Journal of Theoretical and Applied Information Technology
15th April 2020. Vol.98. No 07

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SCRUM-BASED SOFTWARE QUALITY MODEL TO RAISE MATURITY OF SOFTWARE IN SME(S)

[1]SARAH K. AMER, [2]NAGWA BADR, [3]AHMED HAMAD

[1]Demonstrator, Department of Informatics and Computer Science, the British University in Egypt, Cairo

[2]Professor, Department of Informatics and Computer Science, Ain Shams University, Cairo

[3]Professor, Department of Informatics and Computer Science, the British University in Egypt, Cairo

E-mail: [1]sara.amer@bue.edu.eg, [2]nagwabadr@cis.asu.edu.eg, [3]ahmed.hamad@bue.edu.eg

ABSTRACT

Software process improvement (SPI), and by extension Software Quality Assurance (SQA), is the approach to understand the software development process lifecycle and implement necessary changes to the processes to achieve a high-quality, maintainable product. Small software enterprises face enormous challenges to gain a competitive advantage in the software industry, especially with the presence of large conglomerates. Much of these small-to-medium enterprises (SMEs) adopt agile models such as Scrum to quickly react to clients' demands. However, agile methodologies lack direct addressing of maturity in process, project and product that larger enterprises are capable of. It is important for the software engineering community to aid in enabling SMEs to have process maturity without compromising agility. In this paper, we use the Capability Maturity Model Integration (CMMI) – a world-renowned software quality assurance methodology, to address some shortcomings in the Scrum model. Specific practices are selected out of eighteen process areas based on both literature research and field study from the second and third levels of maturity to address missing elements. The proposed model prototype keeps Scrum intact while allowing small enterprises to produce high-quality software without compromising agility or going over budget, thus reducing the 'low-quality' stigma attached to small software developers around the world.

Keywords: *Capability Maturity Model Integration, CMMI, Agile Software Development Model, Scrum, Software process improvement*

1. INTRODUCTION

Small-to-medium software development enterprises (known as SMEs)[1] make up the majority of the software development industry [1][2][3][4], and most do not use software process improvement (SPI) methodologies, relying instead on ad-hoc culture. The purpose of SPI is to understand the software development process lifecycle in an enterprise, and incorporate changes to the processes in order to achieve a mature, high-quality and maintainable product [5]. SPI models are many, including ISO 9000 series of standards [6], Six Sigma [7][8], and CMMI [9] amongst others [10]. SPI's benefits lie in improved quality of software, lower development costs, less rework, higher productivity, shorter project cycle time and most crucially, client satisfaction.

SMEs use agile methodologies to survive the intense competition by deploying solutions quickly. Scrum is a popular choice, with 85% of agile adaptors embracing Scrum as their preferred model [11][12]. Many SMEs' main source of income is maintenance, servicing the client after the software has already been deployed. This is all the more reason for the software engineering community to aid by introducing process maturity practices without compromising agility.

Most studies on the determinants of process maturity in literature are conducted in medium-or-large scale enterprises, and it is common to face enormous challenge in adopting an SPI model in adjacency to its agile model [14][15]. Therefore, it is necessary to find an alternative solution for these enterprises so they can deploy mature software, and guarantee high quality performance and thus returning clients.

Scrum is a popular agile model; agile software development models follow a culture of

[1] Small-to-medium enterprises have no more than 250 employees, and have significant restrictions on time, personnel and funding [2, 13]

adaptive planning and iterative development of functional and non-functional requirements. Its greatest competitive advantage is quick response to change and open, flexible communication channels [16][17]. Like other agile models, Scrum lends itself to a sprint culture, iterative problem-solving and constant communication with clients. [18][19]. A sprint normally lasts between one week and one month and is a full software development lifecycle. A Sprint Backlog is defined for every sprint, defining its objectives and used as a forecast estimate. It derives its case studies and requirements from the Product Backlog, prepared prior by the Product Owner and the client. Scrum mandates close collaboration between the development team, business experts, and clients to produce potentially shippable, fully integrated software that has been tested [15][20].

Agile development is synonymous with speed to market and increased productivity [21], and as SMEs make up the majority of the software industry, they are targeted for offshoring contracts and as such, a globally accepted standard such as CMMI becomes a deciding factor in winning the contract by adopting such models in adjacency with agile [22].

However, there are some gaps in process, project and product quality that Scrum does not address directly: it does not curb rework or slippage, and explicit risk management is not addressed. Therefore, efforts by academics and industry professionals go into incorporating SPI with agile to improve the progress of software development [23][36].

Smaller software enterprises are wary of adopting quality assurance models because of the belief that they are bureaucratic, documentation-heavy and geared towards larger enterprises with better resources [1][14][24]. Both researchers and industry professionals are cooperating to explore novel ways to make agile and SPI work smoothly together. Process improvement models do not avoid the volatile nature of business environments, but are designed to embrace changes and respond timely to them [25][26].

Capability Maturity Model Integration (CMMI) is a world-renowned SPI model adopted by software enterprises around the world with the purpose of processing control, quality improvement, and measurement within an enterprise. Its process areas pertain to every aspect of the development life cycle, and all the personnel and resources involved [9][26][27][28][29]. The purpose is to achieve process maturity, which is an indicator of how well a process is defined, measured and improved. The concept of process maturity is the main form of evaluation. CMMI follows an evolutionary path through 5 maturity levels. Each maturity level other than the Initial level 0 comprises of process areas; a process area is a collection of best practices in a specific area [15][27][30][31]. When implemented properly, it serves to fulfill goals important for quality in that area. In order to achieve a maturity rating recognized globally, the enterprise must undergo an appraisal by an accredited, CMMI-partner external party. Contrary to software developers' fears, quality models do not hurdle agility, and can work well together, as explored in literature [14][18][32]. It was chosen for this work for its positive results, large amount of interest and resources – both in academia and in the industry – and its popularity among clients looking for a trusted standard. The objective of CMMI is to create a methodology of quality assurance that embraces change, addresses maintainability through quantitative measurement, and recognizes risk process [26][33]. CMMI is a living model due to it being constantly updated by its creators. During the writing of this paper, an updated version of CMMI – CMMI Ver. 2.0, was released.

Developers in SMEs do not give SPI the focus necessary, and one recurring factor discussed in literature is that SPI models are more geared towards larger enterprises rather than small startups [34]. Small enterprises need their own modified model that can enable them to increase the maturity and value of their software and future endeavors. They have constraints in terms of time, personnel, funds, training and other resources that do not enable them to embrace the full functionalities of a quality assurance methodology [4][35]. Scrum is a top choice of agile methodology for SMEs, but addressing some of its shortcomings would highly improve the maintainability of their solutions, and improve their clientele and competitiveness [15].

In this paper, we define the most crucial level 2 and 3 process areas and related metrics through researching previous literature and conducting a field study on a sample of the software industry, and propose our concept for a customized Scrum model prototype. The prototype incorporates quality assurance elements to raise its maturity. It must be compact in order to address SME environments realistically.

The rest of the paper is divided as follows: a brief overview of related work in SPI with a focus on CMMI in conjunction with Scrum. This is followed by the research method and proposed model. The model is then discussed and evaluated, followed by the work's limitations and finally the conclusion.

2. RELATED WORK

According to the CMMI Institute in 2015, more than 70% of enterprises with CMMI appraisals use an agile model in their development [28]. Several researchers have attempted to find alternatives for enterprises to introduce manageable SPI models that combine CMMI's comprehensiveness with agile programming's speedy iterations. Due to the limited resources and personnel in those enterprises, they are placed at a serious disadvantage against larger corporations. The software industry is highly competitive, and clients demand top quality especially in critical software solutions. Scrum methodology alone will not bring the SME success or maturity. Scrum is a popular agile model, its strength being its flexibility and quick response to changes by the client. However, it falls back in structured process and process management, engineering practices and quality assurance. The model makes little consideration for things like process quality assurance, building critical software, and maintaining an inventory of reusable artifacts. There are challenges to integration, risk management and project budget control, and there is no addressing of suppliers of outsourced work (e.g. subcontracting). [36][37]

2.1 CMMI and Scrum in Larger Enterprises

Sections (Lina and Dan contribute to the merge between Scrum and CMMI with a focus on project management practices in maturity Level 2 [38]. They show gaps between the two methodologies, and where enterprises adopt practices that make them compliant. Scrum provides the software development how-to's whereas CMMI provides the engineering practices that enables Scrum to be used on larger software projects. They conclude that Scrum is very effective in SMEs, but the larger the enterprise, the less agile it is, hence the model must be customized to accommodate.

