Bachelor’s Thesis

Joonas Heinonen

BROADENING SOFTWARE ENGINEERING PROGRAM CONTENT WITH SELF-STUDY DIGITAL MATERIAL

Examiners: Professor Ajantha Dahanayake

Work mentor: Professor Ajantha Dahanayake
TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Engineering Science
Tietotekniikan koulutusohjelma

Joonas Heinonen

Digitaalisten itseopiskelu opintokokonaisuuksien hyödyntäminen ohjelmistotuotannon koulutusohjelman tarjonnan kehittämiseksi

Kandidaatintyö

2018

38 sivua, 15 kuvaa, 1 taulukko

Työn tarkastajat: Professori Ajantha Dahanayake

Hakusanat: moodle, opettaminen, ohjelmistokehitys, e-oppiminen, e-kurssi

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Joonas Heinonen

Broadening study program content with digital material

Bachelor’s Thesis

38 pages, 15 figures, 1 tables
Examiners:  Professor Ajantha Dahanayake
Keywords: moodle, education, software-development, e-learning, e-Course

The object of this thesis is to investigate how digital materials fit in degree program in software engineering and how these materials can be used to broaden the selection of software development courses. The materials were evaluated by their contents and availability. The availability of the materials was aimed to be free and the content to match the needs of software development. The course was held using Moodle-environment. The outcome of this thesis was a working moodle course, which can be utilized to improve the contents of LUT’s software engineering degree program for future developers.
PREFACE

This work would not be possible without the continuous support of my wife. I’m also very grateful for having such a great tutors, mentors and teachers on my way to become a bachelor graduate. This thesis would not have been possible without the financial support of LUT’s engineering science department. I would like to also express my gratitude to LUT staff for the support I got while working with this thesis.
TABLE OF CONTENTS

1 INTRODUCTION ........................................................................................................... 4
   1.1 OBJECTIVES AND QUESTIONS ........................................................................... 4
   1.2 LIMITATIONS ....................................................................................................... 4
   1.3 THESIS STRUCTURE ............................................................................................ 5
   1.4 INFORMATION GATHERING .................................................................................. 5

2 THEORETICAL BACKGROUND ...................................................................................... 7
   2.1 EARLIER RESEARCH ON SWE EDUCATION ....................................................... 7
   2.2 LEARNING PERFORMANCE IN SELF-STUDY ..................................................... 8
   2.3 EDUCATION MATERIAL ASSESSMENT ............................................................... 8
   2.4 CS AND SWE EDUCATION AS SELF-STUDY .................................................... 9
   2.5 ONLINE-PROGRAMMING LEARNING ................................................................. 9

3 DESIGN PHASE ............................................................................................................ 11
   3.1 RESOURCES ....................................................................................................... 11
   3.2 MOODLE COURSE ............................................................................................... 12
   3.3 MODULES ........................................................................................................... 13
   3.4 WORKFLOW ....................................................................................................... 14
   3.5 STUDENT WORKFLOW ....................................................................................... 14
   3.6 ASSIGNMENT ASSESSMENT ............................................................................. 18
   3.7 EVALUATION COMPONENTS ............................................................................. 19

4 IMPLEMENTATION PHASE ............................................................................................ 21
   4.1 MOODLE AS A PLATFORM .................................................................................. 21
   4.2 VERSION CONTROL ............................................................................................. 21
   4.3 CODE IN MOODLE .............................................................................................. 21
   4.4 IMPLEMENTATION RESULTS ............................................................................. 22

5 TESTING PHASE .......................................................................................................... 27
   5.1 CONTENT IS UP TO DATE .................................................................................. 27
   5.2 VALIDITY ............................................................................................................. 27
# SYMBOLS AND LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>JS</td>
<td>JavaScript</td>
</tr>
<tr>
<td>SWE</td>
<td>Software Engineering</td>
</tr>
<tr>
<td>CS</td>
<td>Computer Science</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>GIT</td>
<td>A version control system</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

1.1 Objectives and questions

Digital education materials for software development are nowadays quite common (Kattimani & Naik 2012). There is a vast selection of different online platforms offering free or paid coding courses. The platforms and other sources offer a broad selection of programming projects, courses and code snippets ranging from beginner to advanced level. This thesis investigates how these platforms and digital materials can be used for academic education purposes in bachelor’s degree level for software engineering. The objective of this thesis is to find out whether these materials are fit to be used for development of the degree program in software engineering. In software development education, the approaches for teaching vary from math- to practical-heavy ways. In this thesis the assessed digital materials were combined as Moodle-course to have very practical-heavy orientation and easy to follow course structure. The thesis objectives aim to examine how the chosen orientation affects on course-attractiveness and how easy it is for students to follow the chosen materials.

The key concerns are how to assess the quality of the course materials without having a comprehensive report on the materials and first-hand experience of the course material usage in academic education. Assessing the source code without excessive knowledge of the language may be a challenge for the teacher in the course. The work can be assessed by the results of the program and how the code looks in general. The question remains for the accuracy of the assessing process. Does the general outline assessment of the source code relate to learning the technology adequately? Another problem is to validate the materials for the academic level. How can one measure the learning of many students without comprehensive resources? The answers for these questions are provided in the sections below.

1.2 Limitations

In the early phases of this thesis, a need for limitations emerged. The vastness of materials found free from online sources needed a proper filter to get the best possible outcome for academic learning purposes. Materials from online sources are very broad in terms of areas they cover in software development. The limitations for different technologies used in software development, is also a need for this thesis to be whole.

Limitations to which areas are taken into this thesis can be formed as the key needs in modern software development trends. In software development, there’s a vast amount of different areas. The areas can be divided roughly in half. There’s always a need for the backend and the frontend. In backend, the focus in development is more into the software core functionalities and the server side. In the frontend, the development work is more about the UI and interactions linked from the backend. There’s also a big need for developers capable of understanding the technologies used in all layers of software, they are called fullstack-developers (Tripp-Carillo et al. 2017).

