Industrial impact on courses in software engineering

An empirical study on the contents of software engineering courses in 1990-2016.

Hannu-Matti Järvinen
professor
Tampere University of Technology
Tampere, Finland
E-mail: hannu-matti.jarvinen@tut.fi

Conference Key Areas: Curriculum development, university-business cooperation

Keywords: Curriculum, thesis, courses

INTRODUCTION

Development of the curriculum is based on academic knowledge and research, but there are often topics where the industrial impact is clearly present. Developers of curricula should keep the contents up-to-date, both in the academic and professional points of view; especially in software engineering, where much of the experiences come from the companies. Institutions like ACM are making regular model curricula to help in the development process, but they are often a little bit delayed when compared with actively developed curricula.

In this paper, we have three points of view. The first one, based on material collected for paper [1], discusses the topics of master's theses, or how the topics have changed during the time. The second one studies the changes made for the offered courses, and we like to see if changes in the master's topics and courses have any connection. The third viewpoint is a short note how the ACM model curriculum of software engineering has changed at the same time.

1 MASTER'S THESES

The study of the theses is based on 540 master's theses made at the Tampere University of Technology (later TUT) on software engineering between 1990-2015. These were evaluated and examined by two professors, covering almost half of the total theses on software engineering of that time. We collected data from the topic,

company, type of the theses, language, orientation (research or constructive), and the gender of the author. Results have been published in [1].

In this study, we concentrate on the topics of theses. The topics were classified into six classes: embedded and operating systems, web, mobile, traditional computer science, communications and protocols (high level, “above sockets”), and information systems. In Figure 1, their share is given in per cent to make the comparison of years easier. After 2000, the number of students varies between 12 and 67 for years, the lowest numbers being 2002; for 2016 only four months is included (7 students). Before 2000, the yearly number of students is very small, from 1 to 6. The overall trend is increasing with a clear exception in the years 2010 and 2011. The last possibility to graduate with pre-Bologna curriculum was in 2010. Hence, the total of 67 students graduated. On the other hand, in the next year, only 24 students graduated.