Tobal and Carvalho conduct research to identify the most crucial CSFs necessary for a software project's success and sustainability [39]. They conclude that three of the top CSFs necessary for software maturity in literature are top management support, a full-time dedicated project manager, and enterprise-level project management maturity. Their scope of the model is limited in the number of variables they study and in the fact that it predicts only project success and not failure. Other variables like project complexity, sector, and lifecycle length should be considered. Sánchez-Gordón and O'Conner clarify the importance and reasons for representing a software process model in an abstract form to be easily adapted by enterprises [1]. They provide evidence in literature and through their own study that small software enterprises do not adopt quality assurance models because of the belief that they are geared towards large enterprises, in addition to a surprising number believing they do not need it.

In Niazi's work [10], a systematic literature review is conducted for the purpose of gaining a deep understanding of success factors crucial for SPI implementation in order to develop SPI methodologies that can be used in different environments and contexts. The researcher forms a comprehensive list of Critical Success Factors (CSFs) necessary for SPI program initiatives. The results of the study show that Senior Management Commitment constantly comes on top as the most important CSF, followed by SPI Awareness, Staff Involvement, Training and Mentoring, and Staff Time and Resources. The CSFs are analyzed and further supported by the data collection directly from the industry, with data collected exclusively reliant on Australian SPI practitioners.

Alharbi and Qureshi propose a solution to incorporate Risk Management into the Scrum model [40]. They create a Risk Register component to be included at each Scrum sprint to manage the risks early on and thus improve the process quality. The Scrum master and team review the risks using the Register during the sprint retrospective meetings, and the risks should go down with each iteration, normally linearly. They validate the proposed solution and conclude after a 70% support rate through cumulative analysis that agility is not diminished through the addition of the risk-managing practice. However, other process areas, such as Requirements Development (RD), were not considered for this model.

Tosun et al. [41] present a software quality improvement project (SQIP) by selecting specific practices from the CMMI maturity levels 2 and 3, then they conduct an internal, non-formal appraisal on two projects – one had no quality processes and another had SQIP and the differences were studied. The selected developer for the case study is a small enterprise, with challenges in several issues such as defect rate and performance measurement, project management, and employee training. Also, there is no set process definition, as work is performed and delivered by some high-level developers. This inevitably leads to bad quality of service and lengthy response time to any change requests. To measure SQIP's performance, the authors observe defect rates in both the project implementing SQIP and the one without, while keeping factors such as team size, cyclomatic complexity and other factors constant. They follow the differences in completion times at each stage of the development life cycle, and discuss interesting variations. One such element of study is during Requirements Management; the practices of the implemented process area took more than seven times the time taken in the SQIP-abiding project. This reduced development time, testing time and effort per person considerably. The authors then define a set of best practices that focus on proper training and management support. Worst practices were summarized in underestimating/ignoring the project planning-related aspects of a project, disregarding team's fears, not using automated tools to save time and increase efficiency, and very importantly, not separating the testing from the development work. These all serve the enterprise's software quality and maintainability, and propose future work on easing metric and bug data collection for the enterprise to use in order to accurately predict project performance in the future.

Engdashet et al. proposed a high-level framework that explores the incorporation of key process areas in CMMI with agile methods, specifically eXtreme Programing (XP) and Scrum [42]. They observed case studies and previous work using agile and CMMI to compensate for each other's shortcomings. As indicated by Jakobsen and Johnson [30], CMMI is suitable for institutionalizing agile methods in a structured manner, whilst the agile methods guide the implementation of CMMI requirements by being the 'how' to the 'what'. Their method of data collection from the industry was through questionnaires and interviews, to prepare for the future step of applying their proposed approach practically to achieve a better evaluation. In the work of Kahkonen and Abrahamsson [43], they explored the possible relationship between eXtreme Programing (XP) and CMMI level 2 in particular. Their work focused on incorporating CMMI as a method of assessment for XP practices, but not the process of maturing the process capabilities of an enterprise.

Elshafey and Galal-Edeen [44] introduce an integrated model combining two approaches, CMMI and general Agile practices in order to balance agility with control and structure. The outcome is designed to have enough flexibility to serve medium-sized enterprises, although no explicit mention of SMEs is made. The authors integrated the two methodologies, leaving out some CMMI process areas that were found to not have a direct connection or relevant tool in any of the popular agile methodologies like Scrum and XP. The result is a model that can increase process maturity, and they recommend testing it in enterprises for several complete project lifecycles. Łukasiewicz and Miler [45] also propose a merge between CMMI and Scrum. Their solution is a combination of a CMMI-Scrum reference model, the P-Sel algorithm to suggest practices from the model based on 2 real life case studies, and the MatureScrum software tool to implement the questionnaire and algorithm. Their work focuses on CMMI V1.2 rather than V1.3.

Farid et al. [11][18] address the industry's need for more structure in their processes without sacrificing flexibility and agility, and provide an approach to map CMMI specific practices to Scrum before applying it to six real companies. They focus only on the project management category, considering activities related to planning, monitoring and controlling the project, and then calculate the CMMI Specific Practice coverage using a set of functions. Every mapped practice is given a rating of being fully satisfied, partially satisfied, or unsatisfied by Scrum's own practices. Their proposed mapping of specific practices selected from specific process areas in levels 2 and 3 to Scrum is done twice - once without enhancements, and another time with the automated solution Team Foundation Server (TFS) incorporated into the process. Their work discusses that Scrum practices successfully supported several CMMI practices certain process such as Project Planning (PP), Project Monitoring and Control (PMC), Risk Management (RSKM) and others. RSKM is remains largely unsatisfied by Scrum practices, which means there is a missing element in the quality assurance of the process and resulting software when Scrum is the model in use. The mapped work does not cover the SAM process area as none of its practices are supported by Scrum; the authors strongly recommend it be addressed due to the increasing size of outsourcing deals in the software development industry.

2.2 CMMI and Scrum in SMEs

Iqbal et al. [2] showcase the importance of SMEs to the global software industry, and literature backs up their finds that small developers are the primary and most common segment of the software industry worldwide. They study the trends of specific practices in CMMI level 2 and explore reasons smaller enterprises are less likely to adopt

Journal of Theoretical and Applied Information Technology
15th April 2020. Vol.98. No 07

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

an SPI model such as CMMI; company size, infrastructure support, cost, management's disinterest in strategic long-term survival are amongst the most recurring reasons. They study SMEs that implement specific practices from Maturity Level 2, and uncover current trends in the borrowal of CMMI practices. The analysed results indicate that specific practices associated with seven process areas of the second level of CMMI are informally followed in many SMEs, thus enabling them to weigh the potential of future CMMI-based SPI implementation. They note that the greater number of small companies do not have potential to become full-fledged CMMI implementers, proving the hypothesis that SPI does in fact have a high cost that puts some enterprises at an advantage against others. They conclude that SPI is inevitable in today's harsh industry.

Tripathi et al. [4] conduct an elaborate research of literature with specific focus on processes in SMEs. They determine that software requirement analysis, project planning, configuration management and lifecycle model management are considered the most necessary processes to be addressed by any SMEs with intent on long-term survival and improvement. SMEs generally face challenges in understanding how to properly collect requirements, and configuration management.

Rad et al. [14] propose a model that incorporates process patterns typically found in SMEs in order to improve the process maturity. This is done by mapping key CMMI process areas against popular agile practices over three stages; analyzing, planning and implementation. The CMMI maturity levels are given contexts that the SME can choose from in order to address the most realistic improvement route they could go down at any one time.

In the works of Chevers et al. [25][46], the most highly-ranked SPI practices of CMMI are identified in relevance to the limited size of the software development enterprise. The authors identify these practices in relevance to small enterprises based on their sample, in order to develop a process improvement model prototype. Amongst these areas were Risk Management (RSKM), Project Planning (PP) and Requirement Development (RD). Chevers uses the Canadian industry for the study, and reports that despite its prolific nature, the local industry still suffers from flawed software. The top-ranked SPI practices are used to determine if they are the most influential in process maturity in Canadian enterprises, and the authors work on simplifying CMMI for small enterprises by reducing the eighteen process areas of maturity levels 2 and 3 by almost half into only ten. They encourage that future work includes proposing a model or framework that addresses the limitations of small developers' by being non-disruptive and cost-effective.

Omran explores CMM/CMMI in small-to-medium enterprise (SME) settings, and points out where the challenges lie in CMMI being originally conceptualized for larger enterprises, and how SMEs can make use of the process areas necessary to improve their quality without interfering with agile model's rapid prototyping nature [15]. The author studies the testament of two such enterprises, and documents a significant improvement in rework, defect density, software productivity, and even unit software cost. To improve upon these results, work was done to create a framework that overlaps XP Programming and CMMI (XCMMI) [47], however XP does not address some process areas, such as Organizational Process Performance (OPP) and does not address Quantitative Project Management (QPM). This is not unexpected given that XP is an agile model. Another direction was to scale down CMM to befit the small size of these enterprises. As with XP, Scrum overlaps with some of CMMI's process areas fully or partially, but not others.

