Research Skills for Software Engineering Undergraduates in Dutch Universities of Applied Sciences

Mortaza S. Bargh
Creating 010
Rotterdam University of
Applied Sciences, The Netherlands
m.shoae.bargh@hr.nl

Annette van Rooij-Peiman
CMI institute
Rotterdam University of
Applied Sciences, The Netherlands
a.c.van.rooij-peiman@hr.nl

Leo Remijn
CMI institute
Rotterdam University of
Applied Sciences, The Netherlands
l.n.m.remijn@hr.nl

Sunil Choenni
Creating 010
Rotterdam University of
Applied Sciences, The Netherlands
r.choenni@hr.nl

ABSTRACT

Undergraduate students who seek a bachelor degree in Dutch universities of applied sciences are supposed to learn also research skills so that they can provide innovative solutions to real problems of the society and businesses in their future careers. Current education and textbooks on research skills are not tuned well to software engineering disciplines. This paper describes our vision about the scope and model of the research suitable for software engineering disciplines in Dutch universities of applied sciences. Based on literature study we identify a number of research models that are commonly used in computer science. Through reviewing a number of graduation reports in our university, we further identify which of the research models are most suitable for the (graduation) projects of software engineering disciplines and also investigate their shortcomings with respect to the desired research skills. Our study reveals that the approach of most graduation works is close to the implementation-based (also called build-based or proof by example based) research model. In order to be considered as a realization of sound applied research, however, most of theses graduation works need to be improved on a number of aspects such as problem context definition, system/prototype evaluation, and critical literature study.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *curriculum, computer science education*

Keywords

Applied science; education; research skills; software engineering; undergraduates

1. INTRODUCTION

In The Netherlands there are two types of universities: Dutch universities of applied sciences and Dutch scientific universities. These universities provide higher-level vocational education and highly specialized (and scientific) education, respectively. There are 38 universities of applied sciences in the Netherlands currently, which are responsible for educating about two-thirds of the country's higher education students [18]. These applied universities deliver mainly bachelor level education in a vast variety of disciplines and prepare their graduates for professions with a hands on experience mentality. In recent years these applied universities have aimed at embedding research skills in their curricula in an attempt to prepare their graduates for the rising dynamicity and volatility that we witness in various professions and expertise areas. The dynamicity and volatility stem from the fast pace of innovations occurring in areas such as Information and Computer Technology (ICT).

Mastering research skills is considered important and necessary for the graduates of Dutch universities of applied sciences nowadays. These graduates will be future experts and practitioners who are supposed to act as knowledge-oriented professionals. In the field of ICT, such professionals should translate and transform scientific results to practical applications within various application domains like healthcare, logistics and transport, education, wellbeing and (business) administration. The objective is to equip these professionals with a set of tools and skills so that they can independently consume computer science knowledge and produce useful and useable solutions that address real problems of individuals and organizations. In this way they directly contribute to innovations in ICT (application) fields.

Following the trend at Dutch universities of applies sciences, the board of Rotterdam University of Applied Sciences (RUAS), for example, has envisioned educating its bachelor students on research skills along the following directions [11]: *having a researcher attitude and mentality* where the student works methodically, interprets relevant data, reflects critically (on, for example, the objectives, assumptions, context, approach and results), forms own opinions and draws conclusions; *having an entrepreneurial attitude and mentality* where the student is problem-oriented and result-driven and tries to find practical solutions for real problems; *being multidisciplinary*, where the student has an eye on a broader context and reflects on the bigger picture than of own work; and *being communicative*, where the student conveys the solutions and the corresponding argumentations to the public and experts. Similar objectives are envisioned within other Dutch universities of applied sciences.

SIGITE'14, October 15–18, 2014, Atlanta, GA, USA.
Copyright © 2014 ACM 978-1-4503-2686-5/14/10…$15.00
http://dx.doi.org/10.1145/2656450.2656477

There is, however, a gap between the envisioned generic research skills (mentioned above) and the desired research skills that are applicable for the computer and software engineering graduates at the bachelor level. Within RUAS there are three disciplines concerned with software engineering, namely: Informatics (INF) for application related software development, Media Technology (MT) for Human Computer Interaction related software development, and Technical Informatics (TI) for infrastructure related software development. In regard to the envisioned and desired research skills, the current curricula of these disciplines do not nurture adequately the education of research skills and some individuals – ranging from lecturers, students, coaches at associated companies and organizations – unjustifiably and ironically get an impression that these ambitions do not match with or even contradict the way that software engineering disciplines and professions work. Therefore it has become a challenge to motivate these individuals to pursue and educate the research skills mentioned.