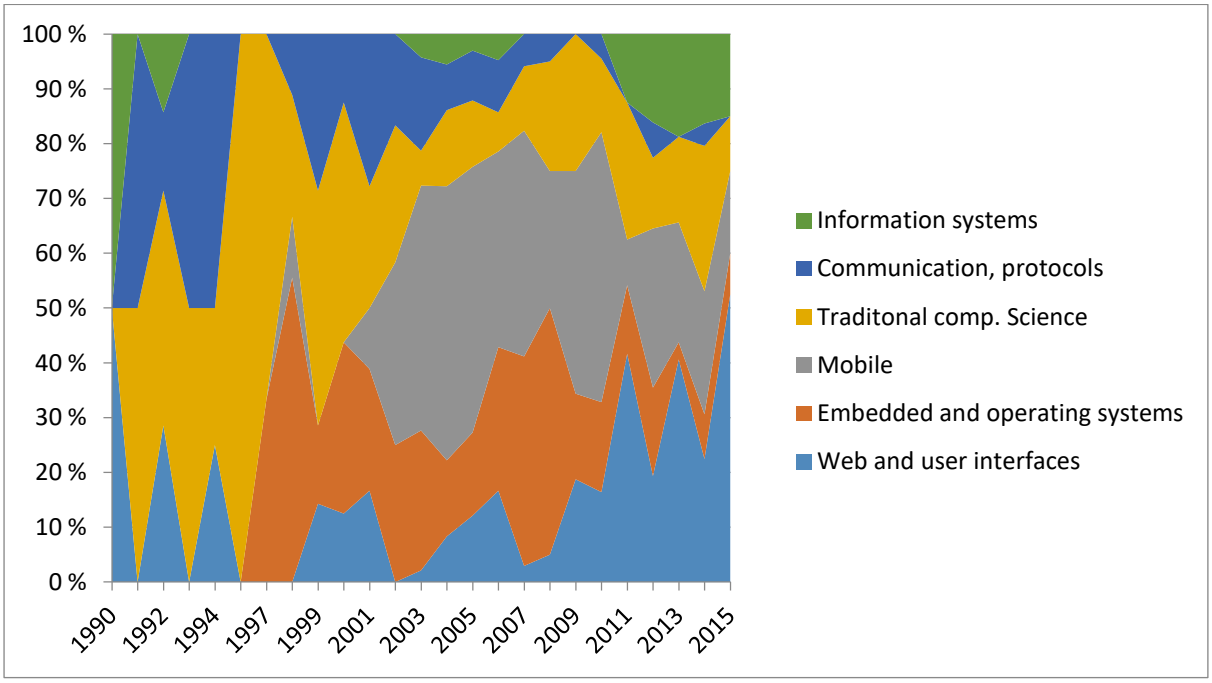

Figure 1. Topics of master’s theses

There are some clear trends in Figure 1, even if yearly changes are big due to a small number of students per year, this affects especially years 1990–1999. Years 2008 and 2009 make the difference. Economic depression took place in 2008 and Nokia selected Windows for their phones instead of Symbian. Web and information systems became more and more popular. The number of theses in embedded and operating systems, and communications and protocols were decreasing, and computer science kept its share. Note that as so many theses were made on industrial topics, their topics were often constructive [1]. Hence, topics that are close to pure software engineering, are classified in "information systems".

2 STUDY OF COURSES

This study covers the courses on software engineering and related topics at TUT from 1990 to 2017 but concentrates on the years 2000–2017. After 2000, there have been two main curriculum reforms of the whole university; the first one in 2005 when the university adopted curricula that are compliant with Bologna agreement and B.Sc. level was introduced; prior to this, all students graduated directly as M.Sc. The second reform took place in 2013 when the faculties of the university were reorganised, B.Sc. studies were decided to be in Finnish, and M.Sc. level courses more and more in English.

Neither of these changes in curricula originated from faculty's needs to change the contents of the studies, but they forced us to redesign the curriculum and decide which are the courses needed in the new situation. The first change was a government decision, and the second one was made mostly because of financial reasons.

Figure 2 shows the relative share of courses classified by their topics. The topics used are the same as in Section 2 but there are no courses on information systems and software engineering is used as one topic.

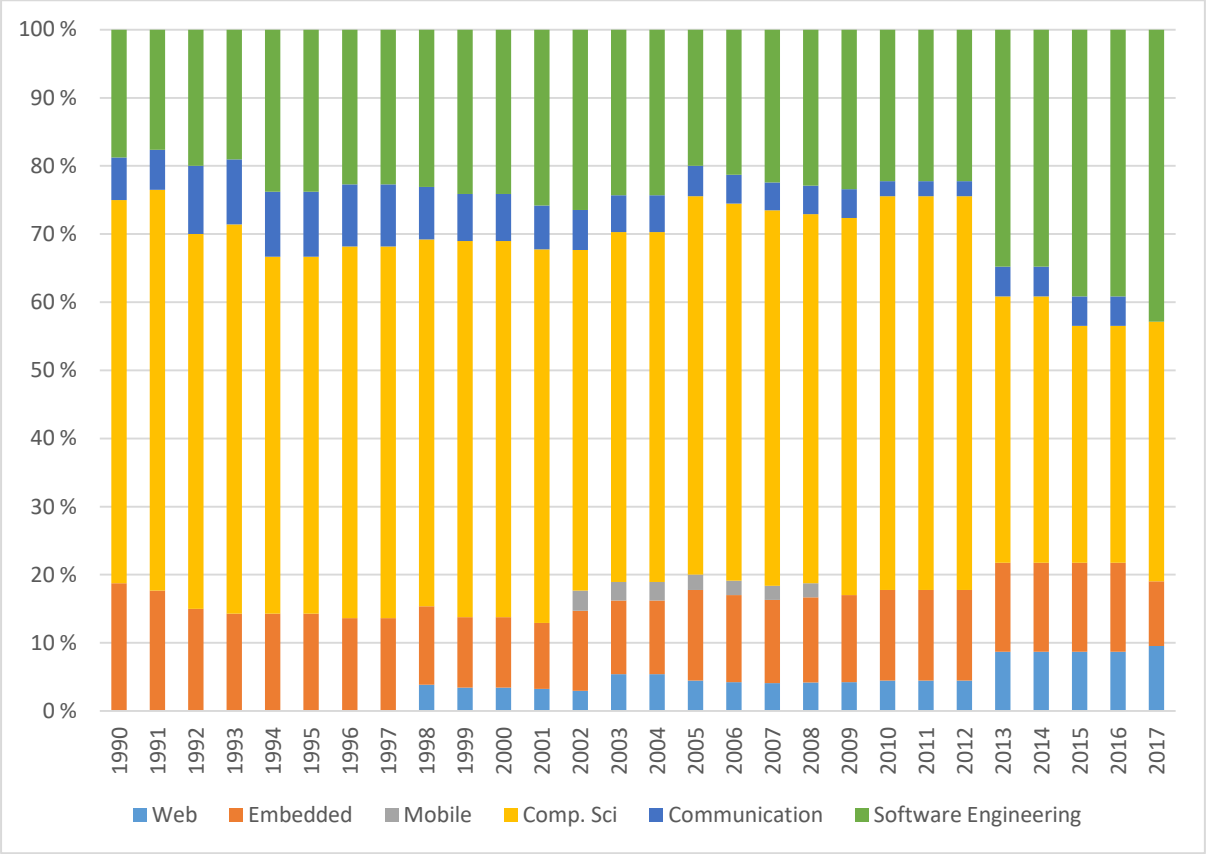

Figure 2. Topics of courses in six classes. Collected from [4].

