FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Concept of a Web-based Application for Cooperative Teamwork in the Field of Teaching of Software Techniques

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Informatikdidaktik

eingereicht von

Mészárosné Csuta Eszter
Matrikelnummer 0727015

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuer: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.nat. Dr.techn. Rudolf Freund

Wien, 16. Oktober 2019

(Unterschrift Verfasserin) (Unterschrift Betreuer)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Statement by Author

Mészárosné Csuta Eszter
József Attila u. 2., 9151 Abda, Ungarn

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

I hereby declare that I am the sole author of this thesis, that I have completely indicated all sources and help used, and that all parts of this work – including tables, maps and figures – if taken from other works or from the internet, whether copied literally or by sense, have been labelled including a citation of the source.

(Place, Date) (Signature of Author)

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout the master thesis. I am thankful for their help and guidance.

I would first like to thank my thesis supervisor Ao.Univ.Prof. Dipl.-Ing. Mag.rer.nat. Dr.techn. Rudolf Freund for the support of my thesis.

I would also like to thank the experts who were involved in the expert interview for this thesis. Without their profound participation and input, the expert interviews could not have been successfully conducted.

I would like to thank my husband for supporting me throughout writing this thesis and in my life in general.

Finally, I must express my very profound gratitude to my father for providing me with continuous encouragement throughout my years of study. I would like to dedicate this work to him.

Kurzfassung

Im Rahmen des Software Engineering-Studiums wird Studierenden vermittelt, zuverlässige Software-Programme zu entwickeln. Ein neuer Trend besteht darin, dass Softwareentwickler meistens in Teams arbeiten. Der komponentenbasierten Software-Entwicklung kommt heutzutage eine entscheidende Bedeutung zu. Um Softwaresysteme aus Komponenten aufzubauen, müssen diese über ihre Schnittstellen gut miteinander kommunizieren können.

In dieser Masterarbeit wird ein Systemkonzept für eine Web-App namens AcadIC vorgestellt. In dieser werden Software-Integrationsprozesse vermittelt, bei denen die Lernenden in kooperativen Online-Teams zusammenarbeiten, um ein voll funktionsfähiges Software-System aufzubauen. Jeder Lernende entwickelt eine Komponente alleine, und danach integrieren die Lernenden in Team-Arbeit diese Einzelkomponenten zu einem funktionierenden Software-System.

Die vorliegende Arbeit beschäftigt sich mit Software-Integration, dem Erlernen von Programmieren, Teamwork und Online-Lernen, um in diesen Bereichen Probleme zu finden, zu identifizieren und zu analysieren. Anschließend wurden die Anforderungen und Merkmale der zu entwickelnden Web-App festgelegt. Zielgruppe, Lernziele und erwartete Auswirkungen wurden definiert. Danach wurde das Design entworfen, in dem die Programmieraufgaben und die einzelnen Prozessschritte, nämlich die Softwarekomponenten, festgelegt wurden.

Zur Evaluierung der vorgestellten Web-App wurden qualitative Experteninterviews durchgeführt. Die Bewertungsergebnisse zeigen, dass die entwickelte Web-App geeignet ist, die Software-Integrationsprozesse durch Teamwork zu vermitteln.

Abstract

Within the framework of the software engineering study program, students are taught how to develop reliable software programs. A new trend is that software developers mostly work in teams. Nowadays, component-based software development is of crucial importance. In order to build software systems from components, they must be able to communicate well with each other via their interfaces.

This master thesis presents a system concept for a Web app called AcadIC. It teaches software integration processes in which learners work together in cooperative online teams to build a fully functional software system. Each learner develops a component alone, and then the learner integrates these individual components into a functioning software system in team work.

This paper deals with software integration, learning programming, teamwork and online learning to find, identify and analyse problems in these areas. Subsequently, the requirements and characteristics of the Web app to be developed were defined. Target group, learning objectives and expected impact were defined. Then the design was elaborated, in which the programming tasks and the individual process steps, namely the software components, were defined.

Qualitative expert interviews were conducted to evaluate the presented Web app. The evaluation results show that the developed Web app is suitable for communicating the software integration processes through teamwork.

Contents

List of Figures

List of Tables

1 Introduction

This chapter gives an overview of the problem description, motivation, and the objectives of this master thesis.

1.1 Motivation

Programming is one of the first classes in the field of the study of computer science [13]. Teaching introductory programming is often problematic because of the large number of learners' heterogeneous groups, the difficulties in understanding the subject, its practical application, and the high course leaving rates of the learners [56]. Novices may be discouraged at the beginning of their learning process [1].

In the software engineering field nowadays component-based systems, shortly CBS, are more widespread and more in industrial fields like Internet banking or development resource planning systems [59]. In these professional field component integration has a crucial role and one of the most challenging problems is the differences when matching two components [59].

In the last years, to support teaching and learning, online education became widespread in the teaching of programming. Nowadays learners can work remotely, far from the classroom (i.e. from home) using web-based platforms, which are also known as online-platforms.

New online-platforms should be designed, which ensure the practice of those topics. Therefore the domain chosen for this thesis is the teaching of introductory programming, online learning, software process integration, and team-work.

1.2 Problem Description

A classical programming class consists of a theoretical and a practical part, where the practical part has a crucial significance. The teaching of programming formerly happens in a classroom environments, where personal presence was expected.

In the publication, *A study of the difficulties of novice programmers* the authors describe an international survey on the difficulties during the teachings of programming, by analysing the learning activities of more than five hundred learners [56]. This section introduces the results of this survey: The results show that learners of the introductory programming class, so-called novices, have problems with applying the gained knowledge into practice. Another problem that appeared in this survey is that novices often overjudge their knowledge, they think they have a much higher knowledge than it is, which can lead to a lack of motivation.

In the publication, *Improving Software Engineering Education through Enhanced Practical Experiences* the author claims that continuously improving education and supporting practical education are crucial points in this field [83].

Nowadays programming teaching happens online with the help of online platforms. An online platform makes it possible to manage the education online, providing and organizing class contents (such as class materials, assignments, quizzes) and providing collaborative and communicative possibilities for learners and teachers on the web [18], which merge tools and services [9]. A large number of learners can access these platforms at the same time and it is also not necessary to have a personal presence during the educational activity. Nowadays several online platforms exist for teaching and learning programming. The developers of these platforms have designed them for both beginner and advanced levels of programming, conveying different programming languages. Some are also developed with games and contests possibilities to motivate learners. These platforms were designed for individual use and teamwork.

In the book, *Didaktik der Informatik* [33], the author points out that motivation is also a main goal of didactics and without adequate motivation the instructional effort is futile. As the author claims in the book, *Reflections on teaching programming*, the proper practical work has a crucial influence on the successful process of teaching and learning programming [13]. So-called learner-centered tasks incite learners and motivate them more in their learning process [13]. They have fascinating goals, in contrary to the so-called "small-scale" exercises, which only deliver exercise for practicing the syntax of a programming language [13]. As the author discusses in the publication *A survey of literature on the teaching of introductory programming*, if learners use a tool during their learning process in an experiment in the field of introductory programming, then their throughput is higher [73].

Computer programs can solve a wide range of problems. The area of software is developing much faster than the area of hardware and the needs of users and the application areas of software are very wide. A good and reliable software can solve those tasks, therefore the qualification of good programmers is necessary and the support of its education is indispensable.

This paragraph is cited from the book *Softwaretechnik* [28], wherein the author describes the software as a ubiquitous and complex notion for 30 years. A successful software implementation needs experience and methods, and collaboration has a top priority in software development, too. As the author points out, a proper software project applies teamwork, which consists of programmers, analysts, architects, user interface designers, project managers, pm-officers, database specialists, test manager, tester and quality manager.

For the training of programming several kinds of tools exist. This paragraph gives an overview of these tools, cited from the article *A survey of literature on the teaching of introductory programming* [73]. So-called visualization tools for programming virtually present the programming structures and the execution of programs on computer screens. As the author claims, " (...) programs and algorithms are dynamic artifacts (...)". Therefore, it is a challenging task for learners to understand their fundamentals. But using only visualization tools for teaching programming has not enough effect on the learning process. Automated assessment tools have relevancy if many learners are present in a programming class. The codes written by the novices are assessed automatically by those tools, easing the workload of the teacher. Programming environments, like Integrated Development Environments (IDE), include, among others, editors for writing program codes and components for supporting the development process of programs, such as test tools, where the written codes can be tested and code analyzers for evaluation of the code. Among the programming environments, Microworlds and Programming support tools are most suitable for novices. Microworlds aims to reduce the gap between mental models of learners and the programming language. Programming support tools have a basic, reduced execution environment, such as IDE. The specified features of programming support tools are the following:

- Interactive incremental code execution: evaluate code expressions and statements, and call methods interactively
- Visualization: graphical presentation of programming structures and execution of programs on computer screens
- Supporting editing functions: editing of program codes
- Supporting syntax: through on-line manipulation of programming environments

Further features of programming support tools are the following:

- Plagiarism detection: it prohibits the copying of coding
- Automated assessment or online evaluator automatic evaluation of code, to assess learners work

Nowadays one of the modern branches of software engineering is Component-based software development. The author in [59] describes CBS development so: *"CBS development is integration-centric with a focus on assembling components through well-defined interfaces to build a software system"*. As it is a quite new branch, therefore it is hard to find supporting educational materials for this topic.

The above-mentioned facts give great significance to programming, software integration processes and the necessity of newly developed Web apps for teaching it, with collaborative, cooperative and motivating features, using electronic educational technologies. An additional reason for newly designing of those Web app is that most current platforms are for advanced programmers and with their complex features and difficult error messages they fit to the needs of advanced, and not to that of the novices and platforms specified for novice programmers are suitable more for teaching introductory programming [73].

This master thesis aims to develop a system concept, which results in a Web app for a Computer-Supported Collaborative Learning pedagogical approach. Computer Supported Cooperative Learning (CSCL) platforms can support and stimulate virtual groups in their cooperative form of learning [74].

The following problem statements will be covered in this master thesis:

- Is the resulted Web app suitable for learning the software integration processes?
- Is the resulted Web app suitable for beginners in programming?
- Does the resulted Web app ease the learning of introductory programming?

1.3 Objectives

The main goal of this master thesis is to develop a system concept for such a Web app, which realizes a cross-section between didactic, informatics and communication. The designed Web app differs from already existed platforms of learning programming: most platforms realize collaborative teamwork, where the most important factor is the common understanding of the tasks. The designed Web app will provide cooperative teamwork for learners, where learners complete tasks in a group to solve a common problem. The Web app aims to ease the entry of the learners to introductory programming and to represent programming more interesting, to motivate learners to learn programming, and to help them to apply their learned programming knowledge in practice.

Further goals with the Web app are to teach the learners the software integration processes, how to integrate autonomously functioning components into a fully functional system, where all those components work together, and to learn in cooperative teams with consecutive tasks. Each learner develops one component individually, the integration of those software components then takes place in cooperative teams to get a completely functional software system. After the development of each component, the learners must integrate them to get fully working software that is capable of handling all the main processes. The output of a component acts as an input for the next one. Learners develop the single components by snapping together predefined blocks representing variables, instructions and other programming elements. It will improve the cooperation skills of learners and their ability in order to make decisions and intensify the learning process. It should provide incentive programming tasks for cooperative group work and communication possibilities for cooperation between learners. With the help of the resulted Web app learners should be able to develop simple data structures (arrays, lists, etc.), integrate components into a fully working software system, and deal with code reusing, architecture, and interfaces. To teach introductory programming with Web app, it must be to be considered, that the Web app has to be designed especially for beginners in programming, because generally, platforms for programming are suitable for the needs of professional coders, and not for novices [73]. The resulted Web app should contain several didactic methods used in the teaching of programming, such as stimulating introductory tasks and performing consecutive tasks.

The resulting Web app has the following objectives:

- Teaching learners the software integration processes through teamwork, from the beginning of programming learning
- Learning in cooperative teams with consecutive tasks
- Supporting learners in their learning process of programming, which is a main subject of computer science study
- Helping learners to apply their theoretical knowledge in programming, and help them in writing programs on their own
- Increasing the motivation of learners, thus decrease the drop-out rate of the introductory programming classes
- Enabling the attendance of a high number of learners in a programming class and making personal presence unnecessary
- Conveying the importance of teamwork in software engineering

1.4 Structure of the Master's Thesis

The thesis is separated into eight main chapters. In the following the chapters are introduced briefly.

The first chapter gives a brief introduction to this master thesis. Objectives and the structure of the master thesis are described here, too.

In the second chapter theoretical fundamentals from the field of component-based development, software process integration, programming teaching, teamwork, online learning, and expert interview are described, with the goal searching problems and gaps in those professional domains.

In the third chapter, the methodology used in the master thesis is described, to develop a system concept for the Web app.

In the fourth chapter related existing online learning platforms for programming are documented.

In the fifth chapter, the requirements analysis is described, to determine the requirements and then the functions of the resulted Web app.

The Web app design is described in the sixth chapter, based on the derived features, and the evaluation process of the Web app is also described here.

In the seventh chapter the evaluation of the Web app, and further results are presented.

The eighth chapter describes a summary and future work of the master thesis.

2 Fundamentals

The master thesis deals with the conception and evaluation of a Web app for learning the software component integration processes through teamwork. However, to understand what it is, the theoretical context, they are embedded in, must be explained in more detail. First, a theoretical introduction to the subject areas is given, which makes up the theoretical background of the research. Those areas of subject are the software integration, programming learning, teamwork, online learning, and expert interviews.

2.1 Component-Based Software Development

This paragraph gives an introduction to Component-based software development, and it is cited from the literature [62]. Component based software development is a specialization of software engineering and it is the improved variation of object-oriented paradigm, where the software consists of several units and the software developers can use those units several times, in another software, too. Therefore, software developers do not have to write this software unit again. This means re-usability. This reusable software unit is called software component. During a software developer process those components come into an existence inside the software development company, as an own development or they originate from another software development company, as bought components. The communication of those components and the method of how a software developer can integrate the components together, are all crucial points during the software development process.

The next paragraph describes some important notions and it is cited from the literature [59]. *"A software component is a fundamental building block for a software system"*. Regards to another definition *"a software component is a unit of composition with contractually specified interfaces and explicit context dependencies only"*. *"A software component can be deployed independently and is subject to composition by third parties that are either developed in-house or purchased off-the-shelf"*. The author describes Component-based system development as following: *"CBS development is integration-centric with a focus on assembling components through well-defined interfaces to build a software system"*. In a Component-based development component integration has a crucial role. Component integration is much more than only putting components next to each other. In connection with component integration some limitations exist. For example, the usage of so called pre-fabricated components has the disadvantage that they fulfill universal goals and maybe they have the kind of features, which are unnecessary regards to the whole software system. The documentation of components is usually not well organized and it lacks of information about the usage of components and its interfaces, too, which would be very important information for the so-called CBS integrators. The author describes the component integration as following: *".. the CBS integration phase usually involves adapting existing software components, implementing missing features and writing glue-code to handle the mismatches between system-to-be requirements and available component features"*. The CBS development approaches has three main phases, those are selection, integration and maintenance. The selection phase involves the choosing and analysing of the appropriate components, which fit the goal of the constructed software system. The component integration phase is the connection of components, based on a so-called architectural infrastructure. The maintenance phase is a life-long on-going activity, which means a development and improvement of the CBS system. The so called glue-code is written to match together components, in order to get a completely functioning software system in the end.

A component model is a standard for illustration, documentation and implementation of software components and it supports the planning process of a software. It also helps to specify the interfaces. Those software component architectures are CORBA, EJB or .NET [62].

A component diagram illustrates the relationship between the certain components and shows how the certain components depend on each other, how they communicate with each other and also the placement of these certain components [82].

2.2 Integration Strategies in Software Engineering

The next section describes the integration strategies in software engineering and it cites from the book *Softwaretechnik* [28]. Integration strategies aim to build the software components together and to make the testing process on the built system. Horizontal and vertical integrations exist. A horizontal integration is the layer by layer integration of a system. There are three kinds of horizontal integrations. This thesis introduces these different types. In software developing projects the developers choose the integrations strategies, which fit the best for the final deadline of the software.

- The Big Bang approach as a horizontal integration strategy is a non-iterative approach, where every system part is integrated simultaneously. The disadvantage of this approach is that early integration and the testing of the system can occur only later. In addition, it is very hard to identify and find the mistakes and errors in the system.
- The Top down integration as a horizontal integration strategy is a layer-oriented, iterative integration form. First the upmost layer is integrated and the layers above are only simulated. The main advantage of this approach is an early integration of the system. The disadvantage of this approach is the huge simulation appropriation.
- The Bottom-up approach as a horizontal integration strategy is an iterative, layer oriented, effective and often used integrations form, where a software tester tests each software parts individually. Then the software parts can be accessed together. First the bottommost layer is tested with test-drivers. By the next integrations layer this test-driver is replaced with a real integration and the new layer, which is needed to integrate, is replaced with a test driver. The main advantage of this approach is that there is no need to do a simulation in the form of Stub. This approach uses test-drivers, whose development is not so difficult and does not take so many times, as the development of a Stub. The disadvantage of this approach is that the system test can only occur later because of the late testing of the uppermost layers.

A vertical integration is a functions-oriented, iterative integrations form. A main characteristic of this approach is that the system has a subassembly (collaboration), in which every part is integrated. There is no need to simulate the system parts and it makes an early system integration possible. Then this collaboration can be integrated, for example with the Big-Bang approach.

2.3 Software Engineering Studies and the Programming

This section introduces the software engineering studies and the programming professional.

2.3.1 Software Engineering Studies

The next paragraph gives an overview about the software engineering studies and it is cited from the literature [17]. Software engineering is one of the most popular studies among universities. The Knowledge Area Software Construction is one of the areas of SWEBOK (Software engineering a body of knowledge) regards to the topic of these thesis. SWEBOK is a guide, which collects the most important knowledge to teach in the field of software engineering. It is aimed to help establish a curricular in software engineering education. The Knowledge Area Software Construction includes fundamental information and technologies about creating an appropriate software. It covers the cases of writing, verifing, debuging and test source coding. There are further so-called software engineering areas according SWEBOK: Software Requirements, Software Design, Software Testing, Software Maintenance, Software Configuration Management, Software Engineering Management, Software Engineering Process, Software Engineering Models and Methods, and Software Quality.

2.3.2 Programming

The author introduces the definition of programming in [77] as follows: *"Programming can be defined as the development of a solution to an identified problem, and the setting up of a related series of instructions which, when directed through computer hardware, will produce the desired results"*. Programming requires a new, special way of thinking and as the author in [77] says *"Computer programming is an art"*. Another author says, *"Software is created by people, with people, and for people"* [50].

A computer program consists of several algorithms represented with programming languages or pseudocode [72]. The author defined algorithms in [20] as follows: *"..an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output"*.

There are three user categories in accordance with cognitive levels in the field of learning programming [87]:

- The beginner, who has not learnt programming before.
- The average, who has some learning experience in the field introductory programming, has gained knowledge in that field and is able to solve algorithmic problems.
- The advanced, who has already gained knowledge in programming and is able to solve complicated programming problems.

2.3.3 Programming Teaching

Next this thesis introduces some programming teaching approaches and methods (relating to this thesis), which help to understand the programming.

Programming teaching approaches

This paragraph introduces one of those programming teaching methods and is cited from the literature [57]. The methodical, algorithmic oriented teaching approach has the focus to create an algorithm. This approach is useful in the whole process of programming (from defining the solvable problem, through the planning, coding and testing process, to the maintenance and final documentation of the software). Due to the exciting programming tasks it offers, this method is most suitable for motivating learners. The advantage of this approach that programming language does not play a role, thus the gained knowledge is more independent.

Another approach is the bottom-up approach, in which learners first learn the syntax and basic elements of a programming language and then later more complicated actions [68].

Programming teaching methods

This paragraph describes some teaching methods, that have great significance in learning programming, too and it is cited from [68]:

- The Problem-based learning (PBL) method helps novices being actively involved in that kind of programming task, that novices need to solve problems. It is derived from the real life of professional programmers. With this method novices prosper their *"higher order thinking", "disciplinary knowledge" and "practical skills"* and use their preliminary gained knowledge in that theme, too. This method helps novices in raising their *"communication skills, creative thinking, motivation and responsibility"*.

- The Puzzle-based learning (PZBL) method supports novices in getting *„critical thinking"* skills and to know how to solve problems in different ways. Within this method novices can work out the used, challenging programming task in different ways. There are more correct solutions for one task. The process of Puzzle-based learning in the case of programming is the following: First, an introduction of the problem, based on the actual learning topic, happens. Then a division of a source code of a programming task into pieces, or so called *"puzzle pieces"*, takes place. After that the learners put the pieces of source code in the appropriate order and at the end the evaluation of the ready source code takes part.

- The pair programming method provides novices a motivating and effecting way of learning programming. Novices build a group of two people and work together on one computer to solve certain programming tasks (designing, developing and testing codes). One novice called *„driver "* writes the source code, while the other novice, called *"navigator"*, takes part in the learning situation by making suggestion for the solution of the programming task and searching for errors in the source code. Novices can often change the roles during the pair programming with each other.

- The Virtual Pair Programming method supports the so called *„flexible collaboration"*. In that case pairs of novices work far from each other, from different computers, and they communicate through an online tool in real time.

- The game-themed programming method helps novices become motivated for coding and understand the different programming concepts. In this method novices write source codes for simple graphic games. The goal of this approach is „*teaching programming concepts through understanding how the games work* “.

Programming Learning Textual and Visual

The next paragraph describes the programming teaching with visual elements and it is cited from the literature [69]. In the case of programming teaching through visual elements, novices can get application knowledge. Tools used in programming teaching must ensure to get both, textual and visual programming learning. For novices it is much easier to learn graphical programming languages (like Appinventor, or Scratch). With the help of visual elements novices can understand the programming better. Through those graphical programming languages novices do not need to know the whole theory of that programming language and they do not need to write codes, which are exactly suitable for syntax and semantics rules, but use drag and drop techniques to make a working program. Novice programmers should start by learning programming with graphical programming languages and advanced learners should continue with learning textual programming language.

In the next paragraph the importance of using animation in teaching is discussed and it is cited from the literature [86]. In connection with the application of animation in the case of teaching algorithms, the author claims that the processes describing algorithms are dynamic and continuously changeful, therefore the teaching process of algorithms should also have the same property. He shows the usage of animation in the algorithms teaching in his work. If teachers would use animations in the teaching process, then it had a motivational effect on the learning process of students. If a tool for learning algorithms uses more than one animations after the other, the efficiency of teaching algorithm will improve. As first step the tools should contain the kind of animations, where algorithms do not perform in such detail, but only the main elements of the algorithms attend. As a second step, the tool should contain that kind of animation, which introduces the algorithms more deeply and more detailed.

Competences of a Programmer

As the author claims in [68] that a programmer needs to have strong *"cognitive skills"* like *"reasoning"* or *"problem solving and planning"*.

In [29] the author has collected some properties of a good programmer and this paragraph will shows some of those properties. Some of the following properties were mentioned: attention, discipline and abstraction ability, sedulity, and self-thinking is important. Rational and logical thinking, the good comprehension and the creativity are also crucial. Furthermore, the importance of analytical and algorithmic thinking, the persistence, and the openness for new technologies were emphasized here.

Next this paragraph describes some competences regarding to pair programming and it is cited from the literature [78]. First, programmers should understand what is written in the software specification. Programmers should be able to communicate about the solvable task with other programmers. They need to write proper codes suitably to the software specification. They should be able to use up coding guidelines properly for their work. They need to know how to test the ready software, clear off the software from bugs and make code refactoring to get a meaningful code. Programmers should be able to communicate with each other, to handle conflicts, and to work and learn in a team.

2.4 Teamwork

This section introduces the necessary teamwork in the software engineering field and shows some characteristics of cooperative teamwork.

2.4.1 Cooperative Teamwork in General

The next paragraph gives an introduction to cooperative teamwork and is cited from the literature [42]. In cooperative learning learners solve a common task and these common tasks are divided into smaller tasks. A group member has the responsibility to solve one task. Learners are discussing how to solve the given tasks with each other during the cooperative work. Learners have the possibility to participate in the learning process actively. Its advantages are: learners are motivated in learning, they can work effectively on the given task and they can make self-assessments. This type of learning gives learners the possibility to be creative, work independently and communicate better during their working process.

The basics of cooperative learning is the humanistic psychology and it appeared first in the United States in the XIX century as one of the first step of reform pedagogy [43].

In comparison with group size, smaller groups can work more efficiently. Six learners working in one group are able to solve bigger, more complex tasks. Furthermore, group forming can happen randomly or the teacher can form the group [44].