One of the reasons behind the existing gap is lack of classical books on the research methodologies and skills needed in the field of computer science and software engineering. Although the existing college books cover a broad spectrum of research skills, for example [3] [4] [15] [10], we have not found a textbook suitable for our target group. Lack of such classical books, in turn, stems from or is related to the fact that computer science and software engineering are relatively new disciplines where research methods are less known compared to those of other disciplines. Also the research in these fields has a volatile and dynamic nature as a result of coping with continuous and rapid developments occurring in ICT fields.

In this contribution we intend to translate the vision of RUAS board into the domain(s) of the software engineering disciplines and move one step towards embedding the required research skills and topics in the curricula of these disciplines. Specifically, the research questions that we would like to address hereto are:

- What does mean research for software engineering disciplines at the bachelor level in universities of applied sciences?
- Which research models are relevant for software engineering disciplines?
- What changes and adaptions are needed to the current approaches of the TI, INF and MT disciplines to enhance the ongoing practices to the level desired?

More specifically, we would like to identify the gaps between how teaching and applying research skills are practiced currently and how they should be practiced in order to prepare RUAS students for finding practical and innovative solutions for real problems. For example, we have to know the shortcomings of the current approaches that students use in executing their assignments with respect to applying the relevant research skills. Hereto it is imperative first to define the set of the research models relevant for software engineering disciplines. Answering these questions will ultimately enable us to develop a scheme for embedding the desired research skills seamlessly in the corresponding curricula. This contribution, which pioneers on its topic in The Netherlands, aims at encouraging the stakeholders (i.e., lecturers, educational staff, students and industry partners) to improve the ways of teaching, learning and applying the research skills within software engineering disciplines.

The paper is organized as follows. Section 2 provides some preliminary information about our research methodology, a generic definition of research and the common research models used in computer science. Our vision of research for software engineering education is described in Section 3. Section 4 presents our analysis of research skill status in INF, MT and TI disciplines, identifies the shortcomings, and elaborates upon improvement aspects. Finally Section 5 draws some conclusions and outlines a number of directions for future work.

2. PRELIMINARIES

2.1 Methodology

This paper is the result of a participatory research, mainly to develop a vision for improving the software engineering education at RUAS. We formed a workgroup consisting of 3 educators who conferred with about 10 other educators from the above-mentioned disciplines in different occasions (including one brainstorming session and three workshops). The workgroup met weekly to brainstorm and share experience about the issues and the (provisionary) solution directions. Through these brainstorming sessions and workshops we took a bottom up approach to elucidate and elicit the tacit knowledge of the practitioners (i.e., carried out a participatory research).

In between these meetings, the workgroup members conducted literature study to learn from the best practices and the state of the art. This study led to identifying 6 generic research models used in computer science and software engineering (see Subsection 2.3). Subsequently we systematically reviewed 60 reports of bachelor graduation projects (see Section 4). We used the reports of the graduation projects, as they are the most comprehensive and representative projects carried out within their curricula. We developed a protocol to review these reports (to be described in Subsection 4.1), whereby we classified the graduation reports according to the identified generic research models. Subsequently we identified the existing shortcomings from the viewpoint of how well the students applied the corresponding research models. Knowing these shortcomings led us to come up with some measures to complement the current approaches so that they accommodate and nurture the desired research skills (see Subsections 4.2).

2.2 Definitions

According to Ellis and Levy [5] research is the process of collecting and analyzing new information/data in order to enhance the body of knowledge, i.e., to create identifiable new knowledge, in an applicable domain. Similarly, Archer [2] considers research as a systematic enquiry in order to produce communicable knowledge. In other words, research is done according to a plan (i.e., being systematic) to find answers to some questions (i.e., being inquiry based). The result of research should be understandable to an audience (i.e., being communicable) and be more than mere information (i.e., being knowledge).

When the acquired information is new for an entity (a person or an organization) but is already known in the literature of that domain, the process is not considered research [8]. In order to determine whether an endeavor is research one has to ask the following questions according to [2]: (1) “Was the activity directed towards the acquisition of knowledge?” (2) “Was it systematically conducted?” (3) “Were the findings explicit?” (4) Was the record of the activity transparent and replicable? (5) “Were the data employed, and the outcome arrived at, validated in appropriate ways?” (6) “Were the findings knowledge rather than information?” (7) “Was the knowledge transmissible to others?” By virtue of the definition of research, see [5], we believe that the term “knowledge” mentioned in question 6 is “new knowledge, in an applicable domain" [5]. If the answers of all these questions are yes, then the corresponding activity is considered as research [2].