The tools for modern web-development is a good starting point for the limitation process. Picking web-development as one of the limiting factors is based on the need from the
industry and the jobs offered for those who are fluent in modern web-technologies. In modern Web-based applications, the traditional approaches for software development needs to evolve or adjust to support the fast pace of emerging technologies. (Uikey & Suman 2015). Another popular topic in the software industry is mobile development. In the past, 2010 to 2014 the trend of developing mobile applications has risen remarkably and the trend is still thriving (Bilal et al. 2015). There are a lot of different ways to develop software in mobile. Differences between the development technologies are eminent (Păvăloaia & Cuza 2013). Assessing the criteria for a student, who has never developed in mobile, leads to selecting Android as the OS and Java as the development language. The selection is based on LUT’s degree program content, which has Java course included.

1.3 Thesis Structure

This thesis is about using theory as a background and making a practical use case of the learning process with digital materials. The use case has three sections: 1) design; 2) implementation; and 3) testing. In the last section, the thesis will wrap as conclusions are presented. In design part the discussion of how and why, is presented. The learning design and research in SWE education is taken into consideration in the design part. In implementation phase the practical work is conducted and the course is built on the Moodle-platform. The changes from design to implementation are discussed. In testing phase the digital materials of each module and other course components are tested. The layout, navigation and support of the course is done with constant feedback of each iteration. After conclusions the thesis will wrap with guidelines of how to continue the course development and how this kind of method can be applied to appeal to bigger audiences.

1.4 Information gathering

Gathering the necessary information from reliable sources is a key part of any thesis including this one. Information gathered in this thesis is found by doing research in the public sites for academic citations such as IEEE, ACM, Springer or similar databases. The area which this thesis is part of is vast. By gathering information from many sources with different views, one must pay even more attention to each source and the origin of the source. Information gathered needs some sort of systematic assessment for the best results to be possible. This systematic way is established by comparing sources with the information available. The comparing of sources included values like year published, citation count and relevance to SWE education or digital materials.

The information gathered about no-cost digital materials, which can be used in the degree program, is aimed to have a positive effect on the development of the degree program in software engineering. The effect can be seen in the skillsets that bachelor students have after they’ve acquired the technologies present in the course, meaning the technologies presented by external digital materials. The effect of broader understanding of popular technology usage through hands on experience may give the student a better chance to succeed in theoretical courses of the degree program. Learning in SWE education and in education in general is done by individuals who are motivated and eager to learn. Motivation can be enhanced by different methods including education gamification and
rewarding (Su 2016). This thesis gathers information about self-study-based learning and the applications for software development it offers. In this thesis, the gamification is bypassed and the focus is moved to research the usage of digital education materials. The “rewards” are given in the form of advice on self-made project usage, when applying for a job.

Offering SWE bachelor’s students a different perspective to learning compared to the usual software development courses available in the current SWE bachelor’s degree program, may give the students more motivation to learn coding by themselves. Clear informing of the benefits gained through the course, is a key part to motivate the participants. Understanding the weaknesses and strengths in self-studying can be done by gathering relevant information about use-cases of self-studying and the results of these cases.
2 THEORETICAL BACKGROUND

2.1 Earlier research on SWE education

The need for advanced computer sciences in university level will be a major problem in the future. The movement on SWE education on lower academy levels has increased globally. The education offered by many schools are based on external digital sources and teachers capable of computing. One of the major problems in SWE/CS education on lower academic is that the students are not mature enough to handle self-study learning. (Yadav et al. 2016)

In quite recent study, Reuter at al. (2017, p. 2) stated, that “The teacher can also be replaced by technology”. The statement was based on the idea that “knowledge can be transferred from teachers to learners” (Reuter at al. 2017). This thesis focuses on lightening the workload of a single instructor. The thesis does not aim on removing the responsibilities of a teacher. Removing the responsibilities of a teacher is possible in theory, but practical experience has shown human teachers being a guidance to the right direction. Currently the comprehensive role of a teacher, is hard to replace by automation or other methods completely.

Each individual learns differently. Students understand and create reality based on their view of the learning process. The experiences individuals have, shape the ways one learns and finds more suitable to self. The learning process develops as one matures. Finding the correct way of learning and conceiving information through open mindedness for different learning methods, is a key factor for improving the ability to learn.

The traditional way of organizing the structure in SWE education is challenging. The traditional education relies heavily on the financial, available space and time-related constraints. (Buffardi 2017). The current state of SWE education does not encourage the advantages of reusability or maintainability to appear for students. The projects made from a scratch in courses are usually very small compared to the projects a student encounters at the first job they get. (Buffardi 2017) mocks these scratch projects as toy-applications with “little use outside the classroom”. The projects students create in the Moodle course context on this thesis, aim to break the frame Buffardi has set. The projects done based on the materials assessed in this thesis will have more use outside the classroom as merits to display the skills one has in a certain technology. The course result can be used to enhance the chance of getting hired in IT-sector.

Understanding the usage of tools in SWE education and research is a key factor for better results in both fields. In every research field as in SWE research, the need for better tools to identify, manage and use information gathered is growing constantly. The importance of tools used in software development is very high. The functionalities and features the tools in software development have nowadays are a vital factor to be recognized. Taking a view on the SWE-education, the tools are an irreplaceable part to assist in the deep understanding of programming a student has. (Wolff 2017)

Personality can be combined with almost anything including SWE or education. Research has been done to gain an understanding of the interactions between humans, which
personality traits can be derived from. In a systematic mapping study conducted by Cruz, et al. (2015), the personality traits of software development related personnel were evaluated. Personality is very complex subject and it’s influences are hard to measure. The evaluation results were not of much use. More research on the personality subject is needed to have a conclusive result of the effects in SWE context. The important note is to understand, that personality can have huge impact on learning behavior in the SWE context.

2.2 Learning performance in self-study

Learning to write code and understanding a programming language or technology can be self-taught. Most of the brightest SWE/CS students among us are self-taught. They have the passion and mindset to reach their goals in a form of self-made projects and experiences. Generally, everyone including undergraduates, have a lot of free digital materials available for them. Open source projects, tools and platforms give plenty of room to learn as a developer without a remarkable financial capital. “The knowledge and implementation skills in CS can be easily digitalized and distributed online” (Voas et al. 2018). This quote represents the current state of SWE education development. The digital nature of CS and SWE is the driver for most of the education materials and project guides being digital from the beginning. Early digital availability offers tremendous potential when studying the learning performance in SWE or self-study.