The next paragraph gives an overview about the characteristics of cooperative groups, cited from the literature [45]. Those characteristic are the following: having positive interdependence inside the group members, respecting each other performances, cooperation of the team members to solve the given tasks and to share responsibility between the group members. A positive interdependence inside a learning group means an incitement of each other to cooperate. Thus, learners in the group have interdependence between each other, namely positive defence. The learning process should be so organized, that the process of gaining the knowledge only happens with cooperation. Positive interdependence could be developed by determining common goals with contest situations between learners, giving cooperative tasks, with a great amount of learning material, giving roles to team members, and by giving prizes for the performance, and with establishing group identity, which means when an individual organize themselves into a group, he/she can identify his/herself with the group, so that the individual has the same goal as the goal of the group. During the cooperative learning several competencies develop, like analytical skills, openness, good interpersonal skills, directness, claim for the social interactions, improving communication skills and feeling more responsible, independence and preciseness.

The four basic principles of cooperative learning are [46]:

- Parallel (or synchronic) interactions inside a group
- Constructive interdependence, when the learning success of a group member depends on the success of another group member or the group, during the learning activity
- Individual responsibility of the members
- Equal participation in the solvable task of the group members

There are some methods of the cooperative group work in the case of learning introductory programming in a classroom environment. An example is the puzzle method and the flash card method [76].

2.4.2 Teamwork in Software Engineering

The next paragraph describes the software development in teams, cited from the literature [50]. One of the main characteristics of software development nowadays is that software developers collaborate in teams and work on the same code sometime remotely. Global software engineering and open source code development requires teamwork, too. In those cases software developers code together far from each other. The author claims about the importance of *„social aspect“* of software development tools and the collaboration between the software developers.

Regards to the definition of team [34], *“a team consists of at least two people, who are working toward a common goal/objective/mission, where each person has been assigned specific roles or functions to perform, and where completion of the mission requires some form of dependency among the group members.”*

There are basic techniques for developing a powerful team in software development field [35]:

- Defining team goals together.
- Strategy determination for how to realize the goals. The division of the given programming problem into parts, determining the task to do for every developer. Team members have to determine the cycles, how to integrate and test the parts, and write documentation about the process they use.
- Giving team members responsibility is crucial and sharing the solvable task with them.
- Communication internal in the group: without regular communication the coordination of the work is not possible because the software developers are not informed about the coding work of other developers and it is always necessary to give the possibility to communicate.
- Communicating external: communication with management and direction is a crucial elements of a successful software development project.

For building an effective team several supports exist [36]:

- Team cohesions: The main characteristic of a cohesive team is the common, free communication between the team members, the assistance of each other and the good relationship between each other.
- Challenging common goals: challenging, common and clear goals play an important role in a team software developing process. Those goals can be *“detailed plans, performance targets, quality objectives, and schedule milestones”*.
- Feedback: getting feedback about the performance of the team members is important in software development process, aswell.
- Common working framework: Knowing exactly what to do in the software development process is crucial, too.

2.4.3 Collaborative Software Development

Nowadays one the favourite software development approach is the collaborative development approach (like pair programming, extreme programming or writing an open source program in team) [51].

In [50] the author defines collaborative software development as following: *„In collaborative software development, the software product gets created by a team, or a community of developers, who keep in close communication throughout development"*.

The main characteristic of this approach is that software developers work in teams, using so called distributed environments and share information between each other in order to solve the given programming problems [51].

The next paragraph introduces some collaboration scenarios used in software development, cited from the literature [27]. The one scenario is the Micro-outsourcing, which means collaboration between software developers and an external professional in order to get some help for the software developing work. The other is the Test-driven pair programming, where programmers work in pairs and this method is combined with the agile method, where first the unit tests are written and then the program codes by the software developers. The third one is applied by a mobile instructor during the collaborative software development which means, that a mobile instructor gives hints and tips to the student during the software development, connecting to the used software development environments via mobile device.

In the case of that kind of environment, which aims for the collaborative learning of programming it is possible to publish and share the source code and the occurrent ideas with other novices in order to evaluate the written source code and give some hints to each other. Wherein social tagging is also a part of collaborative process [87].

2.5 E-Learning

This section introduces the definition of e-learning, its characteristics and the e-learning environments.

2.5.1 E-Learning and its Characteristics

Nowadays the broad adoption of technology results in a wide usage of digital learning resources in education [60]. The definition of online learning or e-learning is as followed: *"E-learning as instruction delivered on a digital device such as a computer or mobile device that is intended to support learning"* [63]. The main goal of e-learning is the same as in the case of presence learning - to reach the appointed learning outcomes, make an appropriate assessment for them, and be more cost-efficient [64]. As the author claims in [15] because of the appearance of new educational needs and modern learning environments even in connection with Industry 4.0, digitalisation of learning has become of crucial significance. Life-long learning can be a basic act of the learning of a human [15].

The students of the future, the digital natives (persons who grown up using digital media), prefer working in a team, they prefer multitasking and they prefer to learn using the media, instead of pure text. Also they like to use rather interactive applications and work in networks [60].

The next paragraph is cited from the literature [60]. E-learning is a part of the distributed and distance learning model, where students can learn in campus based or online. Distributed learning is a modern learning model. It considers the needs of the students in connection with learning and according to those needs, it customises learning environments, where learners are motivated to learn and work together with other students collaboratively. Distance education is a part of the distributed learning approach, too.

Blended learning means a formal education program (or learning modality) and the combination of classroom presence teaching and e-learning [33].

Computer Supported Cooperative Learnings are, as the author says in [74], *" Collaborative forms of learning, (. . .) the learners can profit from each other professional experience and exchange ideas and views (...)"*. It is a pedagogical approach in which technology is used for the learning and the social interaction between the participants is the crucial point. It is characterised, that the participants share and construct knowledge with the help of technology [80].

2.5.2 Environments in E-Learning

Learning environments, e-learning platforms or tools give students the possibility to distribute the learning time freely and those tools use different kinds of communications, like asynchronous (for example email, online learning resources and online libraries) and synchronous (for example chats, and tools for audio and video) [33]. They aim to visualise learning objects, to learn in virtual groups and to provide interactive, practising tasks for students [33].

The following paragraph is cited from the literature [21]. One of the most important requirements of designing e-learning platforms are interoperability and modularity. Interoperability enables the usage of the services of e-learning platforms on any platforms and any environments. In connection to e-learning modularity that means the division of an e-learning system into components, it causes better usage of e-learning systems and eases the work of software developers, too, because software developers can exchange the modules easily and it causes a much more combined system.

There are some standards and models, that can support the design of e-learning platforms. One of those standards is for example IMS Abstract Framework or E-learning Framework (ELF) [21]. An other standard is E-learning Maturity Model (eMM) [31].

The next paragraph gives an overview about the characteristics of e-learning environments and is cited from the literature [60]. An e-learning environment makes it possible for student to practice the given subjects often, independent from time and place, and to receive frequent feedback. It makes education easy to access and cheaper. It makes it possible to wide the numbers of participation, too. Users in those environments can set their own profile and add new contents, make public comments, share information, construct their own knowledge, generate own contents, work collaboratively in a virtual group to solve problems, and regular their own learning. Those environments make it possible to customize the learning. Those tools make it possible to have fun during the learning activity and it can use game elements in the learning process. The most common terms here are construction and discovery. In general e-learning enables the consideration of the learning process as learning-centered. Earlier the traditional classroom teaching aimed teacher-centered perception, which focused on the teacher.

It provides e-books, furthermore it makes it possible for students to listen and to see lectures as many times as they want to, through video lectures [48].

First versions of e-learning platforms appeared around 1993. Those traditional e-learning environments or e-learning platforms were used in distance education support enrolling, fulfilling the administrative tasks around education, creating and maintaining online courses, and reporting the performance of the students [21].

As the author claims in [60] virtual learning environments (VLE) provides the possibility to support the learning process through the interaction between students and instructors, using asynchronous and synchronous communication possibilities.

The upcoming paragraph is cited from the literature [41]. Learning communities are such environments, in which groups of learners learn together. Their main characteristic is that every participant is involved into the learning activity and the group members should understand the given subject collectively. It can be used in a wide range of professional fields, it helps to understand the learning contents and to learn collectively. Furthermore, it helps in creating and sharing knowledge, and it supports the learning process of a learner. Forum is a kind of learning community. It supports students in information-sharing, posting ideas and questions, and adding new comments. Constructionist learning environments support learners to develop physical or digital objects, i.e. a computer game with the aim to understand a given subject. That kind of learning process, where learners construct something, is more motivational, and has a greater effect to the learning outcomes.

Collaborative online learning environments makes it possible for learners to learn together, collaboratively, even if they are far from each other. Students can reach better learning performance using those environments [53]. In the case of collaborative tools, collaboration means mainly using chats or forums, establishing virtual teams to compete aginst [53].

2.6 Expert Interview

For the evaluation of the Web app expert interview was applied. Therefore, this section introduces the expert interview.
Expert interview has appeared in the 1990s and it is a *„method of qualitative empirical research". Its goal is to discover knowledges of professionals* [4]. It has several advantages. As the author in [5] states *„talking to experts in the exploratory phase of a project is a more efficient and concentrated method of gathering data than, for instance, participatory observation or systematic quantitative surveys."* This research method reduces the time of *„data gathering processes"* and provides fast results in the research project [5]. Furthermore, *„expert interviews are a particularly appropriate method in research aimed at reconstructing explicit expert knowledge"* [6]. One type of expert interview is the explorative expert interview, which *„tries to explore a field of knowledge that is still largely unknown to us. The aim of exploratory expert interviews is to get a feeling for the material in question"* [66].

In the case of expert interview, it is important what types of information and knowledge are gathered. Expert knowledge has three dimensions [7]:

- *„Technical knowledge is expert knowledge in the narrower sense; it is explicit knowledge and can be directly communicated in the interview."*

- *„Process knowledge is conceived of as practical experiential knowledge resulting from frequently and repeatedly performed action patterns and interaction routines.“*
- *„Interpretive knowledge is created not only in functional contexts but is additionally shaped by subjective relevance and viewpoints.“ „It is a subjective interpretation of relevance, rules, beliefs.“*

First, in the case of expert interview it is important to search and find relevant experts. In the case of selecting the interviewee the author states: *„if the targeted expert is not only willing to participate, but also holds a key position in the organization, opportunities for expanding the researcher's access to the field may well also be unearthed in the interview.“* [5]. It has a motivational effect to the interviewee, if the interviewer and the interview has the same or similar scientific background or interest [5]. It is important to make more interactive situations during the expert interview, for example the interviewer can ask the interviewee to state his/her opinion more deeply. The interview questions need to be well formulated and systematized [85].

The next paragraph is cited from the literature [85]. In the case of the analyse of expert interview, there are some steps that have to be made. First, the interview has to be transcribed into a written text. Here it is important to summarize the contexts and to identify important themes. The gained text should be labelled and be ordered for example according to themes or interviewees. Then it is crucial to search and to find communalities, divergences, and conflicting between the shared context, thus relations between the contexts can be identified. Finally, the gained informations should be integrated into the theory.

2.7 Chapter Summary

This chapter has introduced the fundamental regards to software component integration, programming learning, teamwork, online learning, and expert interviews. The introduced knowledge areas were considered in the design of the Web app. The important ones in context of this thesis are the Big-Bang integration approach, cooperative teamwork, online learning environments, and the expert interviews.

As the author in [27] claims *"Collaborating programmers should use a development environment designed specially for collaboration, not the same one designed for solo programmers with a few collaborative processes and tools tacked on."*

So, it is necessary, too, that for programming learning in cooperative teams, learners should use those kind of environments, which are designed exactly for the purpose: programming in cooperative teams.

3 Methodology

In the following the methodology of this thesis will be introduced.

3.1 Identification and Analysation of Problems with Literature Analysis

First, this thesis will determine the areas of the researchable topic. The topic should have relevance for the study Didactic of informatics. Therefore, the following areas will be reviewed: online learning, Computer Supported Online Learning Platforms, software engineering, and teamwork.

3.2 Requirements Derivation

After the analysation of the identified problems, some possible solutions will be determined. It will be analysed how the problem could be solved. Based on it, the determination and derivation of requirementswill follow.

3.3 Features Derivation

After the requirements are derivated, some possible functions for the requirements will be determined. It will be analysed, which functions could fit the requirements. Before that some existing features of online platforms are reviewed, too. It will be reviewed, what kind of existing features of online platforms exist, mainly in the field of software engineering. Based on it, the derivation of possible features of the Web app will follow. Traceability matrix will be created for the relationships between the identified problems and the requirements, and between the requirements and functions of the Web app.

3.4 Web app Design

This section introduces the methodology of the Web app design. The target group, goals, expected impacts, main features methodologies of the Web app are discussed here. Furthermore, the programming task and the workspaces of the Web app will be introduced here.

3.4.1 Determination of Target Groups

Based on the above-mentioned research, the target groups of the Web app will be determined. It is important in connection with the features of the Web app.

3.4.2 Determination of Goals

The main learning goals of the Web app will be determined here, based on the above-mentioned research and the target group.

3.4.3 Determination of Expected Impacts

The expected impact will be determined here based on the above-mentioned research, the target group and the learning goals.

3.4.4 Determination of the Programming task to be addressed by the Web app

From the above mentioned features the so-called mandatory features will be selected in order to get a complete system for this thesis, that can be evaluated. In the case of choosing the right topic for the programming task to be addressed within the Web app, the most important factor will be to find that kind of task, which could be well divided into subtasks. It is important that the subtasks should be divided into equal parts, as the subtasks need to have a similar difficulty level. The topic of the task should be a modern, engineering-related topic. Furthermore, the following actions are realized here:

- Determination of the main structure of the programming task: The main directions of the programming task will be determined here and the learners will have the possibility to choose from those processes.
- Determination of the data used by the programming task.
- Determination of the operations of the programming task to be addressed.
- Determination of the Individual Process Steps (also Components) and their connections: Every learner has a process-step of the programming task, whereby the step will be implemented as a component of a software system. The components (or an individual process-step) will have a uniform form. The teamwork of learners will have to learn to link those process-steps together, in order to get a fully working software system. The components will be communicating with each other through their interfaces. Those interfaces will be determined here.
- Creating the State Diagram: Based on the characteristics of the components and their connections the state diagram will be designed.

3.4.5 Individual Workspace Design

In the individual workspace the available components will be listed. From those components the learner can choose in order to program it.

3.4.6 Cooperative Workspace Design

In the cooperative workspace the components will be showed. Here the main task of learners will be to link those components together. This is the main point of the thesis - how learners should communicate when linking components to get a fully working software system.

3.5 Evaluation of System Concept of the Web app

In the next the thesis describes the method regards to the evaluation of the Web app.

3.5.1 Creating of the Interview Questions

First, the main fields of professional areas will be determined, based on the performed research, on the identified problems, and on the resulted Web app.

The questions will be open questions. From the main areas of question, the certain questions of the interview will be derived. In the case of determination of questions, it will be important that the topics of questions should cover all the mandatory features of the system concept of the Web app.

3.5.2 Search and Selection of Interviewee

Appropriately to the professional fields, which the system concept of the Web app covers, the interviewee will be chosen from those professional areas. The selected interviewee will be asked via email to give an interview.

3.5.3 Interviewing

Before the real interview is taken, a probe interview will be performed. Here the aim will be to check whether the presentation is clear and if it conveys all necessary information. Another goal is to find out, whether the questions are suitable or not. After the correction of them, the real interview will be taken part.

The interviews will be recorded. Before the interview a short introduction to the interview and a presentation will be given. It will be imparted that the questions are classified into categories, too.

3.5.4 Transcribe the Interviews

The interviews will be transcribed. With the help of the recorder, the voice content of each interview will be typed into a textual document. Both the probe interview and the real interviews will be transcribed.

3.5.5 Creating Report

A systematized summary of the interviews will be put in writing. Both the probe interview and the real interview will be written down. Then communalities, similarities, divergences, conflicting and individual thoughts will be drawn and noted down.

4 State of The Art

Next, this thesis introduces some existing online platforms. The author of this thesis has classified those online platforms into some categories.

4.1 Online Learning Platforms for Teaching Programming

Codecademy.com is a free online platform, which offers interactive lessons for basics of programming in Python, PHP, jQuery, JavaSkript, Ruby and in mark-up languages. This platform is well organised and it is easy to navigate in it. The fascinating and motivating properties of this platform are the following: providing feedback to learners, consecutive tasks, live demonstration of tasks, badges for completing exercises and the possibility to create courses, and interacting with other members. The codeavengers.com is used in high schools for teaching programming with web technologies. The codehs.com is an online platform, for programming and computer science. Pex4fun.com is an open, browser-based platform used for teaching and learning .NET programming languages, which is based on interactive-gaming. Codeschool.com is an online interactive platform, which teaches programming (Ruby, JavaSkript, HTML/CSS, IOS) with live programming tasks and video lessons.

In contrast with the Web app designed in this thesis, these online platforms for teaching programming do not offer a possibility for cooperative team work.

4.2 Concepts for Teaching of Programming Online

In Nordakademie a concept for a new programming course was created for motivating students learning to program [11]. Within this new course (which is an integration of the courses Programming 1 and Internet basics) students must develop web-applications from the beginning of the course [11]. For the supporting of teaching of the course, basics of programming (mainly functional programming) so-called ProgrammingWiki, a special Wiki-System, was developed and is being applied since 2008 at Zittau/Görlitz University in computer science courses. Source codes can be interactively implemented with this tool (edit, save, implement and evaluate of source codes) [32].

In contrast with the Web app designed in this thesis, these concepts for teaching of programming online do not offer a possibility to cooperative team work and to learn the software integration processes.

4.3 Automatic Assessment Tools

Automatic assessment tools (or online judges) are widely used in teaching programming [81]. But these platforms mainly do not pay attention to instruction materials or to the communication.

SPOJ is an online judge and contester system, which is suitable to teach online computer programming, but it does not provide suitable instructional materials [55]. It has a repository with different programming issues, a judge management system and a user management system [55].

In [38] the author introduces an environment, in which the coding style evaluation, design pattern recognition and plagiarism detection are included.

ZawodyWeb System is used for automatic validation of codes. The students remotely write their codes and send them to the teacher and the codes will be evaluated by the system. It makes italso possible to create assignments, to submit, compile and run programs [71].

So called online judges can be integrated into Learning Management Systems, too. Those resulted systems make it possible to evaluate source codings remotely and to access educational materials. The author presents the integration of Mooshak Open Source Evaluation Engine into Claroline Learning Management System in [26]. The Mooshak system is used for programming competitions as an online judge [26]. In [23] the author presents an assignment management module based on the Moodle Learning Management System.

In [8] the author proposes an automatic assessment as a service, which has the advantage, that it is independent from any Learning Management Systems.

In [19] Pythia, a web-based learning platform is presented, which has a solution grader and is suitable for teaching programming and algorithm design. It also provides direct feedback for students.

The next paragraph is quoted from the literature [3]. The author here proposes an automated assessment system, which was used in a C programming tools course at University of Murcia by extending Mooshak with activities specified for programming tools. The aim of the experiment was to compare the effect of using the automated assessment system in a programming course and the face to face teaching in classroom in the case of testing, debugging, versioning skills. Its compiler is installed on the server, like that it is suitable for using any languages. It extends an automated assessment system with programming tools assignments. This system is grading qualitative factors with an external program, called static corrector, according to quality factors, like efficiency and legibility. During the evaluation a predefined set of inputs codes is used. But this system does not offer automated assessment for grading the robustness and legibility of the source code. ASAP (Automated System for the Assessment of Programming) is based on Java. It assesses programs automatically but also provides learning materials and tools.

In contrast with the Web app designed in this thesis, these automatic assessment tools concentrate mainly on assessing the source code of learners and do not offer a possibility to cooperative team work and to learn the software integration processes.

4.4 Collaborative Learning Tools

HabiPro is an adaptive environment, which offers collaboration and interaction for a group of students in the field of computer programming and it offers different pedagogical methodologies and exercises, for developing good habits in programming [88]. It has the goal to promote skills of students, like observation, and reflections. There the group of students work collaboratively on programming problems, write and correct programs code together, and propose a solution of programming problems together [88].

A complex learning management system especially for computer programming is provided in [70], where a web-based virtual education system is presented for learning C Programming language, which has embedded code execution and assessment capabilities.

The next paragraph is quoted from the literature [51]. Mideaas is a simple web-based integrated development environment (IDE) for learning collaborative software development. It is a cloud-based IDE, which was developed at Tampere University of Technology. The IDE *"is integrated with a hosting solution into which completed applications can be deployed as a service"* and it has two parts. One part is the visual user interface. With the help of it the user can develop new user interfaces, with dragging and dropping the different elements and layouts. The other part of the tool is the code editor, where users can program the necessary functionality. The UI is stored as a XML documents.

In contrast with the Web app designed in this thesis, these learning environments do not offer cooperative learning possibilities and they do not offer the possibility to learn the software integration processes.

4.5 Online Learning Platforms using Gamification

curatr3.com is a social learning platform for business. It is a *"Software as a Service (SaaS)"* development. It makes it possible to transform a learning content into a social game in a fast way. The gamification is realized as followed: the learners earn experience points, by viewing, commenting and contributing learning content. If the learners earn enough points, they can go to the next level and access new learning contents. It works along existing LMS without complex integration. This is used as a social layer for the different kinds of professional contents. It can be customized, according to the user wishes and it also made creating customized gamification possible. It offers social learning tools like object-oriented discussions, group chats, peer reviews, and content curation. It also contains tools for the assessment of the learners: multiple choice questions, discussion questions, peer marking, and upload assignments. It uses Mozillas Open Badge Protocol to issue badges for behaviour, competency, and completion.

The next paragraph introduces a prototype and it is cited from [25]. The prototype called StudyAid, is for an E-learning system using game mechanics. Within this work the author *"explores the opportunity for a new E-learning approach to Game Mechanics based e-learning, which is a mix of CSCL (Computer-Supported Collaborative Learning),CBT (Computer Based Training), and Gamification"*. The prototype is implemented on web application Ruby on Rails. It has a student-centric approach (the students create questions and answer them by their own) and its aim was to build such an e-Learning system, which uses Web 2.0 technology, and centres the cooperation and sharing. Its further aim is to learn with fun and self-driven, and to be motivated. This system was designed so, that the computations are carried out server-side, therefore the system reacted slowly sometimes. The prototype is based on progression/points game mechanics. This was realized through implementing a score counter and a progress bar, which measured experience points (XP). During the evaluation usability and usefulness was evaluated. The system was evaluated in a course Communication Services and Networks at Norwegian University of Science and Technology.

The next paragraph gives an overview on platforms from the gamification field, cited from the literature [61]. In Gamification Platform Matrix the author lists some popular gamification platforms, which have mainly business goals, but they can also be used for educational goals. For example GamEffective is used by enterprises, for engaging employees in sales and costumer services. Cloud captive platform also uses gamification to motivate users. It makes it possible to use gaming elements (points, badges, and live notations) to gift the users. Iactionable is a Software as a Service gamification platform, which uses game mechanics to non-game application. Punchtab is a so-called instant loyalty platform, which was made for website designers and developers to create *"a social and mobile-enabled loyalty program"* within a short time period. SAP Gamification Platform gives a enterprise level platform performance and scalability. It offers gamification concept into enterprise systems, having a sophisticated rule managements and advanced analytics.

In [81] the author proposes an e-learning platform for learning computer programming for a group of students and this platform assumes active teacher involvement. It increases student motivation through gamification mechanism. One major activity of the platform is grading students and the programming exercises here have two types: for individual students and for groups [81].

In contrast with the Web app designed in this thesis, those online learning platforms that apply gamification do not offer cooperative learning possibilities and the possibility to learn the software integration processes.

5 Requirements Analysis

This chapter gives an overview of the requirement analysis used in this master thesis. First, the identified problems, then the derived requirements, and the features of the Web app, and finally the traceability matrixes are described.

5.1 Problems Identification

In this section seven problems are identified. The literature analysis has resulted in those identified problems. At the time of 2015 the following conferences were reviewed: the last 3 years in the field of software engineering and e-learning conferences like ICSE, SECR, ASE, SECRU, ICDLE, ICDEL, and some web search in connection with online platforms in the field of software engineering. The determination of areas of unsolved issues followed. It was determined that teamwork in the field of online learning and software engineering is one of the biggest issues. Within teamwork, the gaps/problems were analysed. In the field of teaching software engineering and the software integration processes the problems/gaps were identified and analysed.

5.1.1 P1: Lack of Feedback

Students of introductory programming get only few feedbacks on their learning process. Students often *"overestimate"*their knowledge and it can result in lack of motivation [56].

This paragraph is cited from the literature [56]. Against the students, the teacher is convinced, that the acquirement of course materials is difficult for the students. Students are not able to recognize all their problems that appear during programming learning. But the teacher, with the help of different assessments possibilities can assess the strength and weaknesses of the learning process of students much better. The knowledge and the experience of the teachers in programming are also an advantage to evaluate the knowledge of students.

A student performance assessment gives students reports and feedback about their learning and it can help students being motivated for completing the next task [58]. In general, performance of students can be measured through oral (with presentation), written (with examination) and practical way of assessment (practical task) [58].