2.3 Research Models

For our study it is crucial to shed light on the ways that research is carried out is computer science. Here we enlist a number of typical research models used within various computer science disciplines, without intending to be exhaustive. Our aim is to build a taxonomy, from which one can derive the research methods/ approaches that are most relevant for software engineering disciplines. We should remind that there are papers like [14] and [7] that present more detailed categorizations of the research methods used in computer science. Our list presented below, summarized mainly from [1] and [9], provides a simple, relevant and to the point categorization based on the feedback that we have received from our colleagues and students. Note that the research models to be mentioned are not exhaustive nor are they mutually exclusive (i.e., a research in computer science may rely on a combination of these models).

Formal-based: Here the researcher develops a mathematical model of the artifact (i.e., software implementation) to be created. Subsequently, mathematical proofs, also called formal methods, are used to verify the properties of the artifact. These properties include time complexity, space complexity, correctness of an algorithm, and the validity of a hypothesis given some evidence.

Model-based: Here the researcher defines an abstract model for a real entity/system. The objective of the modeling is to reduce the complexity of the system under study, while keeping the components and component interactions within the model representative to allow a qualitative or quantitative description of the system properties. The researcher can use the model to carry out experiments (i.e., simulations) to reach a better understanding of the system in a cost effective way.

Empirical-based: Here the researcher follows a specific sequence of steps/stages: hypothesis generation (to define or identify the ideas that the research will test), method identification (to determine the techniques that will be used to establish the hypothesis), result compilation (through execution of the method), and conclusion drawing (to support or reject the hypothesis).

Observational-based: Here the researcher is compelled to observe the operation and use of an artifact within its intended working environment rather than explicitly asking users about the performance of the artifacts. It is similar to the empirical model, however, here the researcher approaches the context of work with an open mind and without any hypothesis to prove or disprove.

Implementation-based: Here the researcher realizes a software system as an example of existing a generic class of solutions. This type of research demonstrates the feasibility of the solution. Often achieving a solution requires conducting iterative refinements based on testing and evaluation. In turn, the testing and evaluation techniques can be for example empirical-based or formal-based.

Process-based: This model is concerned with the study and understanding of activities/processes that involve humans to accomplish tasks in computer science (e.g., studying the ways that humans build and use computer systems). The research in this area aims at discovering proven designs, repeated patterns, or recurring strategies, which can be codified in various procedural ways such as best-practice guidelines, pattern languages, application frameworks, process models, and development tools.

3. VISION ON RESEARCH

3.1 Motivations

The advantages of mastering research skills for software engineering graduates at the bachelor level include: Not reinventing the wheel (this is achieved through investigating the sate of the art works before and during devising any solution intended to solve a practical problem), keeping pace and coping with the fast technological advancements that we witness in the ICT field nowadays (the skills learnt once-upon-a-time, i.e., those learnt through education or experience, may become obsolete in a time window of a few years, costing someone's job and an enterprise's business if no new expertise/skill is acquired), learning about and adapting to the real demands and needs of customers and clients (this requires having a wider view than just focusing on technological aspects, i.e., being multi-disciplinary, as ICT integrates with the fabrics of other disciplines more than ever), continuous learning and improving own and organization's expertise (through learning from other's knowledge); and making innovations in fast cycles (through effectively sharing knowledge with peers or colleagues and learning from others).

3.2 Scope and Boundaries

Engineering is defined as "the branch of science and technology concerned with the design, building, and use of engines, machines, and structures" [13]. A software engineer designs, implements, deploys, and administrates a software-based system and enhances the system to a desired solution to address a real problem. Thus engineering coincides with applicability.

Within our study we should first clarify the relation between research and engineering. Due to the relation between engineering and science (see the definition above), there is clearly a relation between engineering and research in its traditional definition (see Section 2.1). Both engineering and research are concerned with addressing problems and hereto they usually carry out iterative refinements of the solution and research. There are, however, also some differences between engineering and research. Engineering may have no or minimum relation to research when for example an engineer works as a pure practitioner doing routine assignments. A precondition for an engineering task to be considered as research is the engineered artifact to be new or to have some new features. Then the resulting artifact can become a means of gaining new knowledge/insight within a given domain or body of knowledge. Research, on the other hand, may have no (or minimum) applicability. For example, fundamental research, usually carried out by scientific universities, is not assessed by or concerned with applicability of the expected outcomes.

