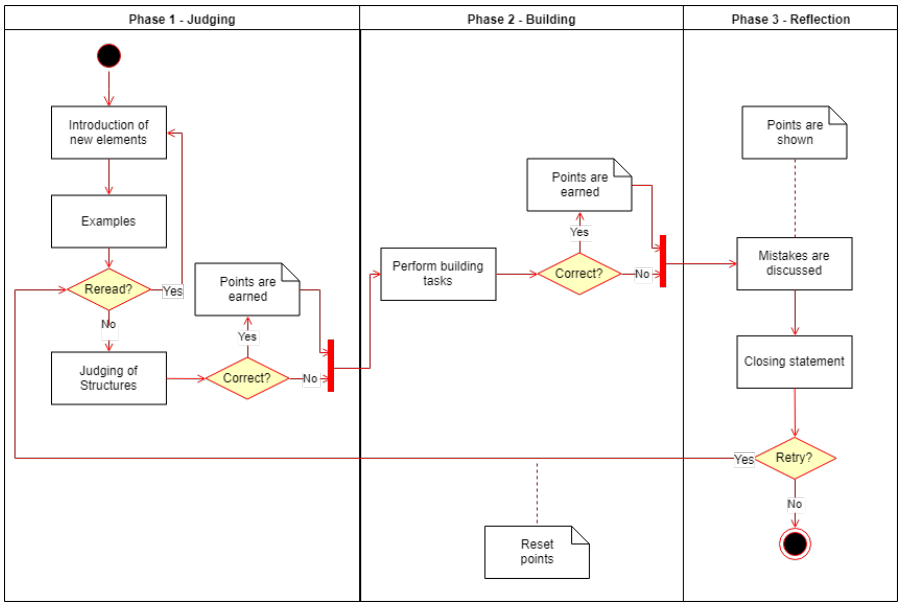

Figure A.1: Activity diagram describing the general layout of a level

A.2 GAME DESCRIPTION

The games main mechanic is dragging and dropping building blocks to create a structure. Correct structures get awarded high scores, incorrect structures can lose you points. More points are awarded the less time is spent. The goal of the game is to get the highest score possible. Before every building phase there is a "learning" phase where the player judges existing structures based on the information and new mechanics provided within this level. Here the player can win points by correctly pointing out whether or not a mistake has been made, and if so where in the structure this error occurred.

A.2.1 *Level 0 – Tutorial*

The player gets introduced to the core mechanics of the game; dragging and dropping building blocks to create a structure. Within this first level, the player gets a quick walkthrough of the general structure of a User Story (Role-Action-Benefit) and is shown how these can be combined into a structure (one of each in specific order, where Benefit is optional). Then, several structures are shown which have a correct structure, following which the player is asked to point out mistakes in some faulty structures. Finally, the player is asked to construct some structures of their own, after which the tutorial is complete. This level will not yet have time pressure, however the following levels will introduce an (increasingly fast) timer for each phase.

A.2.2 *Level 1 – User Stories*

This level does not truly introduce new concepts, but continues on with the new information learned in the tutorial. The first phase of the level introduces some customer requirements and accompanying User Stories, where the player is asked to point out if the User Story correctly captures the requirements as described by the customer, and if not which of the building blocks (Role-Action-Benefit) is incorrect. The building phase then consists of its continuation by creating User Story structures based on similar customer requirements.

A.2.3 *Level 2 – Quality Frameworks (INVEST)*

After successfully completing the former two levels, the player can advance to more complex (User Story related) concepts, the first of which are User Story quality frameworks. This level will start with a general overview of US quality frameworks in general, after which the INVEST framework will be introduced and explained. For each letter in the INVEST acronym the definition is given as well as some examples, after which the player is asked to point out for some structures under which INVEST criterion this structure would be wrong. Finally, during the building phase, the player is asked to create User Stories that do not conflict with any of the INVEST criteria by choosing the right combination of building blocks from a set of comparable blocks, where only one combination would yield an INVEST-sound structure.

A.2.4 *Level 3 – Quality Frameworks (QUS)*

The next two levels have a highly comparable setup to level 2, only now focusing not on the INVEST framework but instead on the QUS. During the introduction phase, the overall QUS framework is introduced, as well as a short introduction as to the priorities of these categories. Then the gameplay phase continues on in the same vein as the former level did, first asking the player to point out which category was violated for some faulty structures, followed by letting the player create/complete their own structures from combinations of existing building blocks.

A.2.5 *Level 4 – Quality Frameworks (QUS), cont.*

This level continues on where the former level left of, introducing the final 6 categories and then continuing on with the same gameplay loop level 2 and 3 introduced, only now focusing on these 6 new categories, while once again stressing the difference in priority between the fist 7 and the last 6.

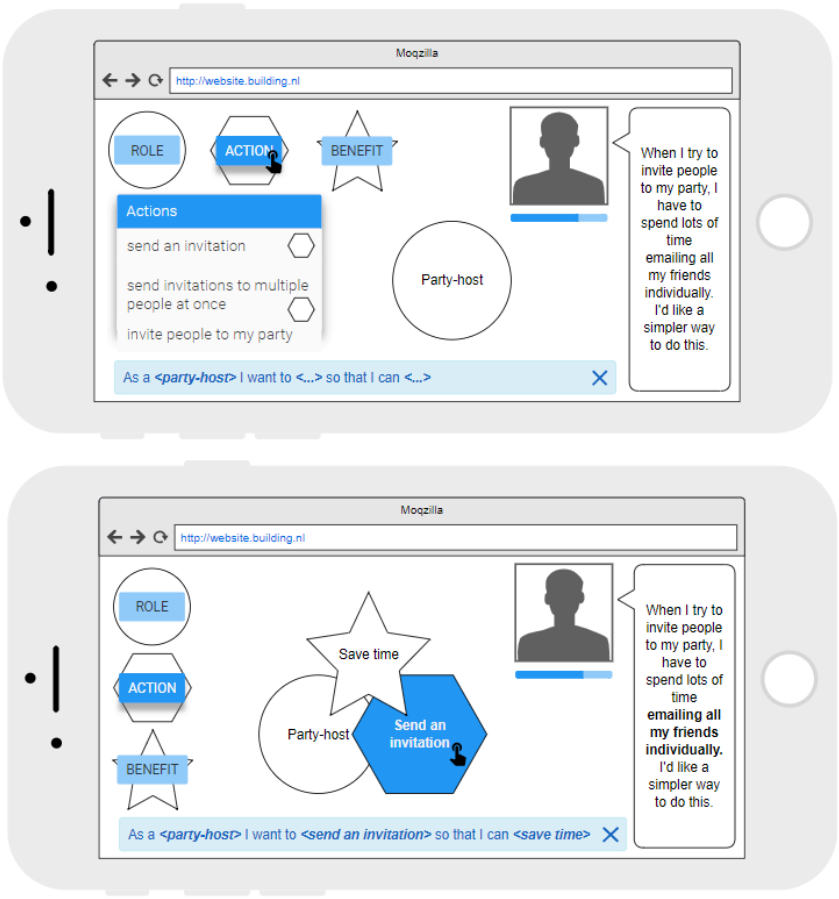

Figure A.2: Mockup/wireframe screenshots of the building and judging phase

A.2.6 *Level 5 – Acceptance Tests*

In the final level, the player gets introduced to the concept of acceptance tests. The first phase here again consists of an introduction to the concept and the general structure of an acceptance tests (combined with some examples), followed by a phase where the player would point out whether or not an acceptance test would be a) structurally correct and b) passing the acceptance test would mean the User Story is fulfilled. Finally the player would create some structurally correct acceptance tests to test some of the User Stories they themselves have created during the game, coming full circle back to the beginning.

A.2.7 *Extensibility*

Depending on time pressure, the game could be extended or reduced by respectively adding or removing levels and/or concepts. For instance, the minimum viable product for this game would only include the first two levels focusing on the structure of User Stories. An

extended version would include the quality framework levels while the ideal version would also include the level on acceptance tests.

