Introducing Software Quality Maturity Models in Software Engineering Education and Small Organizations

George Sharkov[1] and Maya Stoeva[2]

[1] Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski"
24 Tzar Asen str., Plovdiv 4000, Bulgaria, and
European Software Institute – Center Eastern Europe, Sofia, Bulgaria

gesha@esicenter.bg

[2] Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski"
24 Tzar Asen str., Plovdiv 4000, Bulgaria

may_vast@yahoo.com

Abstract. We are currently living in an extraordinary situation. The circumstances, imposed by the ongoing pandemic situation, have led to a variety of unprecedented conditions with unpredictable outcomes. Most of us are working, teaching, and learning online, partially or completely. Therefore, we should be and we are more flexible than ever. What is the best strategy against the chaos, then? We figured out that in terms of software engineering education and small organizations we need at least two things. First, it is always important to search for the intersection between theory and practice, on a daily basis. Secondly, it is of utmost importance to have good described and defined inhouse work processes, no matter whether in consideration of education or of business companies. That is possible through a rigorous establishment of good practices and following the guidelines for models of software quality maturity.

In this contribution, we will introduce some of the more prominent models. These are CMMI (Capability Maturity Model Integration) [1], CMMI with SCRUM [2], or TMMI (Test Maturity Model integration) [3]. Along with this, we will overview a roadmap to transferring and applying this knowledge in practice within the context of small business organizations.

Keywords: Software Quality Maturity Models, CMMI, TMMI, Scrum, Software Engineer Competences, Software Quality, Industrial Standards, Process Improvements, Software Requirements, Project Management, Flexibility, Contemporary Training Methods, Small Organizations.

1 Introduction

The ongoing pandemic provides a plethora of challenges for planning, thus imposing an ever growing need for flexibility and adaptability. The more prepared you are in a given field, the easier the process of decision-making is against the backdrop of unprecedented and constantly emerging diversity of challenges. Thus, to be a good specialist means being able to find the intersection between theory and practice especially if you are working in the field of Information Technologies (IT). Therefore, it is crucial that university programs in software engineering and development can face the challenge of achieving a balance between the theory related to current and future industry demands, and the sufficient practices to develop the required skills, knowledge, and competences. For several decades already, process-oriented models have been used to achieve software and IT services industry maturity [2]. In this paper, we will present our approach and positive experience of introducing models for software quality maturity like CMMI (also combining CMMI with Scrum for a "Disciplined Agile", DevOps) and DevOps, and TMMI within the context of university education, as well as in small business settings.

Understanding the software development lifecycle, its related processes, and roles, specific aspects of teamwork, as well as the variety of organizational models with their related advantages and implementation burdens, are those expected competencies that differentiate developers and software engineers from what students themselves address as "coders". Those gaps and insufficiencies of the IT university education content are frequently underlined by the major IT stakeholders in Bulgaria and the region [6]. Similarly, a strong interest is demonstrated in the introduction of internationally recognized master programs and series of top-class IT courses with a particular focus on IT business, project, and team management. A certain number of courses already address these problematic areas, however, they lack a certain level of synergy and unification, and are often up to the initiative (and availability) of the lecturers. The MSc and BSc programs offered by the main Bulgarian universities are yet more heavily focused on rather technical than managerial profiles.

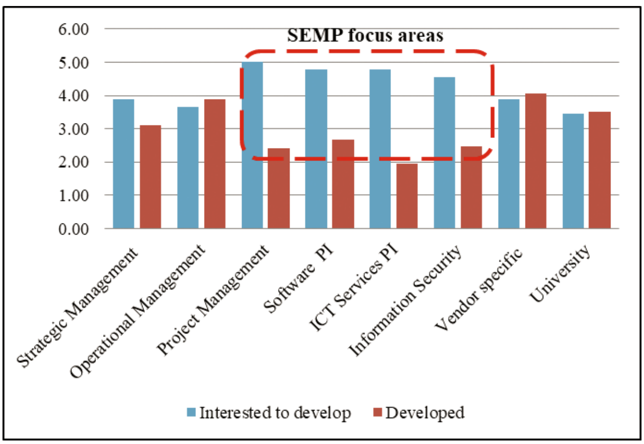

Fig. 1. Aggregated results from a survey with ICT/software industry on skills and competences expectations. *"PI"* stands for "Process Improvement".

Fig. 1 shows the aggregated results from a survey with ICT/software industry (8 countries in Eastern Europe until now) on skills and competences expectations (bipolar scaling method, measuring negative to positive attitudes to a statement*, with values from 0 to 6, where 6 is "high"* [20]). They are compared to the opinion on the status for the period 2009-2015 (*"PI"* stands for Process Improvement, and *"University"* is for academic/theoretical foundation*)*. SEMP is Software Engineering Management Program.