In learning in general, it is important to have different aspects on the subject currently under study. Learning is not just storing information and unloading it when needed. Learning should be understood as an ability to manage demands that environmental changes forces. Research has been done to indicate that students learn better when they engage in projects, where the need is to use the current knowledge to solve more concrete problems. (Telsang & Kulkarni 2015) Student engagement in these projects is most likely done as self-work, which indicates that the results gotten from the study can be used to examine the performance in self-study in the SWE-context. The study shows, that concrete problems related SWE-self-work may have a high probability to enhance the learning performance of a student.

2.3 Education material assessment

There has always been the need to ensure students work being his or her own. Technology is having an impact to written work distribution and to the difficulty of finding cheaters. It is important for teachers to design a course in a way that reflects the learning from the materials. The idea of an assessment is to create some sort of indication, for the student and the teacher, so that the learning which has happened can be measured. (Cox et al. 2014)

The digital material availability is growing in the modern era. Different tools available establish teachers and content-makers to have the ability to create and distribute education content online. The materials include videos, blogs and forums, which can be utilized in an online learning environment such as Moodle. A study conducted by Chilukuri and Raju (2015) indicates that learning regardless from place or time is convenient and may result in better course-attendance. The study concludes to elevate teacher’s flexibility and
knowledge in different areas of teaching methods, including providence of online education materials.

2.4 CS and SWE education as self-study

Earlier the role of a teacher has been the key in education. The teacher has acted as a “dominator” or “controller” for the sake of learning. Self-study removes the teacher from the “dominator” context and rearranges it to a different view. In self-study based learning the teacher acts as a support personnel, ready to provide constructive advice for the students. The students will develop as learners and they will adapt better skills to understand more complex systems on their own. (Xie & Huang 2012, p. 378)

In SWE education various different methods for developing the education has been used. One of these methods were studies with a literature review about SWE education paired with utilization on gaming. Most of the games used in the study were digital. Games are usually operated by one person at a time. This means that in the study context, games in education can be considered as a method for self-studying. SWE education can’t be solely replaced with self-studying or game-based education, but a healthy amount of new methods added to the traditional face-to-face education may result in better education goals. (Mehmet et al. 2016)

2.5 Online-programming learning

Popularity has risen on online programming. In programming education based on an online platform, the quality assurance is a challenge. Massive Open Online Courses (MOOC) has a vast selection of different programming courses, especially in the introductory-level. Usually these courses have a huge student to instructor ratio, which results in problems not found in the traditional programming education done by face-to-face support and communication. Operations, such as course-grading or individual feedback are less likely to be accurate or even available on these platforms. (Li et al. 2016)

In online learning platforms attributes which origin from the role of a teacher, a substitute must be nominated to address the void a teacher’s absence provokes. A variety of different methods have emerged to address the teacher’s absence. One of these methods is to improve the ability to provide guidance on programming via automation. The idea Li et al. (2016) proposed, was to measure behavioral similarities between two programs. One program being the students work and the other being an example solution. The automated solution would give insights to student of how close the students work is to the example solution. The study shows that student learning performance may improve due to gaining more confidence in coding, when moving towards the example solution. Stopping the code solution earlier when moving further from the example solution may save time for the student and possibly avoid bad programming habits from forming. The information gathered in the study reflects the same challenges opposed in this thesis.

A study conducted by Shastri (2015) resulted to better hiring rates of fresh SWE-graduates by giving the students guidance on changing requirements of industry. This study shows,
that by taking industry changes in consideration on SWE-education, will result in graduates having a better skillset. An experience based assumption can be made, that in the current degree program at LUT university, the education of programming learning has remained moderately constant through the recent years. This is caused by various reasons which are not relevant for this thesis to go deeper into. The identification for the needs to broaden the programming courses and materials according to the changes in industry may have been discarded due to the unmentioned reasons. Research shows that having the impact of technologies changing in the industry resulting in affecting the SWE-degree program. The technology changes will assist the development of the degree program for the better, since the availability of different tools rises. The Moodle-course is one solution to address the changes in technologies within the degree program.
3 DESIGN PHASE

3.1 Resources

Project work always has limitations. In this case, the course material needed to be explicitly free, but also available and correspond to the current needs of a future software developer. The needs for a future software developer are assessed by researching the “hot” technologies and the technologies needed in different software industry jobs currently in Finland. The course design principles were heavily caused by the need to minimize resources and maximize value. The most optimal course execution seemed to be a self-study-based solution, since e-learning is usually done by individual effort with the given materials. The design principles followed similar aspects shown in MOOCs. MOOC aims to appeal to grand audience and bring knowledge available to the participants, however this course is not designed to have similar amount of participants as MOOCs, but still have similar learning outcomes. (Gergel & Kustikova 2016) Self-studying needs fewer resources, but the quality of the resources is highlighted. Because the quality of the material needs to be high, the assessment phase of digital materials takes a moderate amount of time.

When choosing the right materials, one must also understand the needs of a student as a learner. In business learning, customer centric business models are praised. (Shah et al. 2006) These models are praised, because that’s the audience which the salesperson wants to appeal the most. In education context, focusing a more student centric approach on the digital software development learning material choosing should lead to better understanding of development work overall.

Understanding the usage of resources leads to a better understanding of the selection for resources. Resources for software development purposes are used by professional developers each day. Professional developers are fluent in searching relevant information and resources for their current problem or case. In this thesis’s course, one of the main goals for a student is to learn how to search from external sources or find help when a student encounters a problem in the development work.

Developers in general, use a lot of different websites such as Stackoverflow for searching development related artifacts (Stackoverflow, 2018). Different search-engines are also commonly used to find other developers with similar problems as the searcher has. Using different search methods will often lead to finding common information for the solution a developer seeks. The other commonly used strategy for solving a software development related problem, is to look up the documentation for the technology currently present in the development. Applying these methods and guidelines to selecting materials and resources for teaching software will result in better suited materials for a software development student. Manner of an approach presented before will give the students the ability to understand the commonly used strategies in software development problem solving and tools to use these strategies with the resources given in the Moodle-course.
The resources and guidelines given in the course try not to just address learning objectives but to give more insight about what is currently important in the IT or software developing sector. Shastri (2015) has researched the skills which companies seek from fresh SWE-graduates. The study shows that few of the important skillsets needed in modern technology based companies, are self-learning and practical application skills in software development. A degree is not a clear indicator for person’s skillset anymore, nor it may have never been. Works indicating persons skills at problem solving and creativity in practical use cases may result in a higher change of getting a job than by SWE-degree. One of the goals in providing the course resources, is to change this perception. Combining theoretical understanding of SWE and the practical programming and problem solving projects will probably result in improvement on the valuation of SWE-degree.