If eventually the game were to be extended, a separate level on, for instance, ambiguity could be added between level 2 and level 3 to serve as an even stronger foundation for the levels to come. Other important principles could be tackled by this game as well, such as the prioritization of User Stories or creating conceptual models, by slightly adapting the game mechanics for the new "gimmick".

Another possible way to further extend the game would be to introduce more "combination" levels, where instead of having fairly conceptually partitioned levels discussing only one level, there would be levels in between that deal with multiple concepts at the same time. This could pose a challenge however considering that the different concepts need different kinds of building blocks.

A.3 IMPLEMENTATION

The game will be created as a mobile web application. The codebase will be created using HTML & JavaScript in Visual Studio, using the toolset Apache Cordova included within this program to assure ease of use and cross-platform development, automatically accounting for Android, iOS and Windows phones.

More information:
https://docs.microsoft.com/en-us/visualstudio/cross-platform/cross-platform-mobile-development-in-visual-studio?view=vs-2017#HTML

A.4 CHALLENGES

This game has to be balanced very carefully to ensure a continuously engaging gameplay experience. As with many simple task-based games, the challenge is with the time limit it poses and the amount of building blocks the player is confronted with. In a game like this flow is incredibly important and therefore the game should be playtested many times over to ensure that this flow is truly there.

B

APPENDIX: GDD – DETECTIVE GAME

B.1 GLOBAL DESCRIPTION

A mystery-esque game where people need to be interrogated to iron out inconsistencies/paradoxes between testimonies. The premise of the game is that a murder has happened and there are four suspects. You, as the player, have to figure out which of the suspects is the killer by comparing their statements against each other and against evidence found on the scene. The evidence can take the shape of information (such as an explanation on User Story structure) or of rules/criteria (such as "every user story starts with a role"). This evidence can then be used to find and prove inconsistencies (or mistakes) made within the users statements (in the form of User Stories). Proving a mistake has been made will update the testimony to include the new information that has been unearthed, until all testimonies have been updated to reflect the right story, which can then be used to unmask the true killer.

B.1.1 *Audience*

This game is meant for students learning the basics of User Stories within the greater field of Requirements Engineering. No prior knowledge of User Stories is required, however the basics of Requirements Engineering's goals and benefits are assumed as prior knowledge.

B.1.2 *Environment*

The game will (ideally) be played in combination with a practical session during a course on Requirements Engineering. The game should be playable on mobile devices, therefore it could be played anywhere and anytime, as long as an internet connection is available.

B.1.3 *Learning Goals*

The following table gives an overview of the knowledge goals and skills that should be attained in (and are at the center of) each level:

	KNOWLEDGE	SKILLS
Witness 1 – Tutorial	General (conceptual) knowledge of US structure.	Writing (structurally correct) US. Seeing whether or not a US is structurally correct.
Witness 2 – User Stories	General (conceptual) knowledge of US structure.	Seeing whether or not a US is correct compared to the original text it was based on & pointing out where in the US the mistake is made.
Witness 3 – INVEST	General knowledge about quality frameworks. The 6 categories of INVEST & the overall meaning of these categories.	Writing US that are correct according to INVEST. Seeing whether or not a US is correct according to INVEST and why.
Witness 4 – QUS	The QUS criteria & the overall meaning of these criteria. The difference in importance between these criteria/which criteria to focus on first and why. Lindland's criteria in relation to QUS.	Writing US that are correct according to QUS. Seeing whether or not a US is correct according to (each criteria included in) QUS and why (focusing on first 7 criteria).

Table B.1: Knowledge goals and skills per witness

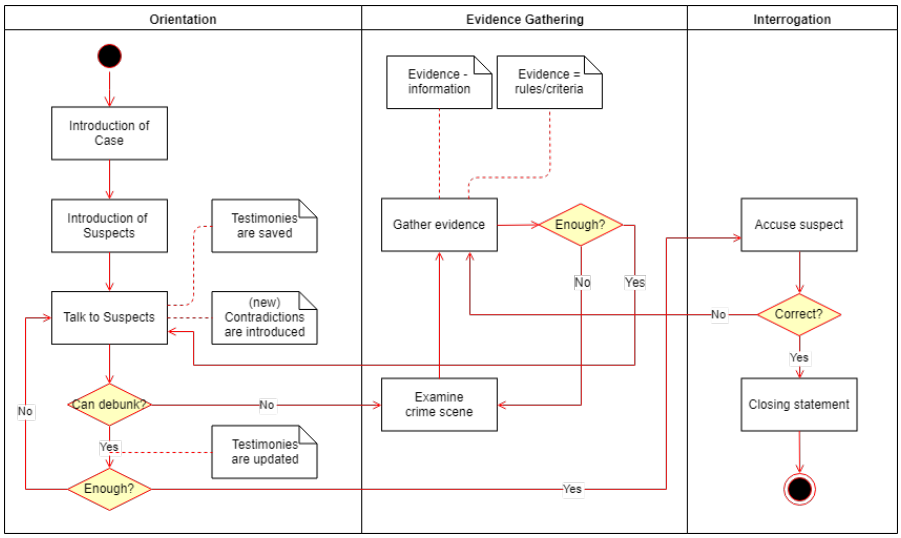

Figure B.1: Activity diagram describing the general layout of the game

B.2 GAME DESCRIPTION

Every witness (or suspect) within the game has a main theme on which their testimonies focus. For instance, one of the witnesses will mainly have structural problems with their testimonies. These kind of issues can be explained in-game as the suspect "being vague" on purpose, therefore hiding something from the inspector. The gameplay focuses mainly on deciding who to talk to during interrogations and clicking on suspicious areas (thus possible evidence) during the investigation phase. More areas to explore will be unlocked as the player progresses through the investigation and more information becomes known.

It is possible to freely switch between the examination phase and the interrogation phase, as well as making it possible to talk to every suspect as often as you want and as many times as you want. However, the suspects will repeat the same lines until progress is made. The player is not supposed to "fix" each suspects testimony individually and then move on to the next one, but should return to the same witnesses several times with new information, gradually unraveling the truth of the case.

B.2.1 *Witness 1 – Tutorial*

The first witness should, in true detective-game fashion, be a terrible liar who makes such obvious mistakes within their testimony that it cannot be missed. This setup can then be used to easily explain the mechanics without telling the player much outside of the information given by the suspect itself, giving the player the opportunity to find this out themselves. This witness will make mistakes based on simple

User Story structural concepts, such as forgetting to use a role, not having any action mentioned or just rambling on without any true points.

B.2.2 *Witness 2 – User Stories*

The next witness will be of the same level of "competence" as the next two, therefore not being as easy to debunk as the first witness, but still mostly focus on the structural parts of User Stories. Their theme revolves around structurally incorrect User Stories, as well as User Stories that do not correctly capture the requirement description they are meant to describe.

B.2.3 *Witness 3 & 4 – Quality Frameworks*

These witnesses will both make mistakes or ask questions based on their respective quality frameworks, for example using User Stories that are not independent for INVEST or User Stories that are not unambiguous for QUS etc.

B.2.4 *Extensibility*

The structure of this game is created in such a way that the addition of more subjects to tackle should be possible, as a new theme can be added in the form of a new witness. However, although this is technically possible, it will become more difficult in later stages of development as the story the game is centered around becomes more clear-cut. An extra character within the story must feel justified for it to work with the narrative, otherwise players might lose interest in the mystery itself and therefore lose motivation to finish the game. Therefore it is not as extensible within one game, however the same systems created for this game could be reworked into new, different games (for instance in the form of an entirely new case) which could be used for future development.