To address that gap and align Bulgarian educational practices against the landscape of the global trends, the SEMP ("Software Engineering Management Program") nationwide project was launched in 2010. It is coordinated by the European Software Institute – Center Eastern Europe (ESI CEE) and in collaboration with the Carnegie Mellon University (Institute for Software Research and Software Engineering Institute), involving 7 leading Bulgarian universities [4], with the University of Plovdiv joining in the 2015 [5]. Upon the initiation of the program, a survey conducted by ESI CEE under the Regional Competitiveness Initiative (RCI) project, supported by USAID, for Bulgaria and 7 countries in South-Eastern Europe showed a clear deficit in IT "management" areas (like Project Management, Software Process Improvement, ICT Services Process Improvement), as well as in Information Security (Fig. 1) [6].

More recently, along with the CMMI models tailored for software development, IT services, and acquisition, a new process-oriented model specialized for software quality testing, the Test Maturity Model integration (TMMI), was introduced by the TMMI Foundation [3]. It comes in a response to the need for system testing and achieving higher and sustainable levels of program quality for the developing software and the new ISO/IEC 25000 standards [8].

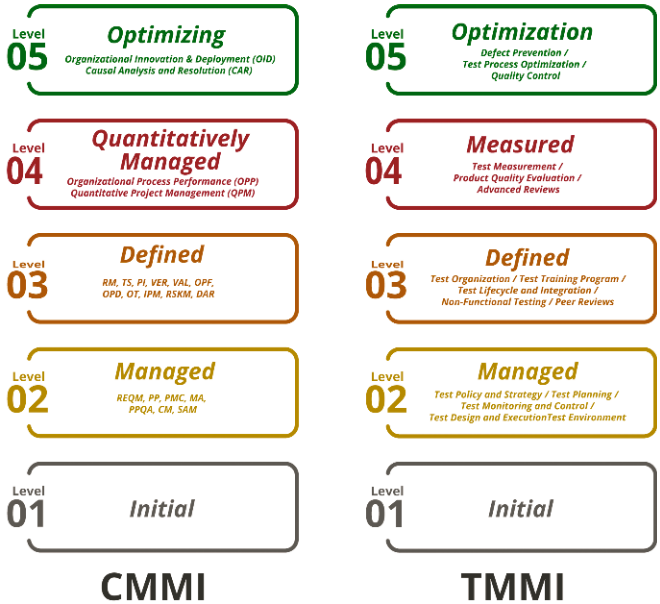

Fig. 2. Compression between CMMI and TMMI maturity levels names. They are almost equal, but as CMMI uses general and specific practices for all processes in one project and company, the TMMI is mainly oriented to software testing.

Software and hardware industries are evolving faster than ever. Only by knowing and choosing the right models for project management, project planning, process optimizations and correct structuring, software quality, and software testing and prevention we can survive in the daily business. Therefore, it is very important how we can provide the right and latest information like teachers to our current and future students. Only in that way they can see and understand how to cross the theory and the practice and become are good professionals in the business companies after their graduation.

2 Background

During the pilot phases of SEMP (the years 2010 – 2021), more than 22 academic courses were introduced or built upon already existing courses to align with the standards and methodologies obtained through the know-how transfer from the Carnegie Mellon University (Institute for Software Research – ISR, Software Engineering Institute – SEI, others). Seven courses were implemented during the first phase (2012-2013), while others were gradually implemented after 2014 (as elective or regular/core courses). Interuniversity and industry-recognized certificates from the SEMP program were issued to all students who completed the courses, supplementary to their scores and credits received from the university.

In total since the beginning of the project, more than 1500 students have completed SEMP courses. Several courses were also offered in an intensive (executive) format option and were attended by both students and industry professionals, thus giving additional benefit from mixed teaming.

The approach chosen by the Faculty of Mathematics and Informatics at the University of Plovdiv (FMI-PU), also a partner in SEMP was one of the gradual implementations. In addition to the continuous improvement of existing courses to bridge the theory-practice gap and meet the new trends, like [10], a new course was first introduced for several years as an introductory elective course under the title "Software Process Quality Management".

The course content was based on the materials and experience of similar courses under the SEMP program, delivered by ESI CEE, or implemented at the Carnegie Mellon University (CMU), Sofia University (FMI-SU), and other SEMP partners [11], [12], [13]. The feedback, however, showed that students with mainly technical expertise could not easily assimilate the abstract process definitions and link the process areas and practices (as in the CMMI model) with a real software project lifecycle and activities. We figure out, that they feel freer with CMMI with Scrum and in some situations with TMMI. The last two are more familiar to them with the increasingly common adoption of the agile development and software testing paradigm as of lately, especially for those who are not only studding but also working in the software industry.