The resource quality must be assessed with caution. When the Moodle-course is self-study oriented, the quality of the materials become even more important than before, since the resources used are the only source of information for learning. When choosing the right resources, the quality attributes need to be selected. The engineering student becomes a problem, when the needs are not clear. In this quality attribute choosing is based on the needs of a software engineering student. Addressing the needs of a software thesis, the needs for a student are moderately clear, which is sufficient enough to base the quality on those needs. The needs can be found from Section 3.5.

### 3.2 Moodle course

Moodle was chosen to be the platform in this thesis due to the usage of it in our university. Moodle as a platform is created as open source. Moodle has tremendous potential as a learning platform, but a lot of the potential is wasted by lecturers that are unfamiliar with the controls in Moodle. As a software engineering student operating Moodle, the controls become more familiar as time passes. The learning curve may be very different from a student who has used the platform to a teacher who has never been on the receiving end as a student. The controls in Moodle can be found on the settings section, using a search engine to refine a search, will result better usability in searching a setting. Various tutorials and other online resources are available for better Moodle usage, but some are highly outdated. The motivation for gaining progress in the platform usage is not that relevant for courses, which are not entirely based on the Moodle-platform. However, in this thesis’s course, the progress of the course structure, navigation and platform aspects must be considered, since the course content is only available online.

Research and development for using the teacher-tool in Moodle is needed, because the UI design in Moodle is rather inconvenient for those users who are not very familiar with software usage. The teacher in Moodle can add different components to the course or even code. By adding components and structure, the course page is being created. Setting the Moodle course up and running is essential to be able to modify the course pages. The necessary settings to have a tab-view for easier navigation can be found from Moodle documentation and is recommended for all courses based solely online.
Moodle has a lot of Plugins and add-ons. They are ideal for small faculties, but for university level, they are insuperable dependencies which can’t be used. A lot of research for Moodle-usage was done in the thesis context. The best solution for this thesis is having a navigation system and pages which can be operated. This was done by adding HTML-code in Moodle with inline CSS. Some dropdown functionality is added with JS to improve the content visibility.

Using code in Moodle was done by running the code in an editor with a preview for the page. This established a better and faster way to develop the course structure. The course pages can be created by pasting the source code to the selected section and submitting it. This is arguably the best way currently available for a software developer to create content in Moodle.

3.3 Modules

Discussion helps understand situations better and the need for new ideas. The initial module selection is based on unofficial discussions with the staff in SWE degree program. These discussions led to an understanding of which studies or technologies are missing from the bachelor’s degree program. The missing key-technologies are cyber-security, mobile-development and advanced web-development.

There are a lot of new technologies and a need for teaching them. A common problem for academic students is to do additional learning for no course credit (Su 2016). Depending on the ability to manage personal life and studies, a student is rarely able to find the motivation from just the pure value of learning to search and execute external developer courses or other digital material. The drivers for learning are either getting course credits or turning the learning into skills that profit money. Identifying the appropriate and valid digital material can be time consuming and hard. Even when considering the value of learning interesting technologies or having a self-made project, students lack the understanding and passion to write code.

The idea of technologies which could be used in modules originated from the need to bring something in to the degree program which is not there yet. The initial idea was to bring cyber-security, IoT, data-analysis and mobile development in the Moodle-course. After a discussion with experts from a local company and researching different technologies in software development, the ideal technologies presented as a package would not be the one’s mentioned earlier. The module package would be better when the technologies presented had common factors or areas. The module package is formed for young developers who are still seeking their passion in software development. In the degree program, courses offered do not have a perspective for front- or backend development. In the course part of this thesis, one of the ideas is to create opportunities for the developers to choose a passion they’ve yet to find. The module package is chosen to contain front- and backend in addition to full-stack in web and mobile.
3.4 Workflow

The design workflow followed the same guidelines as a good software development project has. The guidelines can be seen specifying the requirements, understanding the target audience and collaborating with the audience with many more different variations of the guidelines. When thinking and gathering the requirements for this course project, many ideas emerged. The main driving requirement is that the course must be very simple wrapper for external digital materials. The driving requirement is therefore simplicity.

Keeping the navigation and course structure as minimal and simple as possible may lead to removing some of the aspects hindering students ability to self-study. If the course navigation and layout is hard to use, the students are having a hard time completing the course. In self-study-based course, the course structure and layout will be evaluated by the students even more, since the platform is the only place they can use to complete the materials needed for a passing grade.

Self-studying is not the best way of learning for everyone, since the nature of self-studying is very dependant on the attitude and capabilities of a student. Self-studying gives a lot of freedom, but also responsibilities. A student has to have a strong will and drive to accomplish the difficulties in software engineering. When a course is self-study based, the amount of time put into preparing and assessing the course materials must grow to provide a proper learning experience. Students who struggle with basic time management and drive to learn, should highly consider before starting a self-study-based course.

The workflow of a self-study course is flexible to fit in any curriculum. All materials and education are directly available online and can be accessed through a working internet connection. In addition, students have the ability to contact each other and post questions to the course Discord-forum in problematic occurrences. Establishing a smooth workflow for a student may have a positive effect on the learning process. The course workflow consists of many variables including the course materials and structure, and the students time. This course workflow is designed to support fast access to materials and tools. Accessibility enhances the course experience.

3.5 Student workflow

Student workflow starts by enrolling to the course via the current enrollment system or enrollment method described. Depending on the skills which a student has acquired prior to the course, the workflow can have very different paths. The workflow of a student can be described as five sections: 1) reading the general information; 2) setting up environment; 3) course material execution; 4) creating a self-made project; and 5) giving feedback. In addition to the sections, student is presumed to document their progress through each section in a learning diary. A more experienced programmer may skip sections 2) and 3) and move straight to the self-made project-phase, since the skipped phases are created for students with little to no experience of setting up an environment and understanding the coding domain.
Students are allowed and encouraged to interact through Discord. Discord is used as a platform, where users can post questions and seek help from fellow students. Discord has built-in features to enable voice chatting and typing messages to text channels. These features are very useful since they allow multiple students to share their vision and problems. Discord supports code formatting, images, links and various other contexts. A screen share is also possible through the app, which is helpful in very problematic cases. The platform acts as a fast communication stream between all course participants including teachers.