The evaluation of the performance of students is desired in the case of programming, too. Performance-based assessment is one of the preferred methodologies in the case of programming. It has a motivational effect when students get a clear picture of their performance [67]. Automatic assessment tools or online judges are widely used in teaching programming [81]. Those tools have relevancy, if many students are present in a programming class [73]. The codes written by the novices are assessed automatically, easing the workload of the teacher [73].

5.1.2 P2: Lack of Motivation

Students, who attend the first programming class at universities, can easily lose their motivation on learning to program [2]. In [67] the author claims, that in the beginning of programming learning the motivation of students diminishes rapidly and it is rapport with the appearance of the above-mentioned difficulties of programming.

As the author claims in [2], programming learning is a difficult task. It is typical for the programming process that unique and unexpected situations happen [13]. One of the crucial problems arising in the teaching of programming is the high drop-out rate [56]. To construct a fully working program, it is important to understand the major directions of computers well, otherwise students will be dissatisfied during their learning process [67]. To conceive of the theoretical phrases of programming like variable, or phrases of memory can cause difficulties for the students, too. It is hard for students to connect those phrases to something in the real life [67].

The consequences of losing motivation during the programming learning is the following: students stop to learn and thus they will be success-less during their studies and in the future. They refrain programming and they choose an other profession [67]. Fear from the programming can be the consequence of being frustrated, too [10].

5.1.3 P3: Inefficient Teamwork

Another problem is the inefficiency of teamwork, the communication and coordination problems in software development process and in programming learning.

In software engineering the social aspects are becoming more and more important [50]. During teamwork, software developers write source code together in a team and communicate with each other [50].

This paragraph is quoted from the literature [37]. Not suitable teamwork can cause the break up of software development projects. A lack of suitable leadership means that a problem exists in a software development project. For keeping deadlines, making plans, managing the team work needs proper guidance. Reluctant work of a team member causes a problem, too. Another problem derives from the modification of the software code by team members. This modification is hard to follow up for the software developers and it is hard to consider the original requirements.

Some team members in student projects can be truant and the work of those kind of student needs to be performed by other team members. Thus, the teamwork will be not efficient at the end [75].

In student projects, when two students work together in order to solve a programming task, ineffective or unfair peer evaluation can be a problem, as well, and if a motivated team member feels that he or she gets a wrongful evaluation during a team work, it can result in a competition between the team members and those competition situations can destroy the collaboration inside a team [37]. The effective team size of software development project are 4 to 8 developers [37].

The frequent and proper communication is an important aspect in a student project, too [39]. Collaborative learning in the field of programming has several advantages [39]. It improves creativity, social and communication skills [87]. Students will be much more motivated and they become rather critical thinkers [87]. The collaborative learning or learning in pairs has a crucial significance, otherwise students can feel programming is too difficult [75].

If students learn programming in a collaborative learning environment, their performance will be higher [87]. Applying group work in the case of programming has resulted in the reduction of drop out rates of students from the programming class, too [75].

5.1.4 P4: Applying the Gained Knowledge

A other difficulty is that students of introductory programming have problems applying their gained knowledge [56].

Companies criticize that young workers, who have just finish university and started to work in computer related professionals, do not have all the necessary skills for their work [10]. As the author claims in [75], companies prefer so called *"technology savvy students"* with good collaborative skills, who like to work with other workers together, even from other countries.

A software engineer deals with complicated tasks and those specialists need to deal with multiple domains, like software requirements, architecture of a software, human computer interfaces, and usability [22]. They need to have the knowledge of several programming languages, they need to know how networks and databases work, and how a software should be tested [22].

In a programming course students learn through practical tasks, exercises and student projects [24]. But they have different problems in programming learning. Some students have difficulties solving programming problems, for example with converting pseudo code into a working program [67]. Other students have difficulties making a picture in their mind about the algorithms, in which they work [14]. Students also have difficulties building a program from the components [47]. It is also challenging for them to convert programming problems into a fully functioning program, which has the appropriate syntax [84].

Students of programming need to know how the following actions works: designing programs, writing source code, repairing it, and fixing programming errors [84]. The author claims in [78] that during the programming education course developers need to concentrate on developing the problem-solving skills of students, in order to enable them to apply at companies later, where they will work.

5.1.5 P5: Inflexible Learning

The other problem this thesis discusses is that traditional, presence learning is quite inflexible.

Traditional, presence learning has several disadvantages, for example it does not motivate students insufficiently for the active participation in the learning process, it does not take the different learning styles of them in account [30]. It does not support students to develop skills like problem solving and critical thinking [30]. Another disadvantage of traditional, face to face learning is that it is hard for students to apply the knowledge and skills, they have gained in classroom environment, into the practice, into the working life [54].

Learning habits of students nowadays have changed, they prefer using mobile phone for their studies and the communication happens online with other participants of learning [75]. With the help of the web, they can get a lot of information, they can share documents and files, use online forums to communicate with each other and to share ideas [75].

Nowadays the need for the flexible online learning environments has been increased and the main characteristic of those pliable learning environments are the place and time independence, therefore students can learn anytime and anywhere they want [52]. Those flexible environments help to improve the performance of students in the learning process and make students much more contented [52]. Those learning environments ensure interactivity for student and give much possibility for collaborative problem solving [52].

The active participation of students in the learning process is important in programming, as well [10]. In this case, the face to face lecture is not necessary and those flexible learning environments can motivate students for learning programming, too [10].

Modern online platforms for programming teaching can be used for several purposes, like collaborative problem solving, automated code assessment, plagiarism detection or electronic management of the class materials [73]. Another advantage of online learning versus the traditional learning is that online learning is much more learner-centered [65]. The author claims in [65], that more and more students prefer taking part in online courses.

5.1.6 P6: Software Systems are Complex

The other problem, this thesis deals with, is the complexity of software systems.

During their studies, students need to prepare solving those kinds of complicated tasks. After their studies, students need to face the multitude nature of the tasks to be solved. The market today requires developing more and more complicated software systems [59]. Software developers can construct different kinds of software for a very wide range of usage. For example software developers can establish simple databases, but complex ERP-Systems. They can develop software to operate robots, or to operate embedded systems, like mobile devices. A software can ease the work of the user in a very wide range of field. [28]. For example, working with developing IT Systems for healthcare, industry, but also in the security field, and for governmental, or commercial goals eases the work[28].

It is important to mention that the growing number of free libraries make it possible for the software developers to write more and more complex programs [79]. The growing requirements like rapid production deadlines, good quality, thrift and the conformation to the always changing needs, all result in the need of creating complex software systems. [62].

In order to produce those complex software systems, software developers must write plenty line of source codes. In order to save data of those systems, data must be saved in large computer systems and running the networks, in which computer systems work, means a huge task for professionals, too [16]. The appeared mobile technologies, that are used by humans in the every day life, give software developers lots of task to solve, too [16]. Industry 4.0 with its smart factories and Education 4.0 are nowadays one of the newest and often mentioned issues, which means they all use the wide possibilities of technologies in the industrial and education fields, in order to be much more effective [15] [12]. Those areas give software developers a complex task to code.

5.1.7 P7: Big Bang Approach does not Work Correctly

The next problem, which this thesis discusses, is the problem of the Big Bang approach.

Software testing is a crucial part of software development and the objective of it is to find and recognize errors, decrease them into software systems and to guarantee a good quality of software systems [40]. With applying testing, the quality of the software will be higher [40].

Big Bang is an integration strategy, which aims to build a complete system from software components and to test the system [28]. As it was mentioned earlier, the disadvantage of Big Bang approach is that early integration and test of the system can occur only later, thus it is very hard to identify and find the mistakes and errors in the system [28]. A further disadvantage of the Big Bang approach that the restoration of errors is expensive and the separation of the found errors is problematic [49]. As the author says in [49] the *"observability, diagnosabilty, efficiency and feedback"* is low in the case of the Big Bang approach.

As the author claims in [40], *"to achieve high level of reliability, maintainability, availability, security, survivability, portability, capability, efficiency and integrity the system must need to be properly tested"*. Consequently, if the system testing does not work properly, then we can not get a fully working software system.

5.2 Requirements Derivation

In the next this thesis introduces the requirements derivation from the above-mentioned problems. This requirements will be a base of the features of the designed Web app. For one problem, more requirements were formulated.

5.2.1 R1: The Web app should give for Learners continue Feedback about the Learning Process

This requirement is stem from the P1 problem - lack of feedback.

In this Web app learners should get assessment about their work during the learning process. In general, in the case of programming subject learners can be assessed by automatic assessment tools, which helps them to assess the implemented source codes. Thus, learners can get informations about their syntactical and semantic failures.

5.2.2 R2: With the Web app Learners should learn in Team

This requirement is stem from the P2 problem - lack of motivation.

In the Web app, learners should work in teams. They should implement the task so, that they solve it in a team. They think together on programming problems and find solutions together. They can work collaboratively or cooperatively in teams, too.

5.2.3 R3: Learners should learn through Gamification Elements during their Learning Process in the Web app

This requirement is stem from the P2 problem - lack of motivation.

Competition between teams is a often used method in programming learning, too. If there is a common goal to reach, it has a motivational effect to the learning process.

5.2.4 R4: With the Web app Learners should understand easily the Programming and the Software Integration Processes

This requirement is stem from the P2 problem - lack of motivation.

As the programming is difficult to learn, therefore the Web app should ease its learning process. The Web app should teach programming interactively. Visualization of programming elements is needed, too. Software integration is a new topic in the software engineering education. The Web app should represent the following processes in order to better understand them: linking components together and form a whole system.

5.2.5 R5: The Web app should contain interesting, motivating Elements

This requirement is stem from the P2 problem - lack of motivation.

The Web app should use colours, visual, interactive elements, to attract the attention of learners in order to involve them better in the learning process. The Web app should make the learning process engaging.

5.2.6 R6: Learners should use the Web app easily

This requirement is stem from the P2 problem - lack of motivation.

The Web app should be easy to understand for the learners, thus learners can involve themselves into the learning process much better. Learners should learn to use the Web app intuitive.

5.2.7 R7: The Usage of the Web app should result the Development of Teamwork Skills of Learners

This requirement is stem from the P3 problem - inefficient teamwork.

With the usage of Web app, learners should simulate how a software development works in the reality by companies and how to work and think together on programming problems with other software developers.

5.2.8 R8: With the Web app Learners should follow-up of Modifications of Source Codes

This requirement is stem from the P3 problem - inefficient teamwork.

The Web app should provide the possibility for students to go back to a previous version of the software, during the implementing of source codes (even if more programmers work on a same source code), in order to correct errors that arise during software development. This process should be easy to perform.

5.2.9 R9: With the Web app Learners should work and learn in Team, where 4 to 8 Team Members are present

This requirement is stem from the P3 problem - inefficient teamwork.

In the Web app 4 or 8 learners should work together to build a whole software system. The teamwork means in the case of this Web app, that learners work together in order to integrate components to build a functional software system.

5.2.10 R10: With the Web app Learners should be able to communicate with Each Other frequently

This requirement is stem from the P3 problem - inefficient teamwork.

The Web app should give a possibility for the learners to work in teams, where they can communicate with each other frequently, changing ideas, and views via communication tools (i.e. chat, forum).

5.2.11 R11: With the Web app Learners should work remotely Together on the Same Source Code

This requirement is stem from the P3 problem - inefficient teamwork.

In order to link and integrate software components together, learners should see the source code together with other team members and work on it together with the help of this Web app.

5.2.12 R12: With the Web app Learners should learn online the Programming Subject

This requirement is stem from the P3 problem - inefficient teamwork.

Learner should work with the Web app online, remotely from other learners in order to implement a fully functional software system and they should communicate to solve programming problems through synchronous and asynchronous communication possibilities.

5.2.13 R13: The Teamwork should be well managed on the Web app

This requirement is stem from the P3 problem - inefficient teamwork.

In this Web app the teamwork work should be well managed. For example, a moderator is needed to direct and guide the communication between team members. A task divisions-plan would be necessary to manage the teamwork, as well.

5.2.14 R14: The Web app should ensure the equal division of labour between Learners

This requirement is stem from the P3 problem - inefficient teamwork.

The programming task to be addressed within the Web app should be designed so, the programming task should to be split up so that every learner should get an equal difficulty of programming task, in order to be equitable at the evaluation of their learning process.

5.2.15 R15: The Web app should ensure the suitable Evaluation of the Programming Tasks for the Learners

This requirement is stem from the P3 problem - inefficient teamwork.

The programming task performed by the learners should be well evaluated in order to mark the performance of the learners.

5.2.16 R16: The Web app should support Learners in Learning Programming

This requirement is stem from the P4 problem - applying the gained knowledge.

The Web app should provide programming tasks for learners, where learners can apply the gained theoretical programming knowledge, which they aquired from university studies and they should try out their knowledge in practice, with the help of the Web app.

5.2.17 R17: The Web app should help to prepare Learners to work in IT-Profession / in Software Development Projects

This requirement is stem from the P4 problem - applying the gained knowledge.

The Web app should contain those kind of tasks and it should ensure that kind of environment, where learners are well prepared for the working life. For example, teamwork and the modern software engineering related tasks should be practised with the Web app.

5.2.18 R18: With the Usage of the Web app Learners should get several new Skills

This requirement is stem from the P4 problem - applying the gained knowledge.

With the usage of the Web app learners should get the following skills: teamwork skills, creative thinking, and attention to detail, system thinking, communication skills, and online learning.

5.2.19 R19: With the help of the Web app Learners should apply well the gained Theoretical Knowledge into the Practice

This requirement is stem from the P4 problem - applying the gained knowledge.

In order to fit to the professional needs of companies, this Web app should pay more attention providing learners to get practical knowledges of programming, to get knowledge on how to communicate during a software development project, and to get knowledge about modern software engineering issues.

5.2.20 R20: The Web app should convey to Learner modern Knowledge of Software Engineering (Integration, Modularization etc.)

This requirement is stem from the P4 problem - applying the gained knowledge.

In order to fit to the professional needs of companies the Web app should pay more attention to providing learners to get knowledge about modern software engineering issues. Those modern issues are for example the software process integration and the modularization of programs.

5.2.21 R21: The Programming Task, the Web app contains should be split up into smaller Tasks in order to understand the Task better.

This requirement is stem from the P4 problem - applying the gained knowledge.

The programming task to be addressed within the Web app should be designed so, that the programming task should be split up into smaller tasks that every learner should get an equal difficulty of programming task, in order to learn the software integration processes.

5.2.22 R22: With the help of the Web app Learners should understand how to convert Pseudocodes into a working Program

This requirement is stem from the P4 problem - applying the gained knowledge.

The Web app should contain that kind of functions, where learners can practice how to write pseudo-codes and how to create from programming elements a working program.

5.2.23 R23: Through the Usage of the Web app Learners should imagine better Algorithms in their mind

This requirement is stem from the P4 problem - applying the gained knowledge.

Visualization is important during the learning process in the case of programming learning, too. Learners of the Web app should take a picture in their mind about, how the whole program will work, in order to create a fully functional program.

5.2.24 R24: Through the Usage of the Web app learners should understand how to convert a Problem into a working Program

This requirement is stem from the P4 problem - applying the gained knowledge.

The Web app should prepare learners to work in companies, where they meet with programming problems to solve them and then they must create a fully functional program.

5.2.25 R25: With the Web app Learners should understand how to design Programs, how to write Source Codes

This requirement is stem from the P4 problem - applying the gained knowledge.

The Web app should prepare learners to work in companies. Learners should know how to design programs and how to create a fully functional program.

5.2.26 R26: With the Web app Learners should understand how to fix Programming errors

This requirement is stem from the P4 problem - applying the gained knowledge.

The Web app should help learners in the case of programming errors, to fix them and to make this process easier.

5.2.27 R27: The Web app should apply visual Elements

This requirement is stem from the P5 problem - inflexible learning.

In order to make the learning flexible and independent from time and place, the Web app should contain those kind of visual elements that help in the interaction between learners, even if they remotely work with each other.

5.2.28 R28: The Web app should be applied in Online Learning

This requirement is stem from the P5 problem - inflexible learning.

Learners of the Web app should not sit beside each other during the learning proces. They can use the Web app from different locations, thus they can fulfil the task on different appointments and working in teams with the Web app.

5.2.29 R29: The Web app should ensure Learners the active Participation in the Learning Process

This requirement is stem from the P5 problem - inflexible learning.

In order to engage learner in the learning process and make them motivated, the Web app should contain interesting tasks to solve and should apply teamwork.

5.2.30 R30: The Web app should be that kind of Educational Web app, which fits to the needs of modern Learner

This requirement is stem from the P5 problem - inflexible learning.

The Web app should help modern learners to learn programming in a modern way, learning the subject mobile, working in team, and handling interactive applications.

5.2.31 R31: The Web app should ensure the time and place Independence of Learning Process

This requirement is stem from the P5 problem - inflexible learning.

The Web app should provide a time and place independence for learners during their learning process.

5.2.32 R32: The Web app should be learner-centered

This requirement is stem from the P5 problem - inflexible learning.

The requirement of learners should be favoured in the Web app and the focus should be on the needs of the learners when designing this Web app.

5.2.33 R33: The Web app should ensure cooperative Problem solving for Learners

This requirement is stem from the P5 problem - inflexible learning.

In this Web app the programming task addressed should be split up into smaller tasks, that learners will be able to practice the cooperative problem solvings, which means, individually work on a task, i.e. on software components and then learners turn to work in a team, realizing the integration of software components together.

5.2.34 R34: The Task (Topic) of the Web app should be easy to change, in order to be able to practice for Learners more Programming Task

This requirement is stem from the P6 problem - software systems are complex.

With the Web app learners should practice more kinds of programming tasks, to get to know more areas, and to get more experience.

5.2.35 R35: The Topics of the Programming Task of the Web app should arise from different kind of Domains

This requirement is stem from the P6 problem - software systems are complex.

With the Web app learners should practice more kinds of programming tasks, in order to get to know more areas and to get more experiences.

5.2.36 R36: With the help of the Web app Learners should be able to reuse Codes

This requirement is stem from the P6 problem - software systems are complex.

The Web app should provide the possibility to reuse codes, or a section of codes, existing functions, and procedures, in order to save time and energy for learners.

5.2.37 R37: The Web app should convey Component-based Programming for Learners

This requirement is stem from the P6 problem - software systems are complex.

The Web app should provide a possibility for learners to build systems from components, which are strongly depending on each other, and those components communicate with other components through the interfaces.

5.2.38 R38: With the Web app Learners should learn modularity

This requirement is stem from the P6 problem - software systems are complex.

The programming task of the Web app should base on the modularity, where a system is divided into interdependent and independent modules/components, which reduces the complexity of the software system.

5.2.39 R39: With the Web app Learners should be able to learn Software Integration Processes

This requirement is stem from the P6 problem - software systems are complex.

With this Web app learners should learn linking components or subsystems into a whole, fully functional software system together.

5.2.40 R40: The Web app should be mobile to use

This requirement is stem from the P6 problem - software systems are complex.

The Web app should give the learners the possibility to learn online, independent from time and place, and the ability to interact and communicate with each other.

5.2.41 R41: The Web app should improve the Communication Process between Software Developers

This requirement is stem from the P7 problem - the Big Bang approach does not work correctly.

The Web app should give the possibility for interaction frequent communication between the learners. The main point of this thesis is to research what kind of communication is necessary between the learners in order to create a fully functional software systems at the end.

5.2.42 R42: The Web app should convey the Software Integration Processes

This requirement is stem from the P7 problem - the Big Bang approach does not work correctly.

The Web app should contain those kind of tasks, where learners can learn how to link components together and how to communicate with each other, during this process.

5.2.43 R43: The Web app should ensure the Early Integration of Software Components

This requirement is stem from the P7 problem - the Big Bang approach does not work correctly.

The Web app should provide the easy integration of software components.

5.2.44 R44: The Web app should support identifying and finding Mistakes and Errors in the whole Source Code.

This requirement is stem from the P7 problem - the Big Bang approach does not work correctly.

The Web app should be that kind of tool, where it is easy to find and identify programming errors.

5.2.45 R45: The Web app should support Error Separation

This requirement is stem from the P7 problem - the Big Bang approach does not work correctly.

In order to find errors and restore them, the Web app should have that kind of function, where errors can be separated.

5.2.46 R46: The Web app should ensure the efficient Integration of Components

This requirement is stem from the P7 problem - the Big Bang approach does not work correctly.

In the Web app, during the software development process, it should be ensured that the components are linked together efficiently. The Web app should bring an attention to the relation between implemented software and the applied sources, funds (financial, human-related), too.

5.3 Features

This section shows the features of the Web app. These functions were derived from the requirements above. One feature can come from more requirements. Mandatory and future work features are also determined here.

5.3.1 F1: Peer/Group Source Code Review

This feature comes from the first requirement - The Web app should give continue feedback about the learning process to learners and 15th requirement - The Web app should ensure the suitable evaluation of the programming tasks for the learners.

In the case of peer/group review some learners band together in order to assess the implemented source code of each other. So, it is not the task of the teacher/instructor to review the work of learners and assess them. The Web app should support this process. It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.2 F2: Self-assessment Option

This feature comes from the first Requirement - The Web app should give continues feedback about the learning process of learners.

In the case of this feature the learners will assess their work on their own. Learners can resort the Web app to assess the source code and they can immediately see the result of the assessment.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.3 F3: Ability to work in Cooperative Teams to link Individual Components into a whole Software System together

This feature comes from the second requirement - With the Web app learners should learn in a team, the 7th requirement - The usage of the Web app should result in the development of teamwork skills of learners, the 11th requirement - With the Web app learners should work remotely together on the same course code, the 29th requirement - The Web app should ensure learners the active participation in the learning process, the 33rd requirement - The Web app should ensure cooperative problem solving for learners, the 37th requirement - The Web app should convey component-based programming for learners, the 41st requirement - The Web app should improve the communication process between software developers, the 43rd requirement - The Web app should ensure the early integration of software components, and the 46st requirement - The Web app should ensure the efficient integration of components.

The Web app is so designed, that there is a possibility to work together in a group of 4 to 8 learners. The Web app contains a task, which is split up into subtasks, whereon learners can work individually. The subtasks represent the components of a software system. Learners can form cooperative teams, where they work on integrating, linking together the individual components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

The following picture shows the state diagram of the programming task to be addressed within the Web app.

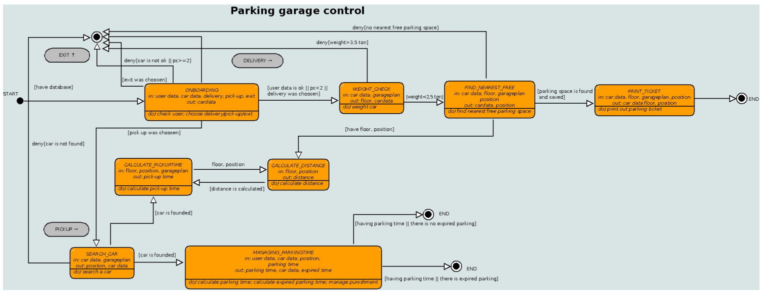

Figure 5.1: Whole Software System

5.3.4 F4: Competition between Teams (with Badges, Scores, and Leader-Boards as rewardings)

This feature comes from the 3rd requirement - Learners should learn through gamification elements during their learning process in the Web app.

Learners form a team, in order to work together and perform the integration of components and the teams can compete. The winner can get a reward for the performance, i.e. badges, scores etc., which are visible in a chart, with a denotation of place and number of points.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.5 F5: Ability to drag and drop and snap together a Block of Programming Elements/Commands

This feature comes from the 4th requirement - With the Web app learners should understand the programming and the software integration processes easily, the 5th requirement - The Web app should contain interesting, motivating elements, the 6th requirement - Learners should use the Web app easily, the 22nd requirement - With the help of the Web app learners should understand how to convert pseudocodes into a working program, the 23rd requirement - Through the usage of the Web app learners should imagine better algorithms in their mind, the 25th requirement - With the Web app learners should understand how to design programs, how to write source codes, and the 27th requirement - The Web app should apply visual elements.

Lists of programming elements are available on the screen, classified into categories, and learners have the possibility to choose elements from it and with the drag and drop technique the programming elements can be pulled to a leer sheet, and at the end from those elements a whole program can be built, by snapping those elements together.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

The following picture illustrates the drag and drop of programming elements contained the Web app.

Figure 5.2: Drag and drop of Programming Elements

5.3.6 F6: Create/manage Teams

This feature comes from the 7th requirement The usage of the Web app should result in the development of teamwork skills of learners.

For the teamwork is necessary to form teams and manage them. This function is necessary, when there is a competition between teams. A new team can be created, giving the name of the team and its functions, and learners can join it. It is possible for learners to join to an existing group, as well. To the team new learners can be added and different rights are set, for example who can view or post, or moderate in the team. Communication and different event organization is also possible here.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.7 F7: Coordinate Group Work

This feature comes from the 7th requirement - The usage of the Web app should result in the development of teamwork skills of learners.