In the viewpoint of IT, the changes can be divided into three phases: 1990-2001 was a growing phase (the growth is not shown in Figure 1, since it gives the share of each topic), 2002-2012 was quite stable (although it contains changes needed

because of Bologna agreement), and 2013-2017 was the shrinking phase. Before going to these phases, a quick look at courses that cover all these phases is taken.

2.1 Courses with a long lifespan

There are some courses that were in the curriculum in 1990 and are still there — or at least a corresponding course exists. The names of the courses may have changed, but their roles in the curriculum have been the same. These courses include mostly traditional computer science courses like "basic programming", "data structures and algorithms", and "the principles of programming languages", but there are also courses related to other topics like "operating systems", "introduction to software engineering", "software engineering methodologies", and "project work". Naturally, their contents have been updated several times over the years.

Of these courses, the basic programming course has been the most problematic one. In 1990, we assumed that new students have some idea of programming in advance. Since this was not the case in the real world, the course got quite early a companion; the courses were named programming 1 and 2. In 1997, IT students were given their own version of the courses, and in 2005 appeared their English equivalents. In Figure 2, courses that were offered both in English and Finnish are treated as one course.

The course of operating systems has had one major revision when concurrency was made a course of its own in 2005. The main reason was that multicore systems were becoming more and more popular, making concurrency issues important for the majority of the students; on the other hand, the need for deep understanding the operating systems was decreasing as a general skill.

Software engineering courses and project work have been the backbone of the curriculum, but there have been changes in their contents.

2.2 The first phase 1990-2001: Growing

In the years 1990–2002, the software industry in Finland was growing fast, Nokia being the leader. IT industry wanted to have more professionals and this message was heard. Intake of the curriculum of information technology at TUT was increasing rapidly being around 220 at the best years. The number of students increased and so increased also the number of courses. In 1990, there were 21 courses in software related topics, in 2002 there were 43 courses. Since some of the courses were also abandoned, this means that on average there were more than two new courses each year. In Figure 2, the relative share of different topics did not change meaning that all areas got new courses. The theses' topics in Figure 1 do not give any clear trend, mostly because of their small number.

Early in this phase, all kinds of skills were needed in companies, the goal was to educate as many masters as possible with quite versatile skills, but there was a growing trend of the companies’ interest turning towards software engineering. This can be seen in the introduction of some new courses: "leading of a software project" (1992, intended for graduated people in companies), "testing" (1998) and "software

architectures" (1999). However, this was not seen in the topics of theses but several years later, software engineering was a skill that was needed, not yet clearly an academic topic.

Introduction of web-based systems can be clearly seen during this period. New courses on computer graphics, graphical user interfaces, web programming and usability were introduced. Some of them were short-lived before a proper set was formed. Also "databases" (1992) and "designing databases" (2001) were created. These two were not new courses in a sense that they were freely available for our students at the University of Tampere, but the increasing number of students and the increasing importance of databases also in technical applications caused their creation.

An interesting detail is that course "artificial intelligence" existed in 1990 but was cancelled in 1994. It was recreated 1999 with different contents, abandoned 2013 only to be recreated again shortly after that.

Some small courses were introduced for practical purposes: e.g. C language, Unix, and Fortran, but they were mostly short-lived and never essential part of the curriculum.

2.3 The second phase 2002-2012: Relatively stable but becoming International

The next phase from 2002-2012 starts with years where so-called "IT bubble" was broken. Interest in IT studies went down, but this is not seen in the courses. Instead, another phenomenon can be seen: more and more courses were lectured in English. This applies especially in the advanced courses on computer science, where lectures in Finnish almost disappear; in other areas, many courses were offered in two languages, Finnish and English. Still, the majority of the courses were offered in Finnish. One reason for computer science switching for English was the fact that at the beginning of this phase, the share of computer science theses decreased (as seen in Figure 1) and the number of students in courses was small. Hence, we did not afford to have them lectured in two languages.

In this phase, creating new courses was not so frequent as before; on average one per year; most of the new courses were created in 2005 when the Bologna system was adopted. In Figure 2, a small portion of courses is in grey to indicate mobile systems. Their topics were covered by several courses, but courses whose names indicate mobile systems were all Symbian-related. Hence, they disappear rapidly when Nokia published in 2008 that they will not develop their own operating systems (Symbian) any more.