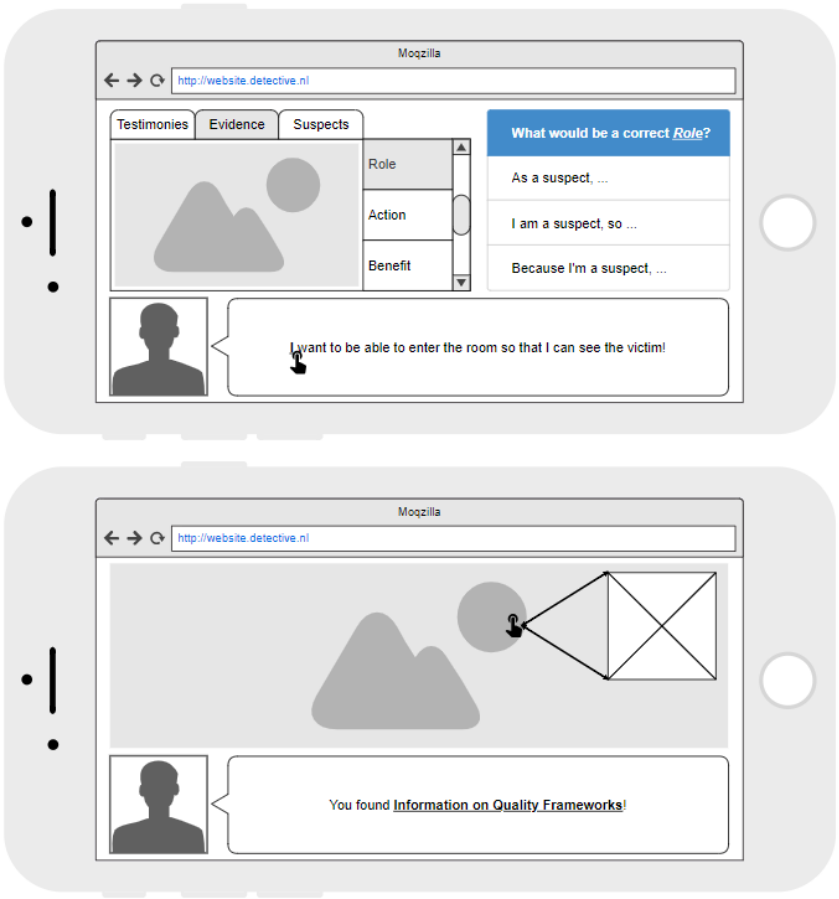

Figure B.2: Mockup/wireframe screenshots of the interrogation and examination phase

B.3 IMPLEMENTATION

The game will be created as a mobile web application. The codebase will be created using HTML & JavaScript in Visual Studio, using the toolset Apache Cordova included within this program to assure ease of use and cross-platform development, automatically accounting for Android, iOS and Windows phones.

More information:
https://docs.microsoft.com/en-us/visualstudio/cross-platform/cross-platform-mobile-development-in-visual-studio?view=vs-2017#HTML

B.4 CHALLENGES

The main challenge during the creation of this game will be how to include the User Story layer in a way that is not too "forced", therefore not completely breaking the immersion. The User Story layer will most likely never seem truly natural, but if the immersion becomes too fragile the game might become (too) childish or unintentionally funny.

Another challenge for this concept is the creation of an enticing mystery that players will want to solve, as well as a system against "gaming the system" and just instantly walking up to the final phase and try every single possible subject combination until you happen to get it right. One way to do this would be to hide the essential information until the endgame, but this could be detrimental to the mystery of the game, as everything before this point becomes "definitely not the killer" because it's too early.

Finally there is a clear limit to the amount of User Stories that can be used within the narrative without becoming superfluous, therefore reducing the number of practice time the player gets to have with every theme of User Story concepts.

C

APPENDIX: DATABASE BAKERE

The following pages contain the database used for BakeRE at the time of the workshop. The sheets are, in order:

1. Epics
2. User stories
3. Mistakes – Atomic
4. Mistakes – Minimal
5. Mistakes – Full Sentence
6. Mistakes – Problem Oriented
7. Mistakes – Unambiguous
8. Mistakes – Independent
9. Acceptance tests (page 1)
10. Acceptance tests (page 2)

Epic Title	Epic
Assignments	*Teachers need to set up assignments for the students to complete, often in groups.*
Tests	*Sometimes, teachers want to challenge students with intermediate tests or quizzes that can be completed in the learning environment.*
Accounts	*Teachers, teaching assistants and students should have accounts in the learning environments, each with different privilege.*
Course material	*The learning environment needs a section that contains the course material.*
Scheduling	*Scheduling information should be available to all students and teaching assistants and should be managed by teachers.*
Communication	*All participants of the learning environment should be able to communicate with one another.*
Recordings	*The learning environment should be able to play back recordings of the lectures.*
Groups	*Groups need to be created in the learning environment to help them collaborate in creating assignment.*

Epic Title	Key	Role	Action	Benefit
Assignments	*1*	Teacher	create an assignment in the learning environment	other external services are not necessary
Assignments	*2*	Teacher assistant	see the list of submissions that I was assigned to grade	I know how much grading work is ahead of me
Assignments	*3*	Teacher	delegate grading rights to my teacher assistants	I can reduce my grading load
Assignments	*4*	Teacher	restrict the file submission types	my teacher assistants can avoid checking file format constraints
Assignments	*5*	Teacher assistant	indicate whether I am uncertain with an assignment's grade	the teacher can review my grading
Tests	*1*	Teacher	create an anonymous quiz for my students	I can test students' knowledge without knowing their performance
Tests	*2*	Teacher	create a graded test within the learning environment for my students	I can include online tests in the course grading
Tests	*3*	Student	receive notifications of test deadlines	I do not forget which tests I need to complete
Tests	*4*	Teacher	generate statistics for a graded test	I can look for too easy or too difficult questions
Tests	*5*	Student	see the test scores of my colleagues	I can compare my performance against theirs
Accounts	*1*	Student	create an account for a course I attend	I can access the online learning materials
Accounts	*2*	Student	set the visibility level of my photo	I can establish who can see this kind of personal details
Accounts	*3*	Teacher	have insight into the current list of students with accounts	I know whether or not all of my students have succeeded in creating an account
Accounts	*4*	Teacher assistant	receive an account with access to grading functionalities	I can help the teacher in grading assignments
Accounts	*5*	Student	search for fellow students who are enrolled in the online learning environment	I can invite them to join a group for working at assignments
Course material	*1*	Student	receive a notification when new material is uploaded	I can immediately download new slides and papers
Course material	*2*	Teacher	upload the slides I use for my lectures to the learning environment	my students can review the slides at home
Course material	*3*	Teacher	upload the literature list to the learning environment	my students know what I expect them to study for the exam(s)
Course material	*4*	Student	have access to the literature list of the course(s) in which I am enrolled	I know what my teacher expects me to study for the exam(s)
Course material	*5*	Student	have access to the slides used during the lectures	I can review the slides at home
Scheduling	*1*	Teacher	create a course schedule within the learning environment	my students and teaching assistents can schedule their time correctly
Scheduling	*2*	Teacher	update the course schedule when unforeseen changes to the planning occur	my students and teaching assistents are aware of changes in the planning
Scheduling	*3*	Student	get a notification when a lecture is dropped	I know I do not have to attend the lecture
Scheduling	*4*	Teaching assistant	get a notification when I am required to assist a lecture or practical	I know for which lectures my assistance is required
Scheduling	*5*	Student	have all scheduling information for my courses in one planner	I can take all courses into account when scheduling my time
Communication	*1*	Student	privately message my teacher or teacher assistants	I can ask clarifications about the course on days when there is no lecture
Communication	*2*	Student	post questions or remarks in a public chat	all other users can see and weigh in with possible answers
Communication	*3*	Student	privately message my colleagues	I can communicate with colleagues in the learning environment
Communication	*4*	Teaching assistant	be clearly identifiable as an assistant when posting messages	students know I am one of the assistants instead of a colleague
Communication	*5*	Teacher	make contact information about me and teaching assistants easily accessible	students know who to contact when they have questions
Recordings	*1*	Teacher	post recorded lectures online	my students can replay the recording of a lecture at home
Recordings	*2*	Teacher	be aware of the acceptable formats for uploading recorded lectures	the lecture recording is suitable for replaying within the learning environment
Recordings	*3*	Student	watch lecture recordings within the learning environment	I can catch up when I have missed a lecture due to illness
Recordings	*4*	Teacher	attach notes to the recordings at certain timestamps	I can provide clarifications to my lecture where necessary
Recordings	*5*	Student	answer simple multiple choice questions attached to the recordings	I can immediately apply the knowledge I have gained from listening to the lecture
Groups	*1*	Teacher	message all student groups	I can share clarifications about the assignments
Groups	*2*	Student	request to join a group	I can find the fellow students with whom I collaborate on an assignment
Groups	*3*	Student	create a new group	I can make a collaboration hub for my assignment partners
Groups	*4*	Student	send group invites	I can add my assignment partners to a group I have created
Groups	*5*	Student	accept group invites	I can become a member of a group