To characterize research, Stokes [17] made a distinction between *generalizability* and *applicability*. The general perception of research until the late 20th century was that research is either applied, i.e., has an application but delivers no new insight, or fundamental, i.e., delivers a new insight but has no application [12]. This view was considered too simplistic according to Stokes [17] who believes fundamental research is highly generalized but applied research can be both generic and specific, as illustrated in the front vertical plane (consisting of the green boxes) in Figure 1. The plane implies that applied research can generate also general insight, as shown by the example of Pasteur as a pioneer at the upper-right quadrant. The lower-right quadrant, exemplified by Edison as a pioneer in such approaches, represents research-based practices that strive for excellence and innovation through doing research to solve a specific/real problem. The box with Bohr as an example pioneer represents fundamental research in the quadrant.

We extend here the abovementioned quadrant from [17] with another dimension of 'research' to define a space consisting of three dimensions *research*, *generalizability* and *applicability* for characterizing research and engineering relations (see Figure 1). Let's consider the case where there is no or minimum research involved, as illustrated in the back vertical plane (consisting of the gray boxes) in Figure 1. When a student graduates from an applied university, he/she is able to carry out some routine tasks because he/she has learnt about the required skills during his/her education (e.g., doing straightforward assignments that require programming skills of Java). When such a practitioner year-after-year does similar assignments he/she obtains some experience-based tacit knowledge about his profession (e.g., Java programming). Gaining such a generalization skill occurs gradually with a rather slow pace and has been a norm in vocational endeavors throughout the years (e.g., leaning family business from parents and learning jobs skills from masters). The initial and gained knowledge is illustrated by a gray box and dashed gray box, respectively, in Figure 1.

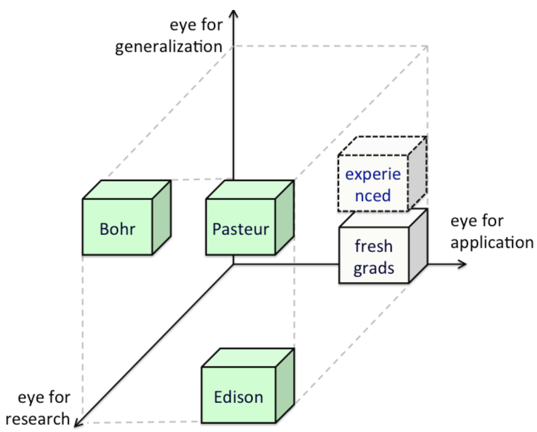

Figure 1: A three-dimensional model, characterizing research and engineering/practice.

To identify the scope of the research within applied universities we first recall Archer's discussion about the research through action for creative practitioners (e.g., artists and designers) [2]. These practitioners claim that what they ordinarily do is research because their outputs, like artworks or design products, constitutes new knowledge. We believe that engineers may belong to this set of practitioners because engineers develop artifacts that, in addition to solving real-problems, may deliver some (new) knowledge (see the implementation-based research-model mentioned in Section 2). Archer [2] distinguishes between 3 cases in regard to research and practice, namely: Research for the purposes of practice, research through practice, and research about practice. In research for the purposes of practice an investigation is conducted through a systematic enquiry to contribute to a practitioner activity like making a software artifact. When the objective hereto is to create communicable new knowledge, then the investigation can be called research. However, in situation where it is sufficient just to show that the outcome is satisfactory, without being concerned about how well the research was done or without evaluating the results obtained the investigation is considered as either "option research" at best or no research at worst (merely being speculation or exploration) by professional scholars and researchers. In research through practice a systematic enquiry is conducted through the medium of the artifact/product (e.g. a software prototype) or the practical action. The product or action is meant for devising or testing new ideas, information, forms or procedures and generating communicable knowledge. In some circumstances constructing something (like in engineering) or enacting a calculated action (like in action research) is the best or the only way to clarify a proposition/principle or to prescribe a material/process. These situations arise in, for example, engineering, medicine, agriculture, education and business. In research about practice a research is carried out about, for example, the history of the practice, the analysis and criticism of the output of the practice, the relation of the practice to people and society, etc. Archer [2] argues that such studies about practice can be considered as research studies if they employ the methods and principles of the-class to which they belong.