Mc Caffery et al. describe a lightweight quality model that they designed for SMEs, particularly geared towards automotive software developers and suppliers. [48] They conduct their case study on the growing Irish and Finnish automotive industries, noting that SMEs unsurprisingly form the majority of all enterprises. Their model, AHAA, assesses an SME for its existing software development practices and how they can be improved. AHAA integrates CMMI, and an agile model named Automotive SPICE™. The process areas were selected based on their own extensive research of the Irish software industry and their compliance with agile models in SMEs. The selected process areas for their AHAA model are Requirements Management (REQM), Project Planning (PP), Project monitoring and Control (PMC), and Configuration Management (CM). They map these process areas to the SPICE™ agile practices and document positive results in their case study applied to a real enterprise. Naturally, AHAA is not formally certified but provides an affordable assessment method, with the authors discussing future improvements with more CMMI process areas instead of only four.

These studies are valuable in their efforts to make enhanced quality a reality for SMEs, however there

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

are limitations where they focus very closely on specific CMMI practices more than others, examples of which are Risk Management and Supplier Agreement Management (SAM).

2.3 Research Questions and Research Method

Given the shortage of qualitative and quantitative information on CMMI adoption in SMEs, literature analysis, field research, and data collection were used to arrive at a prototype for a model that SMEs can adopt prior to taking the plunge into full CMMI expenses and obligations for formal appraisal. The model proposed is titled SQA-SCRUM.

This work uses the EU's official definition of SME, a small-size company comprising of no more than 250 employees [2][13]. In addition to reviewing the work of other researchers, data was collected firsthand from the software industry.

Data was collected from one country as a test example of a growing offshoring destination. Enterprises that formally and successfully implement CMMI are medium-to-large in size, and these are the enterprises from which the authors gathered their data to create a solid base on which to build a model that SMEs can use to mature their processes.

Two methods were used to collect data from enterprises: a closed-ended questionnaire, and one-on-one interview follow-ups. These two were selected for their convenience, in order to encourage the sampled professionals to take the time out of their schedule to answer our questions in a satisfactory manner. The questionnaire was designed to collect both qualitative and quantitative data on several aspects of the enterprise's application of quality assurance before and after CMMI adoption, such as which models are used in adjacency with CMMI, and the types of metrics and indicators they keep track of during the development life cycle. The questionnaire was also crucial in determining the most effective CMMI process areas across the sample by ranking according to their effectiveness in addressing the enterprises' most expensive problems in project, process and product quality. Respondents were project managers, team leaders, and software quality assurance personnel. In enterprises with no independent SQA team, the project manager or team leader managed SQA themselves.

Seven responses and data collection sessions were considered for this study, chosen on basis of unambiguous answers. To protect the respondents' confidentiality, names of CMMI-appraised enterprises are kept anonymous and all respondents are made aware of the use of this data to aid in research and exploration of current quality assurance trends. The authors' empirical study explores findings in the country's local industry in more detail [49]. Interviews allowed the authors to follow up on some interesting answers found in the questionnaire results, and to minimize ambiguity and misunderstanding.

The questionnaire and interviews were necessary to for this work as certain data is not easily found in SPI-related literature, such as the metrics and indicators used by different enterprises. Professionals who answered the questionnaire were also interviewed. In addition to the seven initial responses, two more professionals were approached and interviewed for their experience to review any logical or practicality issues, one being a CMMI partner and the other an official appraiser of CMMI, certified by Carnegie Mellon.

The field research was designed to find top ranking process areas within the sampled enterprises based on their effectiveness in curbing rework and costs. That finding was then used alongside literature analysis summarized in the previous sub-sections to design SQA-SCRUM. The completed model prototype was then evaluated as follows:

a. Obtaining constructive feedback and validation from professionals working with the implementation and appraisal of CMMI

b. Comparison with previous models in literature to determine improvements if any

3. MODEL DESIGN

The three highest ranked process areas according to the field research study results discussed in the previous section were Project Planning (PP), Risk Management (RSKM), and Project Monitoring and Control (PMC). They were followed closely by Product Integration (PI), Configuration Management (CM), and Process and Product Quality Assurance (PPQA). Other process areas were recognized, but our work will focus on utilizing the most crucial to SMEs, considering their limitations.

To support these findings, we returned to literature in order to observe similarities between this process area list and others' works. Using Chevers et al.'s study of the Canadian software industry [25, 46], we compared our findings of effective CMMI process areas in the local industry to Canada's ten most effective process areas. The resulting similarities supported the realism of our data. Other publications consulted were those of Farid et al. which contained information on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Egyptian industry [11]. The proposed models by Farid et al.[11][18], Elshafey and Galal-Edeen, and [40] Łukasiewicz and Miler [45] serve as basis for current mapping trends between agile models and CMMI, and confirm the compatibility of certain process areas and their practices to Scrum.

The diagram in Figure 1 shows common areas between process areas in our findings and those by Chevers et al. and Farid et al. regarding the highly effective process areas in the Canadian and Egyptian software industries respectively.

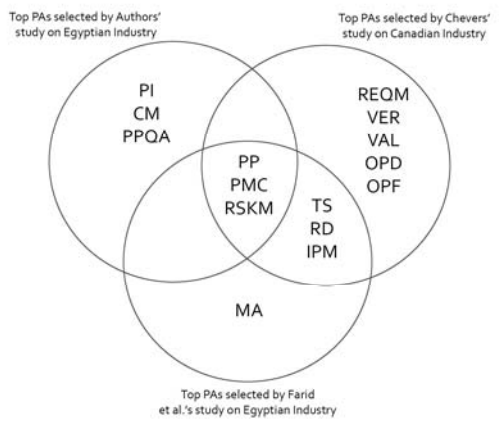

Figure 1: Venn Diagram indicating common PAs across three independent studies

As shown in the figure, the test sample from CMMI-appraised enterprises in our country's test sample – conveyed by the left ellipse, unveiled that implementers ranked Process and Product Quality Assurance (PPQA), Product Integration (PI), Configuration Management (CM), Project Monitoring and Control (PMC), Project Planning (PP), and Risk Management (RSKM) as the five most effective process area to their enterprises' livelihood. The bottom ellipse considers the study of Farid et al. on the Egyptian software industry. The process areas Project Monitoring and Control (PMC), Project Planning (PP), and Risk Management (RSKM) share the position with our own research as highly influential process quality improvement practices. Additionally, Farid et al.'s research also uncovers a strong need for Technical Solution (TS), Requirements Management (REQM), Requirements Development (RD), Integrated Project Management (IPM), and Measurement and Analysis (MA). The right ellipse considers some of Chevers' observations on the Canadian software industry. He noted that the top most crucial process areas as ranked by his sampled enterprises were Project Monitoring and Control (PMC), Project Planning (PP), Risk Management (RSKM), Verification and Validation (VER+VAL), Enterpriseal Process Focus (OPF), Enterpriseal Process Definition (OPD), Technical Solution (TS), Requirements Management (REQM), Requirements Development (RD), Integrated Project Management (IPM), and Measurement and Analysis (MA).

Project Monitoring and Control (PMC), Project Planning (PP), and Risk Management (RSKM) are therefore the top 3 prioritized process areas as a result of the union of these 3 field studies, and take precedence in the proposed SQA-SCRUM model.

SQA-SCRUM was designed while keeping the structure of Scrum unchanged in order to be easily recognized and understood by professionals familiar with the agile development methodology. The selected CMMI process areas are further broken down into their individual specific practices and incorporated where they are needed, thus addressing some of the oversights present within Scrum.

In an environment as limited as an SME, CMMI process areas have to be carefully selected, scaled down and then infused into the correct stage of the typical Scrum model. Those selected process areas are broken down into their component practices and infused into Scrum where they are needed. This creates a model that considers SQA elements without sacrificing agility by not needing to expand teams or manage large numbers of people and not exceeding a planned number of iterations or going over the specified deadline and budget of the project.

Process areas were selected after careful study and analysis of CMMI's maturity levels 2 and 3 in literature, and analysis of the industry field research sample. Integrated Project Management (IPM) as a process area was removed from consideration in SQA-SCRUM due to their time-consuming overhead, which is impractical in SMEs [6][50].

To clarify the process of selecting the appropriate practices, selection criteria were put in place to escalate Scrum's focus on process improvement without sacrificing agility and small team size. The selection criteria can be summarized as follows:

- Added practices must add quality-related benefit previously absent in Scrum model
- Must not compromise agility
- Must not involve enlarging internal teams
- Does not require extensive training, outsourcing and research

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Keeps extra documentation minimal by smart use of tools like Team Foundation Server (TFS) during sprints
- Does not include practices already addressed to an extent by the practices of another process area. Notable example is the Verification (VER) process area, for verification practices will be performed under the umbrella of other process areas

The ten process areas selected based on literature and industry data sample findings are shown in Table 1. These process areas have been identified as 'determinants' by combining both literature and our sampling of the software industry, and are considered by sampled professionals the most significant for easily identifiable positive change in quality within their work environments. These process areas range from being largely addressed by Scrum practices, like Project Planning (PP), to being unaddressed at all, like Supplier Agreement Management (SAM). Process areas that change the accepted structure of the Scrum model, or require the presence of extra specialized personnel have been removed to streamline the process.