	Atomic		
	Role	**Action**	**Benefit**
Assignments	Teacher	create an assignment in the learning environment and in Google Drive	other external services are not necessary
Assignments	Teacher assistant	see the list of submissions and the list of exams that I was assigned to grade	I know how much grading work is ahead of me
Assignments			
Assignments			
Tests			
Tests			
Tests			
Tests			
Accounts			
Accounts			
Accounts			
Accounts			
Course material	Student	have access to the literature and content list(s), as well as my grades and profile information	I know what my teacher expects me to study for the exam(s)
Course material			
Course material			
Scheduling			
Scheduling			
Scheduling			
Communication	Student	post questions or remarks in a public chat, as well as privately message my tutors and peers	all other users can see and weigh in with possible answers
Communication			
Communication			
Recordings			
Recordings			
Recordings			

Minimal		
Role	**Action**	**Benefit**
Teacher	delegate grading rights (including read, write, export) to my teacher assistants	I can reduce my grading load
Teacher	restrict the file submission types	I can avoid additional checks. For example, in the past, we had problems with pdfs
Teacher	upload the lists (such as content, grades etc.) to the learning environment	my students know what I expect them to study for the exam(s)
Student	privately message my colleagues	I can communicate with them (using tools such as PM, DM, email etc.)

Full-sentence		
Role	**Action**	**Benefit**
Teacher	anonymous quiz for my students	add results
Teacher	create a graded test for my students	grades
Student	the literature list of the course(s) in which I am enrolled	I know what my teacher expects me to study for the exam(s)
Teaching assistar	assistant	one of the assistants

Problem Oriented		
Role	**Action**	**Benefit**
Student	see the test scores of my colleagues in an Excel spreadsheet	I can compare my performance against theirs
Student	receive notifications of test deadlines on my Android phone via the push mechanism	I do not forget which tests I need to complete
Teacher	create a course schedule within the learning environment	my students and teaching assistents can pull the schedule into their online agenda
Teacher	attach notes in the form of popups to the recordings at certain timestamps	I can provide additional information where necessary

Unambiguous		
Role	**Action**	**Benefit**
Student	create an account	I have access
Teacher assistant	receive access to grading functionalities	I can perform my duties
Teaching assistant	get a notification when I am required to assist	I know when assistance is required
Student	watch the recordings	I can review the material at home

Independent		
Role	**Action**	**Benefit**
Student	have access to my profile (which contains information as described in US7)	I can check it for incorrect information
Teacher	have insight into the current list of students with accounts	I know if all my students have created an account (as seen in attachment b.)
Student	have all scheduling information (using specified format, see above) in one location	I can take all courses into account when scheduling my time
Teacher	attach notes (see US2) to the recordings at certain timestamps	I can provide additional information where necessary

		Userstory		
Epic title	**Key**	**Role**	**Action**	**Benefit**
Groups	*1*	*Teacher*	*message all student groups*	*I can share clarifications about the assignments*
Groups	*1*	*Teacher*	*message all student groups*	*I can share clarifications about the assignments*
Groups	*1*	*Teacher*	*message all student groups*	*I can share clarifications about the assignments*
Groups	*2*	*Student*	*request to join a group*	*I can find the fellow students with whom I collaborate on an assignment*
Groups	*2*	*Student*	*request to join a group*	*I can find the fellow students with whom I collaborate on an assignment*
Groups	*2*	*Student*	*request to join a group*	*I can find the fellow students with whom I collaborate on an assignment*
Groups	*3*	*Student*	*create a new group*	*I can make a collaboration hub for my assignment partners*
Groups	*3*	*Student*	*create a new group*	*I can make a collaboration hub for my assignment partners*
Groups	*3*	*Student*	*create a new group*	*I can make a collaboration hub for my assignment partners*
Groups	*4*	*Student*	*send group invites*	*I can add my assignment partners to a group I have created*
Groups	*4*	*Student*	*send group invites*	*I can add my assignment partners to a group I have created*
Groups	*4*	*Student*	*send group invites*	*I can add my assignment partners to a group I have created*
Groups	*5*	*Student*	*accept group invites*	*I can become a member of a group*
Groups	*5*	*Student*	*accept group invites*	*I can become a member of a group*
Groups	*5*	*Student*	*accept group invites*	*I can become a member of a group*

Acceptance Test		
Given	**When**	**Then**
a list of groups within the learning environment	I click on a group	I should get the option to send feedback and/or remarks to all group members at once
a list of groups within the learning environment	I click on a group	I should get the option to send feedback and/or remarks to specific group members
a group within the learning environment	I message all group members	all group members should get a notification when one of them replies
a group within the learning environment	I click on the group	I should be able to see the group members
a group within the learning environment	I see that this group contains my group members	I should be able to send a join request
a group within the learning environment	I have sent a join request	I should get a confirmation notification
the learning environment	I have an account within the learning environment	I should be able to create a group
a group within the learning environment	I am creating the group settings	I should be able to set the group description
a group within the learning environment	I have created this group	I should have access to an admin page concerning this group
a group within the learning environment	I receive a join request	I should be able to reject this request
a group within the learning environment	I receive a join request	I should be able to accept this request
a group within the learning environment	I find another student that belongs in my team	I should be able to send a join request
the learning environment	I go to my account	I should be able to access a list of groups
the learning environment	I go to my account	I should see who amongst the students is not yet a member of any groups
a deadline I set on group joining	a student has not yet joined any groups after this deadline	I should receive a notification

D

APPENDIX: JSON DATABASE FORMAT

```
{
  "Title": "Online Learning Environment",
  "Epics": {
    "Number": 8,
    "Epics List": [
      {
        "Epic Title": "Assignments",
        "Epic Text": "Teachers need to set up assignments for the
             students to complete, often in groups."
      },
      {
        ...
      }
    ]
  },
  "Userstories": {
    "Number": 40,
    "Userstories": [
      {
        "Epic Title": "Assignments",
        "Role": "Teacher",
        "Action": "create an assignment in the learning
            environment",
        "Benefit": "other external services are not necessary"
      },
      {
        ...
      }
    ],
  },
  "Mistakes": {
    "Atomic": [
      {
        "Epic Title": "Assignments",
        "Role": "Teacher",
        "Action": "create an assignment in the learning
            environment and in Google Drive",
        "Benefit": "other external services are not necessary"
      },
      {
        ...
      }
    ],
    "Minimal": [
      {
```

```
            "Epic Title": "Assignments",
            "Role": "Teacher",
            "Action": "delegate grading rights (including read, write
                , export) to my teacher assistants",
            "Benefit": "I can reduce my grading load"
47        },
          {
            ...
          }
        ],
52      "Full Sentence": [
          {
            "Epic Title": "Tests",
            "Role": "Teacher",
            "Action": "anonymous quiz for my students",
57          "Benefit": "add results"
          },
          {
            ...
          }
62      ],
        "Problem Oriented": [
          {
            "Epic Title": "Tests",
            "Role": "Student",
67          "Action": "see the test scores of my colleagues in an
                Excel spreadsheet",
            "Benefit": "I can compare my performance against theirs"
          },
          {
            ...
72        }
        ],
        "Unambiguous": [
          {
            "Epic Title": "Accounts",
77          "Role": "Student",
            "Action": "create an account",
            "Benefit": "I have access"
          },
          {
82          ...
          }
        ],
        "Independent": [
          {
87          "Epic Title": "Accounts",
            "Role": "Student",
            "Action": "have access to my profile (which contains
                information as described in US7)",
            "Benefit": "I can check it for incorrect information"
          },
```

```
92        {
            ...
          }
        ]
      },
97    "Acceptance Tests": {
        "Number": 15,
        "Acceptance Tests": [
          {
            "Epic Title": "Groups",
102         "usKey": 1,
            "Role": "Teacher",
            "Action": "message all student groups",
            "Benefit": "I can share clarifications about the
                assignments",
            "Given": "a list of groups within the learning
                environment",
107         "When": "I click on a group",
            "Then": "I should get the option to send feedback and/or
                remarks to all group members at once"
          },
          {
            ...
112       }
        ]
      }
    }
```