Figure 1. Discord platform

The self-made project and the learning diary submission is done via Git repository. A link is submitted to Moodle. The submission can be done by posting a file containing the link, by adding the link as text or code to Moodle submission field. Submission field is created by using the Moodle teacher-tools.

Figure 2. Moodle assignment, repository submission
When the course assignment deadline is met, the evaluator checks the submissions as described earlier. The main focus of the submission is to validate that the demo video and the program is sufficient. The requirements of each module are displayed on the Project section. The evaluated contents are displayed below.
LEARNING DIARY

EXAMPLE STYLE 1

10.9.2018

I checked the general information and understood the main focus of the course, which is to find my passion as a software developer and create a unique project to represent my skills. I chose front-end module because it was the most interesting project offered. I’ve also tried to set up my environment, but I could not decide which code editor I would like to use. I learned to set up a git repository and did my first commit, everything went smoothly after I clicked the banner to watch intro to GIT.

11.9.2018

I have chosen VS Code as my code editor for this course, I learned how to set up addons by googling how to do it. I searched the web for best addons and chose the best addons that I think fits me best. I started to watch the first part of the example project to understand the technologies better.

I did my second commit but somehow it did not go as I planned. I went to stackoverflow and found quite many threads about version control problems. I was able to figure out what was the problem and continued to watch the first part till the end.

Figure 5. Project code in repository

Figure 6. Learning diary example
3.6 Assignment assessment

Assignments are initially designed to be a three-part course. The initial content is designed to base on a tested free web-course from external sources, a learning diary written by a student and a course project. By looking through the first set of material assessment, the workload for overall assessment of all material even considered, would’ve taken too long to fit in thesis schedule.

The assessment phase is divided to smaller parts to support better manageability. The initial narrowing assessment starts by taking a quick look on the contents of the external courses. A few sample snippets of code or other materials are swiftly checked. After the initial narrowing, the amount of valid materials for the course decreased by a fair amount. The assessment process of narrowed digital courses from external sources led to an understanding: assessing multiple courses is too vast of a challenge for a bachelor thesis. Finding a complete course fitting for this thesis context in bachelor level students with free materials is a really challenging task. Whole external course packages are discarded as used digital e-learning sources in this thesis.

The alternative option to learn programming apart from course packages, is example projects. These example projects display a baseline information of the technologies used. By assessing example projects, a need for tutorials to teach the technology basics used in example projects emerged. The technologies are added before every example project section in a form of a tutorial. The project addition enabled students to have a complete course package generated from tutorial videos and tutorial example projects.

When the content of each module becomes ready, the assignment design is checked. A student is required to return a learning diary and a self-made project at the end of the course. The project could also be an extended version of the example project given in the course materials for flexibility reasons. The assignments are uploaded into a git-based
version control system repository, which will act as the final return for the course. The return of the documents is designed to help the evaluator easily access the workflow of a student and the project created. In a version control system, a person can see when the work is committed by date and what was done. Version history enables the students to have a better picture of what they’ve accomplished and what they still need to do for the course.

In the first discussions, the biggest problem turned out to be how to evaluate a course which has new technologies that are not well known with our lecturers. The other problem is to prevent people from submitting copied projects or older projects from the past just to pass the course for the credits. To solve these problems a learning diary is selected as the solution. The diary acts as proofing documentation about the work done. The evaluator can compare the source code and the learning diary contents. If the information matches, then the student is doing the course as it was designed to be executed. If the information does not match, then the student has possibly done something which the course is not intended to be graded from. In problem cases it should be clear to notify the student to re-submit an enhanced version of their project and/or the diary. The evaluator makes decisions based on student responses.

The assessment phase is designed to be managed by one person with moderate knowledge on the development technology areas. This means that the person responsible for the course execution do not need to have expert education on the subjects they are teaching. The course assessment phase is designed to support a person who has knowledge of coding and version control. The main focus is to check whether a student has done the work given. The evaluation is done as a pass or fail evaluation when the criteria to pass the course is set to accomplish only the given tasks.

### 3.7 Evaluation components

The assessments phase results in two items to be evaluated, learning diary and self-made project. The evaluation components are derived from the items. The components are items matching and project quality. These components can be divided into smaller tasks. The task of “items matching” is to check whether the content created in the project is similar to the content described in the learning diary. The matching is done by manually comparing the project results, repository history and documentation from learning diary.

The project quality is assessed by checking that the project can be run or executed. The results of running the project is provided by video and a document containing information of how to run the project. If the project results are provided and they match the evaluation criteria provided in the course page, a student passes the project phase.

Learning diary template is created to assist students in their writing process. The diary should at least have 1) chosen module; 2) date of entry; 3) brief description of the work done; 4) learning outcome. The exact format of learning diary is not defined. A student is given the diary template and is expected to follow the formats provided in the template. The diary evaluation is completed by assessing the sections given above. Each section should
have understandable and clear content, which is easy to follow and read. The content should match the final project content.
4 IMPLEMENTATION PHASE

4.1 Moodle as a platform

Moodle offers a quite adequate amount of tools to support the different structures and assignment a teacher should need. As an open source-based platform it has many options for people with heavy technical background such as software developers. In faculty-level the staff must be able to manage the addons and plugins added to the platform, for security and performance reasons. In the case of this thesis, the access rights to add plugins or addons to Moodle is not available, which has resulted in the course pages being default styled.

4.2 Version control

When creating a course from scratch, it is useful to have a version management for viewing the different versions of the course development process. Version control in this thesis was done by using GIT. The chosen version control offers fast branching and merging with distribution. Version control leads to better management locally and online. Having the version history available, one can quickly restore the version to previous one if needed. Version control gives safety and peace of mind to a programmer when developing any code, that’s why version control is also essential part of the course teaching materials.