Inspired by Archer's categorization [2], we distinguish the following research levels in relation to the practical endeavors (e.g., making a software prototype) that are devised by the students of applied universities: (1) Practical endeavors to come up with a satisfactory result, without using a systematic research enquiry. We call these as No Research (NR). (2) Practical endeavors to come up with a satisfactory result, using a semi-systematic research enquiry (i.e., with conducting some aspects of research like literature study to choose options or evaluating the results obtained). This can be with or without generalization. We call these practices as Weak-sense Research (WR). (3) Practical endeavors through using a systematic research enquiry as characterized in Section 2. This can be with or without generalization. We call these endeavors as Strict-sense Research (SR). Furthermore, we emphasize that the practical endeavor carried out by a practitioner may lead to a new solution or feature, resulting in new insight and knowledge within a body of knowledge. In other words we have two types of outcomes: (a) A solution, solution feature, insight or knowledge that is already known in the body of knowledge, but it is new for the context of the use (e.g., for a company's specific need). (b) A solution, solution feature, insight and knowledge that is new for the body of knowledge in accordance with the definition of research given in Section 2. When a systematic research is applied to achieve outcomes (a) and (b), we can characterize the corresponding endeavors as WR and SR following the research levels defined.

Considering the ambition of applied universities to educate research skills to their undergraduate studies, one can recognize the research level 2 (i.e., WR) as the research suitable for students of software engineering in applied universities. Note that we regard this research as WR solely because the outcome of this research is of type-a (i.e., in being new for the context of use/application). The other aspects of a systematic research should be in place. Scientific universities in The Netherlands, however, aim at type-b outcomes and carry out SR. As such, both scientific universities and applied universities use the same research methodologies, however, their products differ: the former focuses on creating new knowledge while the latter focuses on creating working solutions for real and practical problems. Note that the term "weak" above describes the 'research aspect' and does not imply anything about the 'quality / the added value of the solution delivered (i.e., it does not describe the practicality of the work).

In light of the insight mentioned above, we introduced an intermediary vertical plane in Figure 2 (consisting of the yellow boxes) to represent Weak-sense Research. The intermediary vertical plane represents the research level sought by Dutch universities of applied sciences and the boxes thereon are marked by "WR 2:a", indicating that it Weak-sense Research only because its outcome is of type-a, i.e., delivering new solutions for a given usage context. Note that the right vertical plane in Figure 2 (consisting of boxes marked by 1, 2:a, and 3:b) represents the scope of engineering within the 3D-space defined.

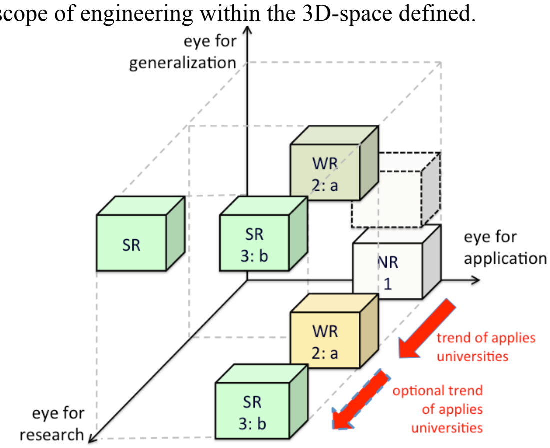

Figure 2: The 3D-model and the trend for research in Dutch universities of applied sciences at the bachelor level.

3.3 Desired Level

Concerning research skills, one needs to fulfill the requirements of the Bachelor level, without being too ambitious to aim at the higher level of Master. There is still a lot of debate going on about what the desired level of research skills for bachelor graduates should be, compared to the level of the research skills required for the graduates of scientific universities (i.e., for the holders of Master of Science degrees).

The European Union has issued a qualification framework, the so-called European Qualification Framework (EQF) [6], whereby all levels of education within Europe can be compared. For the Bachelor level, the EQF requires obtaining "advanced skills, demonstrating mastery and innovation, required *to solve complex and unpredictable problems in a specialized field of work or study*" [6]. For the Master level, however, the EQF requires "specialized problem-solving skills required in research and/or innovation in order *to develop new knowledge and procedures and to integrate knowledge from different fields"* [6]. As seen from these EQF requirements, problems solving skills are relevant for both Master and Bachelor graduates. To this end, therefore, mastering the research skills finds its relevancy for both Master and Bachelor level graduates because research methodologies are devised and used for problem solving for many centuries.

The difference between the problem solving skills (thus research skills) required by EQF for Master and Bachelor levels clearly pertains to the type of the product/output that these graduates deliver. According to the EQF requirements, the product of the graduates with a Bachelor degree is a solution in "a specialized field of work" [6] and that of the graduates with a Master degree is "to develop new knowledge" [6]), while both aiming at real problems and applying (new) solutions in the problem domain. Therefore, our vision for educating research skills in applied universities, described in the previous sections, agrees with the EQF requirements.