E

APPENDIX: PRE- AND POSTTEST

E.1 PRETEST

E.1.1 *User stories*

1. As a student, I want to see my grades on a resizable pop-up window [use Javascript's standard function]
2. As an instructor, I want to update the list of students and insert the list of grades, so that I can easily manage the course I am teaching
3. As an administrator, I want to add new students to a course by clicking on the top-right "Add" button (corresponding stored procedure: addUserToCourse)
4. As a student, I want to be able to manage the courses I am attending in the current period, and to export their syllabus
5. I want to print the list of students who are attending the course I am teaching
6. As a designer, I want to be able to script levels and test them, so that I can improve my performance
7. As a tester, I want to have access to everything, so that I can comprehensively test the game with the toolsets addressed in earlier documents
8. As a player, I would like to finish the game quickly and share the results with my friends
9. As a player, I want game
10. As a developer, I want to access variables at runtime so that I can easily adjust variables during testing

	WELL FORMED	ATOMIC	MINIMAL	PROBLEM ORIENTED	UNAMBIGUOUS	INDEPENDENT	FULL SENTENCE	CONCEPTUALLY SOUND
US1			x	x				
US2		x						
US3			x	x				
US4		x						
US5	x							
US6		x						
US7					x	x		
US8		x			x			
US9							x	
US10								

Table E.1: Options and answers - pretest

E.2 POSTTEST

E.2.1 *User stories*

1. I would like to import the student grades from an external spreadsheet format, so that I can more efficiently calculate the final grade from the individual course components
2. As an administrator, I want to grant observer users access to a given course website (this was requested by the educational manager), so that the exam committee members can efficiently perform their duties
3. As a student assistant, I want to see, print, and export the relevant contents for me
4. As an instructor, I want to receive a notification when a student adds a comment to an existing page, so that I can quickly interact digitally with the students
5. As a veteran player, I want to edit my character's appearance using Blender (version 2.70 or above), so that I can reproduce my own appearance
6. As a developer, I want to use Microsoft Visual Studio
7. I want to customize the appearance and set the initial skills of my character
8. As a developer, I want to save my stuff on Dropbox (we should have a corporate license!), so that I avoid losing my work
9. update profile
10. As a developer, I want to edit my level files (using tools as described in us5) so that I can create new level files

E.2.2 *UES-SF*

1. FA-S.1 I lost myself in this experience.
2. FA-S.2 The time I spent using BakeRE just slipped away.
3. FA-S.3 I was absorbed in this experience.
4. PU-S.1 I felt frustrated while using BakeRE.
5. PU-S.2 I found BakeRE confusing to use.
6. PU-S.3 Using BakeRE was taxing.

	WELL FORMED	ATOMIC	MINIMAL	PROBLEM ORIENTED	UNAMBIGUOUS	INDEPENDENT	FULL SENTENCE	CONCEPTUALLY SOUND
US1	x							
US2			x					
US3		x						
US4								
US5			x	x				
US6				x				
US7	x	x						
US8			x	x	x			
US9							x	
US10						x		x

Table E.2: Options and answers - posttest

7. AE-S.1 BakeRE was attractive.
8. AE-S.2 BakeRE was aesthetically appealing.
9. AE-S.3 BakeRE appealed to my senses.
10. RW-S.1 Using BakeRE was worthwhile.
11. RW-S.2 My experience was rewarding.
12. RW-S.3 I felt interested in this experience.

F

APPENDIX: RESULTS WORKSHOP BAKERE

See the next pages for the database containing the results of the workshop using BakeRE