The work of a team has to be coordinated. For example, to coordinate, which team member works on which components, when they are ready, when they start to work individually and when they start to work in team.

It is a future work feature. It will not appear, neither in the We bapp design, nor in the expert interview and in the evaluation of the Web app.

5.3.8 F8: Version Control Option

This feature comes from the 8th requirement - With the Web app learners should follow-up modifications of source codes.

It means the numeration of source codes versions. The goal of the version control is to follow up the development/modification of the source codes of the learners during the teamwork and to set source codes in a previous state back.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.9 F9: Modular Structure of Programming Task

This feature comes from the 8th requirement - With the Web app learners should follow-up modifications of source codes.

The programming task solved by the learner in the Web app is split up into subtasks, which represent the components of a fully functional software system.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.10 F10: Modularized Contents (using Headings, logically structured)

This feature is come from the 8th requirement - With the Web app learners should follow-up modifications of source codes.

Documents, i.e. education materials and tutorials could be stored in virtual folders and learners can read/download those materials and search in the contents of it.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.11 F11: Ability to work in Cooperative Teams of 4 to 8 Learners to link together Individual Components into a whole Software System

This feature comes from the 9th requirement - With the Web app learners should work and learn in a team, where 4 to 8 team members are present, and from the 37th requirement - The Web app should convey component-based programming for learners.

The Web app is designed so, that there is a possibility to work together in a group of 4-8 learners. The Web app contains a task, which is split up into subtasks, whereon the learners can work cooperatively. The subtasks are represented by the components of a software system. Learners can form cooperative teams, where they work on integrating, linking together the individual components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

The following picture illustrates the cooperative teamwork to link individual components into a whole software system.

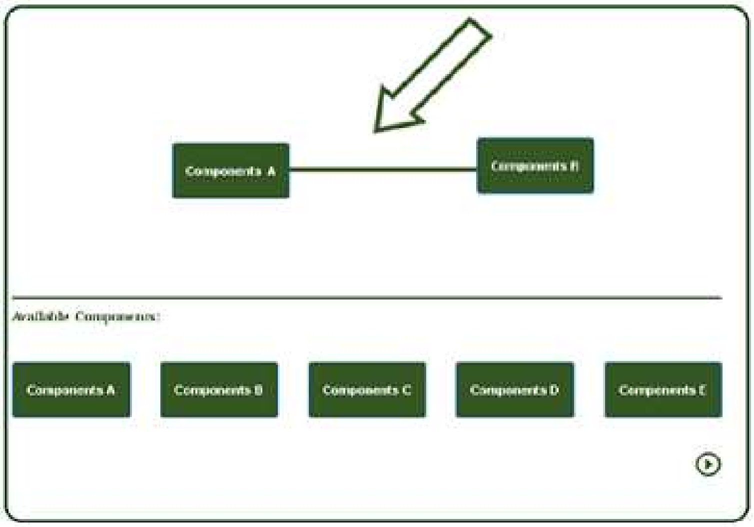

Figure 5.3: Cooperative Teamwork

5.3.12 F12: Upload/ Share audio and video Files

This feature comes from the 10th requirement - With the Web app learners should be able to communicate with each other frequently.

Within the Web app learners have the possibility to upload files (i.e. source codes, tutorials, manuals) and share them between other learners. Different accesses/permissions could be given to those files (read/view only/ share). The files could be stored in virtual, online folders, which learners could access. Learners can make an invitation to call other learners to share/edit files.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.13 F13: Modern Web-based Communication (i.e. Chat Rooms, Forum, Blog, Mailbox, Message Board, Site News etc.)

This feature comes from the 10th requirement - With the Web app learners should be able to communicate with each other frequently.

To share information, ideas, views, and solutions of a programming problem, learners could communicate via those communication channels. Topics that appear on those communication channels could be classified into categories.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.14 F14: Review and retrieve Conversations

This feature comes from the 10th requirement with the Web app learners should be able to communicate with each other frequently.

In the case of web-based communication tools it is possible to review and retrieve old conversation happening between learners/team members, with search function, and the required conversation will be apparent on the screen.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.15 F15: Ability to perform Programming Tasks online, which represent certain Algorithms with Pseudo-Codes

This feature comes from the 12th requirement - With the Web app learners should learn online the programming subject, the 16th requirement - The Web app should support learners in learning programming, the 22nd requirement - With the help of the Web app learners should understand how to convert pseudocodes into a working program, and the 23rd requirement - Through the usage of the Web app learners should imagine better algorithms in their mind.

With this Web app learners perform programming subtasks, which need to be developed so that learners work far from each other and they communicate with each other through the Web app. A subtask represents a component of a software system. Learners create, maintain source codes, and snap together pseudo-codes elements.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

The following picture illustrates the representation of algorithms with pseudo-codes.

Figure 5.4: Representation of Algorithms with Pseudo-Codes

5.3.16 F16: Moderation of Teamwork

This feature comes from the 13th requirement - The teamwork on the Web app should be well managed, and the 46st requirement - The Web app should ensure the efficient integration of components.

The moderator function directs the communication between learners/teams instructions, with its questions. The role of the moderator here is to ensure that the main line of the communication should be not left.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.17 F17: Execute split Programming Tasks

This feature comes from the 13th requirement - The teamwork on the Web app should be well managed, and the 21st requirement - The programming task, the Web app contains should be split up into smaller tasks in order to understand the task better.

One programming task is divided into subtasks and every learner has one subtask to work on. Each subtask appears in a different window. A subtask represents a component. The output of a component acts as an input for the next one. A subtask has the identical description and consist of the following information: purpose, output, input, precondition, pro-condition, operation, success case, possible errors, and contact with other components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.18 F18: Calendar/Group Calendar

This feature comes from the 13th requirement - The teamwork on the Web app should be well managed.

A calendar for individuals and for teams contains a plan for tasks- deadlines, dates of meetings etc.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.19 F19: Equal difficulty of individual Process-Step of Programming Task

This feature comes from the 14th requirement - The Web app should ensure the equal division of labour between learners.

A process-step of the programming task, which represents a component of a software system should be designed/divided so, that it should have equal difficulties compared with other components, because the fairly division of labour between learners should be ensured, keeping the goal to build up a fully functional software system.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.20 F20: Realize online Team-Learning related modern Software Engineering Issues, namely Software Process Integration

This feature comes from the 17th requirement - The Web app should help prepare learners to work in the IT-profession /Software development projects, the 18th requirement - With the usage of the Web app learners should get several new skills, the 19th requirement - With the help of the Web app learners should apply the gained theoretical knowledge into the practice, the 20th requirement - The Web app should convey to learners modern knowledge of software engineering (integration, modularization etc.), the 30th requirement - The Web app should be that kind of learning Web app, which fits to the needs of modern learner, the 37th requirement - The Web app should convey component-based programming for learners, the 39th requirement - With the Web app learners should be able to learn the software integration processes, and the 42nd requirement - The Webmapp should convey the software integration processes.

The Web app contains a task for learning the software integration process. Each learner implements a component of a software system and the teamwork means to integrate all the components together into a whole system.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.21 F21: Performing mathematical related Programming Tasks, which realizes the Operation of an Automated Parking Garage (main Process and Process Steps), to learn the Software Integration Processes

This feature comes from the 19th requirement - With the help of the Web app learners should apply the gained theoretical knowledge into the practice, the 24th requirement - Through the usage of the Web app learners should understand how to convert a problem into a working program, the 29th requirement - The Web app should ensure learners the active participation in the learning process, the 30th requirement - The Web app should be that kind of learning Web app, which fits to the needs of modern learners, and the 35th requirement - The topics of the programming task the Web app should arise from different kind of domains.

The nature of the programming task to be addressed is stem from mathematics. This task includes mathematical problems, like graphs theory, shortest way etc. The task of learners is to represent the certain programming problems with pseudo-codes. Those tasks represent components, where the output of a component acts as an input for the next one. In the case of these programming task, which realizes an operation of an automated parking garage, learners should consider the cars as the nodes of a graph and the ways between the cars as the edges of a graph.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.22 F22: The ability to perform Component-based Programming (Process-Steps, Task-Division)

This feature comes from the 20th requirement - The Web app should convey to learner modern knowledges of software engineering (integration, modularization etc.), and from the 37th requirement - The Web app should convey component-based programming for learners.

The goal with the Web app is the same as the goal of the component-based programming: learners define and implement individual components and then link together those components to establish a whole software system.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.23 F23: The ability to display and select Components from Component's List, where the User can choose Components in order to implement them

This feature comes from the 21st requirement - The programming task, the Web app contains should be split up into smaller tasks in order to understand the task better, the 23rd requirement - Through the usage of the Web app learners should imagine algorithms in their mind better, the 24rd requirement - Through the usage of the Web app learners should understand how to convert a problem into a working program, the 25th requirement - With the Web app learners should understand how to design programs and how to write source codes, the 27th requirement - The Web app should apply visual elements, the 29th requirement - The Web app should ensure learners the active participation in the learning process, and the 37th requirement - The Web app should convey component-based programming for learners.

In the case of individual work, learners choose components (also subtasks) from the offered components list and start to implement them. It is also possible to visualize/view the completed components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

The following picture illustrates the list, contains the components of the software system, whereout the components can be chosen.

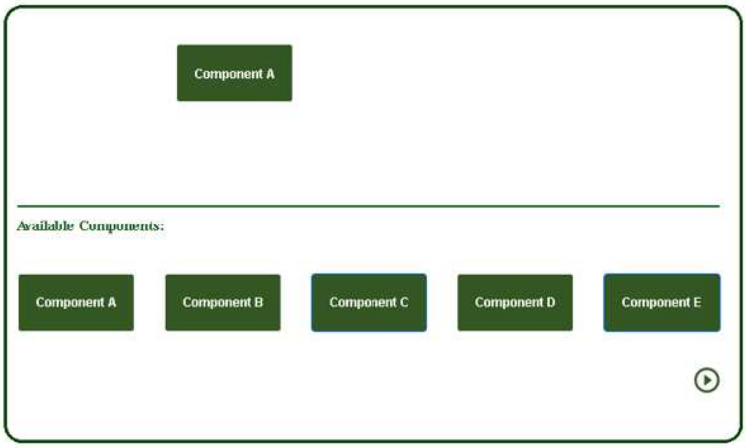

Figure 5.5: List of Components

5.3.24 F24: The ability to view/edit Components

This feature comes from the 21st requirement - The programming task, the Web app should be split up into smaller tasks in order to understand the task better, and from the 37th requirement - The Web app should convey component-based programming for learners.

In the components' list it is also possible for learners to visualize/view/edit the existing components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.25 F25: The ability to implement Components severally and together

This feature comes from the 24th requirement - Through the usage of the Web app learners should understand how to convert a problem into a working program, and from the 37th requirement - The Web app should convey component-based programming for learners.

The Web app contains a programming task, which is split up into subtasks (which subtask represent a component of a software system), whereon learners can work individually and while linking those components together they work in teams cooperatively.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.26 F26: The final Solution of Programming Tasks appears in a separate Window

This feature comes from the 24rd requirement - Through the usage of the Web app learners should understand how to convert a problem into a working program.

When the program (software system) is ready all components are well linked together and the off-come of the programming process should be visible in a separate window on the Web app.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.27 F27: Web Recorder

This feature comes from the 26th requirement - With the Web app learners should understand how to fix programming errors.

This feature is suitable for archiving webpages (where the implemented source codes are visible), for example in order to locate errors in the source code.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.28 F28: The ability to separate Errors (i.e. in a separate Window)

This feature comes from the 26st requirement - With the Web app learners should understand how to fix programming errors, the 44rd requirement - The Web app should support identifying and finding mistakes and errors in the whole source code, and the 45th requirement - The Web app should support error separation.

In order to find and identify errors appeared in the source code, errors should be highlighted so, that they are visualized in a separate window in the Web app.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.29 F29: The possibility to learn online

This feature comes from the 28th requirement - The Web app should be applied with online learning, the 31st requirement - The Web app should ensure the time and place independence of learning process, and the 40th requirement - The Web app should be mobile to use.

Learners remotely use this Web app, they learn programming far from each other, work in a team and communicate with each other via the Web app, through internet.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.30 F30: Idea Generation Organizations

This feature comes from the 29th requirement - The Web app should ensure learners the active participation in the learning process.

In order to make use of all good ideas of learners during the software development process, ideas and a suggested solution for a programming problem should be well organized. This function can be realized with an internet forum or message board, where learners can impart their views, ideas in connection with the given programming problems, and brainstorming could be organized, too. Mindmapping can be used for helping to imagine connections between programming problems. The goal is to make it easier to save, to organize, and to handle the ideas, that come suddenly from learners.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.31 F31: Upload/share/search distributed Learning Materials

This feature comes from the 30th requirement - The Web app should be that kind of learning Web app, which fits to the need of the modern learner.

This feature is a feature of a learning management system. It provides capabilities for uploading documents into a virtual folder, where learners can share those files, documents and they are able to share learning materials between the other participants, too.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.32 F32: Customizable Workspace Area (for individual and for Teamwork)

This feature comes from the 32nd requirement - The Web app should be learner-centered.

The workspace, where the learners work on components, could be modified regarding to the personal expectations of the learners. For example, styles, colours and fonts can be set individually on the workspaces, and elements of the workspace windows can be arranged according to the need of learners.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.33 F33: Individual Workspace Area

This feature comes from the 32nd requirement - The Web app should be learner-centered.

The individual workspace area contains a surface, where the code blocks are listed with a drag and drop possibility and run and trash buttons can be also found here.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.34 F34: „About me“ Section

This feature comes from the 32nd requirement - The Web app should be learner-centered.

This feature should provide information about the learners - what they do, which skills competences, education and professional background they have related to programming.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.35 F35: Identical Description of Programming Tasks

This feature comes from the 34th requirement - The task(topic) of the Web app should be easy to change, in order to be able to practice more programming tasks.

Every programming sub-task has an identical description. These descriptions are: purpose, output, input, precondition, pro-condition, operation, success case, and a connection with other components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.36 F36: The ability to reuse the Individual Components

This feature comes from the 36th requirement - With the help of the Web app learners should be able to reuse codes, and from the 37th requirement - The Web app should convey component-based programming for learners.

In the programming task addressed in the Web app, there should be components, that can be used by more than one component (or its results can be got). Thus, the source codes of those components does not have to be written again, just only one time.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.37 F37: The ability to build a completely functional Software System from Components

This feature comes from the 38th requirement - With the Web app learners should learn modularity, the 37th requirement - The Web app should convey component-based programming for learners, and from the 39th requirement - With the Web app learners should be able to learn software integration processes.

With the Web app learners have the possibility to implement the components of a software system and work in a team to build up a fully functional software system from components.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.38 F38: Error Detection Option

This feature comes from the 44th requirement - The Web app should support identifying and finding mistakes and errors in the whole source code.

An automatic error detection automatically recognizes syntactical and semantic errors in the source code and highlights them with different text colours.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.3.39 F39: The ability to implement Components, which are communicating with each other through their Interfaces

This feature comes from the 37th requirement - The Web app should convey component-based programming for learners, and from the 46th requirement - The Web app should ensures the efficient integration of components.

In the case of a successful software development it is necessary to define the interfaces between components properly. The proper communication needs to be ensured and it is crucial to research which communication would be the best in this case. This thesis searches the answer to this question.

It is a mandatory feature. It will appear in the Web app design, in the expert interview and in the evaluation of the Web app.

5.3.40 F40: Source-Observer

This feature comes from the 46th requirement - The Web app should ensure the efficient integration of components.

Developing tools are used for observing the used sources, founds and human-resources during the software development process.

It is a future work feature. It will not appear, neither in the Web app design, nor in the expert interview and in the evaluation of the Web app.

5.4 Traceability Matrices

This section shows the traceability matrices relating to the identified problems and the requirements, and the features of the Web app.

Table 5.1: Traceability Matrix: Requirements and Problems I.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
P1	X															
P2		X	X	X	X	X										
P3							X	X	X	X	X	X	X	X	X	
P4																X
P5																
P6																
P7																

Table 5.2: Traceability Matrix: Requirements and Problems II.

	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	R31	R32
P1																
P2																
P3																
P4	X	X	X	X	X	X	X	X	X	X						
P5											X	X	X	X	X	X
P6																
P7																

Table 5.3: Traceability Matrix: Requirements and Problems III.

	R33	R34	R35	R36	R37	R38	R39	R40	R41	R42	R43	R44	R45	R46
P1														
P2														
P3														
P4														
P5	X													
P6		X	X	X	X	X	X	X						
P7									X	X	X	X	X	X

Table 5.4: Traceability Matrix: Features and Requirements

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	F34	F35	F36	F37	F38	F39	F40
R1	X	X																																						
R2			X																																					
R3				X																																				
R4					X																																			
R5					X																																			
R6					X																																			
R7			X			X	X																																	
R8								X	X	X																														
R9											X																													
R10												X	X	X																										
R11			X																																					
R12															X																									
R13																X	X	X																						
R14																			X																					
R15	X																																							
R16															X																									
R17																				X																				
R18																				X																				
R19																				X	X																			
R20																				X		X																		
R21																	X						X	X																
R22					X										X																									
R23					X										X								X																	
R24																					X		X		X	X														
R25					X																		X																	
R26																											X	X												
R27					X																		X																	
R28																													X											
R29			X																		X		X							X										
R30																				X	X										X									
R31																													X											
R32																																X	X	X						
R33			X																																					
R34																																			X					
R35																					X																			
R36																																				X				
R37			X								X									X		X	X	X	X											X	X		X	
R38																																					X			
R39																				X																	X			
R40																													X											
R41			X																																					
R42																				X																				
R43			X																																					
R44																												X										X		
R45																												X												
R46			X													X																							X	X

6 Web app design

This chapter shows the process of the Web app design.

6.1 The Concept

A concept is a Web app for learning software integration processes through teamwork.

Each learner develops one component individually and the integration of those components then takes place in cooperative teams in order to get a completely functional software system.

The idea was to implement an operation of an automated parking garage, where each learner will develop a process-step of the automated parking garage, whereby the steps are implemented as the components of a software system.

After the development of each component, the learners must integrate them to get a fully working software which can handle all the main processes. The integration is accomplished cooperatively in teams of 4-6 learners. The output of a component acts as an input for the next one. It is important to mention that some components may be used several times and that learners work remotely.

During the Web app design in the 5th capital determined requirements and mandatory features were taken account and build in.

6.2 Determination of Target Groups

Based on the researches in the 2nd, 4rd, and 5th capitals, the target groups of the exemplary system concept was determined. The target groups are learners, which have a basic programming knowledge with the goal of learning software integration processes.

6.3 Determination of Goals

The main goals of the Web app are to learn how to integrate autonomously functioning components into a fully functional system, where all those components work together. Another goal is to learn in cooperative teams with consecutive tasks. A further goal is to change the attitude of learners in learning to program, in order to be more motivated during the programming learning, and to make it possible for a high number of learners to be attendant in programming learning, through the Web app.

6.4 Determination of Expected Impacts

This thesis lists the most important impacts of using the Web app.

After using the Web app:

- Learners can learn the software integration processes through teamwork.
- Learners can learn programming in cooperative teams with consecutive tasks.
- Learners can apply their theoretical knowledge about introductory programming in practice and they are able to write programs on their own.
- The attitude of learners in learning how to program will be changed so, that they will be motivated to learn programming.
- A high number of learners can learn programming with the Web app

6.5 Determination of the Programming Task

This section describes the programming task to be addressed within the Web app.

6.5.1 The Main Structure of the Programming Task

The designed task can be well divided into subtasks. The task was designed so that it could be divided into equal parts, as the subtasks need to have a similar difficulty of levels, in order to give every learner a task with equal difficulty. The programming task realizes mathematical related problems, like graphs-theory, i.e. shortest way calculation.

6.5.2 Introduction to the Automated Parking Garage

The programming task to be addressed is an operation of an automated parking garage.

The main processes are the deliverys, pick-ups of cars and exits from the Web app. The learners can choose between those possibilities.

The task of learners is to represent the certain algorithms with pseudo-codes. If there is a task, which needs to be solved with graph theory, learners should consider the cars as the nodes of a graph and the ways between the cars as the edges of a graph.

A certain automated parking garage is a parking garage of a university, that has been just installed. It has 4 floors and it works automatically. In this parking garage only teachers, students and the staff of the university can park. This parking garage works like this: After the car driver has entered successfully, the driver must leave the car and with a help of a conveyor, the car is taken into the elevator automatically, which is in the ground floor of the automated parking garage. The elevator then takes the car in the required floor. In a certain floor there is a right-hand and a left-hand conveyor. After that the car will be taken to one of the conveyors and the conveyor takes the car into a free parking space and secures it there. When the parking time ends and the car leaves the parking garage a conveyor takes the car to the elevator and with the help of the elevator the car is taken into the ground floor, where there is a conveyor, which gives back the car to the car driver. The parking time is checked, the crossing gates go up and the car driver can leave the parking garage.

6.5.3 Data in Connection with the Automated Parking Garage

A database which is updated every day contains the car and the user data.

The user datas are the following: the name of the drivers, the car registration number, the occupation of the driver, the parking time interval, the parking charge paid, and the punishment counter.

The car datas are: the owner of the car, the car registration number, the occupation status, the weight, the type, and the colour.

There are additional variables, that can be found in the description of the components.

6.5.4 Operation of the Automated Parking Garage

First it needs to check, whether the user and car data are correct. Correct data means that the user who wants to use the parking garage appears in the database and the user has her/his own car. Furthermore, the user has less then 2 expired parking times.

The user has the possibility to choose whether he/she wants to deliver or pick up his car or exit from the program. In the following the main processes of the automated parking garage are described:

- Delivery: In the case of a car delivery, the weight of the car is checked first, to decide which floor the car will be parked. In one floor the allowed weight is 100 tons. In one floor there are only 50 parking spaces. So, in a ideal case the cars who come into the parking garage should weight around 2 tons on one floor. If there is a difference, the program should check the actual load of the given floor and decide on which floor the certain car should be placed. (It can happen that the car will be taken to another floor). Then the program must determine the position of the nearest, free parking space on the given floor. At the end of delivery-process of a car, the data of the parking is printed out.
- Pickup: In the case of the pick-up of a car, the car must be searched and it must be determined on which position it parks. It must be also calculated how long the user should wait for his/her car. A car is not allowed to park more than 12 hours in one day. To determine whether the user exceeds this normal parking time, the parking time of the car needs to be calculated and from this, it can be determined whether the users parking time has expired. A user can only have one expired parking, in this case the program gives out only a warning message. But if the user has already exceeded the parking time twice, then the user is not allowed to come in the parking garage again.
- Exit: If the exit was chosen, the program writes out a message, that the usage of the exemplary system finished.

6.5.5 Determination of the Process Steps of the Automated Parking Garage, so the Individual Components of the Software System, and their Connections

Every learner develops a process-step of the automated parking garage, whereby the steps are implemented as the components of a software system. The process-steps or individual components have a uniform form. From here this thesis uses the component denomination.

The components have the following information:

- Purpose: the main goal of the component is described here and the tasks that should be solved by the component.
- Input values: data that the component gets for its processes.
- Output values: data the component gives out after its operation.
- Precondition: preconditions that must be fulfilled that the component is able to start working.
- Post condition: after the component has fulfilled its task, which conditions should be performed.
- Operations: detailed description of the actions, which the component must fulfil.
- Success case: the best outcome of the operation of the component.
- Error message: the possible error messages, in which case the component should give out a error message.
- Remark: any additional information regards to the component.

First the components of the process Delivery are introduced, then the components of the process Pickup, and at the end the process End is described.

Component WEIGHT CAR: Process-step 1 of main process Delivery

- Purpose: the main point here is to find the suitable floor for the car, based on the weight of the car. On each floor the permitted maximal weight is 100 tons and there are 50 parking spaces. For example: If a car comes, which has a weight of 2,5 tons, then this car is needed to be taken, where those kind of cars are parking which generally have less weights. Here the task is to write an algorithm, which searches that floor, where: if we put the next incoming car there, the average weight of the cars will be around 2 tons. If a car arrives, which has a weight of around 2.5 tons and there is a free parking space, then in the future on that floor, those kind of cars should be parked, which do not weigh more than 1.5 tons. And if a car arrives, that has a weight higher than 3.5 ton, then this car is not allowed to park in the parking garage.
- Input: car data, garage plan.
- Output: floor, car data.
- Preconditions: the delivery was chosen in component ONBOARDING USER.
- Post conditions: weight is less than 2.5 tons and the program has found the floor, where the car can be put in.
- Operations: The weight of the car should be checked. It has to be compared to the referred value of weight (2 ton). The weight of the floor should be calculated. The floor that is suitable for the car should be determined and written down.
- Success case: having that floor, that is suitable for the weight of the car.
- Error message: if the weight is more then 3.5 tons.