Moodle has different components for assignment submission. Students can submit their work as different files or combination of multiple files. When assessing a coding project a student has made, the amount of files and the process of managing those files for assessing grows too big for Moodle to handle. Hence the Moodle course conducted in this thesis, has an assignment submission as a git-repository. The submission form of the repository is a link to the repository posted as a text or a file containing the link. When the submission is carried out this way, the assessing becomes very developer friendly and fast.

4.3 Code in Moodle

The ability to add code as HTML and inline CSS to the course pages is generally very useful. Creating and managing the course pages with each tab as a file in a code-editor is way more efficient and faster than the traditional way. The traditional way is to use the built-in mechanics in Moodle to add content. Some of the components needed to create the course as whole, one can find from the built-in tools.

Using a code-editor allows the course creator to have different technologies available for developing. Code-editors have add-ons which can be utilized even for developing a plain text-based course page. The add-ons for the chosen code-editor in this thesis are different formatting libraries from code color styling to bracket matching. Different add-ons are available for displaying the developed version of the course page quickly, without pasting the source code to Moodle. In future iterations for this course, using a CSS preprocessor for inline styles is worth to consider for easing the workload on layout.
4.4 Implementation results

The implementation completion is done by relying on the decisions on design phase. The results formed through 3 iterations. The iterations phases were creating, testing and gathering feedback. The next iteration was based on the feedback from the previous iteration. The banners creation is done by using Canva and materials found within the platform (Canva, 2018). The results are the displayed as the course layout.

Figure 8. Course landing page
If you are familiar with version control, this can be skipped.

Install Git. Version Control

Install SublimeText (Optional)

Create account in Bitbucket

Setup Git and create a public repository. Commit your work in the repository and keep track of your code.

**Recommended Code Editors**

Download VS Code

Extensions for VS Code

Download SublimeText

Download Atom

Download Android Studio, for android mobile development

**Join the course community**

**Figure 9. Environment setup**

**Course goal**

This course aims to give you an edge in the job market by providing tools for creating unique projects and to help you find your position as a software developer.

In this course you will choose one module of your liking.

You can choose between Frontend, backend, mobile (Android) or full-stack development.

The course has 4 mandatory assignments for a student to pass this course.

1. Watch the tutorial and choose one that you like.
2. Create your own code and commit it to your repository.
3. Submit your code to the course coordinator for review.
4. Review feedback from your peers and make necessary improvements to your code.

**How to complete the course actually?**

1. Watch the tutorial and choose one that you like.
2. Create your own code and commit it to your repository.
3. Submit your code to the course coordinator for review.
4. Review feedback from your peers and make necessary improvements to your code.

**How does the evaluation process work?**

You will submit a link with the assignment containing a link to your public repository.

1. Submit tutorial and submit code.
2. Review feedback from your peers.
3. Submit final work to the course coordinator.

Evaluations will be based on the following criteria:

1. Your ability to complete the assignments.
2. Your understanding of the concepts covered.
3. Your ability to apply the concepts in practical scenarios.
4. Your ability to communicate your thoughts and ideas clearly.

**Using tutorials as a source of learning**

1. Tutorials are a great way to understand the key concepts. Make sure to follow along with the tutorials and take notes.
2. Review the concepts by revisiting the tutorials and practicing the exercises.
3. Practice problems with solutions to solidify your understanding of the concepts.
4. Don't just stop with the solutions. Try to understand the reasoning behind the solutions.

**Figure 10. General information**
Module tasklist

Complete the tasks below and remember to reflect on your work in the learning diary.
Follow the steps and complete the ending table in each video.

1. You'll learn to develop websites for coding
2. What is react and how to use it
3. What requirements before the classes

Figure 11. Module information

Front and your goal is to understand how to combine front and side with JavaScript to create various effects in web pages.
First go through with the environment setup and read the general source information page carefully.

Module navigation

Figure 12. Front end module
In mobile development, your goal is to understand the basics of Android Studio and how to develop an Android app. First, go through with the environment setup and read the general course information page carefully.

**Module tasklist**

Complete the tasks below and remember to reflect your work in the learning diary. Follow the steps and complete the coding tasks in each video.

**Introduction (Part 1)**
1. In this video, you will learn how to set up an Android Studio Project.
2. Your Android Studio can be used to create a simple app.
3. How to debug and run an app in Android Studio.

**Core Elements (Part 1)**
1. In this video, you will learn about core elements of Android development.
2. You will learn what are Activities, Views, Intents, Services, and Broadcast Receivers.

**Layouts, Layouts, and Images (Part 2)**
1. In this video, you will learn how to use images.
2. How to create a custom layout component.
3. How to incorporate images into design with View.

**Project task**

Your project should have the following:
1. Functionality with components (e.g., buttons, text fields, loggers, etc.)
2. Multiple views.
3. A component to display information (e.g., loggers).

You are free to add anything you think is necessary. Your project task involves the app. You can add text, buttons, images, and other logic.

**Android Staging (Android Studio)**

Make your project represent your skills, so that you may use this project as a solid proof of your skills when applying for a developer position.

---

**Figure 13. Mobile development module**

**Figure 14. Backend module**
In Fullstack your goal is to understand how to combine MongoDB, Express and Node.js with Angular to create a MEAN-stack project.

First go through with the environment setup and read the general course information page carefully.

Module tasklist

Complete the tasks below and remember to reflect your work in the learning diary. Follow the steps and complete the coding-tasks in each video by coding along.

- Node.js
- MongoDB
- Express.js

Express JS Crash Course [Crash Course].

You will learn:
1. Express Installation and Setup
2. Middleware
3. Routing
4. Template Engines
5. Forms Input
6. MongoDB w/ MongoDB

**NOTE:** don't use the error formatter code, instead copy this: `app.use(expressValidator());`

If you're having trouble with the tutorial you can compare your code to the source code link to sourcecode.

- Angular
- MEAN-Stack

Project task

Your project should have the following:

1. MEAN-stack

Remember that you can use the example code in your own project. You are also free to add anything you think of in addition. Make your project represent your skills, so that you may use this project as a solid proof of your skills when applying for a developer position.