Participant	Gender	Attended Lecture 08-05	Reviewed Material	Prior Experience	Flipped	Pre (adjusted)	Post	Difference	Progress	UES	FA	PU	AE	RW	Session	Levels completed
1	Male	Yes	No	I have used them in a project	Yes	6	6,25	0,25	Yes	3,5	3,67	2	4	4,33	1	4
2	Female	Yes	Yes	I have used them both in a course and in practice	No	7	7	0	No	2,58	3,33	2,33	1,67	3	2	4
4	Female	No	Yes	I have used them in a course, but not recently: more than 6 months ago	No	7	7	0	No	3,08	3	3	3,33	3	2	1
5	Male	Yes	No	I have used them in a project	No	6,5	5,25	-1,25	No	2,83	3	1,67	3,67	3	1	2
6	Female	No	No	I have used them in a course, but not recently: more than 6 months ago	Yes	5,75	7	1,25	Yes	3,5	3,67	2,67	4,33	3,33	1	0
7	Male	Yes	No	I have used them in a course, but not recently: more than 6 months ago	No	6,75	4,75	-2	No	3,42	3	3	4	3,67	1	3
8	Male	Yes	No	I have used them both in a course and in practice	Yes	5,75	7,75	2	Yes	3,67	3	3,67	4,33	3,67	1	4
9	Male	Yes	No	I have used them in a course, but not recently: more than 6 months ago	Yes	5,75	5,75	0	No	3,92	4,33	4	3,67	3,67	1	4
10	Male	Yes	No	I have used them in a recent course	No	8,25	7,5	-0,75	No	2,58	3	2	2,67	3,67	1	4
11	Male	No	No	I have used them in a project	No	7	7,5	0,5	Yes	3,42	4,33	3,33	2,67	3,33	2	3
12	Male	Yes	No	I have used them in a recent course	No	7,75	8,25	0,5	Yes	2,5	2,67	2	2	3,33	1	2
13	Male	Yes	No	I have used them in a project	Yes	5,5	5,75	0,25	Yes	1,42	1,67	1,67	1	1,33	1	3
14	Male	Yes	No	I had no experience before the lecture	No	5,25	6,25	1	Yes	2,58	3	2,67	2	2,67	1	1
15	Female	No	Yes	I had no experience before the lecture	No	9	7,5	-1,5	No	2,83	4	2,67	2	2,67	2	4
16	Male	Yes	No	I have used them in a course, but not recently: more than 6 months ago	No	7,75	6,75	-1	No	3,08	3,33	2,67	2,67	3,67	1	4
17	Male	Yes	Yes	I had no experience before the lecture	No	7,75	7	-0,75	No	2,83	3,67	1,67	3	3	1	4
18	Female	Yes	No	I have used them in a course, but not recently: more than 6 months ago	Yes	5,25	4,5	-0,75	No	2,83	2,67	2	3,67	3	2	2
19	Male	No	No	I have used them in a course, but not recently: more than 6 months ago	No	6,5	7,75	1,25	Yes	2,42	2	2	2,67	3	2	2
20	Male	No	No	I have used them in a course, but not recently: more than 6 months ago	No	6,75	8	1,25	Yes	3,17	3,67	2,33	3,33	3,33	2	3
21	Female	Yes	No	I have used them both in a course and in practice	Yes	6,25	5,25	-1	No	2,25	2,33	1,67	2,33	2,67	1	4
22	Female	Yes	No	I have used them both in a course and in practice	Yes	5,75	6,25	0,5	Yes	2,08	1	3,67	2,33	1,33	2	4
23	Male	No	No	I had no experience before the lecture	No	8,25	7,5	-0,75	No	3	1,67	2,67	4,33	3,33	1	2
24	Female	No	Yes	I had no experience before the lecture	No	7,75	8	0,25	Yes	3,42	4,33	2,33	3,67	3,33	2	3
25	Male	Yes	No	I have used them both in a course and in practice	Yes	6,75	6,5	-0,25	No	3,42	3	2,67	4	4	1	1
26	Male	Yes	No	I had no experience before the lecture	No	8,25	6,5	-1,75	No	3	2	2,67	4	3,33	2	4
27	Male	No	Yes	I have used them in a project	Yes	5,25	6	0,75	Yes	2,42	2,67	2,33	2,33	2,33	2	1
28	Male	Yes	No	I have used them in a recent course	No	4,5	6,5	2	Yes	3,58	4	2,33	4	4	1	2
29	Female	Yes	No	I had no experience before the lecture	Yes	6,5	7	0,5	Yes	3,08	3,33	1,67	4	3,33	1	3
30	Female	Yes	No	I had no experience before the lecture	No	8,5	8,75	0,25	Yes	3,25	3,67	2	3,33	4	1	3
31	Male	No	No	I have used them in a course, but not recently: more than 6 months ago	Yes	5,25	5	-0,25	No	3,67	4,33	2,67	4,33	3,33	2	0
32	Male	Yes	No	I have used them in a course, but not recently: more than 6 months ago	No	6,75	5,75	-1	No	3	2	3	4	3	1	2
33	Female	Yes	Yes	I had no experience before the lecture	Yes	6,5	4,75	-1,75	No	3,58	4,67	1	4,33	4,33	1	2
34	Male	Yes	No	I have used them in a project	Yes	7	7	0	No	3,83	4	3,33	4,33	3,67	1	4
35	Male	Yes	Yes	I have used them in a course, but not recently: more than 6 months ago	Yes	5	6,5	1,5	Yes	3,5	3,33	3	3,67	4	12	0
36	Female	Yes	No	I have used them in a course, but not recently: more than 6 months ago	No	7,25	5,5	-1,75	No	3,83	3,33	3,67	4	4,33	1	1
37	Female	Yes	No	I have used them in a recent course	Yes	5,75	5,75	0	No	3,5	3,33	3,33	4	3,33	1	3
38	Female	Yes	Yes	I had no experience before the lecture	No	5	5,5	0,5	Yes	3,75	4,67	2	4,33	4	12	0
39	Female	Yes	No	I have used them both in a course and in practice	No	6,75	4,75	-2	No	2,92	2	2,67	3,33	3,67	2	2
40	Male	No	Yes	I have used them in a project	No	7,5	8	0,5	Yes	4,08	4,67	3,33	4	4,33	2	4
41	Male	Yes	No	I have used them in a recent course	Yes	7	8,25	1,25	Yes	3	2,67	2,67	3,67	3	1	4
42	Male	No	No	I have used them in a course, but not recently: more than 6 months ago	No	6	6	0	No	3,42	4	3,67	3	3	2	4
43	Male	Yes	Yes	I have used them both in a course and in practice	No	8	8,25	0,25	Yes	3,08	2,67	2,67	3,33	3,67	1	2
45	Male	No	No	I have used them in a course, but not recently: more than 6 months ago	No	5,75	3,75	-2	No	2,58	3,33	2,67	2	2,33	2	0
46	Male	No	Yes	I have used them in a recent course	Yes	5	6	1	Yes	3,83	3,67	4	3,67	4	2	1
47	Male	Yes	No	I had no experience before the lecture	No	7,5	8,5	1	Yes	3,75	4,67	2,33	4	4	1	4
48	Female	Yes	Yes	I have used them both in a course and in practice	Yes	5,75	6,75	1	Yes	3,08	2,33	3,67	3,33	3	2	4
49	Male	No	Yes	I have used them in a course, but not recently: more than 6 months ago	Yes	6	7,25	1,25	Yes	3,75	4	2,67	4,33	4	2	3
50	Male	No	No	I had no experience before the lecture	Yes	6,25	7,5	1,25	Yes	3,75	4,33	2,67	4	4	2	4
51	Male	Yes	No	I have used them in a recent course	No	7,75	7	-0,75	No	3,42	2,33	3,67	3,67	4	2	2
52	Male	Yes	No	I have used them in a recent course	No	6,5	7,25	0,75	Yes	3,42	3,67	3	3	4	1	4
55	Male	No	Yes	I have used them in a project	Yes	5	5,25	0,25	Yes	3,25	2,33	3,33	3,33	4	2	4
56	Male	Yes	No	I have used them in a project	No	6,5	6	-0,5	No	1	1	1	1	1	2	1
57	Male	Yes	Yes	I had no experience before the lecture	Yes	6,75	6	-0,75	No	2,67	2,67	2,67	2,67	2,67	2	0
58	Female	Yes	Yes	I had no experience before the lecture	No	6,25	7,5	1,25	Yes	3,67	4	3,33	3,67	3,67	2	2
60	Male	No	No	I have used them in a project	Yes	5,25	6,25	1	Yes	3,67	5	3	2,67	4	1	1
63	Male	Yes	No	I had no experience before the lecture	Yes	7	4,25	-2,75	No	2,83	2,67	3	3,33	2,33	2	1
64	Male	Yes	Yes	I have used them in a project	No	6,5	6,25	-0,25	No	3,75	3,67	3	4,33	4	2	4
65	Male	No	Yes	I have used them both in a course and in practice	No	8	8,25	0,25	Yes	3,58	3,67	3,33	3,67	3,67	2	4