4. GRADUATION REPORTS REVIEW

This section presents the result of our analysis of the graduation reports of the students that were graduated in 2012 and 2013 within the INF, MT, and TI disciplines.

4.1 Review Protocol

In Section 2.3 we identified 6 research models that are used in computer science from literature. These research models, in turn, encompass the specific methods (e.g., literature study, case study, user study, etc.). As research in universities of applied sciences and the research in scientific universities only differ in the outcome, we concluded that the identified set of research models is applicable and relevant for the students of universities of applied sciences to define the research processes and methods.

As baseline we would like to know which of the research models are (more) applicable to the typical assignments of INF, MT, and TI software engineering disciplines. For these typical assignments we have chosen the INF, MT, and TI graduation projects carried out in recent years. The motivations for choosing these graduation projects are: They are defined and executed based on the curriculum profiles and guidelines of these disciplines [16] and, as such, they represent the corresponding education profiles, they are the most comprehensive practical works that these students carry out during their education, and they should (ideally) be executed according to the most relevant and appropriate research methods as required by the guidelines of RUAS board.

For the baseline study we classified these graduation assignments to the identified research models. Our objective is to see which research model is most relevant for each of these disciplines. Knowing the relevant class(es) enables us primarily to shed light on the research methodologies that can be used in each of these disciplines. Moreover, we can hereby identify the existing shortcomings in exercising the corresponding research model(s). Based on literature study and conferring with peers, we used the following research models for our classification of the graduation reports: formal-based (M1), model-based (M2), empirical-based (M3), observational-based (M4), implementation-based (M5), implementation-based for influencing experience and/or behavior (M6), process-based (M7) and any other model than those mentioned above (M8). In order to have a more representative set of classes we defined research model M6 as a variant of model M5 to cover the current scope and direction of the MT discipline. Class M8 was defined to control whether this taxonomy covers the space of all assignments.

We used the following protocol to classify the reports: Per discipline we chose 20 graduation reports for review and classification from the last two years (i.e., 2012 and 2013). Thus we studied 60 graduation reports in total. Two members of the workgroup reviewed every paper. Per report we (1) read the introduction to evaluate the problem statement and the research questions, (2) browsed through the report to identify the closest research method, asses the quality of the analyses of the problem context and existing solutions (and if relevant, assessed the validity/grounding of the design choices made and the quality of the evaluation of the results obtained), (3) read the conclusions to evaluate the quality of the reflection on the results achieved and of the answers to the research questions.

4.2 Results and Discussion

The result of our classification of the graduate reports for the three disciplines is presented in Table 1. We see that the implementation-based research-model (sum of classes M5 and M6) is the closest one for the assignments of TI (90%), INF (76%) and MT (77%). These figures imply that the majority of these graduation assignments should closely follow the research method(s) used in the implementation-based research-model in order to enhance their research-related qualities. For enhancing the research quality of the majority of software engineering graduation project we need to identify the existing shortcomings of these assignments with respect to this research model.

Table 1: Result of our classification of the graduation reports of the three technical disciplines.

%	M1	M2	M3	M4	M5	M6	M7	M8
MT	–	–	7	13	13	64	3	–
INF	–	–	14	–	58	18	10	–
TI	–	–	6	4	90	–	–	–

Our review of the old graduation reports and also the experience of our colleagues reveal that the way that our technical oriented students used to carry their graduation works adhered to a process model consisting of design and implementation stages. The input to the assignment was a number of requirements, often determined by the company coach who envisioned how the expected software (or software-hardware) system (component) to be developed. The output of this process model was often a prototype with no or minor evaluation of its performance and functionality. This traditional process model is depicted in Figure 3. In this figure the height of each block conveys the amount of work put in the corresponding stage of the process.

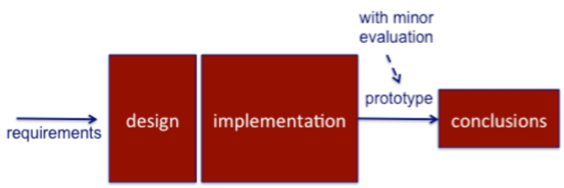

Figure 3: Illustration of the old work process for graduation assignments.

Based on our analysis of (the shortcomings of) the existing graduation reports, we envision the research model of Figure 4 to be appropriate to capture and encompass the implementation based research model that is applicable for majority of the (graduation) assignments of the software engineering disciplines. The model represents the typical development lifecycles of an information system, i.e., analysis, advise, design, implement, and manage, according to [16]. As such, the model resembles the implementation-based research-model directly. One can also show that the model is compatible with the advise-based and service-management-based types of assignments [16]. Showing this compatibility, however, is out of our scope in this paper.