Figure 15. Fullstack module
5 TESTING PHASE

5.1 Content is up to date

When assessing the chosen materials for this thesis the up-to-dateness is considered as one of the most important values to take into account. When the materials are based on teaching coding, the freshness of the materials is a key value. If the materials displayed have outdated information, a student has to find out what exactly has changed and more importantly, how to fix it. This will cause unnecessary work for the students, which may harm the motivation to learn coding.

Testing the teaching materials by doing them, is the best way to inspect the up-to-dateness of the material. In some materials outdated information about software development can be found. When outdated information is found, teacher has two options. First option is to find the change and post the change to the Moodle page. Second option is to discard the current material as outdated and find a replacement material. In these cases, the first run of this course used the first option and posted code changes to course pages.

5.2 Validity

The materials must be valid for academic education purposes. This is evaluated based on the knowledge a bachelor-level student has when joining to the course. The knowledge is based on the courses a student should have passed, when starting the second or the third year studies.

The materials need to have a low level of understanding as a starting point of the topic. Ease of understanding is to ensure, that students with minimal to none level of knowledge of the new technologies presented, are able to complete and learn the materials. The materials need to be sufficient enough to explain the technology and the example uses of the technology. From a students view, the material examples need to be easy to follow and explanatory, so students can learn the answer for “why something is written as it is written” and not just mindlessly follow and type the code. It is important to understand the reasons behind coding decisions. Pausing the video and reflecting what’s actually happening is key ability in e-learning. By pausing and reflecting the content learned, a student is more likely to learn the technology.

The material quality must be visually attractive and all software used should be told. Steps of the tutorials must be clear and followable. If the material is a video tutorial, then the video quality must be clear enough for a student to differentiate the code syntax. Having the video material including a repository of the project is helpful, since even if the video may be outdated the repository may be updated to match the current versions of external libraries.
5.3 High interest level

The contents of the video material were a high priority value during testing phase. A student perspective for the matter is critical for material that a student is able to watch without pausing the development. The voice and atmosphere of the content creator is a key factor for students who learn audiovisually. The material creator’s voice may have very negative effects on the endurance a student has for learning. A student may lose focus and drive to learn if a voice is hard to understand (Craig & Schroeder 2017).

In the testing phase, the voice and overall feeling on the video was tested by completing the video tutorials. Completing the tutorials is done by coding along with the content creator and following the creator’s steps along the tutorial. By testing the materials from a students view, the interest level of a video is evaluated. Some tutorials are not fit due to the lack of video quality, the creator inadequacy, video content or the technics used in the videos.

Each tutorial evaluation is done by the course developer. The evaluation is based on hands-on experience and expert consulting from professionals, who are more familiar with the subject than the course developer, if necessary. The latter part mentioned is not mandatory for having a successful course, but is recommended.
6 CONTINUOUS DEVELOPMENT

6.1 Digital material research

The material search is based on using the following web sources: Youtube (Youtube 2018), Udacity (Udacity 2018), Codeacademy (Code Academy 2018), Khan-academy (Khan Academy 2018) and the homepages of different web-technologies such as Angular (Angular 2018). Youtube is the most used source in the context of this thesis. The variety, quality and availability are significant when comparing the course needs to the material sources.

The sources which offer free or partly free digital materials can be found searching the web for free coding courses, using search engines or video platforms such as Youtube. Many of the big companies have education available for free, these materials are not included in the search of this thesis, but should be considered in the next iterations of content searching.

Re-assessing the previous iteration course-materials is important for maintaining the course quality with a certain level. Even if the material works and the project compiles, that does not instantly clarify the material as valid. Technologies develop as do the materials, so both areas must be checked to gain a firm picture of the material quality.

Possibility to create own updated materials based on the outdated ones, is one option for managing the course development. Re-creating the materials with similar learning objectives can be a lot of work, but it may be the best solution to maintain the best quality level of the course. Further research is needed to conclude the legal matters when creating free or paid materials for education purposes only.

6.2 Course results as a part of development

This course is designed to develop with each iteration. The feedback section is mandatory for each course execution to gain the best possible data to develop the course better. After the first run, the course staff can analyze the data and the projects students did. The analysis will result in better understanding of the course management, structure and improvement ideas. The data may affect as extending the content of some modules or even lead to deletion of a module due to popularity.

Different course improvement opinions are gathered through individual discussions between the course assistant and students. The discussion topics have varied from the module contents to deadlines design. One common subject is the student work distribution. Should the work timeline offer more strict, but more followable solution. Currently a single deadline offers a lot of flexibility to students, but the freedom may not be the best solution for all students. The students who perform best under pressure may get better course results by multiple deadlines, since the work is divided by the course structure.
After the first set of data has been acquired for processing, the improvement areas will be more clear. The need to develop certain areas or to add more modules will also be easier to identify.

6.3 Course material maintainability

Maintaining the course material is an essential part for continuous development. The idea to have the same modules for different iterations requires reassessment of the “old”-modules in order to keep the content up-to-date and valid. Doing the reassessment can be done with different workloads. A fast assessment is sufficient to check the course materials validity after a short period of time has passed after the course has been held. A fast assessment includes testing for running the example projects found in git-repositories. If a project fails to run, a fast check for available fix may be faster than finding a completely new material and assessing it.

6.4 Tools selection development

The tools section of the course introduces different code-editors and version control software available for free. The code-editors have different settings and layouts for each individual to find their favourite. Different add-ons for editors are also available, some easier to add than the others. Further research for developing the tools section may have a positive effect on the course, since tools enable better performance on programming.

Creating good code comes primarily from the skills, which a programmer has. The other formidable component to produce good code, is the tools one uses. The tools provide shortcuts, auto completion, different searches and many other useful hints for the developer. Learning to use an editor or version control effectively will be very rewarding on the long run. Of course editors and programming are also developing, but the fundamental properties an editor has, should stay moderately the same.

Developing the tools section has multiple angles for development. One can find other code-editors to extend the current selection. The other way of approaching the tools development, is to extend the materials given for each code-editor. There are many tutorials of how to use a code-editor efficiently. Many useful blogposts of different configs for shortcuts can also be found from various sources online.

A good free agile project management software could be a healthy addition to the course package. It would help students divide the workload more evenly for the course duration and help managing the internal deadlines for the phases in the project. Management softwares are widely used across the whole IT-industry in agile domains (Wagenaar et al. 2018). By getting students familiar with the industry settings, students use and learn these management techniques early and develop a healthy habit of time management.