Additionally, being an elective discipline, the course was unable to provide the targeted common understanding and unified project management "language" that links and relates to the other disciplines in the university program. Thus, the beneficial opportunity to unite practical tasks from other disciplines was not utilized at all and sometimes they were partly overlapping each other. They remain formal as "homework", exercising some individual skills but yet too far from the real industrial projects. Nevertheless, the course was considered a significant asset for students when seeking employment in the IT sector (see feedback results in the next section, Fig. 3) or just using the acquired knowledge in their current job or projects.

To better align the course with the development strategy of the Faculty and to address the above-mentioned shortcomings, a new "Software Quality Assurance" course was introduced as a core course of the 4th year of study in the BSc program of Informatics. Despite the usual association of "QA" with "testing", the course focuses on the quality of processes as a prerequisite for the quality of the product and the roadmap from process improvement to excellence. This course is also based on the CMMI for Development (CMMI-DEV) model, benefitting from its completeness and recognition by the industry as a de-facto standard, but further adapting the relatively "heavy" model to the knowledge and skills level of undergraduate students. The model was applied as a generic sample model, to which other modern methods and techniques such as Scrum, "Disciplined Agile" with CMMI, Six Sigma, Kanban are consequently addressed in the course. The course also introduces business aspects such as "cost of quality" along with the need to balance process improvement and product quality with a brief introduction of Kaplan and Norton's Balanced Scorecards [14], applied to the IT and software industry.

We also include in the current academic year (2020/2021) CMMI with Scrum and TMMI model, together with the software-testing tool to be aligned with the last development tendencies. The CMMI with Scrum just expands the CMMI model with the agile paradigm. We further included TMMI, because as it well complements the previous too. It can be used in the testing phase of the software. It contains the five maturity levels, known from CMMI (see Fig. 2). The initial phases of both models are the same – business processes are unpredictable, poorly controlled, and reactive. All companies start at level 1 of maturity. After integrating and applying the corresponding specific and general practices to each level, we go to the next upper one and levels cannot be skipped. Following the "generic" nature of the processes and practices in CMMI, in addition to Scrum and Agile, we illustrate to students how they map to another popular work and team organization, namely DevOps.

To face the deficit in "Information security" area training, especially in software engineering profile, we have also introduced the engineering aspects (process areas) from another capability and maturity model, developed by CERT at Software Engineering Institute (Carnegie Mellon University) – CERT RMM (Resilience Management Model) [19]. As expected, it naturally maps the cybersecurity and resilience aspects to software and IT systems requirements (Development and Management), as well as with Resilience Technical Solutions Engineering. This way, we add to the typical "requirements" (customer and product requirements) also the "security by design" and "resilience by design" principles, secure coding, and design principles, which are usually not explicit in neither the customer nor the product requirements. The process-oriented description, goals, and practices' structure, made it easier for the student to absorb and master these new areas. A separate new course "Cybersecurity and Business Resilience" was also included in the software engineering program, entirely based on CERT-RMM and specific technical information security standards.

For six years of delivery, the core undergraduate course "Software Quality Assurance" outlined and demonstrated to more than 550 students how process improvement factors into the quality of software programs. The goal, however, is not to study the CMMI or TMMI only. They complete each other and they are used as a reference framework to describe the main processes along the lifecycle of one typical software project and cultivate software development discipline by exercising in life-related realistic projects. In addition, help us to build good software products and successful business.

Within the next sections, we will outline the main milestones of combining university education and small business companies.

3 Motivation, approach, and goals

With the global shift towards remote work and education, we well realize that through targeted efforts, we can work, study and teach online, not worse than in an offline setting. Zoom, Google Meet, Microsoft Teams, Skype, and Slack became a part of our daily routine and way of life. Software development technologies, frameworks, and platforms have rapidly evolved since the onset of the pandemic, revealing a need for drastic revisions in the academic content to remain "contemporary" and "outstanding".

On the other hand, the logic behind most software projects' lifecycle is quite generic and the challenge is how to translate the abstract processes into practical implementation rules in an academic environment, which can be used in further real projects and small organizations. Undergraduate students in informatics are typically interested in acquiring a wider range of practical competences and obtaining good grades, so they can jump quickly into higher positions in the industry [6]. Naturally, they are more interested in attending lectures and courses that tackle more practical, rather than theoretical aspects. Most of them have certain practical experience working in companies, which, unfortunately, are not always good examples of best practices in project organization and management.