The selected process areas are too large to be implemented in their entirety in a small scale agile enterprise; they are reduced into specific practices, and the most necessary practices as deemed by our literature research and field study are included in SQA-SCRUM. All other specific practices are removed to streamline the model. Specific practices that are already wholly covered by Scrum's practices as indicated in works [11][18][40] are not addressed in this model to avoid redundancy. Only partially supported and unsupported practices are incorporated into the Scrum model in the positions that require them so they would be implemented effectively. Tables 2 to 11 specify all selected practices based on the selection criteria mentioned prior, meant to improve the process definition, and thus process quality of Scrum's practices where an agile model falls short. Table 2 addresses the Project Planning (PP) process area, Table 3 addresses Requirements Development (RD), Table 4 addresses Project Monitoring and Control (PMC), and Table 5 addresses Measurement and Analysis (MA). Table 6 addresses Technical Solution (TS), Table 7 addresses Risk Management (RSKM), Table 8 addresses Configuration Management (CM) and Table 9 addresses Product Integration (PI). Finally, Table 10 addresses Process and Product Quality Assurance (PPQA), and Table 11 addresses Supplier Agreement Management (SAM).

In the tables, each process area is referred to by its designated abbreviation. Specific Goals are shortened to SG, and Specific Practices are shortened to SP. The left column represents the content presented to its right. The Specific Goal refers to the name of the specific goal from a specific process area. Specific Practice refers to the individual practices within the Specific Goal that are necessary for SQA-SCRUM's design. Level of Support refers to how much the typical Scrum model supports that specific goal. It can either be partially supported, or not supported at all. Finally, the Proposed Positioning Within Scrum explains where the specific practices need to be implemented within typical Scrum to create the more quality-conscious version of SQA-SCRUM.

We use the illustration of the Scrum model by [51] as underlying structure for our proposed model SQA-SCRUM due to its high quality design and readability across all the models studied by the authors. The Specific Goals selected from each of the process areas shown in Table I are incorporated into the model in Figure 2 using the abbreviation SG, and Specific Practices are shortened to SP.

The model considers the collaboration between the Product Owner with the Client to produce a comprehensive Product Backlog. They discuss and estimate the costs of the final product (in Project Planning SG1 SP4) whilst identifying and analyzing all possible risks to the project's success and timely delivery (Risk Management SG1 SP1). Next, all configuration management items and baselines must be identified and added the Sprint Backlog and/or Product Backlog by the Product Owner, and communicated directly to the Scrum Master (Configuration Management SG1 SP1 SP2 SP3).

It was taken into consideration that the SME may need to deal with other developers and suppliers of solutions or tools on a specific, limited scale. Suppliers must be selected carefully and agreements forged, which are the purpose of (Supplier Agreement Management SG1 SP2 SP3). It is the Product Owner's responsibility to communicate with and validate the required offshored product. Within the sprint itself, the offshored work product or tool is to be incorporated into the current product iteration after being accepted and transitioned, and that is where the practices of (Supplier Agreement Management SG2 SP3 SP4) come into play. Not every project will need a supplier, but the option is available due to the rising prevalence of offshoring solutions, technological tools and parts of development.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

During the Sprint Planning Phase of Scrum, the Scrum Master and Product Owner communicate to their team which metrics are to be kept track of and the procedures of doing so in order to build historical data for future forecasts, and project analysis (Measurement and Analysis SG1 SP1 SP2 SP3 SP4). They must be kept to a minimum to avoid any disruption to the speedy process, made faster with disciplined use of TFS or other alternatives. When it comes to gathering data to be used as historical reference for future projects, documentation is not needed or performed. The design artifacts resulting from all the iterations are kept in proper backup and used as necessary. Technical solutions and possible alternatives and scenarios are selected to begin development (Technical Solution SG1 SP1 SP3 SP4). Risks to process, project and product are assessed are determined using both (Risk Management SG1 SP1) and (Project Planning SG2 SP2) and are categorized before start of development as well. The costs of the project are estimated in more detail (Project Planning SG1 SP4), and how data will be managed throughout the sprints and after project completion (Project Planning SG2 SP3). Also during the same Sprint Planning phase, the Scrum Master and their team should determine how the product will be integrated through reviewing all interface descriptions and planning how they will be integrated in the most seamless way possible. They agree upon the best metrics and processes to use to objectively evaluate the product and process (Process and Product Quality Assurance SG1 SP1 SP2).

Several small but very important practices have to be considered during the Daily Scrum meeting and the everyday work on tasks during the sprint cycle, which can last between 1 and 4 weeks depending on the project's length and complexity. Requirements are taken to the next level by incorporating the Requirements Development practices (Requirements Development SG1 SP3, Requirements Development SG2 SP1, Requirements Development SG3 SP2) during sprints to establish the requirements for the product and product-components that can be used as future baselines to better serve functional analysis. All versions of the product iteration must be kept as a form of documentation for product support (Technical Solution SG3 SP2). As mentioned earlier, if a portion of the project is offshored and must be integrated into the iteration of the current product, this is where the practices of (Supplier Agreement Management SG2 SP3 SP4) come in.

The metrics to be collected as planned for during the Sprint Planning are revisited during sprints (Measurement and Analysis SG2 SP1 SP2 SP3) in order to make the necessary measures, analyze them and store them as historical data. In the daily meeting, metrics are revised, kept clear and stored (Measurement and Analysis SG1 SP2 SP4), and the costs of the project are further refined and made more accurate (Project Planning SG1 SP4) along with defining project risks again if necessary and maintaining the ways of data collection and management (Project Planning SG2 SP2 SP3). Several risk management practices need to be carried out during the Daily Scrum meeting, and during development and testing. Sources of risk are determined during the Client and Product Owner meeting, and again during Sprint Planning and Daily Scrum meeting (Risk Management SG1 SP1). Afterwards, they are prioritized and categorized during the daily meetings. This is done in parallel with the project monitoring and control practice of monitoring the risks of the project (Process Monitoring and Control SG1 SP3), and corrective action is taken in the Sprint Review. Furthermore, the daily meetings are when plans are developed to mitigate rising risks (Risk Management SG2 SP1 SP2) and then these plans are executed during the sprint itself (Risk Management SG3 SP1 SP2).

During the Sprint Review and after the sprint ends, the development team, Scrum Master and Client meet to review the product iteration. For this, product integration practices must be incorporated (Product Integration SG3 SP3 SP4) to evaluate the assembled product and its components, and allow the Client to test that iteration. Configuration management practices (Configuration Management SG2 SP1 SP2) must be performed to track the Client's change requests across all iterations and maintain control on all items that fall to configuration tracking, and Burn Charts can perform this job well. And of course, project monitoring and control (Process Monitoring and Control SG2 SP2) is necessary to take any corrective actions deemed necessary to make sure the iteration works as intended and risks are safely deviated. If the Client is satisfied with the iteration, the product is deployed and maintained. If not, a new sprint begins, incorporating necessary changes and fixes.

After the product is successfully deployed, the Scrum Master and their team look back on all previous iterations with their challenges and inconveniences, and perform two very important goals to improve future project performances: they

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

develop a baseline integrity by creating configuration management records in the simplest way possible, including keeping different versions of the product at different completion stages (Configuration Management SG3 SP1), and they develop or upgrade their risk mitigation plans and put them in action before starting a new similar project.

4. EVALUATION AND DISCUSSION

In this section, we evaluate the model's logical standing and potential in a real world setting through evaluation by a formal CMMI appraiser and an official partner, and by comparison to previous work in literature.

4.1 Industry Experts' Evaluation

The model was analyzed in detail with two professional software quality assurance consultants, one of which is an official CMMI appraiser and the other an official partner. Both have experience incorporating elements from agile methodologies into CMMI and designing models to befit different cultures and work practices in different enterprises. These two professionals are independent of the data sample collected through closed questionnaire, as discussed previously in the Methodology section. They were asked to judge the performance of the model prototype based on its target users (SMEs), and the logic behind the choices of practices based on what the specific agile model Scrum does not directly address.