Component FINDING THE NEAREST FREE PARKINGSPACE: Process-step 2 of main process Delivery

- Purpose: writing an algorithm to find the nearest parking space (on the floor which the component WEIGHT CAR gave back), which is free. Determine the value of the position of that parking space.
- Input: car data, floor, garage plan (garage plan variable has a changing value and it always has to be updated).
- Output: car data, position.
- Preconditions: the car is weighted.
- Post condition: the nearest parking space was found and saved on the given floor.
- Operations: The position of the user (entrance) and the car should be picked up. The distance should be measured between these two positions. Take the smallest one and search that one, which has the free status.
- Success case: having the free and nearest parking space (with the position and the floor).
- Remark: the distance between the two positions should be calculated in a separate component (Component CALCULATE DISTANCE).
- Error message: there is no nearest free parking space.

Component CALCULATE DISTANCE: Process-step 3 of main process Delivery

- Purpose: this component calculates the distance between two given positions in meters.
- Input: floor, position.
- Output: distance.
- Preconditions: having the value of the variable of floor and position of the car.
- Post condition: the distance is calculated.
- Operations: the distance between two positions (even on different floors) should be calculated.
- Success case: the value distance is available (in meters).

Component PRINT PARKING TICKET: Process-steps 4 of main process Delivery

- Purpose: this component prints out a parking ticket on the screen, containing the information on which floor and in which position the car parks.
- Input: user data, car data, floor, garage plan.

- Output: user data, car data, floor, position.
- Preconditions: the parking space is found.
- Post condition: -
- Operations: the floor and the position, where the car parks, should be written down.
- Success case: the floor and the position of a parked car appears on the screen.

Component SEARCH CAR: Process-step 1 of main process Pickup

- Purpose: This component searches the position of a car.
- Input: car data, garage plan.
- Output: position, car data.
- Preconditions: Pick-up was chosen by the user in component ONBOARDING USER.
- Post conditions: car was found.
- Operations: based on the car data and garage plan the car should be searched.
- Success case: the position of the car is available.
- Error message: car was not found.

Component CALCULATE PICKUP TIME: Process-step 2 of main process Pickup

- Purpose: the component calculates the awaiting time, that means how long the user should wait for his/her car in the entrance point.
- Input: floor, position, garage plan.
- Output: pick-up time.
- Preconditions: the car was found.
- Post condition: pick-up time is calculated.
- Operations: the pick-up time should be calculated, based on the distance between the user (entrance point) and the car in meters. The suggestion is that 1 meter way takes 2 seconds. The shortest way should be counted.
- Success case: the pick-up time is written down.
- Remark: this component uses the component CALCULATE DISTANCE, which calculates the distance between the two position, based on the given data (floor and position).

Component MANAGING PARKING TIME: Process-step 3 of main process Pickup

- Purpose: this component calculates, how long the car has been parked in the given parking space. Furthermore, it checks if the car has an expired parking time. If existing, then the punishment counter of the car has increased, if not the program ends.
- Input: user data, car data, position, parking time.
- Output: parking time, car data, expired time.
- Preconditions: car was found.
- Post condition: The parking time is available.

  End1: It is determined that there is no expired parking time.

  End2: An expired parking time exits.

- Operations: The parking time of the car should be calculated, based on the elapsed time the car spent in the parking garage. The expired parking time should be calculated as followed: The parking time of the car should be checked and compared with the referred value, which is 12 hours.
  There are two possibilities:
  - If the parking time is less than 12 hours within a day, than the program should be ended and the car can leave the parking garage without any punishment.
  - If the parking time is more than 12 hours within a day, then the program has to increase the punishment counter of the certain user.
- Success case: Having written down the parking time of the leaving car. Determining and writing down whether the car has an expired parking time or not. The punishment counter is increased if it is necessary.

Process-steps of main process Exit

This process contains only an exit command, which makes it possible for users to leave the exemplary system.

6.5.6 Creating the State Diagram of the Automated Parking Garage

This thesis introduces the state diagram of the automated parking garage.

Two components are communicating with each other through their interfaces. In the state diagram the connection between components can be seen, namely those connections need to be programmed by the learners working in a team, as the one of the main point of this thesis. It is important to mention that there are two components, the CALCULATE PICKUPTIME and CALCULATE DISTANCE that are linked to more components.

The following picture shows a diagram for the programming task of the automated parking garage.

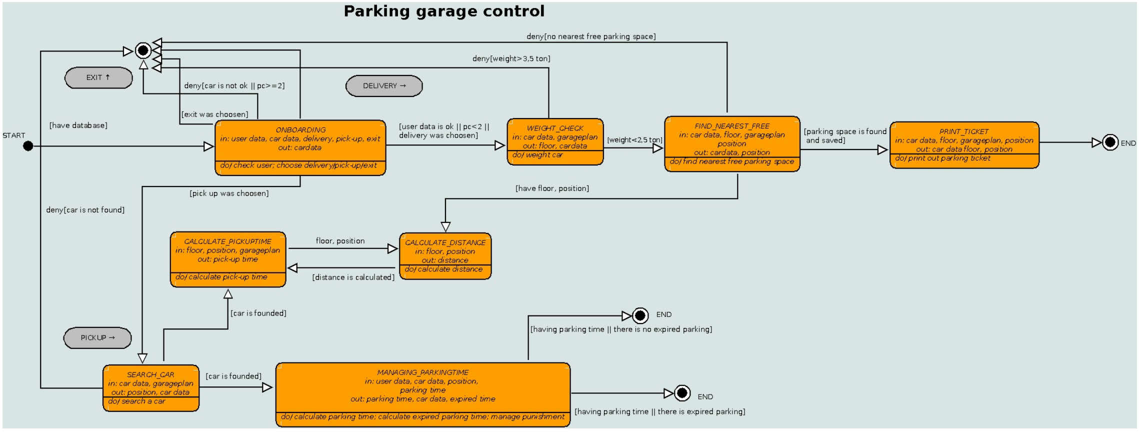

Figure 6.1: Whole Software System of the automated Parking Garage

6.6 Workspace 1: The Individual Workspace

In the individual workspace the components are listed. From those components the learner can choose in order to code them. The following picture illustrates the choosing of Component A.

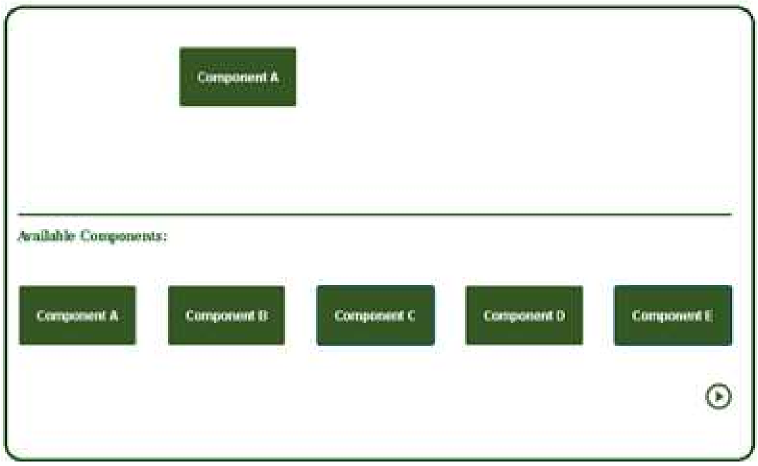

Figure 6.2: Individual Workspace

6.7 Workspace 2: The Cooperative Workspace

In the cooperative workspace two components can be showed, for example Component A and Component B. Here the main task of learners is to link the two components together. The main point of the thesis is how to link components and how to communicate with each other during this process to get a fully working software system. The following picture illustrates it.

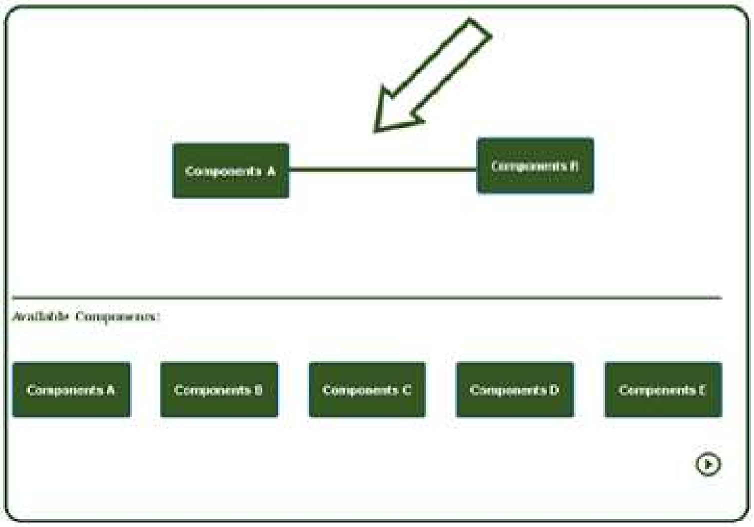

Figure 6.3: Cooperative Workspace

6.8 Workspace 3: Development of Individually Components

Following, this thesis shows a working space, where learners can develop the individual components, with a drag and drop of programming elements. Here learner need to snap together predefined blocks representing variables, instructions and other programming elements to get a functioning component.

It can be seen on figure 6.4 that from the Logic code blocks the if do statement is chosen and the necessary parameter (Pcounter) is given.

Figure 6.4: Drag and Drop of Programming Elements I.

It can be seen on the picture 6.5 that from the Text code blocks the print instruction is chosen and the suitable text is typed in.

Figure 6.5: Drag and Drop of programming Elements II.

6.9 Evaluation of the System Concept of the Web app

The Web app were need to be evaluated. For the evaluation expert interviews were chosen. The following describes those processes.

6.9.1 Creating the Interview Questions

First, the main fields of professional areas were determined. Based on the performed research, on the identified problems, the formulated requirements, and the determined features of the Web app, the following question-areas were determined:

- Personal questions
- Software integration
- Learning to program
- Teamwork
- Software development in a team
- E-learning

The questions are open questions and it was important that the interviewee will be able to answer them with long answers. From these main areas of question, the certain questions of the interview were determined. In the case of the determination of the questions, it was important that the topic of questions should cover all the mandatory features determined.

The area of Personal Questions contains questions related to academical experience and current professional of the interviewees. It was discussed whether the interviewees have experience with online learning or in the field of programming education. This information was important so it was known which professional areas the interviewee represented from, as the within this thesis designed Web app contains different professional areas, like e-learning, programming teaching, software integration and teamwork.

In the case of the determination of the Software Integration questions area, the aim was to get information about the problems occuring during the software integration processes. It was especially important, which problems could occur that are communication and personal related and not primarily technical related. It was also important to know here, which role could fulfil the designed Web app in the field of teaching the software integration processes.

The area of Learning to program a field of questions aims to get to know which level of programming education the Web app can be used for. The other goal here was to get more information from the possible target group of the Web app. Furthermore, in this area questions are inquired to get the information about the effect of the Web app containing visual elements for the learning process of the learners.

The Teamwork fields of the questions area examines, what kinds of teamwork can be realized there and what the gained skills related to teamwork are.

The Software development in a Team questions area analyses whether the exemplary Web app can have later effects for the work of learners after using it. Furthermore, the skills which can be developed by the conceptualized system were also asked here.

E-learning related areas of questions examine the possible extension of the Web app related to online learning. Furthermore, here further proposals for the possible future work of the Web app were made, too.

6.9.2 Search and Selection of Interviewee

Appropriately to the various professional fields which the Web app covers, the interviewees were chosen from different professional areas. It was the aim, that the professional knowledge of the expert should cover all the three knowledge dimensions (technical, process and interpretive knowledge).

The selected interviewees were asked via email to give an interview, in which the title of the thesis and some details about the interview, like goals and the time interval were also imparted. After that the agreement of the time and place of the interviews followed.

6.9.3 Interviewing

Before the real interview a probe interview was made. Here the aim was to check whether the presentation is clear and conveys all necessary information. The other goal was to try out the interview, to check whether the questions were suitable or not. As a result of the probe interview one question had to be corrected and another question had to be divided into two parts. After the correction of them, the real interviews took part. In Appendix A1 and A2 the thesis contains the probe and the final interview questions.

First, a short presentation was given to introduce the concept of the Web app for the interviewees. After the presentation the interviewees were asked, if they have any questions in connection with the presentation. If the answer was yes, the questions were answered.

The interviews were recorded and the interviewees were informed about it. A consent form was filled out and undersigned by the interviewee. After the presentation the recorder was switched on. The duration of the presentation and fulfilling the interview was around 60 minutes.

Before the interview, a short introduction was given. The interviewees were informed, that the interview can be held in English or in German.

All the documents of the interview were provided in English and in German, too. It was imparted that the questions are classified into categories and to one category one or two questions belongs to.

To summarize the following documents that were used during the expert interview can be found in the Appendix: an expert interviews questionnaires, a consent form, a drawing of the system concept of the Web app, and the presentations slides, namely in Appendix A1, A2, A3, A4 and A5.

6.9.4 Transcribe the Interviews

The interviews were transcribed. With the help of the recorder and the voice content of each interview was typed into a textual document. Both the probe interview and the real interview were written down.

6.9.5 Creating Report

From the transcribed text a summary was written down and this summary was ordered according to the question categories.

From the summaries the communalities, similarities, divergences, conflicts and the individual thoughts were drawn and noted down, and are listed in chapter seven.

7 Results

This capital discusses the result of this master thesis. The result of this master thesis will be a proposal for further development of the designed Web app. The Web app was named AcadIC. This proposal consists of three parts:

- The demonstration of the objectives of this thesis and the features of the AcadIC
- The report from the expert interviews: the summary and the presenting communalities, similarities, divergences and individual thoughts of the expert interviews
- The list of future work features
- An idea for teaching the software integration processes, which came up during writing the thesis

7.1 A demonstration of the objectives of this thesis and the features of the Web app

In the following a table will be introduced, containing the relationship between the objectives of this thesis and the determined mandatory features of AcadIC.

Objectives	**Features of AcadIC**
Teaching learners the software integration processes through teamwork, from the beginning of programming learning	F3, F11, F20, F22, F23, F24, F25, F36, F37, F39
Learning in cooperative teams with consecutive tasks	F9, F17, F19, F35
Supporting learners in their learning process of programming, which is a main subject of computer science study	F5, F11, F15, F20, F21, F22, F23, F24, F25, F29, F33, F36, F37, F39
Helping learners applying their theoretical knowledge in programming and helping them to write programs on their own	F5, F15, F19, F20, F21, F22, F25, F29, F33, F36, F37, F39
Increasing the motivation of learners, thus decrease the drop-out rate of the introductory programming classes	F11
Enabling the attendance of a high number of learners in a programming class and making personal presence unnecessary	F15, F20, F29
Conveying the importance of teamwork in software engineering	F3, F11, F20

Table 7.1: Objectives and the Features of the Web app

7.2 Report of Expert Interviews

This section consists of two parts. One gives a summary of the expert interviews and the other one of the analyzation of the expert interviews.

7.2.1 Summary of the Expert Interviews

The summary of the expert interviews is classified regarding the questions and their categories.

Personal Question: **Educational background and current profession and practical knowledge of interviewee in teaching programming and e-learning**

Interviewee 1 has been studying in the field of informatics and he has already worked as a tutor in software engineering related studies at TU Vienna, so he has some experience in teaching. He was using the Moodle-based TUWell online platform during his teaching activity.

Interviewee 2 works as a software engineer at a software engineering company and currently works on his Ph.D. He works as an adviser, supervisor in bachelor, and master thesis in the field of software engineering at TU Vienna, too. He has used the Moodle-based TUWell online platform during his teaching activity.

Interviewee 3 works as a software engineer and he is a lecturer in the field of software engineering in bachelor and in master, too. He is involved in teaching the following subjects: Software Engineering and Project Management, and Advanced Software Engineering. He has used the Moodle-based TUWell online platform during his teaching activity, among others using for uploading course materials.

Interviewee 4 has a master's degree in communication media and educational science. She has been working as a media and educational scientist in research and development projects. She is a lecturer at a university, too. Her research area is innovative pedagogical approaches in open and responsive learning environments.

Software Integration I.: **Non-technical problems arise during the integration of software components**

Interviewee 1 says, interfaces used by linking together components differ from the interfaces, which stay in the requirements documentation. The suitable communication between the components would be useful for finding problems between the interfaces, too. He sees the problem in the requirements and the communication between the software development teams. Regarding his experience, there is a problem when a team implements a component together with regards to the requirements and this team does not communicate with the team, who will use those components. He thinks AcadIC could be used well in software testing.

Interviewee 2 thinks the main problem is the lack of communication between the team members during the implementation of software components. He feels, that during developing software systems, the main problem is to define the interfaces, which link the components together. When developer teams work remotely, far from each other, in this case, the communication can take a lot of time, with writing emails to the developers, etc. It means a problem, too, that the right person cannot be connected directly, only through another person, and in the end, it causes a problem in the communication, and he or she can not get the answer. It should be specified, how the components should be integrated. He has summarized the answer to this question so: the main problem is in the communication and the documentation.

Regarding Interviewee 3 his experience, in students project, when different features of software systems need to be developed, in the end, because of time constraints, they are not able to emerge those features and they are not able to get a working element, because they do not have much time for it within the lecture. He means it is because students start working on those projects works only later in the semester and they are not able to finish those tasks in time. He feels when implementing components in a team, and team members are far from each other, they only work remotely. The task division between the team members is the main problem. It is necessary to talk through who is responsible for which task. He thinks, there are also problems in the documentation and the support.

Interviewee 4 thinks working together with technical, social and didactic people is often problematic. It does not work, that technical people develop something and just pass it to the users for using it. It is better if the users are involved from the beginning of the software development process. In the case of a product, that needs to be programmed, first, the user must use this product and the developer/didactic people have to watch how they use it, and through this observation, they will know what the most important tasks are, that they have to fulfill. After that, a list has to be made. Thus they will know which functions are really necessary and what these functions mean. In the best case some empirical experts are involved, who know how to work together with the user and someone technical, who knows how to implement those kinds of systems. If this collaboration does not work between the target groups, when designing a system, then normally the resulted/implementable system will also not work at the end. All those target groups must be present so that the technical people will know which tasks they have to do. All three target groups have their vision and their visions are different from each other's visions.

Software Integration II.: **The effect of the described concept for teaching the integration process to students**

Interviewee 1 thinks that the system concept would be beneficial to teach the software integration processes. It is a good idea and it can be a benefit for the teamwork, to work in a team with those kind of people, who come from different domains. Those people have different perspectives. AcadIC could give students the ability to outline various areas, through the communication of interfaces.

As Interviewee 2 sees it, the AcadIC is valuable to teach in the integration processes. During integrating components in a team, team members must talk about incompatibilities. The most important thing is that team members should communicate with each other and find a solution for the given integration problem together. In practice, it is an advantage that developers talk with each other right from the beginning of the component developing process. But in the case of AcadIC, students develop components individually and then they form a group and work together to link components. It is a good lesson, to realize that before developing the components, the students should talk to each other and not just when they link the components.

Interviewee 3 thinks that it would be better to not apply AcadIC in a software engineering course, but in the quality insurance lecture, because of the interfaces and contract between the components. It is a benefit to show the students how to develop a large scale of flow, which is built up from small components. It would be very interesting to show, how the parts fit together because the main problem is usually defining the interfaces and contracts between the components. He suggested two solutions. One is to show students how to implement components, focus on the components and focus on, how to test them. The other solution is when the communication is not defined at all and students must define the contract between the components. When making the contracts between two components, students must define a common interface, where the output of a component is the input of the other one.

Interviewee 4 does not answer this question (because of time pushing).

Learning to program I.: **The effect of the presented system concept for learning programming in relation to varying knowledge levels**

Interviewee 1 means, that in a student team it is a common problem, that students have different knowledges, different experiences in programming or different motivations to learn, and it can cause problems during teamwork. But maybe with AcadIC, through the communication of integration of individual components, weaker students can get the possibility to talk and improve their knowledge and their motivation, too. As a result, they could play a better or more productive role in the team.

Interviewee 2 thinks AcadIC would be suitable for beginners and for advanced because of its visual programming nature. But those visual editors are suitable for beginner programmers. But students, who have already had some experience in programming do not need visual programming elements to learn programming. But for those advanced, that kind of visual programming can be fun, that they could use as a game. For integrating components AcadIC is suitable because the interaction is a good experience for students.

Interviewee 3 feels this GUI based programming is suitable for beginner programmers (who only have some theoretical knowledge). AcadIC is not about getting to know the way of thinking of programming and how to treat problems during coding. He means it is just a UI editor for the action code. As an example, he mentioned Apple, which has been developing programming guides for children. Those guides begin with simple programming tasks and then it is getting more complex and so, they can teach the basic behavior of programming. For example, a task can be there in the case of AcadIC: organize a game, where students must implement a robot, which moves from one place to another and the first step is to tell him to go three times forth and one to the left and so on. Then it gets more and more complex, using also loops, etc. In that case, students learn programming paradigms and they do not write codes first. Snapping together blocks of elements is a good idea to teach programming. He thinks it would be nice to do it at different levels. The first level would be: students should design Lego and they need to write it on their own. Then the contract between the components needs to be determined. AcadIC could be used for varying steps the in the software engineering process. He thinks, that AcadIC is too complicated to learn it. AcadIC would work by adult people, but they would need to write the source code on their own. In the case of AcadIC students, they do not need the documentation in their mind, so it is a little bit easier to learn programming. But he is not sure if AcadIC would work in the case of real beginners, who have no previous knowledge in programming. He means at least AcadIC simplifies the language, which is available. In his opinion, AcadIC would work better with 20 code blocks, but it would not work with 100 code blocks.

Interviewee 4 did not answer this question (because of time pushing).

***Learning to program II.:* The effect of the approach with prefabricated code blocks and the drag and drop technique to the learning progress**

Interviewee 1 means, this approach would help students to learn programming. It depends on the knowledge level of students in teams, but for students, who do not understand the programming yet, AcadIC would be more helpful. Sequence diagrams would be more suitable to use in AcadIC.

Interviewee 2 has no additional information on this question. He feels that he has already answered this question.

Interviewee 3 thinks the effect depends on the target audience. For children, it would work fine. For adults, it would be too easy. AcadIC would work in the first couple of hours during a programming lesson because in a programming course, students need to learn a lot of things and AcadIC would be only useful in the beginning of programming learning.

Interviewee 4 did not answer this question (because of time pushing).

***Teamwork I.:* The effect of the system concept on the cooperation and team skills of students**

Interviewee 1 thinks he has already answered this question.

Interviewee 2 feels, the most important thing is that students have to communicate with each other and make the components work. Students must discuss and negotiate about how to write the source code, to fit on both sides of the components. The communication during this work would depend on the personality of team members because based on the communication skills and personalities of team members, there are different kinds of programmers. This process would be individual. He does not believe, that a standard trend would exist, that could be used in every group. It would be good if the communication would be individual for every team. He thinks all depends on, how each group member reacts to the other team member. At the end of this experiment (when AcadIC will try it out), there will be a lot of individual results.

To Interviewee 3's mind, AcadIC does not affect the cooperation and team skills of students at all. He would rather make a design meeting inside a team and discuss the integration. He thinks it is not good that everybody just individually implements a component and in the end, the whole will work. Between this process, a lot of communication has to be performed and the contract between components defined. Either defining contracts would be done by AcadIC and the integration is not needed to do it at all or the above-mentioned design meetings must be done within team members. In his opinion, there is no need for an application. He thinks, talking to other people can not be replaced with a system.

Interviewee 4 thinks, that it is hard to judge the effect without any observation, where she could see how students work together, and what would work and what would not work. Into AcadIC a chat tool, an information tool can be implemented, where students can leave data or a forum can be implemented, where students can store information. In her opinion, it would be useful to implement that kind of tool in AcadIC, where it can be seen, at what time which person does something on the integration job. From this communication, a list should be made and then an empirical study should be done to see which communication tools are used the best. A comparison should be made between similar applications and AcadIC to see what already works and how, and what does not work. It should be useful to see, how the communication process in an already existing automated parking garage control system works. Information can be derived from it, which communication functions work really and which would be useful for the system concept. A good approach would be, when a research would be done in connection with the parking garage control system, to see how people can learn with this. A better way would be, to make an empirical study, to give AcadIC to students and to let them test it, and students could be observed, to figure out how the communication between them works. After that empirical data could be won from it and then a convincing statement could be done.

***Teamwork II.:* Elements of the showcased system concept, that have the greatest significance in relation to teamwork**

Interviewee 1 thinks to work in a team, talking about each component, finding bugs and other problems, and working together to improve the definitions or source code are all useful activities. It is better to work in teams from the beginning and then working alone. He has referred to Redmine, an open-source tool, which is used for defining i.e. the concepts, where every component has its own user interface and people can comment or link components together. He feels, it is a good idea that students in a group come together and talk about the whole system or the components. It does not matter if students communicate remotely or hands-on mode. He mentioned plug-ins, too. In software projects the communication should be available and private. The components diagram (showed in the presentation) is very useful in the case of teamwork.