After collecting and analyzing our experiences from the elective course "Process Quality Management", similar courses of the SEMP program, and examples from the software industry (see Fig. 1), we have proposed for the core course „Software Quality Assurance (Q.A.)" the following approach and statements:

1. **Introduce students to the more prominent models for software quality maturity**, such as CMMI, CMMI with Scrum, TMMI, corresponding to ISO/IEC 25000 standards, to focus students on process improvement as a key factor for software product quality. This process provides all necessary definitions and vocabulary regarding processes and company maturity, institutionalization, goals and practices, software testing and quality software standards, as well as a generic approach and practices for organizational process definition and management.
2. **Overview selected process areas** related to software project management and further demonstrate the logic and interconnectivity of project phases with the respective process areas and specific practices, with good practical examples of their implementation, "what if not" discussions, and appropriate tools.

3. **Demonstrate real projects with all aspects of the software lifecycle** from low- and high-fidelity prototyping, extracting software requirements to the black/white and grey software testing.
4. **Exercise theory in a realistic collaborative setting**, by assigning the development of similar projects in teams, where students can assign and perform different project roles, with their respective responsibilities (role-playing), and think about developing innovative applications based on well-prepared documentation, compliant with other internationally approved standards, such as ISO/IEC 25000 [9]. Students are provided with all templates, tools, and documentation needed to perform this task.
5. **Encourage creativity** by inventing fictional software projects that focus on real business problems, especially if the supply of industry-related project proposals and ideas are insufficient (typical for such an initial course stage).
6. **Challenge** established business models (kind of "hackathon"). Provoke and cultivate the sense of a startup, but always available to support them via social and meeting platforms to eliminate ambiguities.
7. **Foster self-organization, leadership skills, and team management** deciding what to do and with which technologies, plans, and deliverables. Provide mentorship (organizational and technical, involve industry and professionals from other areas – from Q.A. to software and design companies) [16].

An important factor that supported the successful incorporation of such practical team projects was the approval of the course as a core course for BSc students in Informatics. This allowed stable team composition and result-oriented collective impetus (turned to be impossible with elective format). Such an approach is widely used in master's programs [13], where project and team management are in focus, and reinforced by earning credits of critical importance for graduation. However, since the major source of developers for the industry comes from undergraduate programs, the teaming and project culture is already a critical asset, as opposed to just an advantage.

Thus, it was beneficial for students to introduce it earlier in the undergraduate program (see feedback in Fig. 3). Another motivation was that most of the students from the last year in the university already have working experience in IT companies and have observations on "bad practices" as well and learning online they were able to take part in all lectures and exercise we had. They are prepared to extend the understanding of quality assurance beyond quality code producing and testing, and the importance of the established processes, quality of requirements, planning, monitoring, and control for successful project delivery in time.

The main goal of this approach is via introducing models for software quality maturity in software engineering education to support students in finding this intersection between modern theory and practice and lead this valuable knowledge to the small organizations and IT industry in summary.

To achieve that goal, we focus on bringing the real in-company working atmosphere to the university classroom by applying modern teaching methods, training techniques, and style of organization to break the patterns of the ordinary teach-exercise-grade education system. Through a proper role-playing and teamwork, we address the involvement of the different stakeholders – students, professors, future employers, government, end-users, IT companies, and clients working together based on the established "common language" – the language of quality.

4 The intersection between university courses and small business organizations

Through the course of the development and implementation of these sets of courses, we concluded that to achieve a certain level of flexibility and success in the area of software engineering education and the small organizations we need at least two things. First, to find the intersection between theory and practice, and secondly, to have well-described and defined inhouse work processes, no matter whether in an educational or business setting. For both, we can rely on the already introduced CMMI (or in the specific agile situation CMMI with Scrum), TMMI models, and the famous quality software group of standard ISO/IEC 25000. The question is what exactly we can use from them in the education area and how we can map that in the companies. In other words, how this knowledge can be easily transferred from our students and applied to small businesses. We will describe in detail and we will approach an example with a real company, in which one of the authors of the current contribution has been building a team for the last four years – with an initial number of team-members of 2 to currently 15 people.

We started with CMMI, which provides a variety of well-structured and complete reference frameworks for the organization and pathway to industry maturity for IT/software companies. Of course, this process requires careful selection and gradual introduction, with sufficient practical examples and exercises. The maturity of the processes as a differentiator for mature companies brings a natural basis for a deeper introduction to CMMI.