Both consultants commend the proposed model for its compactness and addressing of SMEs' limitations, which are often overlooked in literature in favor of large-scale mapping that would befit larger enterprises. They approve of the addition of Configuration Management (CM), Risk Management (RSKM), Measurement and Analysis (RSKM) (Particularly in Sprint Planning), and Technical Solution (TS) practices, and highlight a major detail other models often miss – improving process quality without forcing extra documentation, for the proposed models makes sure to encourage developers to log their work correctly with the correct required metrics and descriptions during the sprint itself. If it is done correctly, no extra documentation will be needed in the presence of automated solutions such as TFS, SVN, or GIT.

They wholly agree to the definition of specific Supplier Agreement Management (SAM) practices to the model, confirming its relevance to current industry practices and trends regarding partnering with external developers. Despite agreeing that Integrated Project Management (IPM) is very important to product quality, given the SMEs limited resources they agreed to its removal from SQA-SCRUM to maintain speedy iterations, for Integrated Project Management (IPM) requires elaborate documentation of plans regarding every aspect of the lifecycle, including the plan of requirements gathering, the plan of architectural design, the planning of various testing approaches, etc.

The consultants suggested that future versions of the model incorporate Risk Management as an ongoing set of practices across the entire Scrum life cycle, and Product Integration (PI) SG1 SP1 SP2 SP3 can be removed from the model without problems due to them being focused on integration strategy at the enterprise level. This is redundant in an SME environment and thus was done without it to maintain a streamlined simplicity. Process Monitoring and Control (PMC) should be kept at a minimum in the SME context as the enterprise does not need to do more than keeping backups of the ongoing versions and final project are needed. They approved of our inclusion of Process and Product Quality Assurance (PPQA) SG1 SP1 SP2, which was one of their previous suggestions, as a way to address how the processes and working product will be evaluated during the Sprint Review.

One consultant alluded to a common problem where new recruits to come to an enterprise and have no idea what the model of work is. Therefore, they suggested inclusion of specific practices from Enterprise Process Definition (OPD) and Enterprise Process Focus (OPF) in future improvements to the model in order to set the ground basis for the establishment and approval of a set of standard processes and their descriptions, and maintain them across all project teams in the enterprise.

4.2 Comparison with Previous Models

Direct mapping between CMMI process areas and agile methodologies: our proposed prototype strives to improve upon the models suggested in literature by Elshafey and Galal-Edeen [44], Łukasiewicz and Miler [45], and Farid et al. [11][18] the most closely due to their practicality. Comparing our model to Elshafey and Galal-Edeen's, SQA-SCRUM focuses on compactness to maintain small team numbers and agility in a realistic manner. Their work focuses on finding areas of similarities but not explicitly whether or not it would be realistic in an SME, especially without the hiring of any new personnel or an SQA team. Our model also does not import Six Sigma tools. However, the use of TFS is highly recommended to get the most

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

out of our model. Furthermore, Supplier Agreement Management (SAM) was incorporated successfully into our model to address the growing issue of offshoring, whereas it is absent from previous works.

Comparing our model to Łukasiewicz and Miler, there are some similarities in the logic used to suggest suitable practices for the proposed Scrum model. However, our algorithm considers CMMI V1.3 rather than V1.2, and only the practices that need no extra personnel, SQA team, or more documentation are used in the final model design. Łukasiewicz and Miler also created a CMMI-Scrum reference model combining both suggested and additional practices. They do however warn that additional practices pose a risk to agility, and around 25% of the practices they suggested were incompatible with SME culture and resources.

Comparing our model to Farid et al., who applied their mapped CMMI-practices-to-Scrum to six enterprises, they focused on the Project Management (PP), Project Monitoring and Control (PMC), and Risk Management (RSKM) process areas in particular. Their work proposed a CMMI-Scrum mapping solution and determined the level of compliance, whether it is absent, partially satisfied or fully satisfied. Our model takes it further by consciously incorporating practices that are absent or only partially satisfied into Scrum's model. Table 12 displays the percentage of specific CMMI practices incorporated into our model in comparison with Farid et al.'s, and Figure 3 further illustrates this comparison.

In the figure, the x-axis indicates each process area considered in both models, and the y-axis indicates percentage of each successfully-incorporated CMMI practice. Farid et al.'s model included the practices of Integrated Project Management (IPM), but we consciously removed them from our model, due to their expensive nature in terms of time and personnel for SMEs. Supplier Agreement Management (SAM) is not incorporated in their model, whereas it is an important addition in ours. This is highly important because it allows small teams to successfully take on large projects they could not beforehand.

Risk Management (RSKM) process area practices: Chevers et al. [25, 46] studied and identified (qualitatively and quantitatively) the ten most crucial process areas in the Canadian software industry defined RSKM as the number one process area necessary for process and software quality. Even in the absence of full-scale CMMI, the practices of this process area are strongly recommended to be incorporated in development methodologies regardless their culture being agile or otherwise.

Alharbi and Qureshi [40] present an interesting solution for Scrum's lack of early recognition and mitigation of risks to the software being developed. Referring to their validation method, they used cumulative analysis and surveying of the industry to determine the level of usefulness of their proposed 'Risk Register' solution, and 70% of their surveyed professionals agreed upon the importance of the process area incorporated with Scrum without loss of agility. This value is an indicator of the effectiveness of applying practices of Risk Management (RSKM) to the proposed SQA-SCRUM model. However, Alharbi and Qureshi placed the process area in whole into the Sprint Planning phase, not considering other times in the sprint's cycle that risks may need to be recognized and managed. SQA-SCRUM addresses this aspect, with no loss to Scrum's agility. Risk Management (RSKM) is considered during Sprint Planning, very briefly during the Sprint, and in the Daily Scrum Meeting.

5. LIMITATIONS

This study faced some limitations in its implementation due to the small size of the industry sample. Another limitation lies in time-consuming data collection due to prevalence of red tape in the industry; it takes a long time to gather information to be used for scientific research. To address this, we intentionally selected only personnel that works with quality assurance to minimize the risk of variation in answers and overviews. This proved to be beneficial to the homogeneity of the collected data, and minimized outliers. To ensure the reliability of the results, the answers provided by the respondents were double checked by the interviewer to further reduce any misunderstanding or faulty communication.

Another limitation is the small number of experts whose knowledge and expertise with CMMI implementation and appraisal was consulted to validate the model. These experts are not in abundance and are sought after in the industry.

Furthermore, this study and the proposed model focus on the second and third maturity levels of CMMI only, the reason for this being to maintain agile culture, and due to the majority of CMMI-appraised enterprises falling within levels 2 and 3.

6. CONCLUSION

SQA-SCRUM is a customized model prototype that honors Scrum's agility and sprint culture whilst incorporating necessary practices for improved software quality and process maturity.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The added practices allow SMEs – which form more than 80% of the world's software industry – to produce quality software capable of competing with high-end large enterprises without the need to hire new personnel and expand documentation. Our model is also the first to our knowledge to incorporate practices of Supplier Agreement Management (SAM) into a Scrum-based model. The model compares positively when analyzed against previous attempts in literature, and is commended by two quality assurance experts with deep knowledge of software industry trends. The incorporation of Risk Management (RSKM), Configuration Management (CM) Measurement and Analysis (MA), and Technical Solution (TS) practices without specific and elaborate documentation is considered a success in itself. Risk Management (RSKM) is given a lot of attention in particular, due to its effect on quality as studied and tested by many researchers and SCRUM addresses a shortcoming in previous literature by being revisited and incorporated at various points in the development life cycle rather than only the Sprint Planning Phase.

Future research and work involves incorporating Risk Management across the entire Scrum life cycle, and testing the model practically within SMEs, covering whole projects from start to finish and preferably across several countries, in order to develop a dataset of results that can be eventually used to create a more objective and quantitatively-managed model.

REFERENCES:

[1] Sánchez-Gordón, Mary-Luz, and Rory V. O'Connor. "Understanding the gap between software process practices and actual practice in very small companies." *Software Quality Journal* 24.3 (2016): 549-570.

[2] Iqbal, Javed, et al. "Software SMEs' unofficial readiness for CMMI®-based software process improvement." *Software Quality Journal* 24.4 (2016): 997-1023.

[3] Pino, Francisco J., Félix García, and Mario Piattini. "Software process improvement in small and medium software enterprises: a systematic review." *Software Quality Journal*16.2 (2008): 237-261.

[4] Tripathi, Nirnaya, et al. "Exploring processes in small software companies: a systematic review." *International Conference on Software Process Improvement and Capability Determination*. Springer, Cham, 2016.

[5]I. Sommerville, Software Engineering, 9th ed. Pearson, New York, 2011.

[6] Hoyle, David. *ISO 9000 Quality Systems Handbook-updated for the ISO 9001: 2015 standard: Increasing the Quality of an Organization's Outputs*. Routledge, 2017.