Interviewee 2 thinks that the greatest significance is the part, where students must talk to each other. Maybe timing would be useful to determine, how long each phase should be last. Students must be allowed to work on the second component, only if the first (or actual) component is ready and it works correctly. So, the component must be tested and must it be ensured that components meet the requirements. This would be one possibility. He means the other possibility would be to let students work out and finish components afterward (so do not make a ready component), but in this case, other approaches are necessary instead of it. If students continue to work on the previous component (which does not work correctly), it perhaps would be a programming lesson for the students. With this experiment, lots of information transfers from one student to another students could be performed. In this case, the teamwork depends on the single person and to which person this person is matched to.

According to the Interviewee 3' way of thinking, the contract between components has the greatest significance. Because there it is necessary to communicate with each other, exchange ideas and make an input-output exchange. It has to be defined: the components, the contract between the components, and to how to communicate.

Interviewee 4 did not answer this question (because of time pushing).

Software development in a team: **The effect of the system concept to the ability of a student to work in a team in an IT profession, and the particular skills, which the concept can facilitate**

Interviewee 1 means, that when people work alone on a project, they would always perform better, then doing the same work in a team, because the communication faults will not appear. Maybe another person has a different level of education and one of the team members may work quicker than the other. If people have to split the work into a team, then they must learn how to communicate and learn how to talk to each other. If some developers work in a team and they want to model something together, they can have different opinions and it can cause problems. People must learn, how to be objective about some requirements. Their egos must be put away. If students learn how to communicate and they implement models together, they should practice it and they can practice, how to talk to each other.

Interviewee 2 thinks, in practice, developers need to talk to each other during developing the components and not just when they link the components. It would be a good lesson for students to learn how to talk constantly to each other from the beginning, to make the system, where components are well connected. About the gained skills, he has mentioned the importance of good communication skills in a team and about the right level of detail from the technical side.

Interviewee 3 would apply this system concept in the quality engineering field or in the architecture design perspective, where components have to be defined. If that kind of example is given to students, they come to the task submission with one big program, which does everything and they do not have this components-divisions or architectures. It is a good thing to teach or force students to think about components - how to implement them, how is the contract between them and how they can test them. It would be nice to do test developments, where students can learn and practice, more in the architecture design direction. For beginners maybe AcadIC would be too overvalued. He could imagine that the kind of system with some automated tests in the background, which check the implementations and when everything is implemented, the program can be run. An example: in the case of a robot, the students first learn how to drive and then the robot must turn left and right. He feels, it is an interesting approach to have a big program, which is built up from very small tasks, which students can implement in different ways, on their own, and then they have the big picture in the end. This is one of the key aspects, that students must learn in software engineering. There is a bigger problem and it has to be divided into smaller parts. In his opinion, the whole process is probably too complicated for one person, especially at the beginner level. One of the benefits of it is to learn by doing it and to see how to design those contracts, interfaces and those communications.

Interviewee 4 did not answer this question (because of time pushing).

***E-learning I.:* Further developing possibilities of the presented concept in relation to E-Learning**

Interviewee 1 means he does not know the key benefits of e-learning systems in general, so he does not answer this question. But in his opinion when people who remotely work together on a system, their most important goal is communication.

Interviewee 2 means regarding his experience, in the case of developing software in distributed teams, if developers do not sit beside each other, it is hard to communicate with each other, i.e. hard to talk to the microphone. It would be a very interesting study, to observe two kinds of group, where one group sits next to each other and tries to integrate the system and the components, and there would be a remote group, where the team members sit in front of the computer and communicate via Skye or Web chat. He thinks, in practice, there are many more components, more complicated ones, so it will not be so easy to integrate them. In the case of AcadIC, there are much fewer and not so complicated components. In the case of more complicated components making a well-made documentation is very important. It would be a good lesson for students to learn writing good documentation, but in the case of AcadIC the components are not so difficult, maybe it is not suitable for it.

Interviewee 3 feels it is a nice example to learn the basics of programming, where students can learn to implement small parts. In his opinion, it would be a very good idea, to start with a small program, extend it with features and then build really big applications. Here people could learn, how to think in a big picture, how to split tasks into small parts. For beginners, it is nice to split the task into small parts and learn programming so. This would be the first step. The next step would be to let them design that communication and implement that. It would be a nice idea, first to teach to implement small tasks, then to teach them how to divide small tasks and after that teach them how to connect them. This process can be made good with an e-learning platform. Because with that students can get instant feedback on what they do, because all those things are automated testable. Automated tests can be run and instant feedback can be given to the students. For students, information can be brought along, for example, if a component does not work correctly. He feels, AcadIC can give ideas to on the user how to test the components. He likes AcadIC more from the QE perspective.

Interviewee 4 believes that AcadIC should be compared with some reference projects like for example Scratch. It needs to be compared as followed: to watch what is the best case, what is there already in AcadIC, what are the differences in AcadIC from a reference project, and then to make a judgment: it is better, or worse, or what is an added value. The most important question is, what the added value of AcadIC is, compared with already existing similar systems. She feels communication depends on the behavior of people and if they like to work together or not. Staying flexible is always good in the case of communication. An e-learning process is normally based on a concept. This concept should have a learning goal and in the end, there are results. The results should be tested and it can be seen, which knowledge can be achieved. She has suggested a thesis as a reference, written by Michael Derntl, the title is Patterns for Person-Centered e-Learning, which contains UML Diagrams and methodologies for e-learning. In his thesis, the author has created UML Diagram methodologies, for making it visible, how e-learning can work or not work. In connection with AcadIC, first, it must be decided, what the main added value of it is. If the goal is to make a methodology, to make e-learning visible, to make an e-learning concept visible, then e-learning can work based on such abstract work like this, like in the above-mentioned UML bases dissertation. Or, the other possibility is to find out, which kind of communication is needed to integrate those components. If the main question is, what kind of communication it needed, then it is important to see already existing systems. She has suggested a process: first, collecting empirical data, so give students such a system to test it and do not give them a theoretical concept, because it is hard for them to make it visible. So, give this system to students, observe them, and derive empirical data from it. After that, the decision can be made, which kind of communication tool it needs and which tool would not work at all. It needs to be analyzed, too, how to derive empirical data, which empirical methodology has to be used, and it needs to see how a parking control system works. She feels, as a first step, the system must be described, what students do, how they interact with each other, and how it works. After that, based on this description, an analysis can be made, what did work, what did not work, and what could be optimized. As a first step, it is always better to get to know an existing system/concept, which works in a similarly, and some related work. In order to have ideas, of which systems already exist, how they work, and what works there and what does not.

***E-learning II.:* Incompletion, and directions, where the platform should be developed further**

Interviewee 1 thinks a discussion board would be good to implement into AcadIC. It would be nice to track information: who modeled and what in the diagram/system. It would be nice if students could click on certain relations, or the components and write their comments, or requests for comments. And other students can also comment and answer those. GoogleDocs has similar functionality. Comments and that information would be useful because after some time, students start to work or call the components again and they will be able to know (from the comment section) why it was modeled like this.

As Interviewee 2 sees it, AcadIC should be developed further so: AcadIC should be suitable to learn together as many students, where many of the components need to be implemented, so making crowded learning with it. The components in AcadIC have to be increased to around 50 components. Maybe experienced programmers could use a text editor and not graphical editor to establish program codes. He feels it would be a good lesson, to compare when student codes using graphics editor and using text editor, too.

Interviewee 3 thinks one aspect would be only to define those components, the second aspect is to define the contracts between components, and the third aspect is to define the whole work-flow. Then a test should be written for those components, maybe this test could be written before writing the component itself. Thus, a testing development could be received, and with the 3 different kinds of aspects, the teaching can be performed in AcadIC. There is potential in these different aspects. In his opinion, it would be nice to make a step by step guide, to see how to develop a software. First, a little program needs to be written by the students, then two processes have to be defined, which are communicating with each other. Then, as a next step, a bigger component has to be defined and then design architectures. In the end, students can learn how to test components and not the whole work-flow. Because this whole work-flow is developed on its own, students could only test the whole work-flow and not a simple component. And if a component breaks, students know, where it breaks, so they will know where exactly the failure in the system is.

Interviewee 4 did not answer this question (because of time pushing).

7.2.2 Analysation of the Expert Interviews

To analyze the expert interviews, communalities, divergences, and conflicting points of view were looking for. The author of this thesis has extended the analyzation with two further categories. This thesis describes these categories:

- Communalities: Here this thesis lists the answers of the experts, where experts agreed, that they have the same opinion and same thoughts on a given topic.
- Similarities: Sometimes those experts, who come from different domains, have different professional views and thus they have similar opinions on a topic.
- Divergences: This thesis classified those answers, that had some disagreements between the experts.
- Individual thoughts: During the expert interviews valuable thoughts and ideas of the experts arise, which are listed here.

1. Communalities
    - All experts have the same opinion, the most common problem arising during the integration of software components is the method and the amount of communication between software developers.
    - From the 4 experts, 3 experts have agreed that AcadIC would be a benefit to teaching the software integration processes. Thus, showing the learners how to develop a large scale of flow in a team, when a software system is built up from small software components. Those interactions are good experiences for learners. Learners can get to know the software components-division or system architecture, too.
    - From the 4 experts, 3 experts have agreed that, before the developing of the software components, learners should talk to each other and not just when linking together software components.
2. Similarities
    - As a didactic expert's point of view, it is better, if users are involved from the beginning in the software development process. As a developer expert's point of view, it is an advantage if developers are talking with each other right from the beginning of the software component developing process and not just when they link together the software components.
    - As a didactic expert's point of view in the case of designing that kind of Web app the best case is that people who come from different domains (like the user of the Web app, technical people, and didactic people) work together, but it is often problematic. As a developer expert's point of view in the case of developing those kinds of software integration processes, the best case is to work in a team with those kinds of people, who come from different domains

of software engineering, because those people have different perspectives. As another developer's expert point of view, this can be problematic because a different person has a different level of education and it can happen, that one of the team members is working quicker than the other one. If some developers work in a team and they want to model something together, they can have a different opinion and it can cause problems. But if people must split the work into a team, then they must learn how to communicate and learn how to talk to each other.

- As the didactic expert's point of view, it would be useful to implement into AcadIC that kind of tool, where it can be seen, at what time which person does what on the integration of software components. A developer expert's point of view timing would be useful to implement in AcadIC in order to determine, how long each software component development phase should be taken. As another developer expert's point of view, from an education domain, in the case of a software engineering course, when learners implement a software, because of time constraints, learners are not able to produce a working software at the end of the course.
- As one of the developer expert's points of view, the communication in software projects should be private. As another developer experts point of view teamwork depends on the person and on which person this person is matched to, and the communication would depend on the personality of the team member because the communication skills and personalities of programmers can be different. As the didactic expert's point of view, the communication depends on the behavior of people and if they like to work together or not.
- As one of the developer expert's point of view, it would be better to not apply AcadIC in a software engineering course, but in the quality insurance engineering lecture/field, or in the architecture design perspective, and to do test developments, where learners can practice the architecture design. And as another developer expert's point of view, AcadIC could be used well in software testing.
- As one of a developer expert's point of view, talking - in the case of software developing in team - other people cannot be replaced with a system. As another developer experts point of view, it does not matter if learners communicate remotely or hands-on mode. Another developer expert's point of view for distributed teams, if software developers are not sitting beside each other, it is hard to communicate with each other, for example, it is hard to talk with microphone.
- As the didactic expert's point of view, it is hard to judge the effect of the AcadIC for the learning process without any observation, where it could be seen how learners work together. As developer expert's point of view, it would be very interesting to observe two kinds of teams, wherein one team, the team members sit next to each other and try to integrate the system from software components together, and in the other case a remote team would be observed, where the team members sit in front of the computer and communicating via Skype or Web chat with each other.

3. Divergences

- Regarding the target group of AcadIC there are some divergences between the answers of two experts. As one of a developer expert's point of view programming editor of AcadIC, with the prefabricated code blocks and drag and drop technique, is rather suitable for beginner programmers, who have no previous knowledge in programming and GUI editor would be fun for advanced learners, as a game. The other developer expert thinks, beginners only need some theoretical knowledge to use those kinds of graphical programming editors. This expert thinks that for beginners – who have no previous knowledge in programming - the presented AcadIC would be too overvalued and its whole process with software component integration is probably too complicated and the presented AcadIC would work by adult people, but they would need to write the source code on their own.

- Regarding the elements of AcadIC, having the greatest significance concerning teamwork, two developer experts think that the communication between the learners and the other expert thinks the contract between software components has the greatest significance in connection with teamwork.

4. Individual Thoughts, Ideas of the Interviewees

- As one of the developer expert's point of view, during the software development, software developers have to learn, how to be objective about some requirements, otherwise it can cause problems during the software development process.
- As a developer expert's point of view during developing software components and their integration, the not suitable documentation and the support of the software developers' work are crucial and can be problematic.
- As an experts point of view, as a didactic expert, generally when a product has to be programmed, first users have to use this product and the developer and didactic people have to watch them use this product and through this observation, they will get to know what the most important features are, that needed to be implemented into the system. So AcadIC should be given to learners to test it (but do not give them a theoretical concept, because it is hard to make it visible). Then, a list has to be made about this observation, to see which communication tools are used the best: so it is necessary to describe what learners do, how they interact with each other and how it works, what could be optimized, and thus the features, that are really necessary could be found out. It also has to be analyzed, in which empirical methodology has to be used. When designing those systems, if those collaborations do not work between the involved empirical experts, the user and the developer, then normally the implementable system will also not work in the end.
- One of the developer experts has suggested two solutions regarding to make the communication between the learners during the software component integration process easier. One is when showing learners how to implement software components, focusing on the software components and focusing on how to test them and here the communication is defined. Another solution could be when the communication is not defined at all and the learners must define the contract between the software components. When making the contract between two software components, learners must define a common interface and during this process, they need to communicate with each other somehow. For the communication about the integration of software components a design meeting has to be done between the team members, but in this case, there is no need for a Web app.

- In software engineering studies learners have different knowledges, different experiences in programming or a different motivation to learn and it can cause problems during teamwork. But with AcadIC weaker learners can probably get the possibility to talk and maybe improve their knowledge and their motivation. As a result, they could play a better or more productive role in the team.
- One of the main problems in software development in a team, is the task division between the team members, namely, to talk it through, who is responsible for which tasks.
- A software component diagram is very useful, to see and to know everything during the teamwork in software development.
- Using sequence diagrams in the case of blocks of programming elements drag and drop would be useful.
- Into AcadIC a chat tool, information tool, or forum can be implemented.
- It has a lot of potential to make a real step by step guide, how to develop software in AcadIC, where those steps are getting more and more difficult. Starting by writing a little program, where there are two components, which are communicating. Here learners design and implement interfaces between the components. And then bigger components could be defined, extending with features and designing architecture. Here learners could learn, how to split tasks into smaller parts, learn how to build big applications, how to think in a big picture, and how to get a whole work-flow at the end. In the end, learners will learn how to test components, but not the whole work-flow. If a component breaks, learners will know, which component breaks in a work-flow, and where exactly to find the failures in the system. This process can be made easier with an e-learning platform. Automated tests can be run and instant feedback can be given to the learners and they can get information whether the software component works correctly or not.
- AcadIC would work well with fewer software components, i.e. with 20 code blocks, but with 100 code blocks, AcadIC would not work. In the case of more complicated software components, making well-made documentation is very important and it would be a good lesson for learners to learn writing good documentation. AcadIC should be developed further to learn together, where many of the software components need to be implemented, to make crowded learning with it.
- The learning process that AcadIC offers would be individual. A standard trend, that could be used for every team, would not exist. It would be good if the communication was individual for every team. Everything depends on, how each team members react to the other team member. At the end of those experiments, there would be a lot of individual results.
- As the didactic expert's point of view, a comparison should be made between similar applications, i.e. Scratch learning environment, and AcadIC, to see what works already and how what does not work, what the differences from a reference project are, and then to make a judgment: it is better, worse, or what is an added value. It should be useful to see, how the communication process in already existing automated parking garage control systems work and information can be derived from it, which communication functions work really and to see how people can learn with it.

- During developing software components, learners only have to be allowed to work on the second software components, if the first (or actual) software component is ready and it works correctly. So, the software components must be tested and it must be ensured that the software components meet the requirements. The other possibility would be to let learners work out and finish software components afterward (so do not make a software component ready), but in this case, more other approaches are needed for it. If learners continue to work on the previous software component (which does not work correctly yet), it would perhaps be a programming lesson for the learners. With this experiment, lots of information transfer from one learner to other learners could be performed.
- Every software component should have an own user interface and learners could comment or link software components together, like in the case Redmine, the open-source tool.
- The following have to be defined during the software integration processes: the software components, the contract between the software components, and how to communicate.
- A thesis was offered as a reference by an expert from a didactical domain, written by Michael Derntl and the title is Patterns for Person-Centered e-Learning. The author of this thesis has created UML Diagram methodologies, for making it visible, how e-learning can work or not work. In connection with AcadIC, first, it must be decided, what the main added value of it is. If the goal is to make a methodology here, how to make e-learning visible, an e-learning concept visible, how e-learning can work based on the AcadIC, like in the above-mentioned UML bases dissertation, or to find out, what kind of communication is necessary to integrate those software components.
- A discussion board would be good to implement into the AcadIC.
- It would be useful to track information in AcadIC: who modeled and what.
- It would be useful if learners could click on certain relations, or the software components and write their comments, or requests for comments there. Other learners can also comment and answer those (GoogleDocs has similar functionality). Comments and that information would be useful because after some time learners start to work again or call the software components and they will be able to know (from the comment section) why it was modeled like this or that.
- In the case of AcadIC, it would be a good lesson, to compare two kinds of processes - one is where the coding happens with a graphical editor and the other is when the coding happens with text editor.

In the capital introduction described objectives have almost fulfilled, because the above-listed answers show, that teamwork in the field of software engineering is problematic. This field has to be researched and the AcadIC would be suitable for teaching learners the software integration processes. Using graphical programming editor in the case of AcadIC, it should be thought all over again, because of the divergences of the answers. Observations on teamwork in AcadIC would be recommended. The ideas and thoughts that come up within the expert interviews could be converted into further features of AcadIC in the future.

7.3 Future work Features

Future work features are derived from the identified problems, too, presented in chapter three. These future work functions did not appeare in the Web app design and they were not evaluated with the help of the expert interviews, because those features came up during the research work, after AcadIC was designed. This thesis consists of those future features and their descriptions in section 5.3. Here in this section this thesis only lists them:

- F2: Self-assessment Option
- F4: Competition between Teams (with Badges, and Scores, Leader-Boards for rewarding)
- F6: Creating/managing Teams
- F7: Coordinate Group Work
- F8: Version Control Option
- F10: Modularized Contents, using Headings, logically structured
- F12: Files upload and share
- F13: Modern Web-based Communication, like Chat Rooms, Forum, Blog, Mailbox, Message Board, Site News
- F14: Review and retrieve Conversations
- F16: Moderation of Teamwork
- F18: Calendar/Group Calendar
- F26: Final Solution of Programming Task appears in a separate window
- F27: Web Recorder
- F28: Ability to separate Errors (i.e. in a separate Window)
- F30: Idea generation Organization
- F31: Upload/share/search distributed Learning Materials
- F32: Customizable Workspace Area (for Individual and for Teamwork)
- F34: „About me“ Section
- F38: Error Detection Option
- F40: Source-Observer

7.4 An Idea for Teaching the Software Integration Processes

This thesis summarizes the idea of the author, that came up during this research work and when fulfilling the expert interviews.

The idea is to teach the software integration processes for learners based on teaching children how to build a puzzle from its pieces. This idea could be realized for example with the help of a Web app.

First, this thesis lists some similarities – that were discovered – between a completed picture built from puzzles and the software integration processes.

A completed picture built from puzzles	**Software integration process**
The product here is: a completed picture built from puzzles.	The product here is: a completely functional software system.
A completed picture is made up separate pieces of puzzles.	A software system is made up from individual working software components.
A piece of puzzles has an edge.	A software component has an interface/ input and output.
Goal of puzzles: to build a completed picture.	Goal of components: to build a completely functional software system from them.
Target group of learning how to build a completed picture from puzzles: children who are beginners in building a puzzle.	Target group of learning the software integration processes: learners, who are beginners in learning software integration processes.

Table 7.2: Puzzles and the Software Integration Processes

Hypothesis

If there are similarities between a completed picture build from puzzles and the software integration processes, it can be, that their teaching is also similar.

This thesis shows a derivation of a possible way to teach learners the software integration processes from a way of teaching children building a completed picture from puzzles.

The listed ideas could be future features of AcadIC.

Following, this thesis introduces the method of teaching children how to build a completed picture from puzzles and it is taken from a single method from the website www.ot-mom-learning-activities.com. The next tables show those methods.

Teaching children how to build a completed picture from puzzles	**Teaching learners how to build a completely functional software system from components**
Method 1:	**Idea 1:**
1. Learning for children: what means to put two things together with one piece of puzzle	1. **Show and illustrate learners what to integrate one component into a software system exactly means**
2. Learning with 2-pieces puzzles how to build a completed picture:	**2. Learning with 2 pieces of components how to build a completely functional software system:**
Step 1: Demonstrating for children how to build a completed picture from puzzles.	Step 1: Showing learners a fully working software system, build up from 2 components.
Step 2: Let the children look at those completed picture.	Step 2: Let the learners analyze those software systems.
Step 3: Then comes an individual task for children: to let them to complete 2 pieces of puzzles.	Step 3: Then let the students complete a 2 pieces software system on their own.
3. Learning with 4 pieces of puzzles how to build a completed picture:	**3. Learning with a software system, which consists of 4 components:**
Step 1: Building a complete picture from 4 pieces of puzzle in front of the children and after that show them the completed picture.	Step 1: Building a software system, which consists of 4 components in front of the learners and after that show them the fully working software system.
Step 2: From the completed picture one piece of the puzzle must be taken away and next, children need to fit that piece of puzzle back to the completed picture. This step has to be repeated with a different piece of the puzzle.	Step 2: From the fully working software system one component must be removed and learners need to integrate back this component again in order to have a fully working software system again.
Step 3: Step 2 needs to be repeated by taking away two pieces (always different ones) of puzzles.	Step 3: Step 2 needs to be repeated by removing two (always different ones) components each time.
Step 4: The completed picture needs to be taken into pieces, but the pieces have to be kept rightly orientated, then the children need to build the completed picture again.	Step 4: The interfaces have to be erased, but the interfaces of source codes of the components still have to be visible/available and then learners need to integrate those 4 components again, with writing the source codes for the interfaces, in order to get a fully working software system.
Step 5: Then the completed picture needs to be taken into pieces again, but the pieces have to be mixed up and then the children have to build the completed picture again from those mixed pieces of the puzzles.	Step 5: Then the interfaces of components have to be erased again, but the interfaces source codes of the components are not visible/ available, then learners need to integrate those 4 components again, writing the interfaces source codes, in order to get a fully working software system.

Table 7.3: Method 1

Method 2: Outside edge first	Idea 2:
Step 1: The pieces of puzzles have to be turned over and the completed picture, that has to be built up, has to be studied by the children.	Step 1: The components of an existent software system needs to be shown for the learners, by showing all the components one by one with their interfaces/input/output and let learners analyze/read it.
Step2: The pieces of puzzles with straight edges and pieces of puzzles without edges have to be picked out.	Step 2: Those components from an existed software system have to be shorted out, which can be treated as outside components. Or short out components, based on other properties of the components. (Here it first has to be determined which components can be considered as outside or inside components.)
Step 3: Children have to find the puzzle-corners and place them correctly, using the predetermined picture of the puzzle for reference.	Step 3: First learners have to find those components which need to be considered as outside components and start to integrate those components together using an existent (similar/same) software system for reference.
Step 4: Then children have to start building the whole puzzle as followed: the outside puzzles first and then the inside puzzles have to be built in. For the children is necessary to be showed how to look for pieces of puzzles which have similar colors in order to match the puzzle pieces up.	Step 4: Then learners have to start to integrate the software system as followed: first the outside components have to be integrated and then the inside components. It is necessary to show the learners how to look for the right component which has a similar property.

Table 7.4: Method 2

Method 3: Build Sections First	**Method 3: Build parts of a software system first**
Step 1: The pieces of puzzles have to be turned over and the completed picture, that has to be built up, has to be studied by the children.	Step 1: The components of an existent software system have to be shown for the learners, by showing all the components one by one with their interfaces/input/output and let learners to analyze/read it.
Step 2: Children have to arrange puzzle pieces that are similar together and then children have to start building one section.	Step 2: Learners have to integrate components with similar purpose together and have to start building one part of the software system.
Step 3 : As the next step children have to build the next section of the puzzle.	Step 3: As the next step learners have to integrate the next components with similar purpose to build the next subsystem.
Step 4: If the children are finished with building the different sections together, then they must search which puzzle pieces miss and start to place those puzzles, i.e. the outside edges.	Step 4: If the learners are ready with integrating the different sections of the software system, then as a next step, learners can integrate the remainder components into the software system.