For the sake of better clarity, we use staged representation with Maturity Levels. We are limiting the scope to Levels 2 and 3. Maturity Level 1, also named "Performed", is a good illustration of companies in a "survival" mode. Here we make a discussion session with students that work on their observations over such a typical picture – processes and deliverables are unpredictable, we have no control, and the mode is reactive, most of the people do not know what and when or even how to do, so project delays and unsatisfied customers are common.

For education purposes, an essential part of the material is dedicated to Maturity Level 2 Managed (also referred to as a Repeatable). All process areas are reviewed in detail as this maturity level is dedicated to successful project management. Goals and practices are to implement managed, institutionalized processes and with continuous improvement. We have six important process areas from Maturity Level 2: *Requirements Management (REQM), Project Planning (PP), Project Monitoring and Control (PMC), Process and Product Quality Assurance (PPQA), Configuration Management (CM), and Measurement and Analysis (MA).*

A prerequisite for the start of real projects and group work is the knowledge of two process areas, namely *Requirements Management (REQM)* and *Project Planning (PP)*. This is the necessary minimum to start transforming the business ideas into project specifications, work breakdown structure with respective estimates, project plans (with Gantt charts). We provide templates for all documents, which result from these areas, and demonstrate the use of appropriate tools. We further include an introduction to the ISO/IEC 25000 standards at this stage as well.

Consequently, with the progress of work, the topic of what progress monitoring methods could be applied and why corrective measures are needed is discussed, thus introducing naturally the *Project Monitoring and Control (PMC)* process area with its respective specific practices. Afterward, students understand through experience what a "significant deviation" from the project plan means, how to follow corrective measures, and why *Configuration Management (CM)* is more than source control. Particular attention is paid to the specific goals and practices (that define the scope of the process areas), the generic goal, and the ten generic practices are repeatedly illustrated, finally giving the logical justification and link specific practices to be implemented for managing the quality of processes under *Process and Product Quality Assurance (PPQA)* process area. Some basic indicators needed for process improvement are also underlined by the mentors, so students can recognize the value of the practices under the *Measurement and Analysis (MA)* process area. The seventh process area from Maturity Level 2, *Supplier Agreement Management (SAM*) is only briefly introduced, as in several cases; a need to delegate or "outsource" some studio project tasks to other teams is identified.

From Maturity Level 3 (Defined) we cover only a selected subset (mainly around the *Requirements development and validation, peer reviews under Verification*) as the students need them for their team projects. Since the very first initiation stage of discussing the project ideas, the teams already need some guidance on formalizing "customer" requirements. Some hints come logically with specific practices from *Requirements Development (RD)* process area.

The *Validation (VAL)* process area becomes critical in few cases of real industry or business-related project ideas. Some of them are about "patching" or improving real systems in use (such as ERP software, for example) – a challenging task even for mature companies. From the higher Maturity Levels 4 and 5 (Quantitatively Managed and Optimizing), only the goals and business benefits are discussed (again referring to the Cost of Quality structure).

After having introduced the basics of CMMI and the ISO ISO/IEC 25000 standards, we further include TMMI [3] for the testing phase. As the maturity levels are the same, we can discuss straightly Level 2 (again called Managed), where the fundamental testing approach is established and managed, within an organization with test policy and strategy, test planning, test monitoring and control, test design, and execution and corresponding environment. We further introduce Level 3 (Defined), where all projects, which should follow the same standards and procedures throughout the organization and its units, are established. A particularity of Level 2 is that it is going on to Execute, however, teams are now organized, test-training programs exist, all tests are integrated into the development life cycle, and into all projects from early in development. Non-functional testing is planned, executed and reviews are used in each project as well. Because Level 4 (Measured) and Level 5 (Optimization) are again the responsibility of the higher management, we again just brief students shortly. By referring to industrial models and standards (vendor-neutral), we give a realistic picture of what the industrial environment is (or should be).

The overview of existing international software development and IT services standards and models, their applicability in different industries and for different purposes, as well as cross-mappings, is discussed. A good example that supports our approach is one of the latest feedbacks received from a small software outsourcing company in which the second author is working in addition. After passing the „Software Quality Assurance" course for the first time a couple of years ago, she received an opportunity to build a new team consists of 2 people in the beginning. In the first year, the company was at Level 1 and we had no structured workflow or defined processes. The author was starting methodically to use *REQM, PP, PMC, PPQA* from Level 2*,* combined with the *RD* process area from Level 3. With the consistent implementation of these processes, as well as with the help of the general and specific practices, at the end of the first year, the team grew to 6 persons. In the next 3 years, they have increased respectively to 11 and then 13 until now when they have reached 15 (web designers, 3D designers, and front-end developers. Most of the team members are students from FMI-PU and have passed that "Software Quality Assurance" course, which greatly supports the processes' institutionalization as the teammates are already familiar with the model and its practices.