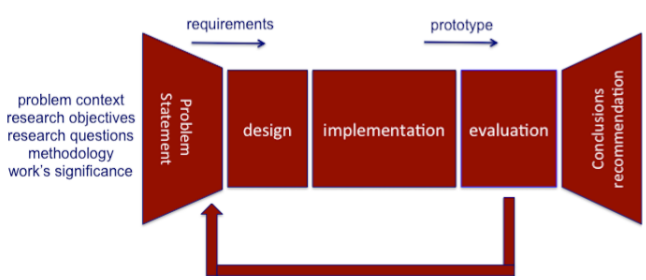

Figure 4: Illustration of the envisioned work process for graduation assignments.

Comparing the old and envisioned models in Figure 3 and Figure 4 reveals the shortcomings (i.e., improvement aspects) of the way that previous assignments were used to be carried out. This comparison emphasizes the importance of project initiation stage, evaluation stage and conclusion/recommendation stage. The student should start with an initial analysis and investigation of the problem context and understanding the motivations behind the assignment (analysis of the current situation). The student should also investigate the existence of a real problem and the importance/relevancy of the sought solution in its bigger context (in overall system architecture and/or society) in the early stage. This investigation enables the student to define the corresponding research objectives and research questions properly. The design and implementation stages should aim at achieving the objectives (most often, realizing an artifact properly) by examining possible options and making decisions based on some (predefined) criteria.

The evaluation of the achieved results (e.g., a software prototype) is an integral part of the process, which is concerned with validation, verification and test of the artifact realized. The evaluation can be done often in quantitative ways or sometimes in qualitative ways. More generally stated, one can deploy any subset of the formal-based, model-based, empirical based, and observational based research models for the evaluation stage. This evaluation can also be considered as an analysis stage after the prototyping and it usually follows/completes the implementation-based approach. The result of the evaluation may reveal the need for revisiting the assumptions, re-design or reimplementation of the prototype. This is shown with a feedback line in Figure 4, emphasizing the iterative aspect of the whole process. Multiple iterations are common particularly in those designs where human factors are dominant and determinant. Finally, the conclusion process involves a reflection on the obtained results in illustrating the degree to which the research questions are addressed during the work and which lessons are learnt. This concerns also production of some (new) knowledge gained through the whole exercise, possibly with an eye on generalization of the results. The conclusion phase provides also an insight for future directions system-wise, business-wise, etc.

In summary, concerning the analyzed graduation reports, we identify the following improvement aspects: The evaluation (test, verification and/or validation) is often weakly done, the problem context is not well studied, the problem statement is not adjusted to the work carried out (or the question that was solved), the problems and the solutions are often not positioned in a larger context such as society, business and/or overall system architecture, the literature study is often done at a micro level (e.g., what the design options are per component), not at a meta-level (i.e., positioning own work as a whole with respect to similar works to learn from those), and embedding multi-disciplinary aspects in the graduation work is limited (the last one, we suspect, seems inevitable considering the limited timeframe available for executing such projects).

One of the objectives of applied universities is to teach student multi-disciplinary research and communication skills. In the model depicted in Figure 4 we showed this aspect schematically in the form of a funnel in the problem statement and conclusion components. Here the student needs to position his/her work in a larger and wider context (e.g., through investigating the problem context and providing advice across disciplines). We admit that this is a limited exercise for developing multidisciplinary skills, imposed due to limited duration of graduation projects. We believe that there should be some other courses, where students get enough opportunity to carry out such multidisciplinary works in collaborative teams. Communication skills, related to effective communication with professionals and nonprofessionals, become relevant when writing graduation reports and presenting the results.

The extent of the implementation stage (e.g., amount and complexity of the software code written) varies per discipline. It may be the main focus of the work (e.g., in those software engineering assignments that aim at creating new ICT artifacts) or may be of minor focus (e.g., in those assignments that aim at studying the existing ICT artifacts in their (new) settings). In some HCI projects, for example, implementing a mockup prototype might be enough to prove/demonstrate the idea behind. This variation of implementation-depth depends also on the research model chosen. For example, in observational model one can use (or reconfigure) an existing system and study its usability.

Classical research methods like literature study, user study, descriptive research, comparative research, explorative research, evaluative research, explanative research, desktop research, field study, lab research, quantitative research, qualitative research, … can be relevant and applicable in each stage of the model depicted in Figure 4. Depending on the discipline, these methods can be relevant to these stages in a varied extension. For example, user studies are particularly relevant in the begin stage to derive system requirements and in the evaluation stage to evaluate/validate the system functionality and/or performance. We will not elaborate on defining the research methods appropriate for a specific research.