Table 7.5: Method 3

8 Summary and Future Work

This thesis introduced a system concept for a Web app. The Web app was named AcadIC. AcadIC was designed for learning the software integration processes for beginners in programming. The creation of AcadIC is composed of several tasks. The basics of component-based development software integration processes, programming teaching, online learning, teamwork in software engineering and cooperative teamwork were researched. The state of the art of online platforms in the field of teaching and learning programming was introduced. Problems, gaps were identified in the above mentioned researched fields.

Based on the related works, the research in the different domains, and the problem identification, requirements were derived, and from those requirements features for AcadIC Web app were designed. A Web app design was given, possible target group, learning goals, expected impacts, and a determination of programming tasks of AcadIC were described.

AcadIC was evaluated by expert interviews, which was given by 4 experts. The evaluation indicated that AcadIC has generally benefit to teach the software integration process, and there are several further possibilities to extend its features. Observation of those online teamwork would be necessary in order to determine the right communication process. The target group and the used programming editor of AcadIC must be thought over again.

Future work could be analyzing the results of the expert interviews and derive possible further features, as well as, AcadIC could be developed further with the described future functions, and the introduced idea for teaching the software integration processes. Further improvements could be to develop AcadIC with features of learning management systems.

Bibliography

References

[1] Marzieh Ahmadzadeh, Dave Elliman, and Colin Higgins. „The impact of improving debugging skill on programming ability“. In: *Innovation in Teaching and Learning in Information and Computer Sciences* 6.4 (2007), pp. 72–87.

[2] Marzieh Ahmadzadeh, Dave Elliman, and Colin Higgins. „The impact of improving debugging skill on programming ability“. In: *Innovation in Teaching and Learning in Information and Computer Sciences* 6.4 (2007), pp. 72–87.

[3] José Luis Fernández Alemán. „Automated assessment in a programming tools course“. In: *IEEE Transactions on Education* 54.4 (2011), pp. 576–581.

[4] Wolfgang Menz Alexander Bogner Beate Littig, ed. *Interviewing Experts*. 1st ed. ECPR Research Methods. Page 17. Palgrave Macmillan UK, 2009. ISBN: 978-0-230-24427-6 (e-book). DOI: 10.1057/9780230244276.

[5] Wolfgang Menz Alexander Bogner Beate Littig, ed. *Interviewing Experts*. 1st ed. ECPR Research Methods. Page 2. Palgrave Macmillan UK, 2009. ISBN: 978-0-230-24427-6 (e-book). DOI: 10.1057/9780230244276.

[6] Wolfgang Menz Alexander Bogner Beate Littig, ed. *Interviewing Experts*. 1st ed. ECPR Research Methods. Page 5. Palgrave Macmillan UK, 2009. ISBN: 978-0-230-24427-6 (e-book). DOI: 10.1057/9780230244276.

[7] Wolfgang Menz Alexander Bogner Beate Littig, ed. *Interviewing Experts*. 1st ed. ECPR Research Methods. Page 38. Palgrave Macmillan UK, 2009. ISBN: 978-0-230-24427-6 (e-book). DOI: 10.1057/9780230244276.

[8] Mario Amelung, Katrin Krieger, and Dietmar Rösner. „E-Assessment as a Service“. In: *IEEE Transactions on Learning Technologies* 4.2 (2011), pp. 162–174.

[9] Carmelo Ardito et al. „Usability of e-learning tools“. In: *Proceedings of the working conference on Advanced visual interfaces*. ACM. 2004, pp. 80–84.

[10] Shahdatunnaim Azmi, Noorminshah A Iahad, and Norasnita Ahmad. „Gamification in online collaborative learning for programming courses: A literature review“. In: *ARPN Journal of Engineering and Applied Sciences* 10.23 (2015), pp. 1–3.

[11] Jan Bartelsen and Johannes Brauer. *Kooperatives Lernen mit moodle in der Programmierausbildung*. Tech. rep. Arbeitspapiere der Nordakademie, 2008.

[12] Martin / Stephan Waba Bauer. „eEducation - Digitale Bildung für alle“. In: *Schule Aktiv Coding Ein Baustein der Informatischen Bildung* (2016), p. 16.

[13] Jens Bennedsen, Michael E Caspersen, and Michael Kölling. *Reflections on the teaching of programming: methods and implementations*. Vol. 4821. Springer, 2008.

[14] Anders Berglund and Raymond Lister. „Introductory programming and the didactic triangle“. In: *Proceedings of the Twelfth Australasian Conference on Computing Education-Volume 103*. Australian Computer Society, Inc. 2010, pp. 35–44.

[15] Zinn Bernd. „Editorial: Conditional variables of 'Ausbildung 4.0'–Vocational education for the future.“ In: *Journal of Technical Education (JOTED)* 3(2). (2015).

[16] David Berry. *The philosophy of software: Code and mediation in the digital age.* Springer, 2016.

[17] Pierre Bourque, Richard E Fairley, et al. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0.* IEEE Computer Society Press, 2014.

[18] Sandy Britain and Oleg Liber. „A framework for pedagogical evaluation of virtual learning environments“. In: *Research report* (2004).

[19] Sébastien Combéfis, CLÉMENT de SAINT-MARCQ, et al. „Teaching Programming and Algorithm Design with Pythia, a Web-Based Learning Platform.“ In: *Olympiads in Informatics* 6 (2012).

[20] Thomas H Cormen et al. *Introduction to algorithms second edition.* The MIT Press, 2009.

[21] Declan Dagger et al. „Service-oriented e-learning platforms: From monolithic systems to flexible services“. In: *IEEE Internet Computing* 11.3 (2007), pp. 28–35.

[22] Balakrishnan et al. DASARATHY. „The past, present, and future of MOOCs and their relevance to software engineering“. In: In Proceedings of the on Future of Software Engineering. ACM, 2014, pp. 212–224.

[23] JM Del Alamo, A Alonso, and M Cortés. „Automated Grading and Feedback Providing Assignments Management Module“. In: *ICERI2012 Proceedings.* IATED. 2012, pp. 3784–3793.

[24] Paul Denny, Diana Cukierman, and Jonathan Bhaskar. „Measuring the effect of inventing practice exercises on learning in an introductory programming course“. In: *Proceedings of the 15th Koli Calling Conference on Computing Education Research.* ACM. 2015, pp. 13–22.

[25] M Gåsland. „Game mechanic based e-learning“. In: *Science And Technology, Master Thesis (June 2011). Retrieved October* 4 (2011), p. 2014.

[26] Katerina Georgouli and Pedro Guerreiro. „Integrating an Automatic Judge into an Open Source LMS“. In: *International Journal on E-Learning* 10.1 (2011), pp. 27–42.

[27] Max Goldman, Greg Little, and Robert C Miller. „Collabode: collaborative coding in the browser“. In: *Proceedings of the 4th international workshop on Cooperative and human aspects of software engineering.* ACM. 2011, pp. 65–68.

[28] Thomas Grechenig. *Softwaretechnik: mit Fallbeispielen aus realen Entwicklungsprojekten.* Pearson Deutschland GmbH, 2010.

[29] SÁGI GYÖNGYI. „Született programozó. Mítosz és valóság“. In: *Bitport* (2015). URL: http://bitport.hu/programozonak-szuletni-kell-mitosz-es-valosag.

[30] Wallace H Hannum and Leslie J Briggs. „How does instructional systems design differ from traditional instruction?“ In: *Educational Technology* 22.1 (1982), pp. 9–14.

[31] Ilkka Haukijärvi. „E-Learning Maturity Model–Process-Oriented Assessment and Improvement of e-Learning in a Finnish University of Applied Sciences“. In: *IFIP Conference on Information Technology in Educational Management.* Springer. 2014, pp. 76–93.

[32] Michael Hielscher et al. „ProgrammingWiki in der Praxis–ein Erfahrungsbericht“. In: *Informatik mit Kopf, Herz und Hand–Praxisband zur INFOS 2011* (), pp. 91–100.

[33] Peter Hubwieser. *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. Springer-Verlag, 2007, p. 15.

[34] Watts S Humphrey. *Introduction to the team software process*. Page 20. Addison-Wesley Professional, 2000.

[35] Watts S Humphrey. *Introduction to the team software process*. Page 23-24. Addison-Wesley Professional, 2000.

[36] Watts S Humphrey. *Introduction to the team software process*. Page 20-21. Addison-Wesley Professional, 2000.

[37] Watts S Humphrey. *Introduction to the team software process*. Page 17-18. Addison-Wesley Professional, 2000.

[38] Petri Ihantola et al. „Automated assessment of programming assignments: visual feedback, assignment mobility, and assessment of students' testing skills". In: (2011).

[39] Jouni Ikonen et al. „Is the world ready or do we need more tools for programming related teamwork?" In: *Proceedings of the 15th Koli Calling Conference on Computing Education Research*. ACM. 2015, pp. 33–39.

[40] Syed Roohullah Jan et al. „An Innovative Approach to Investigate Various Software Testing Techniques and Strategies". In: *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN* (2016), pp. 2395–1990.

[41] David Jonassen and Susan Land. *Theoretical foundations of learning environments*. Routledge, 2012.

[42] Orban Jozsefne. *A kooperativ tanulas: szervezes es alkalmazas*. Ed. by 1. Page 23-26. Orban es Orban, 2009. ISBN: 9789630677325.

[43] Orban Jozsefne. *A kooperativ tanulas: szervezes es alkalmazas*. Ed. by 1. Page 10-11. Orban es Orban, 2009. ISBN: 9789630677325.

[44] Orban Jozsefne. *A kooperativ tanulas: szervezes es alkalmazas*. Ed. by 1. Page 41-42. Orban es Orban, 2009. ISBN: 9789630677325.

[45] Orban Jozsefne. *A kooperativ tanulas: szervezes es alkalmazas*. Ed. by 1. Page 54-63. Orban es Orban, 2009. ISBN: 9789630677325.

[46] Orban Jozsefne. *A kooperativ tanulas: szervezes es alkalmazas*. Ed. by 1. Page 26. Orban es Orban, 2009. ISBN: 9789630677325.

[47] Jens J Kaasbøll. „Exploring didactic models for programming". In: *NIK 98–Norwegian Computer Science Conference*. Citeseer. 1998, pp. 195–203.

[48] Mary Kalantzis and Bill Cope. „Learning and New Media". In: *The SAGE Handbook of Learning* (2015), p. 373.

[49] Mohd Ehmer Khan. „Different Software Testing Levels for Detecting Errors". In: *International Journal of Software Engineering (IJSE)* 2.4 (2011), pp. 70–80.

[50] Terhi Kilamo, Imed Hammouda, and Mohamed Amine Chatti. „Teaching collaborative software development: A case study". In: *Proceedings of the 34th International Conference on Software Engineering*. IEEE Press. 2012, pp. 1165–1174.

[51] Terhi Kilamo et al. „Knowledge transfer in collaborative teams: experiences from a two-week code camp". In: *Companion Proceedings of the 36th International Conference on Software Engineering*. ACM. 2014, pp. 264–271.

[52] Antti Knutas, Jouni Ikonen, and Jari Porras. „COMPUTERSUPPORTED COLLABORATIVE LEARNING IN SOFTWARE ENGINEERING EDUCATION: A SYSTEMATIC MAPPING STUDY“. In: *Journal on Information Technologies & Security* 7 (2015), p. 4.

[53] Ralph Kölle, Glenn Langemeier, and Wolfgang Semar. „Programmieren lernen in kollaborativen Lernumgebungen“. In: *Effektive Information-Retrieval-Verfahren in Theorie und Praxis. Ausgewählte und erweiterte Beiträge des Vierten Hildesheimer Evaluierungs-und Retrievalworkshop. Universitätsverlag Konstanz* (2006), pp. 147–170.

[54] Utku Köse. „A web based system for project-based learning activities in “web design and programming” course“. In: *Procedia-Social and Behavioral Sciences* 2.2 (2010), pp. 1174–1184.

[55] Adrian Kosowski, Michał Małafiejski, and Tomasz Noiński. „Application of an online judge & contester system in academic tuition“. In: *International Conference on Web-Based Learning*. Springer. 2007, pp. 343–354.

[56] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. „A study of the difficulties of novice programmers“. In: *ACM SIGCSE Bulletin*. Vol. 37. 3. ACM. 2005, pp. 14–18.

[57] Szlavi Peter–Zsako Laszlo. „Methods of teaching programming“. In: ().

[58] Martin Lehner. *Allgemeine Didaktik: Eine Einführung*. Vol. 3245. UTB, 2009.

[59] Sajjad Mahmood. „Empirical study of software component integration process activities“. In: *IET software* 7.2 (2013), pp. 65–75.

[60] Rennie Frank Maison Robin. *E-learning and social networking handbook*. Taylor and Francis, 2008. ISBN: 978-0-415-42606-0.

[62] Dr. Charaf Hassan Marossy Kálmán. *Komponens alapú programozás COM+ és .NET környezetben*. Tech. rep. BME, Automatizálási és Alk. Informatikai Tanszék, 2001. URL: https://nws.niif.hu/ncd2002/docs/ehu/28/.

[63] E. Richard / Clark Colvin Ruth Mayer. *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. Ed. by 3. Edition. John Wiley & Sons, 2008.

[64] Terry Mayes and Sara De Freitas. „Review of e-learning theories, frameworks and models“. In: *JISC e-learning models desk study* 1 (2004).

[65] Daniel McFarland and Diane Hamilton. „Factors affecting student performance and satisfaction: Online versus traditional course delivery“. In: *Journal of Computer Information Systems* 46.2 (2005), pp. 25–32.

[66] Harald A Mieg and Matthias Näf. „Skript-Experteninterviews in den Umwelt-und Planungswissenschaften“. In: *Eine Einführung und Anleitung. ETH Zürich. Download unter: http://www. mieg. ethz. ch/education/Skript_Experteninterviews. pdf* 20 (2005). Page 22, p. 2013.

[67] Iwona Miliszewska and Grace Tan. „Befriending computer programming: A proposed approach to teaching introductory programming“. In: *Informing Science: International Journal of an Emerging Transdiscipline* 4.1 (2007), pp. 277–289.

[68] Sanja Mohorovičić and Vedran Strčić. „An Overview of Computer Programming Teaching Methods“. In: *XXII Central European Conference on Information and Intelligent Systems*. 2011, pp. 1–6.

[69] Nyirő Péter Molnár György. *A gyakorlati programozás tanításának játékfejlesztésen alapuló, élménypedagógiai alapú módszerének bemutatása*. ISBN 978-80-89691-32-6. Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest, 2015. URL: http://www.irisro.org/pedagogia2016kotet/31MolnarGyorgy-NyiroPeter.pdf.

[70] Ilhyun Moon et al. „Virtual education system for the c programming language“. In: *International Conference on Web-Based Learning*. Springer. 2008, pp. 196–207.

[71] Marek Nowicki et al. „On-line Contests Hosting Service as a Tool to Teach Computer Science Students Programming“. In: ().

[72] Anne L Olsen. „Using pseudocode to teach problem solving“. In: *Journal of Computing Sciences in Colleges* 21.2 (2005), pp. 231–236.

[73] Arnold Pears et al. „A survey of literature on the teaching of introductory programming“. In: *ACM SIGCSE Bulletin* 39.4 (2007), pp. 204–223.

[74] Margit Pohl et al. „Computer supported collaborative learning and vocational training: adapting the technology to the learners? needs“. In: *Universal Access in the Information Society* 7.4 (2008), pp. 259–272.

[75] Loreen M Powell and Hayden Wimmer. „Evaluating the effectiveness of student group work for mobile application development learning, productivity, enjoyment and confidence in quality“. In: *Proceedings of the EDSIG Conference*. 3456. 2015.

[76] Pap-Szigeti Robert. „Kooperativ modszerek alkalmazasa a felsooktatasban“. In: *Iskolakultura* 17.1 (2007). Pannon Egyetem Gondolat Kiado, pp. 56–66.

[77] Lesley Anne Robertson. *Simple Program Design, A Step-by-Step Approach*. Page 2. Cengage Learning, 2006.

[78] Michael Schumm et al. „Required competences in software engineering: Pair programming as an instrument for facilitating life-long learning“. In: *Interactive Collaborative Learning (ICL), 2012 15th International Conference on*. IEEE. 2012, pp. 1–5.

[79] Sharanjit Singh, Amardeep Singh, and Sandeep Singh Samson. „Component Based Software Engineering“. In: (2016).

[80] Gerry Stahl, Timothy Koschmann, and Dan Suthers. „Computer-supported collaborative learning: An historical perspective“. In: *Cambridge handbook of the learning sciences* 2006 (2006), pp. 409–426.

[81] Jakub Swacha and Paweł Baszuro. „Gamification-based e-learning platform for computer programming education“. In: *X World Conference on Computers in Education*. 2013, pp. 122–130.

[82] Sparks System. *THE COMPONENT MODEL*. Tech. rep. Enterprise Architect, 2004. URL: http://www.sparxsystems.com.au/downloads/whitepapers/The_Component_Model.pdf.

[83] Kun Tian, Kendra Cooper, and Kang Zhang. „Improving Software Engineering Education through Enhanced Practical Experiences“. In: *Computer and Information Science ICIS, 2011 IEEE ACIS 10th International Conference on*. IEEE. 2011, pp. 292–297.

[84] HH Tie and Irfan Naufal Umar. „The impact of learning styles and instructional methods on students' recall and retention in programming education“. In: *Proceedings of the 18th International Conference on Computers in Education*. 2010, pp. 191–195.

[85] Leo Van Audenhove. „Expert interviews and interview techniques for policy analysis“. In: *Vrije University, Brussel Retrieved May* 5 (2007), p. 2009.

[86] SR VÉGH Ladislav. „INTERAKTÍV ALGORITMUS ANIMÁCIÓK AZ OKTATÁSBAN“. In: *XXIX. DIDMATTECH 2016* (2016). URL: http://didmattech.inf.elte.hu/wp-content/uploads/2016/08/VL_Interactive-algorithm.pdf.

[87] Maria Virvou and Sotirios Christos Sidiropoulos. „Collaborative tools in learning a programming language“. In: *e-Learning and e-Technologies in Education (ICEEE), 2012 International Conference on*. IEEE. 2012, pp. 162–165.

[88] Aurora Vizcaino et al. „An adaptive, collaborative environment to develop good habits in programming“. In: *Intelligent Tutoring Systems*. Springer. 2000, pp. 262–271.

Online References

[61] Herger Mario. *Gamification Platform Matrix*. Reviewed: 05.02.2015. 2012. URL: http://enterprisegamification.com/index.php/en/resources/platforms.

A Appendix

A.1 Interview questions, a probe interview

Personal questions

Please summarize your educational background and your current profession.

Report on your practical experience in teaching programming and e-learning.

Persönliche Fragen

Erläutern Sie Ihren Bildungshintergrund und Ihren aktuellen Beruf.

Berichten Sie über Ihre praktische Erfahrung im Bereich der Programmierlehre und im E-Learning.

Software Integration

Which problems arise in your experience during the integration of software components?

In your opinion, how can the described concept help to teach the integration process to students?

Softwareintegration

Welche Probleme treten Ihrer Erfahrung nach während des Integrationsprozesses von Softwarekomponenten auf?

Inwiefern kann Ihrer Meinung nach das beschriebene Konzept helfen Studierenden den Integrationsprozess von Softwarekomponenten zu lehren?

Learning to program

In your view, how can the presented concept help students to learn programming in relation to varying knowledge levels?

The presented concept employs visual elements to assemble prefabricated code blocks into a functioning program with a drag and drop technique. In your opinion, how is this approach beneficial to learning progress?

Programmieren lernen

Inwiefern kann Ihrer Meinung nach das vorgestellte Konzept Studierenden helfen Programmieren zu lernen in Bezug auf unterschiedlichen Wissensstand?

Das vorgestellte Konzept sieht die Verwendung visueller Elemente und DragandDrop vor um vorgefertigte Codeblöcke zu einem funktionierenden Programmteil zusammenzufügen. Inwiefern ist dieser Ansatz Ihrer Meinung nach dem Lernfortschritt zuträglich?

Teamwork

In what way does the showcased concept affect the cooperation and team skills of students in your judgement?

In your opinion, which elements of the showcased concept have the greatest significance in relation to teamwork and why?

Teamarbeit

Inwiefern beeinflusst das vorgestellte Konzept Ihrer Meinung nach die Kooperations- und Teamfähigkeit von Studierenden?

Welche Elemente des vorgestellten Konzepts haben Ihrer Meinung nach die größte Bedeutung in Bezug auf Teamarbeit und warum?

Software Development in a Team

In your opinion, how does the concept affect the ability of a student to work in a team in an IT profession and can the concept facilitate particular skills, if yes which ones and why?

Softwareentwicklung im Team

Inwiefern beeinflusst das Konzept Ihrer Meinung nach die Befähigung des Studierenden in einem ITBeruf als Teil eines Teams zu arbeiten und kann das Konzept bestimmte Kompetenzen fördern, wenn ja welche und warum?

E-Learning

In what way can the presented concept be developed further in relation to E-Learning in your view?

In your opinion, what is missing from the platform? In which direction should the platform be developed further?

E-Learning

In welcher Weise könnte das vorgestellte Konzept Ihrer Meinung nach, in Bezug auf E-Learning, weiterentwickelt werden?

Ihrer Meinung nach was fehlt auf der Plattform? In welche Richtung soll die Plattform weiterentwickeln?

A.2 Interview questions

Personal questions

E1:Please summarize your educational background and your current profession.

E2:Report on your practical experience in teaching programming and e-learning.

Persönliche Fragen

D1:Erläutern Sie Ihren Bildungshintergrund und Ihren aktuellen Beruf.

D2:Berichten Sie über Ihre praktische Erfahrung im Bereich der Programmierlehre und im E-Learning.

Software Integration

E3:Which non-technical problems arise in your experience during the integration of software components?

E4:In your opinion, how can the described concept help to teach the integration process to students?

Softwareintegration

D3:Welche nicht-technische Probleme treten Ihrer Erfahrung nach während des Integrationsprozesses von Softwarekomponenten auf?

D4:Inwiefern kann Ihrer Meinung nach das beschriebene Konzept helfen Studierenden den Integrationsprozess von Softwarekomponenten zu lehren?

Learning to program

E5: In your view, how can the presented concept help students to learn programming in relation to varying knowledge levels?

E6:The presented concept employs visual elements to assemble prefabricated code blocks into a functioning program with a drag and drop technique. In your opinion, how is this approach beneficial to learning progress?

Programmieren lernen

D5:Inwiefern kann Ihrer Meinung nach das vorgestellte Konzept Studierenden helfen Programmieren zu lernen in Bezug auf unterschiedlichen Wissensstand?

D6:Das vorgestellte Konzept sieht die Verwendung visueller Elemente und DragandDrop vor um vorgefertigte Codeblöcke zu einem funktionierenden Programmteil zusammenzufügen. Inwiefern ist dieser Ansatz Ihrer Meinung nach dem Lernfortschritt zuträglich?

Teamwork

E7:In what way does the showcased concept affect the cooperation and team skills of students in your judgement?

E8:In your opinion, which elements of the showcased concept have the greatest significance in relation to teamwork and why?

Teamarbeit

D7:Inwiefern beeinflusst das vorgestellte Konzept Ihrer Meinung nach die Kooperations- und Teamfähigkeit von Studierenden?

D8:Welche Elemente des vorgestellten Konzepts haben Ihrer Meinung nach die größte Bedeutung in Bezug auf Teamarbeit und warum?

Software Development in a Team

E9:In your opinion, how does the concept affect the ability of a student to work in a team in an IT profession?

E10: Which skills can the concept facilitate and why?

Softwareentwicklung im Team

D9:Inwiefern beeinflusst das Konzept Ihrer Meinung nach die Befähigung des Studierenden in einem IT-Beruf als Teil eines Teams zu arbeiten?

D10: Welche Kompetenzen kann das Konzept fördern und warum?

E-Learning

E11:In what way can the presented concept be developed further in relation to E-Learning in your view?

E12:In your opinion, what is missing from the platform? In which direction should the platform be developed further?

E-Learning

D11:In welcher Weise könnte das vorgestellte Konzept Ihrer Meinung nach, in Bezug auf E-Learning, weiterentwickelt werden?

D12:Ihrer Meinung nach was fehlt auf der Plattform? In welche Richtung soll die Plattform weiterentwickeln?

A.3 Interview Consent Form

Interview Consent Form

This expert interview was conducted by Mészárosné Csuta Eszter as part of a master thesis for the master study Didactics for Informatics. Thank you, for agreeing to be interviewed.

In order to facilitate the evaluation work, the interview will be recorded and a transcript will be produced.

All data collected will be treated confidentially and will not be passed on to third parties. The results will appear in my master thesis but they are anonymous.