Currently, the business workflow is well defined and everybody knows what, when, and how to do it. That leads to successful results, respected deadlines, despite the remote work setting for more than a year, and the constantly changing requirements. All achieved through well-defined processes and practices in different process areas and consistency.

One of the other company departments (there are five in summary) is starting to use CMMI repeatable maturity model, after being introduced with the good experience of the described team.

5 Our methodology in depth

Familiarizing with the de-facto industry standards (CMMI) and TMMI, exercising them through collaborative realistic projects with the use of the right tools, manages to convey an "industrial" atmosphere in the university classroom. Each of the lectures challenges students with practical industrial challenges through exercises structured around the development of a real project. A list of contemporary tools (most of them fully open-source) and techniques with proper samples are being used and introduced not only for their university projects but also in real work business environments [16], [17].

Our main function as mentors is to align those profiles with the respective process areas' activities and responsibilities and to make sure those members with more than one "hat" (needed for smaller teams) act appropriately. Another engagement is to make sure that we systematically develop the ICT professional competences based on the newest European standard e-CF (European e-Competence Framework) [18]. *In our educational course, we use the following methodology:*

- Demonstrate and outline the work on real software development projects in teams. The important factor here is for the students to know what to do, how, and with what in their team projects.
- **Encourage teamwork and role-playing** by involving students in studio projects.
- **Provide students with the freedom to creatively decide on their projects' scope and development phases**, while notwithstanding providing assistance and facilitating the collaboration, as needed.
- **Allow students to engage in the documentation and presentation** of their work, thus training them on responsibilities, native to project management roles.
- **Demonstrate the use of contemporary tools and techniques** [17] to achieve all of the above.
- **Provoke, model, and adapt team challenges** to encourage innovation and creativity, as well as to a realistic business arena and "competitive" environment.

- Last, but not least, we allow students to decide on the awards and provide additional "bonus" scores and unique certificates by category. Completed projects also receive a SEMP-recognized sign. Those competitions and certificates turned to be a huge motivator for students and an inspiration for innovative and amazing ideas.

For the six academic years of running the course "Quality Code Assurance" as described, with 36 student projects completed. Although the circumstances, imposed by the Coronavirus pandemic, required entirely remote collaboration, we had more teams and successful projects than ever.

Furthermore, a noticeable evolution in the project ideas can be observed as well. In the beginning, projects were more trivial, concerning online shopping services or e-commerce systems, however, since the latest course, we have had business-to-business applications, game portals, mobile applications, ticketing systems, teachers' diary, and a Python-based AI project.

6 University Course Feedback

Due to the pandemic allowing remote participation in the classroom activities, a much bigger interest, presence, and motivation of students to engage proactively in the different software projects within the course "Software Quality Assurance" was observed, further showing the relevance of our conclusions and approach.

Our students succeeded to put theoretical knowledge on process quality improvement into practice and develop valuable skills:

1) creating, developing, and managing software project specifications and plans, through different industrial de-facto standards like CMMI-DEV, TMMI, and CMMI with Scrum,
2) understanding and applying mature software development processes, analyze them for improvements, focus on defects prevention, work in teams (even manage them online) with roles and responsibilities, and 3) immerse in a real industrial atmosphere and style of work (including a "bonus" system for performance and creativity).