Crashed?	Total Score	Level 1	Level 2	Level 3	Level 4	Ttut	T1-1	T1-2	T1-3	T1-4	T1-5	T2-1	T2-2	T2-3	T2-4	T2-5	T2-6	T2-7	T3-1	T3-2	T3-3	T3-4	T3-5	T3-6	T3-7	T4-1	T4-2	T4-3	T4-4	T4-5	Total time
No	**9958**	2995	2286	2291	2386	51	38	35	54	52	88	68	56	48	58	77	96	93	27	31	29	34	41	49	55	36	22	29	19	34	1220
No	**7816**	2954	1124	1846	1892	194	44	66	80	65	96	103	71	22	29	83	53	61	25	23	24	45	44	68	45	26	23	38	21	45	1394
No	**2948**	2948	-	-	-	52	135	52	73	78	76	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**4114**	3348	766	-	-	60	43	77	56	55	74	76	53	29	105	71	206	95	-	-	-	-	-	-	-	-	-	-	-	-	-
Yes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**5497**	2915	1754	828	-	41	35	37	89	69	93	53	23	32	40	44	44	76	27	49	41	58	30	29	238	-	-	-	-	-	-
No	**10425**	3453	2138	2467	2367	64	39	33	56	56	47	45	24	22	33	29	61	72	59	60	46	81	54	70	112	34	28	26	24	20	1195
No	**9000**	3304	1666	1677	2353	113	45	54	46	64	58	64	41	29	57	71	64	47	80	42	40	47	37	31	108	19	30	48	36	29	1300
No	**8457**	3393	1454	1880	1730	55	38	39	63	55	72	36	25	59	80	67	67	56	48	29	53	31	51	44	66	55	23	90	36	40	1278
No	**6247**	3219	1975	1053	-	67	58	30	84	41	54	104	56	33	33	41	54	46	53	36	42	69	32	43	339	-	-	-	-	-	-
No	**4237**	3065	1172	-	-	91	67	40	57	67	81	113	60	76	56	96	74	170	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**5469**	3106	1496	867	-	34	31	31	41	63	85	39	47	29	67	97	36	63	38	24	95	27	43	156	68	-	-	-	-	-	-
No	**2431**	2431	-	-	-	85	59	68	92	230	90	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**10593**	3480	2537	2224	2352	32	29	27	46	40	47	28	20	27	38	26	21	19	49	71	48	72	49	31	121	17	14	44	23	24	963
No	**9761**	3470	2232	2068	1991	44	36	38	56	43	58	61	29	23	30	29	51	34	52	25	51	24	22	42	46	19	19	59	21	37	949
No	**8619**	2879	2085	1688	1967	66	56	38	97	51	76	40	42	29	40	52	40	39	73	22	49	43	25	40	83	37	21	94	24	39	1216
No	**3987**	2993	994	-	-	148	64	58	70	81	111	39	42	56	33	50	63	80	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**4482**	3026	1456	-	-	85	79	34	71	61	108	111	55	41	62	50	71	43	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**4240**	2816	663	761	-	62	45	82	73	50	93	26	29	27	35	199	134	64	42	108	78	50	99	52	80	-	-	-	-	-	-
No	**9346**	3098	2149	1872	2227	69	46	55	37	39	82	65	37	29	37	27	24	22	57	25	21	21	27	29	60	44	14	30	20	34	951
No	**8716**	3222	1620	1490	2384	31	32	39	88	34	89	36	26	30	55	23	114	66	55	40	56	98	100	119	190	28	15	61	18	28	1471
No	**4547**	2809	1738	-	-	50	38	50	88	64	125	44	47	59	75	51	168	51	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**6101**	2774	1695	1632	-	56	65	76	95	198	95	50	49	35	33	71	79	35	38	67	67	51	37	58	224	-	-	-	-	-	-
No	**2405**	2405	-	-	-	45	88	540	63	106	70	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**7966**	3146	1115	1261	2444	46	43	36	80	73	66	54	73	36	98	35	85	60	42	32	46	48	85	39	84	55	17	37	28	31	1329
No	**2819**	2819	-	-	-	45	135	62	63	71	105	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**4330**	2808	1522	-	-	55	77	53	75	81	129	71	74	47	39	85	128	58	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**5761**	2977	1944	840	-	43	51	51	53	60	104	28	52	41	57	65	72	27	50	54	36	81	111	72	84	-	-	-	-	-	-
No	**6162**	3031	1797	1334	-	44	32	55	55	73	79	50	47	35	57	28	48	84	50	44	81	21	35	39	190	-	-	-	-	-	-
Yes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**3823**	2606	1217	-	-	129	58	40	116	68	114	52	27	38	53	88	122	95	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**4449**	2980	1469	-	-	51	55	29	79	49	104	152	42	36	44	85	98	26	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**9141**	3564	2130	1158	2289	39	27	34	49	32	56	28	35	28	67	29	39	52	20	30	48	38	24	104	83	40	17	25	23	98	1065
No	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**2821**	2821	-	-	-	97	148	152	196	198	444	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**5988**	2485	1792	1711	-	50	41	204	56	66	74	44	48	52	54	28	51	68	48	34	41	40	146	65	35	-	-	-	-	-	-
No	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**3975**	2669	1306	-	-	66	54	44	80	49	115	37	37	23	89	31	49	55	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**7642**	2833	1355	1444	2010	50	43	74	62	84	128	57	96	48	38	85	50	32	76	38	49	39	61	38	151	30	27	97	38	83	1574
No	**9793**	3334	1886	2193	2380	49	37	43	54	85	54	36	37	22	50	48	54	55	37	45	42	42	26	27	46	84	23	44	27	31	1098
No	**8971**	3103	1753	1924	2191	69	74	48	70	58	73	89	74	70	74	107	129	139	25	30	34	34	27	97	36	34	17	37	44	48	1537
No	**4326**	2900	1426	-	-	200	45	63	106	137	69	50	103	44	75	62	107	107	-	-	-	-	-	-	-	-	-	-	-	-	-
Yes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**2613**	2613	-	-	-	108	119	92	129	101	141	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**9051**	3333	1659	1684	2375	98	36	36	65	54	59	38	35	28	89	27	57	69	27	36	47	41	76	23	57	58	30	40	30	35	1191
No	**9984**	3408	2464	1916	2196	33	29	36	43	38	62	110	57	79	49	87	71	53	43	33	33	74	30	23	29	22	16	48	21	32	1151
No	**5103**	2993	946	1164	-	70	36	30	122	68	80	93	63	100	53	189	53	29	56	38	46	37	40	118	192	-	-	-	-	-	-
No	**7391**	3229	984	929	2249	49	36	37	50	62	64	118	72	44	48	70	95	116	116	31	40	125	68	53	113	27	19	54	25	32	1564
No	**3606**	2472	1134	-	-	177	41	103	85	60	178	80	72	61	60	41	126	58	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**8495**	3547	2209	1006	1733	46	31	31	49	39	59	43	22	21	22	20	31	79	38	31	30	43	69	70	178	76	58	48	49	50	1233
No	**8365**	3206	1273	1692	2194	60	49	72	67	58	102	32	65	29	123	37	91	84	68	59	67	44	41	48	56	27	18	59	23	31	1410
No	**2675**	2675	-	-	-	152	81	47	58	90	160	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Yes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**4005**	3224	781	-	-	47	42	53	67	68	90	66	194	32	92	321	103	48	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**3221**	3221	-	-	-	66	49	68	137	54	74	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**2855**	2855	-	-	-	59	115	57	84	80	71	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
No	**8782**	3308	1364	1867	2243	45	41	51	58	45	110	29	35	28	81	117	49	106	21	54	39	54	35	42	60	21	20	60	20	23	1244
No	**9440**	3149	2342	1895	2054	51	40	39	55	68	85	71	106	43	161	68	164	81	42	39	39	54	48	28	81	32	28	44	20	82	1569

BIBLIOGRAPHY

Alaoutinen, S. and K. Smolander (2010). 'Student Self-assessment in a Programming Course Using Bloom's Revised Taxonomy'. In: *ITiCSE'10*, pp. 155–159.

Anderson, L. W., D. R. Krathwohl and B. S. Bloom (2001). *A taxonomy for learning, teaching, and assessing : a revision of Bloom's taxonomy of educational objectives*. Longman (New York).

Atkinson, R.C. and R.M. Shiffrin (1968). 'Human memory: A proposed system and its control processes'. In: *The Psychology of Learning and Motivation* 2.

Baric, R., J. Vlasic and Sasa Cecic Erpic (2014). 'Goal Orientation and Intrinsic Motivation for Physical Education: Does Perceived Competence Matter?' In: *Kinesiology 46:117-126* 46, pp. 117–126.

Beatty, J. and M. Alexander (2008). 'Games-Based Requirements Engineering Training: An Initial Experience Report'. In: *Proceedings of the 16th IEEE International Requirements Engineering Conference*, pp. 211–216.

Beck, K. et al. (2001). *The Agile Manifesto*. http://www.agilemanifesto.org/.

Beckers, K. and S. Pape (2016). 'A serious game for eliciting social engineering security requirements'. In: *International Requirements Engineering Conference*, pp. 16–25.

Bloom, B. S., M. D. Engelhart, E. J. Furst, W. H. Hill and D. R. Krathwohl (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain*. David McKay Company (New York).

Bogdanova, D. and M. Snoeck (2017). 'Domain Modelling in Bloom: Deciphering How We Teach It'. In: *The Practice of Enterprise Modeling*, pp. 3–17.

Brabeck, M., J. Jeffrey and S. Fry (2011). *Practice for knowledge acquisition (not drill and kill)*.

Cohn, M. (2004). *User Stories Applied*. 1st. Pearson Education (US).

Csikszentmihalyi, M. (1990). *User Stories Applied*. 1st. Harper and Row (New York).

Dalpiaz, F. and K.M.L. Cooper (2018). 'Games for Requirements Engineers: Analysis and Directions'.