2.4 The third phase 2013-2017: Shrinking

In 2013, there was a reorganisation of the faculties in the university, mostly motivated by financial problems. There were strict rules about curriculum development, which resulted in fewer courses. This was done by merging courses to bigger courses and dropping courses with a small importance and a small number of attendees out of the curriculum. This meant that parallel implementations for IT

students and other engineering students were cancelled. Computer literacy was also cancelled since students were expected to know more about computers in advance than before. From 47 courses in 2012, only 28 were left in 2013 (not counting parallel implementations in Finnish and English). During this period, only one new course has been introduced ("data-intensive programming", 2015) and two have been abandoned but of their contents were covered by other courses whose credit points were increased.

Most of the courses that were combined or abandoned were programming courses or traditional computer science courses. Since some of them had only a handful of students, they were removed from the curricula without having any corresponding course. Only two courses related to software engineering were removed from the curriculum: "maintaining software" and "the seminar of project management". This can be clearly seen in Figure 2, where the relative share of traditional computer science is dropping, but information systems have increased its share significantly. The share of web-related courses was doubled, too. This change was actually not as dramatic as how it appears, because the majority of students selected software engineering courses anyway.

2.5 Conclusion of courses

Until 2012, the number of courses increased, but the relative number of courses of different topic areas did not change. In spite of its popularity in theses, there were never many courses dedicated to mobile systems, but the topic was covered by other courses related to embedded and distributed systems. Basic knowledge of computer science courses is needed by all software engineers, explaining the number of courses on computer science compared with the number of theses.

Although software engineering courses were offered from the very beginning, theses that discussed improving the process were rare. Their amount has been increasing over the past few years. Switching for agile methods has been one of the reasons for this since they require new way leading projects and new tools for continuous integration, delivery, or automated testing to mention some. Anyway, this prevented the decrease in courses in software engineering, which can be seen in Figure 2 as an increase in their share.

3 ACM MODEL CURRICULA

There are two ACM model curricula available for undergraduate level software engineering during this period, the curriculum of 2005 [2] and later one in 2014 [3]. These reports do not represent their curricula in the same way, hence a detailed comparison is not easy. Since the above study concentrates actually on the master level of studies, the applicability of the ACM model is limited but it gives some hints to the trends.

If the “look and feel” of the reports is studied, a clear difference in describing software engineering is found. The tone of the 2005 version is to convince that software engineering is an important subject, but that tone is not found in the 2014

version. Clearly, the subject of software engineering had become mature by then. As expected, the curriculum still includes a lot of basic skills inherited from computer science, but there is a bigger emphasis on software processes, quality, large-scale programming related topics (e.g., architecture) and tools needed. Of course, most of these existed also in the 2005 version in some way, but there a clear difference in their relative importance.

4 SUMMARY

Compared with the development of the curriculum at TUT and our topics of theses, the same phenomenon as in ACM software engineering curriculum development can be seen: software engineering is clearly matured as a discipline. With maturity comes also the increasing need for professionals in the area, as our topics of theses indicate.

We were forced to make big changes in our curriculum in 2013, and it can be seen that they follow the spirit of the model curriculum of 2014. In the details, there are differences; we might have decreased the share of traditional computer science courses more than the model curriculum, but their absolute amount is above the minimum described by the model curriculum.

The conclusion is that the model curricula are excellent ways to check the direction of development. However, if the development is based on them, the curriculum is a little bit late. An actively updated curriculum should naturally lead to the same direction as the model one. At TUT, a close connection to the industry is formed with master's theses made of companies' topics. This connection helps to direct the courses and the content of the curriculum to correspond to the future needs of professionals.

REFERENCES

[1] Järvinen, H.-M. & Mikkonen, T. Sep. 2016 Industrial impact on topics and types of Master's theses: Empirical study of software engineering theses made in 1990-2016. *Proceedings of SEFI 2016 Annual Conference.* Järvinen, H-M. & Clark, R. (ed.). European Society for Engineering Education SEFI.

[2] Computing Curricula 2005, the overview report. A Volume of Computing Curricula Series. The Association for Computing Machinery, the Association for Information Systems, and the Computer Society. 30 September 2005. Available at https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2005-march06final.pdf.

[3] Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of Computing Curricula Series.

February 23, 2015. IEEE Computing Society and Association for Computing Machinery.

[4] Annual study guides of the Tampere University of Technology, 1990–2017. Tampere University of Technology, Tampere, Finland.