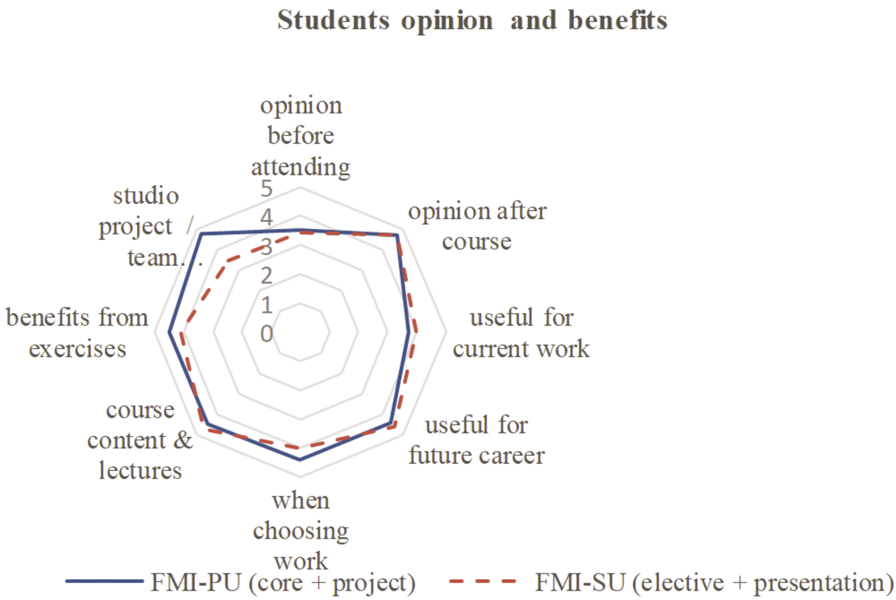

Fig. 3. Final results from software quality management courses at FMI, Plovdiv University (core course with project teamwork, online for the last "pandemic" year) and FMI, Sofia University (elective, with group study and presentations as teamwork, again online this academic year).

We have done two different types of summary surveys. First, was made from FMI-PU students, participating in the last 6 years of the course delivery as a core course with studio projects is shown in Fig. 1, in comparison to the aggregated feedback from a similar CMMI-based course at FMI-SU. The essential difference between the two courses' formats is that the latter is still elective (also for BSc, but mixed diverse years of study and specialties), and the practical teamwork is limited to group study and presentations (e.g., on process area or implementation mixed techniques with Agile, DevOps, Kanban).

In both cases, the feedback on the course benefits and value for real work and employment are comparable (high). However, the significant difference in students' opinion on teamwork format clearly shows the advantage of the studio project.

We can also add from this group of students some open comments related to the benefits of the course, such as:

- "The teamwork and the practical knowledge gained on different process-oriented models",
- "The knowledge I received for my future projects",
- "The knowledge I gained, which I can apply in my practice",
- "The teamwork and the simulation of an actual work environment",

- “The very motivating teachers who want to teach us in a way we can easily understand the material, not only take the exam after the course”.

The second type of study (shown in Fig. 4) was made based on how many teams were involved in projects this year (made only for FMI-PU), comparing the last 5 years, when we studied in the auditoriums, not online.

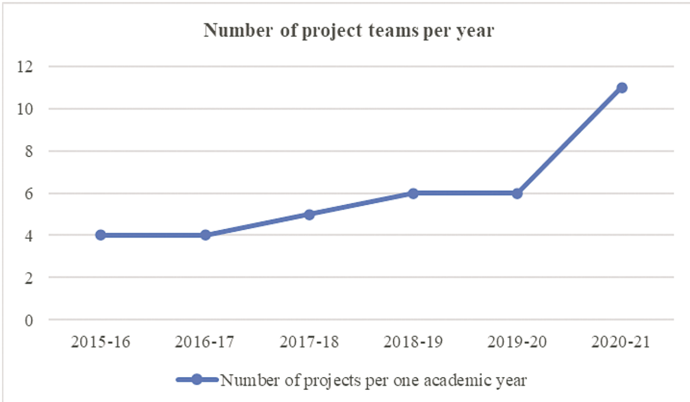

Fig. 4. The blue curve shows how many project teams with students were involved in the last 6 academic years. One course with students contains between 90-100 people for each year.

It is easily observable that in the current academic year we have our peak in the number of projects – 11 teams with students. That is double the year before. We may conclude, supported by the student’s feedback that this higher interest is mainly due to the forced by the pandemic online teaching/studying process. Through the remote setting of education, our working colleges can also participate in lectures and exercises, even from their offices or at home, without the traveling downtime, or being absent from their job.

Moreover, the use of different social platforms for communication and video conferencing provided both teachers and students, with the ability to keep in touch constantly and to be much more flexible through various collaboration platforms. In addition, this is additional experience for their future teamwork, especially within project management tools and environments, which was previously limited to the in-class exercises.

7 Results and conclusions

This paper introduced authors approach to the models for software quality maturity in software engineering education. It represented how they can be transferred and applied from students in small business organizations, especially functioning within the IT industry. We described our core course „Software Quality Assurance (Q.A.)” for the last (4th) year of study at the Faculty of Mathematics at the University of Plovdiv [5], with the feedback from the last six years of delivery as compared to other similar courses and formats. The most appreciated advantages of the course consist of: (1) Introduction to industrial standard models CMMI, CMMI with Scrum and TMMI and (2) Applying model to small business organizations with the survey showing how one company department started with two workers 4 years ago and increased in size to 15 as of currently.

We have demonstrated how such a course could be used by graduated students as a framework to find an intersection between theory and practice in their current and future projects. Trough the studio-projects we have encouraged the teamwork and practical exercising of roles and functions in software development teams, thus cultivating at university the so much desired by the industry behavioral competencies, such as leadership, conflict management, communications, and teaming [21].