[7] Green Belt Body of Knowledge. (2018, 10/2/2018). *Green Belt Body of Knowledge - International Association for Six Sigma Certification.* Available: https://www.iassc.org/body-of-knowledge/green-belt-body-of-knowledge/

[8] What is Six Sigma? (2017, 1/12/7). *What is Six Sigma? - International Six Sigma Institute*. Available: https://www.sixsigma-institute.org/What_Is_Six_Sigma.php

[9] Team, CMMI Product. "CMMI® for Development, Version 1.3, Improving processes for developing better products and services." *no. CMU/SEI-2010-TR-033. Software Engineering Institute* (2010).

[10] Niazi, Mahmood. "A comparative study of software process improvement implementation success factors." *Journal of Software: Evolution and Process* 27.9 (2015): 700-722.

[11] Farid, Ahmed Bahaa, AS Abd Elghany, and Yehia Mostafa Helmy. "Implementing Project Management Category Process Areas of CMMI Version 1.3 Using Scrum Practices, and Assets." *International Journal of Advanced Computer Science and Applications (IJACSA)* 7.2 (2016).

[12]Giudice, D.L. (2015, 5/2/2017). *"The 2015 state of agile development."* Available: https://www.forrester.com/report/The+2015+State+Of+Agile+Development/-/E-RES122910

[13] Definition. (2005, 15/2/2017). *SMALL AND MEDIUM-SIZED ENTERPRISES (SMES) Definition*. Available: https://stats.oecd.org/glossary/detail.asp?ID=3123. Accessed: 2018

[14] Rad, Babak Bashari, AL-Ashmori Ammar, and Zahra Ahanin. "Software Process Enhancement Model for SMEs." *IJCSNS*17.7 (2017): 8.

[15] Omran, Ahmed. "AGILE CMMI from SMEs perspective." *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*. IEEE, 2008.

[16] Fowler, Martin, and Jim Highsmith. "The agile manifesto." *Software Development* 9.8 (2001): 28-35.

[17] Schwaber, Ken, Jeff Sutherland, and Mike Beedle. "The definitive guide to scrum: The rules of the game." *Recuperado de: http://www. scrumguides.*

org/docs/scrumguide/v1/scrum-guide-us. pdf (2013).

[18]Helmy, Y.M., Farid, A.B., Abd Elghany, A.S. "SIMPLIFYING CMMI VERSION 1.3 IMPLEMENTATION BY USING AGILE PRACTICES: AN EMPIRICAL STUDY." *International Journal of Intelligent Computing and Information Science (IJICIS)*, 2015.

[19] Alliance, Scrum. (2016, 5/2/2016). *"What is Scrum? An Agile Framework for Completing Complex Projects-Scrum Alliance."* Available: https://www. scrumalliance. org

[20] Gangji, Arif, and Bob Hartman. "Agile SCRUM For Denver Web Development." *Neon Rain Interactive* (2015).

[21] Fraenkel, B. (2015, 5/2/2016) *Agile for SMEs - Silicon Valley Software Group*. Available: from https://svsg.co/agile-for-smes/

[22] Khan, Muhammad Ishfaq, Muzaffar Ali Qureshi, and Qaisar Abbas. "Agile methodology in software development (SMEs) of Pakistan software industry for successful software projects (CMM framework)." *Educational and Network Technology (ICENT), 2010 International Conference on*. IEEE, 2010.

[23] Pressman, Roger S. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.

[24] Silva, Fernando Selleri, et al. "A reference model for agile quality assurance: combining agile methodologies and maturity models." *Quality of Information and Communications Technology (QUATIC), 2014 9th International Conference on the*. IEEE, 2014.

[25] Chevers, Delroy. "The Impact of CMMI Levels 2 and 3 Practices on Process Maturity in Canadian Software Development Firms." *International Journal of Engineering Science and Innovative Technology (IJESIT)* 3.4 (2014).

[26] Chrissis, Mary Beth, Mike Konrad, and Sandy Shrum. *CMMI guidlines for process integration and product improvement*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[27]Phillips, Mike. "CMMI® Version 1.3 and Beyond." (2010).

[28] Glazer, Hillel, et al. "CMMI or agile: why not embrace both!" (2008).

[29] Potter, Neil, and Mary Sakry. "Implementing SCRUM (agile) and CMMI together." *The Process Group-Post newsletter* 16.2 (2009): 1-6.

[30] Jakobsen, Carsten Ruseng, and Kent Aaron Johnson. "Mature Agile with a twist of CMMI." *Agile, 2008. AGILE'08. Conference*. IEEE, 2008.

[31] Index of CMMI. (2018, 1/3/2018). *Index of CMMI*. Available: http://hci-itil.com/CMMI/

[32] McFall, Donald, et al. "Software processes and process improvement in Northern Ireland." (2003).

[33] Richardson, Ita, and Christiane Gresse Von Wangenheim. "Guest editors' introduction: Why are small software organizations different?." *IEEE software* 24.1 (2007): 18-22.

[34] Liou, Jing-Chiou. "On Improving CMMI in an immature world of software development." *J. Inf. Sci. Eng.* 27.1 (2011): 213-226.

[35] Turner. (2007, 20/3/2018). *CMMI Mentoring Service.* Available: http://www.tcg.com/services-CMMI mentoringoverture.html?OVRAW=elements% 20.

[36] Turk, Dan, Robert France, and Bernhard Rumpe. "Limitations of agile software processes." *arXiv preprint arXiv:1409.6600*(2014).

[37] Giudice, D. L. "The 2015 State Of Agile Development." *Forrester Research, August* 3 (2015): 2105.

[38] Lina, Zhang, and Shao Dan. "Research on combining scrum with CMMI in small and medium organizations." *Computer science and electronics engineering (iccsee), 2012 international conference on*. Vol. 1. IEEE, 2012.

[39] Berssaneti, Fernando Tobal, and Marly Monteiro Carvalho. "Identification of variables that impact project success in Brazilian companies." *International Journal of Project Management* 33.3 (2015): 638-649.

[40] Alharbi, Eman Talal, and M. Rizwan Jameel Qureshi. "Implementation of risk management with SCRUM to achieve CMMI requirements." *International Journal of Computer Network and Information Security* 6.11 (2014): 20.

[41] Tosun, Ayse, Ayse Bener, and Burak Turhan. "Implementation of a software quality improvement project in an SME: a before and after comparison." *Software Engineering and Advanced Applications, 2009. SEAA'09. 35th Euromicro Conference on*. IEEE, 2009.

[42] Engdashet, Tatek, Ricardo J. Machado, and Dida Midekso. "Integrated Framework of Agile and CMMI: An Alternative Path towards Product Focused SPI for Small

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Companies." *Lecture Notes on Software Engineering* 4.1 (2016): 1.

[43] Kähkönen, Tuomo, and Pekka Abrahamsson. "Achieving CMMI level 2 with enhanced extreme programming approach." *International Conference on Product Focused Software Process Improvement*. Springer, Berlin, Heidelberg, 2004.

[44] Elshafey, Lamis Atia, and G. H. Galal-Edeen. "Combining CMMI and agile methods." *Proceedings on the 6th International Conference on Informatics and Systems*. 2008.

[45] Lukasiewicz, Katarzyna, and Jakub Miler. "Improving agility and discipline of software development with the Scrum and CMMI." *IET software* 6.5 (2012): 416-422.

[46] Chevers, Delroy A., et al. "Identifying key software development practices in the English-speaking Caribbean using the nominal group technique." *ACIS 2008 Proceedings*(2008): 94.

[47] Emery, Patrick. (2018, 1/2/2018) *"eXtreme CMMI."* Available: http://www.members.cox.net/cobbler/XCMMI.ppt

[48] McCaffery, Fergal, Minna Pikkarainen, and Ita Richardson. "Ahaa--agile, hybrid assessment method for automotive, safety critical smes." *Proceedings of the 30th international conference on Software engineering*. ACM, 2008.

[49] Amer, Sarah K., et al. "Empirical study on the effectiveness of adopting software quality frameworks in the Egyptian software industry." *Intelligent Computing and Information Systems (ICICIS), 2017 Eighth International Conference on*. IEEE, 2017.

[50] Schwaber, Ken, Sutherland, Jeff, Beedle, Mike. (2013, 2/2/2018) *"The definitive guide to scrum: The rules of the game."*. Available: http://www.scrumguides. org/docs/scrumguide/v1/scrum-guide-us. Pdf.