5. CONCLUSIONS

Mastering research skills is crucial for the graduates of universities of applied sciences nowadays because these skills will enable the graduates to become knowledge-oriented professionals. In this contribution we sketched our vision on what research means within universities of applied science, particularly for software engineering disciplines, at the bachelor level. The main conclusion of this study is that while scientific universities and applied universities produce different outcomes – the former focus on creating new knowledge while the latter focus on creating working solutions for problems – both use the same research methodologies. Based on our analysis of the existing graduation reports we identified the implementation-based research to be the most appropriate research model for majority of assignments and projects in software engineering disciplines. Based on and compared to our envisioned research model, we identified the shortcomings of the current practices, specially those in the project initiation phase (to investigate and define the context of the problem), evaluation phase (to evaluate and reflect on the results achieved) and conclusion/recommendation phase (to capture the main outcomes and the lessons learnt, and to define future directions).

Research skills and competences improve with practice. For students to get better at research, we see the necessity of embedding instances of such practices within the curricula next to classic and theoretical courses. These instances can be accommodated within explicit courses on research skills as well as those courses that touch upon research skills implicitly. We are currently busy with adapting the curricula of our software engineering disciplines to properly accommodate these software skills explicitly and implicitly. It seems also necessary to establish some connections among these explicit and implicit courses in order to motivate students and make them aware of the fact that research isn't something far from their reality and that it is indeed something related to and part of their study and future work.

6. REFERENCES

[1] Amaral, J.N., et al.. About computing science research methodology. Available: http://webdocs.cs.ualberta.ca/~c603/readings/research-methods.pdf.

[2] Archer, B. 1995. The nature of research. *Co-Design Journal*, 2, 11.

[3] Baarda, D.B., and Goede, M.P.M. de. 2006. *Basisboek Methoden en Technieken* (in Dutch), 4th edition, Groningen: Noordhoff Uitgevers.

[4] Delnooz, P.V.A. 2010. *Creatieve Actie Methodologie* (in Dutch). Boom Lemma Uitgevers.

[5] Ellis, T.J. and Levy, Y. 2008. Framework of problem based research: a guide for novice researchers on the development of a research-worthy problem. *Information Science: the International Journal of an Emerging Trans-discipline*, 11.

[6] European Qualifications. 2008. *The European Qualifications Framework for Lifelong Learning leaflet*. Available: http://ec.europa.eu/eqf/home_en.htm.

[7] Glass, R. L., Ramesh, V., Vessey, I. (2004). An analysis of research in computing disciplines. *Communications of the ACM*, 47(6), 89-94.

[8] Hart, C. 1998. *Doing a Literature Review: Releasing the Social Science Research Imagination*. Sage Publications, UK

[9] Johnson, C. 2014. What is research in computing science? http://www.dcs.gla.ac.uk/~johnson/teaching/research_skills/research.html.

[10] Oates, B.J. 2012. *Researching Information Systems and Computing*. SAGE publications, London, UK

[11] O&K document. 2013. De HR-visie op eindnieveau: onderzoek op zijn Rotterdams (in Dutch). Ver. 1.0, February.

[12] Offermann, P., et al. 2009. Outline of a Design Science Research Process. *In Proc. of the 4th International Conference on Design Science Research in Information Systems and Technology*, ACM, 7.

[13] Oxford, 2013. *Oxford Dictionaries*. Available http://www.oxforddictionaries.com/definition/english/engineering?q=engineering

[14] Ramesh, V., Glass, R. L. and Vessey, I. 2004. Research in computer science: an empirical study. *Journal of systems and software*, 70, 1, 165-176.

[15] Saunders, M., Lewis, P. and Thornhill, A. 2008. *Methoden en Technieken van Onderzoek* (in Dutch). The 4th edition, Amsterdam: Pearson Education Benelux.

[16] Schagen, J.D, van der Kwaak, W., Leenstra, E., Smit, W. and Vonken, F. 2010. *HBO-I Bachelor of ICT Domain Description*. HBO-I foundation, Amsterdam, the NL.

[17] Stokes, D.E. 1997. *Pasteurs Quadrant: Basic Science and Technological Innovation*. Brookings Institution Press

[18] Vereniging Hogescholen (Applied Universities Union). *Over Hogescholen (About Applied Universities)*. Available: http://www.vereniginghogescholen.nl/hogescholen/over-hogescholen