References

1. CMMI for Development model – version 1.3, http://cmmiinstitute.com/resources/cmmi-development-version-13, last accessed 2021/05/18.
2. Improving Agile Performance with CMMI, https://cmmiinstitute.com/cmmi-and-agile, last accessed 2021/05/18.
3. TMMI model, https://www.tmmi.org/tmmi-model/, last accessed 2021/05/18.
4. SEMP (Software Engineering Management Program), https://semp.esicenter.bg/partners/, last accessed 2021/05/18.
5. Stoeva, M., Sharkov, G., Specialized page of "Software Quality Assurance (Q.A.)" course, http://edesign-bg.com/quality-software-2021.html, last accessed 2021/05/18.
6. Sharkov G., Stoeva M.: Software Engineering Management Education – Bringing Industry Standards to the University Program, Tenth Jubilee International Conference Information systems and grid technologies – ISGT 2016, p. 184-198, ISSN 1314-4855, Sofia, Bulgaria (2016).
7. Pichler, R., Agile Product Management with Scrum: Creating Products that Customers Love, ISBN-13: 978-0-321-60578-8, Massachusetts (2010).
8. Rubin, K., Essential Scrum: A Practical Guide to the Most Popular Agile Process, ISBN-13 978-0-13-704329-3, Michigan (2012).
9. ISO/IEC 25000 – SQuaRE (*System and Software Quality Requirements and Evaluation*), https://iso25000.com/index.php/en/iso-25000-standards, last accessed 2021/05/18.
10. Stoeva, M., Krushkova, M.: Modeling and implementation of an interactive web-based system for education and entertainment with games, From DeLC to VelSpace. In: International conference, FMI, Plovdiv University "Paisii Hilendarski", Plovdiv town, Bulgaria, ISBN: 0-9545660-2-526-28, p. 265-274 (2014).
11. Kaloyanova, K.: Design from Data: How to Use Requirements for Better IS Analysis and Design. In: Proceedings of the Int. Conference Informatics in Scientific Knowledge, pp. 189-197., (2012).
12. Kaloyanova, K.: Including Real Stakeholders at Students Projects. In: Proceedings of the 9th International Conference Computer Science and Education in Computer Science (CSECS), Fulda/Wurzburg, Germany, p. 55-59, (2013).
13. Sharkov, G., Asenova, P., Ivanova, V., Gueorguiev, I., Varbanov, P.: Evaluation of ICT Curricula using European e-Competence Framework. In: 10th Annual International Conference on

Computer Science and Education in Computer Science 2014 (CSECS), pp. 267-286, https://www.ceeol.com/search/article-detail?id=469775 (2014).
14. Kaplan, R. S., Norton, D. P.: The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment. Boston, MA: Harvard Business School Press (2000).
15. Top Online Meeting Software, https://www.techfunnel.com/information-technology/11-best-virtual-meeting-platforms-for-business/, last accessed 2021/05/18.
16. Stoeva, M.: Interactive Multimedia tool for dynamic generation of web interfaces with HTML5/PHP/MySQL and JavaScript. International Journal of Emerging Technology & Advanced Engineering, Vol. 4, Iss. 9, p. 412–418 (2014).
17. Contemporary tools for prototyping, project planning, Gantt charts, software testing, and sharing information, including, but not limited to: moqups.com, www.uxpin.com, figma.com, docs.google.com, www.smartsheet.com, www.teamgantt.com, www.browserstack.com, www.atlassian.com/software/jira, https://loadfocus.com/, https://kardsort.com/, last accessed 2021/05/18.
18. e-CF (European e-Competence Framework) version 3.0, adopted in 2016 as European standard and published by CEN as the European Norm EN 16234-1, https://www.ecompetences.eu/, last accessed 2021/05/18.
19. Caralli, R., Allen, J. and White, D. *CERT Resilience Management Model (CERT-RMM)*. SEI Series in Software Engineering, Addison-Wesley Professional (2011).
20. Kamoen, N., *Positive, negative, and bipolar questions: The effect of question polarity on ratings of text readability*, Survey Research Methods 7(3):181-189 (2013), https://www.researchgate.net/publication/261340634_Positive_negative_and_bipolar_questions_The_effect_of_question_polarity_on_ratings_of_text_readability, last accessed 2021/05/18.
21. Kanabar, V., Kaloyanova, K.: Identifying and Embedding Behavioral Competencies in Information Systems Courses. In Proceedings of the 25th European Conference on Information Systems (ECIS), Guimarães, Portugal, June 5-10, 2017 (pp. 3115-3122). ISBN 978-0-9915567-0-0 Research-in-Progress Papers. https://aisel.aisnet.org/ecis2017_rip/59.