[51] MM1. (2018, 25/3/2018). *The Consultancy for Connected Business. Agile Development*. Available: https://mm1.com/about-us/methods/agile-development/

Table 1: Top 10 process areas selected based on literature and industry findings

Process Area	Process Area Category	Available Corresponding Scrum Practices
Project Planning (PP)	Project Management	Sprint planning Product backlog Sprint backlog development
Requirements Development (RD)	Engineering	Product backlog development Sprint backlog development Recurring client communication Daily Scrum meeting Sprint review
Project Monitoring and Control (PMC)	Project Management	Recurring client communication Daily Scrum meeting
Measurement and Analysis (MA)	Support	Daily Scrum meeting Metric collection during sprint
Technical Solution (TS)	Engineering	Sprint backlog Client feedback and review Sprint cycle/iteration Sprint planning Product increment/prototype
Configuration Management (CM)	Support	Sprint backlog Client feedback and review Sprint cycle/iteration Sprint planning Product increment/prototype
Product Integration (PI)	Engineering	Sprint cycle/iteration Product increment/prototype Sprint review Client feedback and review
Process and Product Quality Assurance (PPQA)	Support	Unaddressed by Scrum practices
Risk Management (RSKM)	Project Management	Daily Scrum meeting
Supplier Agreement Management (SAM)	Project Management	Unaddressed by Scrum practices

Table 2: Project Planning (PP) specific goals and practices mapping for SQA-SCRUM model

Goal	PP-SG1	PP-SG2	PP-SG2
Specific Practice	**Establish and maintain the planning parameters of the project** *SP4: Estimate the costs and effort*	**Develop project plan** *SP2: Identify project risks and analyze them*	**Develop project plan** *SP3: Determine how project data will be managed*
Level of Support	**Partial Support** Cost estimations vague Little documentation Frequent changes	**Partial Support** Project risks vague Mitigation plans absent from org's policy	**Partial Support** Metrics undetermined Project data management plan absent from org's policy
Proposed Positioning within Scrum	Product Owner meeting with the Client/Product Backlog development Sprint Planning Daily Scrum meeting	User stories development Sprint planning Sprint backlog development	User stories development Sprint planning Sprint backlog development

Table 3: Requirements Development (RD) specific goals and practices mapping for SQA-SCRUM model

Goal	RD-SG1	RD-SG2	RD-SG3
Specific Practice	**Develop customer requirements** *SP3: Develop the customer requirements*	**Develop product requirements** *SP1: Establish product and product-component requirements*	**Analyze and validate requirements** *SP2: Establish the required functional analysis*
Level of Support	**Partial Support** Customer requirements collected with no analysis	**Partial Support** Requirements collected without analysis No datasets to mine/forecast information for future projects	**No Support** Absent analysis of product purposes, functions and logical groupings Absent datasets and no use of historical data
Proposed Positioning within Scrum	Sprint cycle/iteration	Sprint cycle/iteration	Sprint cycle/iteration

Table 4: Project Monitoring and Control (PMC) specific goals and practices mapping for SQA-SCRUM model

Goal	**PMC-SG1**	**PMC-SG2**
Specific Practice	**Monitor project against plan** *SP3: Monitor risks to project*	**Make corrective actions** *SP2: Take corrective action as pre-planned when risks and deviations were assessed as possibilities*
Level of Support	**Partial Support** Little foresight Not following potential issues early on	**Partial Support** No planned corrective actions in case of risks
Proposed Positioning within Scrum	Development team communication/Daily Scrum meeting	Sprint review

Table 5: Measurement and Analysis (MA) specific goals and practices mapping for SQA-SCRUM model

Goal	**MA-SG1**	**MA-SG2**	**MA-SG2**
Specific Practice	**Align measurement and Analysis activities** *SP1: Establish measurement objectives* *SP2: Specify measures to be collected* *SP3: Specify procedures for data analysis* *SP4: Specify procedure for data collection and storage*	**Provide measurement results** *SP1: Collect measurement data*	**Provide measurement results** *SP2: Analyze measurement data* *SP3: Store data and analysis results*
Level of Support	**No Support** No defined parameters Therefore industry-wide metrics used	**Partial Support** Only Burndown charts used No quantitative metric use in any part of lifecycle	**No Support** No defined parameters and methodology Therefore industry-wide metrics used Absence of historical data
Proposed Positioning within Scrum	Sprint planning Daily Scrum meeting	Sprint cycle/iteration	Sprint cycle/iteration

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6: Technical Solution (TS) specific goals and practices mapping for SQA-SCRUM model

Goal	TS-SG1	TS-SG3
Specific Practice	**Select product-component solution** *SP1: Develop alternative solutions and selection criteria* *SP3: Improve operational concepts and possible scenarios* *SP4: Select product-component solutions*	**Implement the product design** *SP2: Develop documentation for product support*
Level of Support	**No Support** Vague definition of needed tools and software for projects No planning for future events while using necessary tools and technology	**Partial Support** No documentation of software and tools to heighten maintainability and usability No historical data
Proposed Positioning within Scrum	Sprint planning Spring backlog development	Sprint cycle/iteration

Table 7: Risk Management (RSKM) specific goals and practices mapping for SQA-SCRUM model

Goal	RSKM-SG1	RSKM-SG2	RSKM-SG3
Specific Practice	**Prepare for risk management** *SP1: Determine sources of risk and categorize them*	**Identify risks and analyze them** *SP1: Identify risks* *SP2: Evaluate categorize and set priorities to risks*	**Mitigate risks** *SP1: Develop plans for risk mitigation* *SP2: Implement plans for risk mitigation*
Level of Support	**No Support** Absent active risk categorization and recognition	**Partial Support** Little focus dedicated to active risk identification in any project	**No Support** No mitigation plans
Proposed Positioning within Scrum	Product Owner meeting with the client/Product Backlog development Development team communication/Daily Scrum meeting Sprint planning	Development team communication/Daily Scrum meeting	Development team communication/Daily Scrum meeting Sprint cycle/iteration Sprint retrospective

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Configuration Management (CM) specific goals and practices mapping for SQA-SCRUM model

Goal	CM-SG1	CM-SG2	CM-SG3
Specific Practice	**Establish baselines** *SP1: Identify all configuration items* *SP2: Establish a configuration management system* *SP3: Create baselines*	**Track and control changes** *SP1: Track change requests* *SP2: Control configuration items*	**Establish baseline integrity** *SP1: Establish configuration management records*
Level of Support	**No Support** Undefined process for clarifying configuration management items and baselines	**Partial Support** Ad-hoc response to client changes and management of items, finished and unfinished	**No Support** Absent historical data Repeat of configuration errors in new projects due to poor management
Proposed Positioning within Scrum	Product Owner meeting with the Scrum Master Sprint planning Sprint Backlog development Product Backlog development	Sprint review	Sprint retrospective

Table 9: Product Integration (PI) specific goals and practices mapping for SQA-SCRUM model

Goal	PI-SG2	PI-SG3
Specific Practice	**Ensure compatibility of interfaces** *SP1: Review interface descriptions for completeness* *SP2: Manage interfaces*	**Assemble product components and deliver final product** *SP3: Evaluate assembled product components* *SP4: Deliver the product and components*
Level of Support	**No Support** Absent descriptions of interfaces necessary for project completion	**Partial Support** Ad-hoc assembly of components after testing
Proposed Positioning within Scrum	Product owner meeting with the Scrum Master Sprint Planning Sprint Backlog development	Sprint review

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 10: Process and Product Quality Assurance (PPQA) specific goals and practices mapping for SQA-SCRUM model

Goal	**PPQA-SG1**
Specific Practice	**Objectively evaluate process and work products** *SP1: Objectively evaluate processes* *SP2: Objectively evaluate work products*
Level of Support	**Partial Support** Little to no use of quantitative metrics to measure and evaluate expected process, project and product results
Proposed Positioning within Scrum	Sprint Planning Sprint Backlog development

Table 11: Supplier Agreement Management (SAM) specific goals and practices mapping for SQA-SCRUM model

Goal	**SAM-SG1**	**SAM-SG2**
Specific Practice	**Establish agreements with supplier** *SP2: Select suppliers* *SP3: Establish agreements*	**Satisfy agreements with supplier** *SP3: Accept the acquired product* *SP4: Transition the product*
Level of Support	**No Support** No offshoring/off-location support	**No Support** No offshoring support No support for integrating offshored components
Proposed Positioning within Scrum	Product owner meeting with the Scrum Master Product Backlog development	Sprint cycle/iteration

Table 12: Comparison between specific CMMI practices incorporated into SQA-SCRUM and Farid et al.'s work.