7 SUMMARY AND CONCLUSIONS

7.1 Initial course results

The first run of this course had 15 students participating. The results from the first run will be gathered through evaluation process, where the main data source is a feedback form presented in the Moodle page. The feedback form answering rate and other results will be discussed in further research.

<table>
<thead>
<tr>
<th>Participants</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Front end</td>
</tr>
<tr>
<td>7</td>
<td>Mobile</td>
</tr>
<tr>
<td>2</td>
<td>Backend</td>
</tr>
<tr>
<td>3</td>
<td>Full-stack</td>
</tr>
</tbody>
</table>

Table 1. Module participation

The most common module was mobile module. Mobile module had android technologies present, which indicates that students are mostly interested in full-stack mobile development. The relation for the amount of mobile development job-positions rising can be a research question for future research. The full-stack module offered most diverse technologies. The projects created in this module are expected to be great examples of how a complete software project looks like. Front-end module offers a more design oriented approach to coding. The way one creates a layout and how the components are manipulated through the UI are the key learning objectives. However the module participation rates indicate that creating mobile UI and functionality may been more interesting choice for a student when comparing to the web based front end module.

7.2 Conclusions

The study on this thesis is based on the assumption, that external digital materials can be used for academic education purposes. E-learning has been researched extensively with very promising results (Rubin & Brown 2019, p.238-239). The reason driving the study lies on the roots of wanting to richen the skills, which SWE students have after completing the bachelor’s degree in LUT. The SWE degree program needs all the help and research it can get to match the growing need of developers in the digital era. This study is an example of student influencing the contents of the future SWE degree program.

Many successful cases in e-learning research exists. As one example, Le et. al. (2010) researched e-learning through pedagogical analyses. The results of this thesis can be seen as concrete evidence when using external digital materials on academic education purposes successfully. The results can be used to increase the amount of possibilities to learn different technologies within the LUT SWE curriculum. The results can be also seen as a different way of teaching coding. A lot of student work is already done individually, but usually not by a whole course.
7.3 Reliability

The possibility of human error in the research done within the thesis, is present. Misconceptions in the material evaluation or other aspects of the course being deficient compared to the goals are possible, but extremely unlikely. The core decisions affecting the course outline are based on prior successful experiences and research in software development education and the usage of self-study material. The failure of interpretation of the research materials are present in every thesis, however the guidance of other parties involved in this thesis can be seen sufficient enough to verify the outcome of this thesis.

The technologies chosen in the course materials are a little part of the available mass of development technologies. This thesis is sampling only the most popular Node.js and Android technologies coupled with MEAN-stack and RESTful architecture. A lot of different technologies and frameworks are available for further course-development.

7.4 Extending research

How different materials and sources affect the learning is still open for more research. The amount of different methods and the outcomes of the usage of these methods remain open for more research. This thesis operates in the domain of self-study based learning, where the teacher or teacher’s assistant role is heavily based on the questions for the course structure and the ability for the student to complete this course. Different strategies used when designing the support roles in e-learning environment may have better results in future learning outcomes.

The major impact affecting further research comes from the first course results. The course is developed to support an iterative development model. The iterative approach does not aim for the course being perfect at first try. The idea is to use the information gained from course participants to enhance the course in the next iteration. The research of how this iterative model affects the course structure, learning outcomes and other aspects covered in this thesis remain unconcluded.

When a course is based on digital On-Line materials, which are available with a working internet connection the course scalability is great. The research of extending the course to appeal for bigger audiences through scalability is still open for more research.
8 REFERENCE LIST

Angular, 2018, Available at: https://angular.io/, Accessed: 28.11.2018


Canva, 2018, Available at: https://www.canva.com/, Accessed: 28.11.2018

Chilukuri, K., C., Raju, K.V.L., 2015, PPP: A Paradigm for Online Education in Engineering Colleges

Code Academy, Available at: https://www.codecademy.com/, Accessed: 28.11.2018

Cox, K., Imrie B.W., Miller, A., 2014, Student assessment in higher education: a handbook for assessing performance

Craig, S.D., Schroeder, N.L., 2017, Reconsidering the voice effect when learning from a virtual human

Cruz, S., da Silva, F.Q.B and Caprerz, L.F, 2015, Forty years of research on personality in software engineering: A mapping study

Gergel, V., Kustikova, V., 2016, Internet-Oriented Educational Course “Introduction to Parallel Computing”: A Simple Way to Start


Le, D., Tran, V., Hung, D., Nguyen, A., Hung, A., 2010, Applying Pedagogical Analyses to Create an On-Line Course for e Learning

Li, S., Xiao, X., Bassett, B., Xie, T., Tillmann, N., 2016, Measuring Code Behavioral Similarity for Programming and Software Engineering Education


Rubin, A., Brown, A., 2019, Unlocking the Future of Learning by Redesigning Educator Learning

Shastri, K., 2015, Transforming Engineering Students into Industry-Ready Professionals

Stackoverflow, 2018, Available at: https://stackoverflow.com/, Accessed: 28.11.2018

Su, C., 2016, The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study

Telsang, M., Kulkarni, S., 2015, Improving Classroom Dynamics Through Teaching–Learning (Academic) Audit

Tripp-Carrillo, M., Alzarev-Riviera, L., et al., 2017, HTMoL: full-stack solution for remote access, visualization, and analysis of Molecular Dynamics trajectory data

Uikey, N., Suman, U., 2015, A Lifecycle Model for Web-based Application Development: Incorporating Agile and Plan-driven Methodology

Voas, J., Kuhn, R., Paulsen, C., Schaffer., 2018, Computer Science Education in 2018

Wagenaar, G., Overbeek, S., Lucassen, G., Brinkkemper, S., Schneider, K., 2018, Working software over comprehensive documentation – Rationales of agile teams for artefacts usage


Xie, A., Huang, X., 2012., Advances in Computer Science and Education.

Yadav, A., Gretter, S., Hambursch, S., Sands, P., 2016 Expanding computer science education in schools: understanding teacher experiences and challenges

Youtube, 2018, Available at: https://www.youtube.com/, Accessed: 28.11.2018