The recording will only be used internally and for my master's thesis, to demonstrate the results. The recording will not be published.

Your participation in the interview is voluntary. You have the right to stop the interview or withdraw from the interview at any time.

Please read the following statement and sign it below. Thank you!

I agree that I will be recorded (audio) while being interviewed.

I expressly permit this record to be used for the purpose of examining and demonstrating the interview results as part of a Master thesis.

- Surname, First name (please in PRINTED LETTERS):
- Date, Place:
- Signature:

Einverständniserklärung

Dieses Experteninterview wird von Eszter Csuta Mészárosné im Rahmen einer Masterarbeit für den Studiengang Informatikdidaktik durchgeführt. Vielen Dank, dass sie mir helfen die Anforderungen der Funktionalitäten für eine Lernsoftware zu erheben.

Um mir die Auswertungsarbeit zu erleichtern, werde ich während des Interviews das Gesprochene aufzeichnen und notieren. Alle von Ihnen erhobenen Daten werden vertraulich behandelt und nicht an Außenstehende weitergegeben. Die Ergebnisse werden in der Masterarbeit aufscheinen, sie sind jedoch anonymisiert.

Die Aufzeichnung wird nur intern und für meine Masterarbeit verwendet, um die Ergebnisse zu demonstrieren. Die Aufzeichnung wird nicht veröffentlicht.

Ihre Teilnahme an dem Interview ist freiwillig. Sie können die Bereitschaft zur Teilnahme jederzeit widerrufen, beziehungsweise die Teilnahme an dem Interview abbrechen.

Bitte lesen Sie die folgende Erklärung und unterschreiben Sie darunter.

Vielen Dank!

Ich bin einverstanden, dass ich in Ton aufgezeichnet werde, während ich interviewt werde.

Ich gestatte es ausdrücklich, diese Aufzeichnung zu Zwecken der Untersuchung und Demonstration der Interviewergebnisse im Rahmen einer Masterarbeit zu wenden.

- Name, Vorname (bitte in DRUCKBUCHSTABEN):
- Datum, Ort:
- Unterschrift:

A.4 Drawing of the System Concept

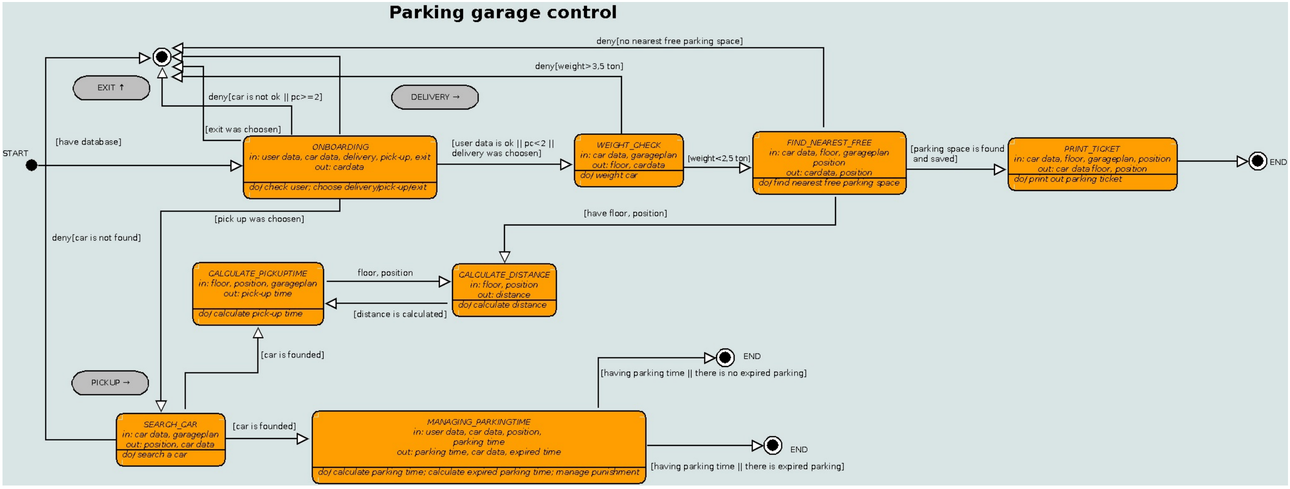

Figure A.1: Drawing of the System Concept

A.5 Presentations Slides

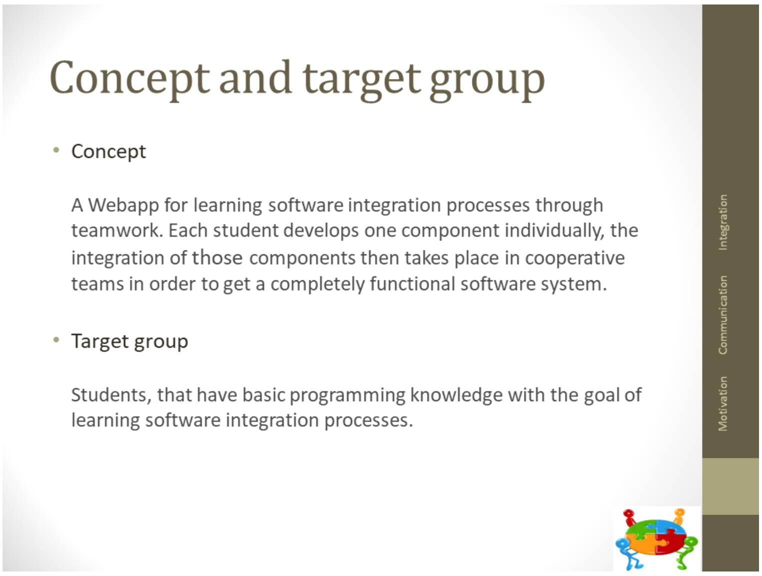

Figure A.2: Presentations Slides I.

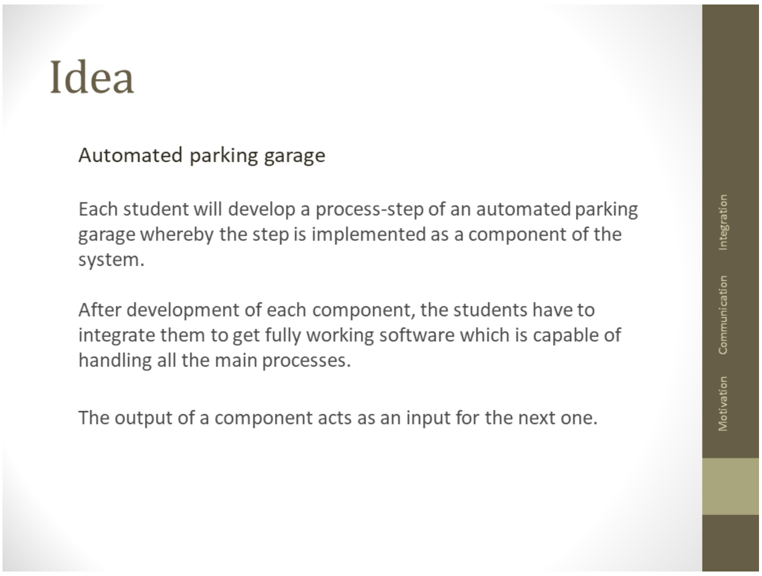

Figure A.3: Presentations Slides II.

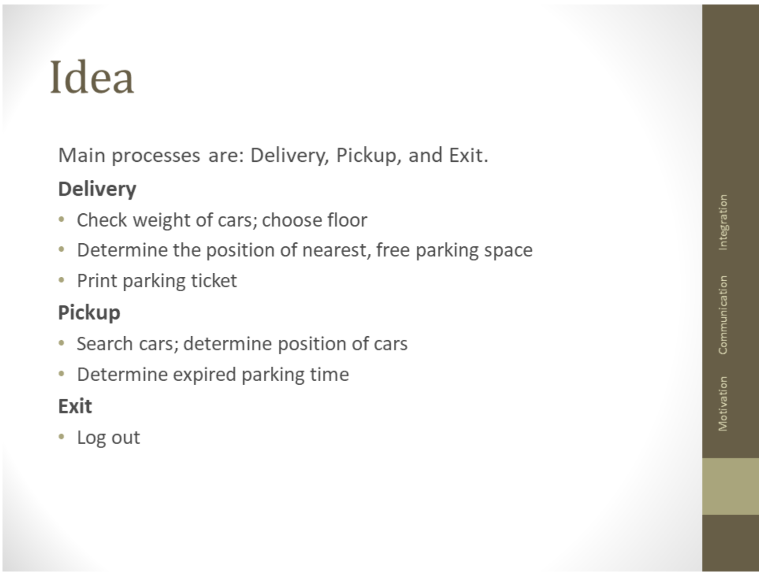

Figure A.4: Presentations Slides III.

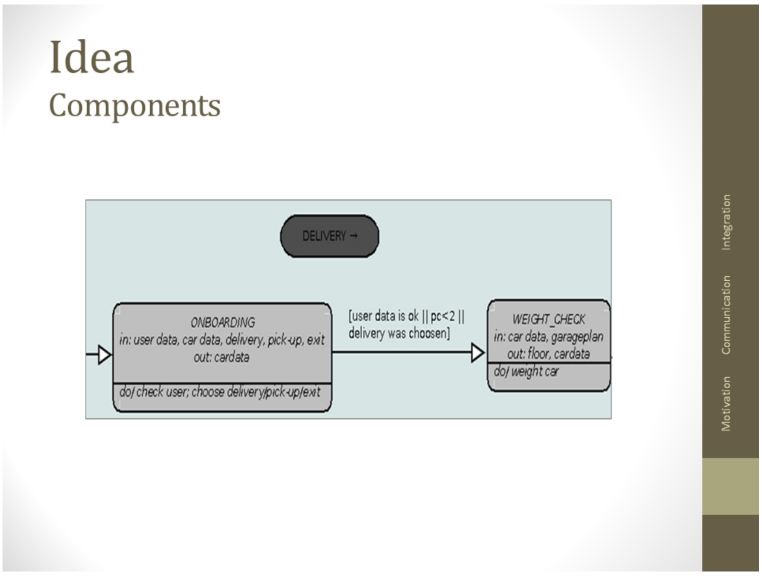

Figure A.5: Presentations Slides IV.

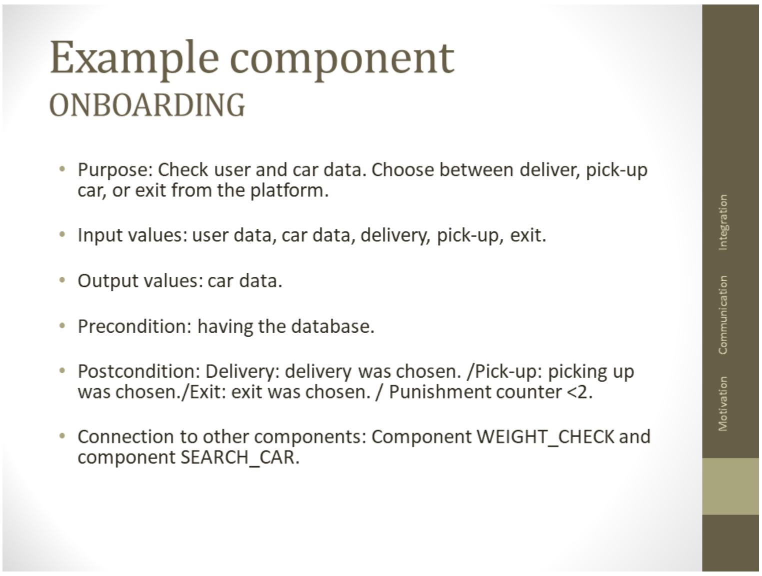

Figure A.6: Presentations Slides V.

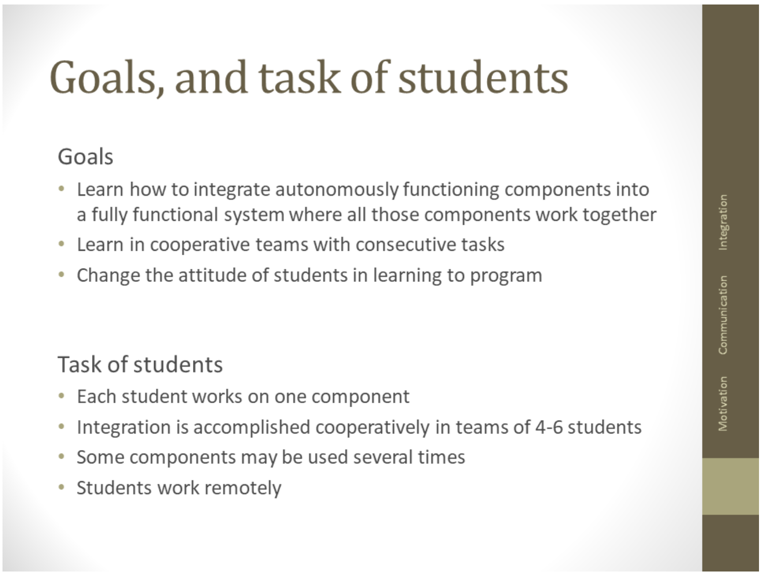

Figure A.7: Presentations Slides VI.

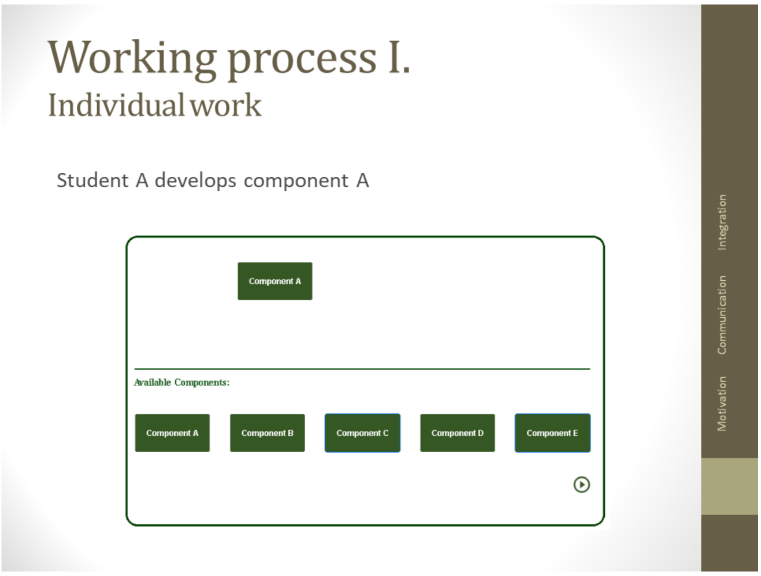

Figure A.8: Presentations Slides VII.

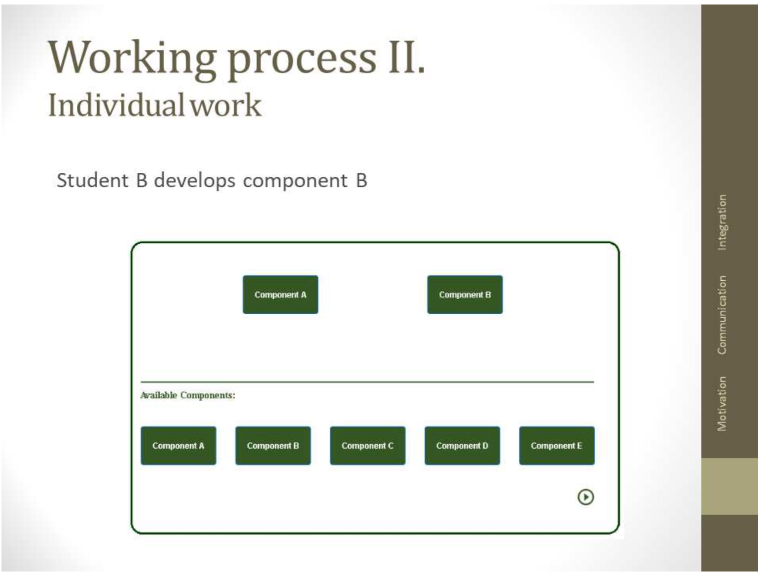

Figure A.9: Presentations Slides VIII.

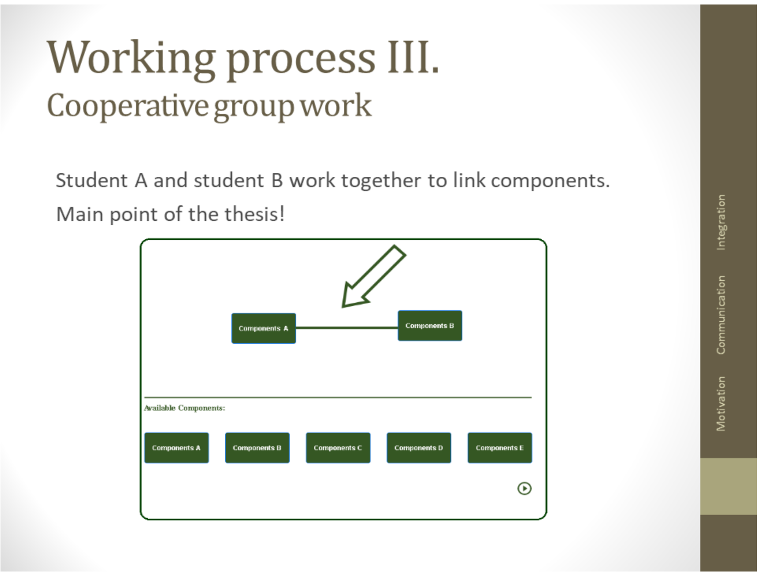

Figure A.10: Presentations Slides XI.

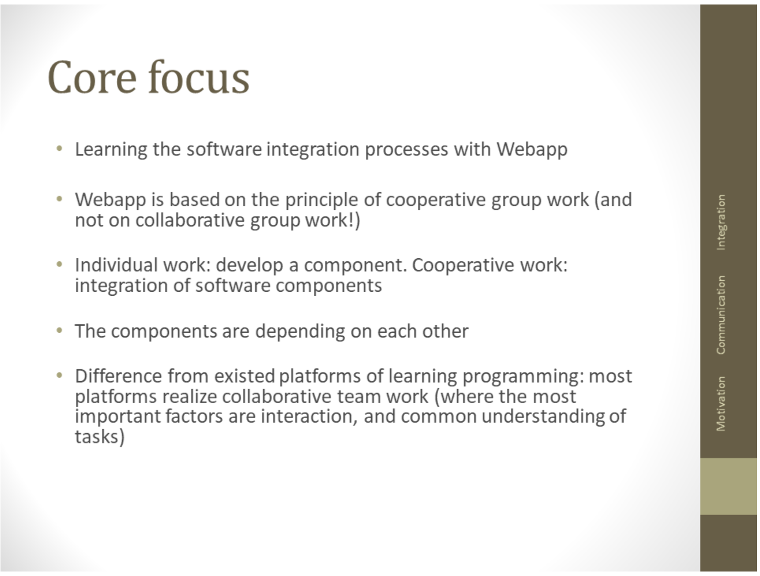

Figure A.11: Presentations Slides X.

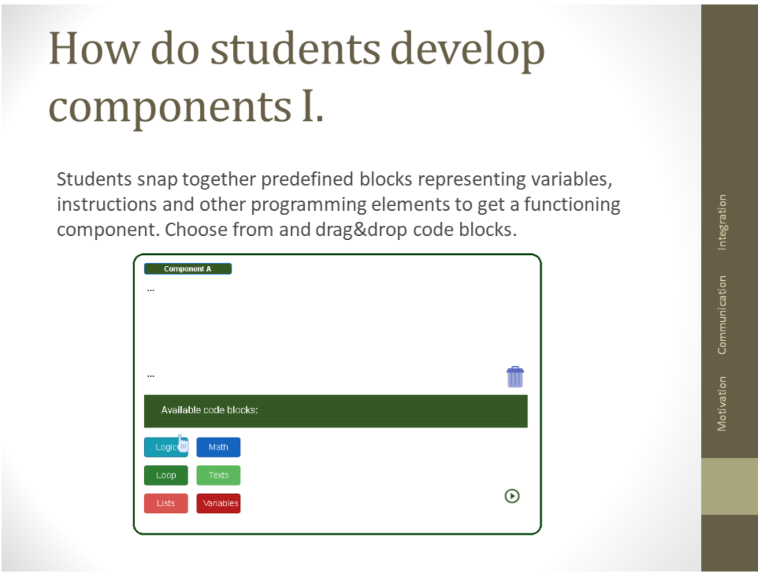

Figure A.12: Presentations Slides XI.

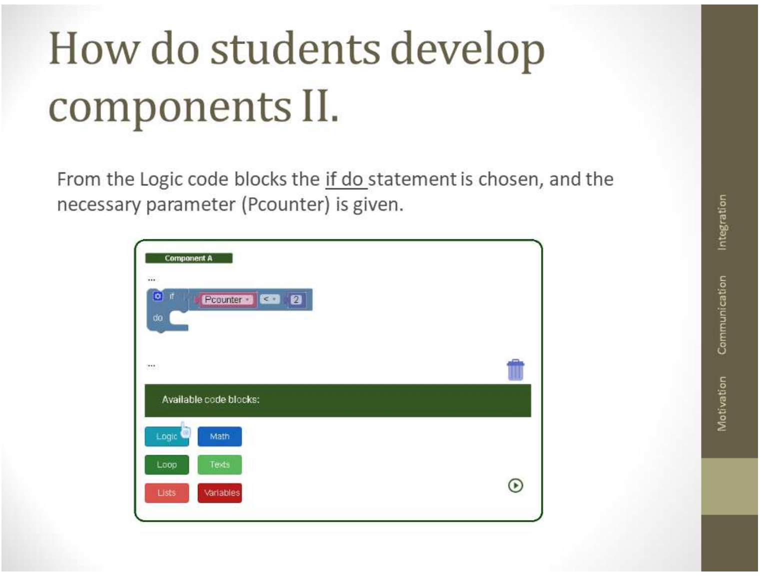

Figure A.13: Presentations Slides XII.

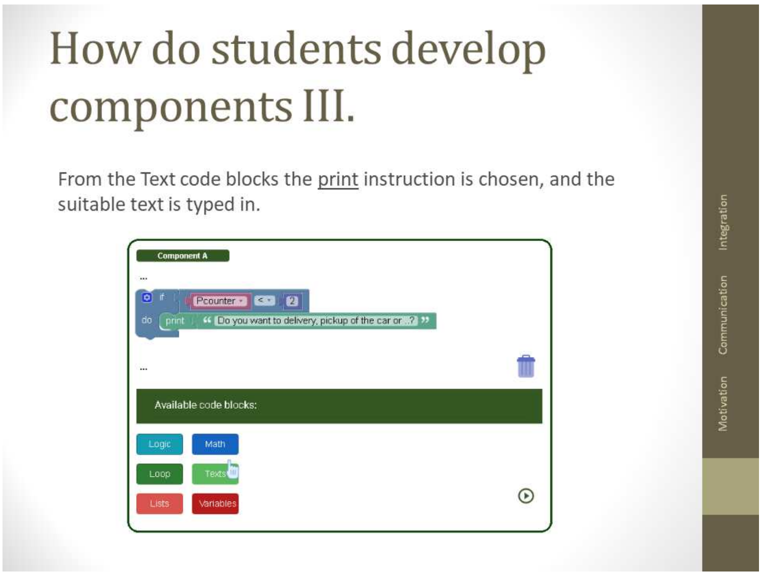

Figure A.14: Presentations Slides XIII.

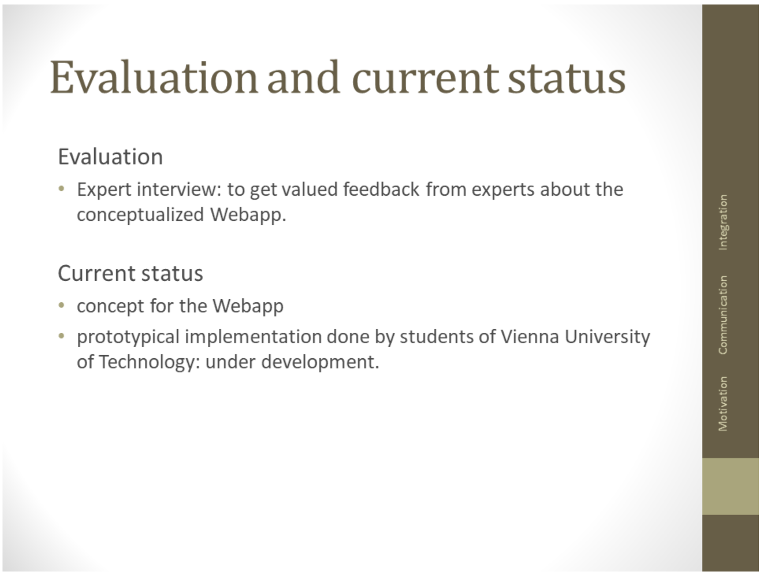

Figure A.15: Presentations Slides XIV.