Deterding, S., D. Dixon, R. Khaled and L. Nacke (2011). 'From Game Design Elements to Gamefulness: Dening Gamication'. In: *Proceedings of MindTrek: Envisioning Future Media Environments*, pp. 9–15.

Douglas, B., E. Tanner-Smith and S. Killingsworth (2016). 'Digital Games, Design, and Learning; A Systematic Review and Meta-Analysis'. In: *Rev Educ Res.* 86, 79—122.

Greitzer, F.L., O.A. Kuchar and K. Huston (2007). 'Cognitive Science Implications for Enhancing Training Effectiveness in a Serious Gaming Context'. In: *ACM J. Edu. Resources in Comput.* 7.3, pp. 1–10.

Hainey, T., Stansfield M. Connolly T.M. and E.A. Boyle (2011). 'Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level'. In: *Computers & Education* 56, pp. 21–35.

Huber Kolpondinos, M.Z. and M. Glinz (2017). 'Behind Points and Levels – The Influence of Gamification Algorithms on Requirements Prioritization'. In: *Proceedings of the IEEE Requirements Engineering Conference (RE)*, pp. 332–341.

Huynh-Kim-Bang B.and Wisdom, J. and J. Labat (2008). 'Design Patterns in Serious Games: A Blue Print for Combining Fun and Learning'. In:

IEEE recommended practice for software requirements specifications (1998). IEEE Computer Society.

Kassab, M. (2015). 'The changing landscape of requirements engineering practices over the past decade'. In: *Proceedings of the International Workshop on Empirical Requirements Engineering (EmpiRE) IEEE*, pp. 1–8.

Knauss, E., K. Schneider and K. Stapel (2008). 'A game for taking requirements engineering more seriously'. In: *International Workshop on Multimedia and Enjoyable Requirements Engineering*.

Kotovsky, K., J.R. Hayes and H.A. Simon (1985). 'Why are some problems hard? Evidence from the Tower of Hanoi'. In: *Cognitive Psychology* 17, pp. 248–294.

Lakoff, G. and M. Johnsen (2003). *Methaphors we live by*. The university of Chicago press (London).

Lamsweerde, A. van (2009). *Requirements Engineering, from System Goals to UML Models to Software Specifications*. 1st. John Wiley And Sons Ltd.

Lawrence, B., K. Wiegers and C. Ebert (2001). 'The Top Risks of Requirements Engineering'. In: *IEEE Software* Volume 18.6, pp. 62–63.

Leffingwell, D. (2010). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. 1st. Addison-Wesley Professional.

Lindland, O.I., G. Sindre and A. Sølvberg (1994). 'Understanding Quality in Conceptual Modeling'. In: *IEEE Software* 11 (2), pp. 42–49.

Lombriser, P., F. Dalpiaz, G. Lucassen and S. Brinkkemper (2016). 'Gamified requirements engineering: Model and experimentation'.

In: *Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 171–187.

Lucassen, G., F. Dalpiaz, J.M.E.M. van der Werf and S. Brinkkemper (2016). 'Improving Agile Requirements: The Quality User Story Framework and Tool'. In: *Requirements Engineering* 21, pp. 383–403.

Marne, B., J. Wisdom, B. Huynh-Kim-Bang and J.M. Labat (2012). 'The Six Facets of Serious Game Design: A Methodology Enhanced by Our Design Pattern Library'. In: *Proceedings of EC-TEL 2012: 21st Century Learning for 21st Century Skills*, pp. 208–221.

Mayer, R. E. (2004). 'Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction'. In: *American Psychologist* 59, pp. 14–19.

Michael, D. and S. Chen (2005). *Serious Games*. 1st. Course Technology PTR.

Millbower, L. (2003). *Show Biz Training: Fun and Effective Business Training Techniques from the Worlds of Stage, Screen, and Song*. AMACOM/American Management Association (New York).

Miller, G.A. (1956). 'The magical number, seven, plus or minus two: Some limits on our capacity for processing information'. In: *Psychological Review* 63, pp. 81–97.

Mouraz, A., A.M. Vale Pereira and R. Monteiro (2013). 'The Use of Metaphors in the Processes of Teaching and Learning in Higher Education'. In: *International Online Journal of Educational Sciences* 5, pp. 99–110.

North, D. (2006). 'Introducing Behaviour Driven Development'. In:

O'Brien, H.L., P. Cairns and M. Hal (2018). 'A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form'. In: *International Journal of Human-Computer Studies* 112, pp. 28–39.

Peters, A.K. (2014). *The Art of Game Design: A Book of Lenses*. 2nd. Ltd. Natick.

Pinto-Albuquerque, M. and A. Rashid (2014). 'Tackling the Requirements Jigsaw Puzzle'. In: *Proceedings of the 22nd IEEE International Requirements Engineering Conference (RE)*, pp. 233–242.

Regnell B.and Brinkkemper, S. (2005). 'Market-driven requirements engineering for software products'. In: *Engineering and managing software requirements*, pp. 287–308.

Rieber, L.P. and D. Noah (2008). 'Games, simulations, and visual metaphors in education: Antagonism between enjoyment and learning'. In: *Educational Media International* 45, pp. 77–92.

Rusu, A., R. Russell and R. Cocco (2011). 'Simulating the Software Engineering Interview Process using a Decision-Based Serious Computer Game'. In: *Proceedings of the 16th International Conference on Computer Games*, pp. 235–239.

Ryan, R.M. and E.L. Deci (2000). 'Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions'. In: *Contemporary Educational Psychology* 25, pp. 54–67.

Salen, K. and E. Zimmerman (2003). *Rules of Play: Game Design Fundamentals*. MIT Press.

Schneider, W. and R. Shiffrin (1977). 'Controlled and automatic information processing: I. Detection, search and attention'. In: *Psychological Review* 84, pp. 1–66.

Shiffrin, R. and W. Schneider (1977). 'Controlled and automatic information processing: II. Perceptual learning, automatic attending, and a general theory'. In: *Psychological Review* 84, pp. 127–190.

Sitzmann, T. (2011). 'A meta-analytic examination of the instructional effectiveness of computer-based simulation games'. In: *Personnel Psychology* 64, pp. 489–528.

Smith, R. and O. Gotel (2008). 'Gameplay to Introduce and Reinforce Requirements Engineering Practices'. In: *Proceedings of the 16th IEEE International Requirements Engineering Conference (RE)*, pp. 95–104.

Snijders, R., F. Dalpiaz, S. Brinkkemper, M. Hosseini, R. Ali and A. Özün (2015). 'REne: A Gamied Platform for Participatory Requirements Engineering'. In: *Proceedings of the 1st IEEE International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*.

Vogel, J. J., D. S. Vogel, J. Cannon-Bowers, C. A. Bowers, K. Muse and Wright M. (2006). 'Computer gaming and interactive simulations for learning: A meta-analysis'. In: *Journal of Educational Computing Research* 34, pp. 229–243.

Wang, X., L. Zhao, Y. Wang and J. Sun (2014). 'The role of requirements engineering practices in agile development: An empirical study'. In: *Proceedings of the Asia Pacic Requirements Engineering Symposium (APRES)* 432, pp. 195–209.

Wautelet, Y., S. Heng, M. Kolp and I. Mirbel (2014). 'Unifying and Extending User Story Models'. In: *Proceedings of the 26th International Conference on Advanced Information Systems Engineering*.

Wieringa, R.J. (2014). *Design Science Methodology*. 1st. Springer, Berlin, Heidelberg.

Wouters, P., C. van Nimwegen, H. van Oostendorp and E.D. van der Spek (2013). 'A meta-analysis of the cognitive and motivational effects of serious games'. In: *Journal of Educational Psychology* 105.2, pp. 249–265.

Zowghi, D. and S. Paryani (2003). 'Teaching requirements through roleplaying: lessons learned'. In: *Proceedings of the 11th IEEE International Requirements Engineering Conference (RE)*.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/