Process Area	% of Process Area Incorporated (SQA-SCRUM)	% of Process Area Incorporated (Farid et al. [11, 18])
Project Planning (PP)	93%	83%
Requirements Development (RD)	33%	N/A
Project Monitoring and Control (PMC)	80%	90%
Integrated Project Management (IPM)	N/A	40%
Measurement and Analysis (MA)	75%	N/A
Configuration Management (CM)	86%	N/A
Technical Solution (TS)	82%	N/A
Risk Management (RSKM)	71%	14%
Product Integration (PI)	33%	N/A
Process and Product Quality Assurance (PPQA)	50%	N/A
Supplier Agreement Management (SAM)	71%	0%

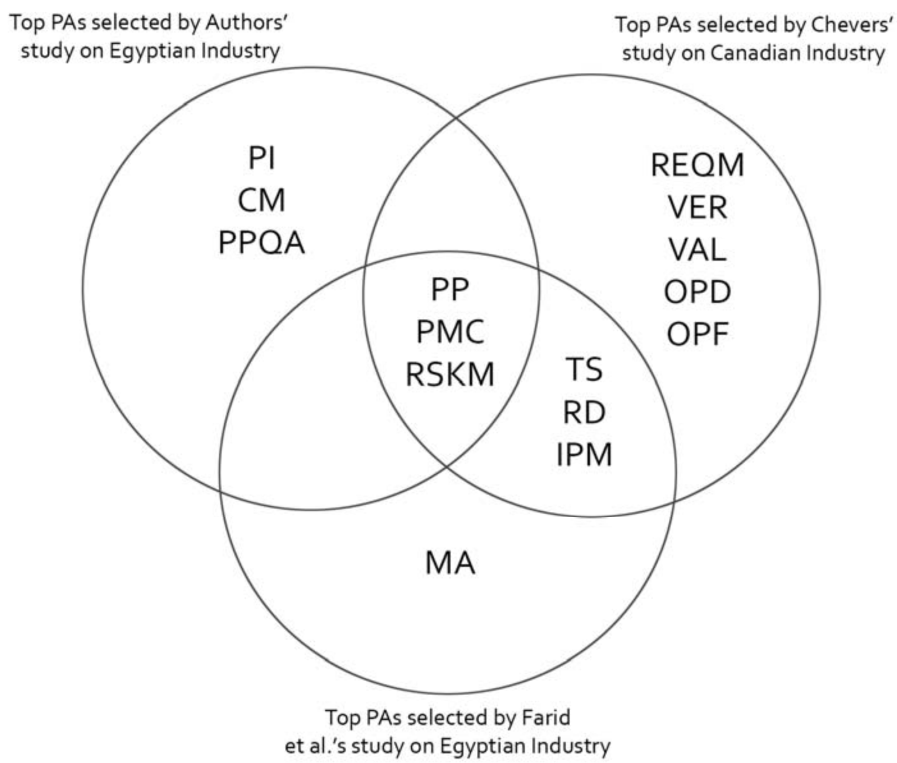

Figure 1: Venn Diagram indicating common PAs across three independent studies

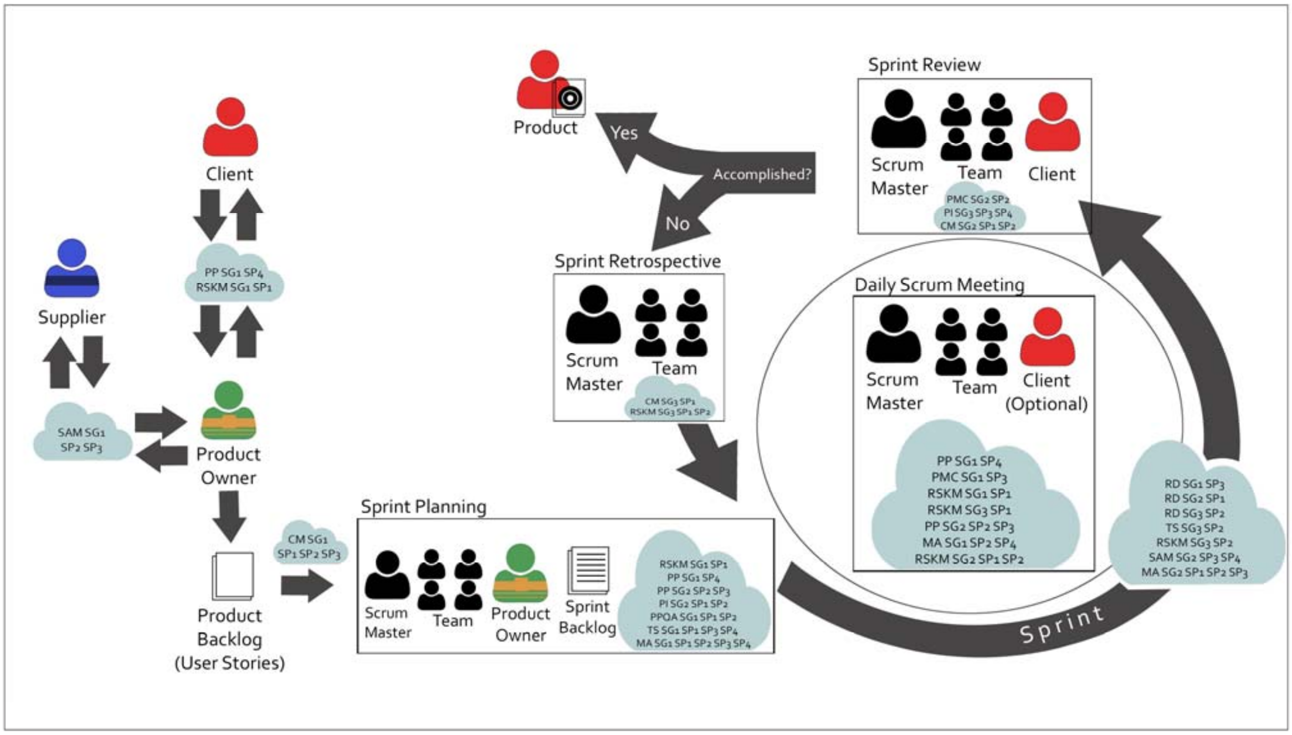

Figure 2: Illustrative representation of the proposed SQA-SCRUM model

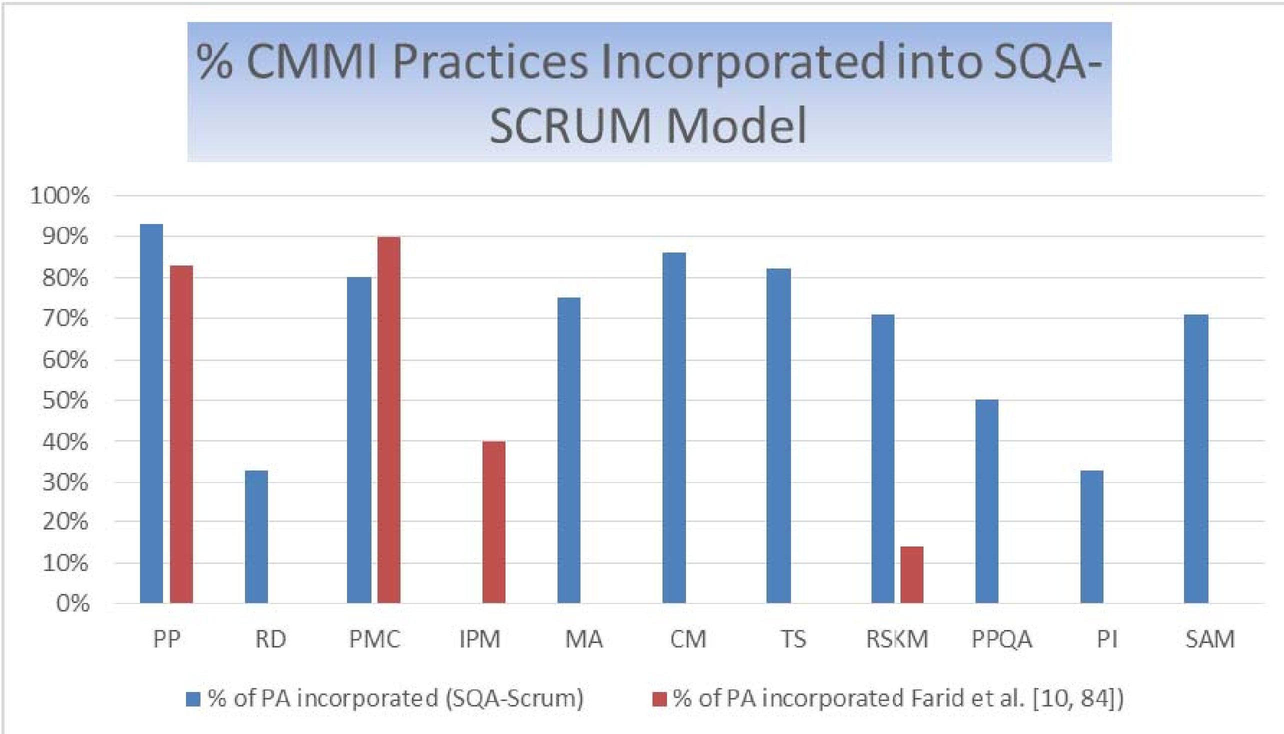

Figure 3: Percentage of CMMI Practices incorporated into SQA-SCRUM in comparison with Farid et al.'s model