Supplier Qualification Model (SQM): A Quantitative Model for Supplier Agreements Evaluation

Mohammed Omar[1], Yehia Helmy[2], Ahmed Bahaa Farid[3]
Information Systems Department, Faculty of Computers and Information, Helwan University[1]
Microsoft Company, Technology Solution Professional (TSP), 32511 Helwan, Egypt[2]
Management Information System Department, Faculty of Commerce and Business Administration[2]
Helwan University, 32511 Helwan, Egypt[2]
Information Systems Department, Faculty of Computers and Artificial Intelligence, Helwan University, 32511 Helwan, Egypt[3]
Information Systems Department, Faculty of Computers and Artificial Intelligence[3]
Beni-Suef University, 62511 Beni-Suef, Egypt[3]

Abstract*—Recently software outsourcing has increasingly widespread due to the valuable economical and technical benefits it introduced to the software development industry. Where the software development organizations adopt a third party to acquire a software project component (product, service). In the acquisition, process companies rely on the CMMI supplier agreement management (SAM) process area to select the potential supplier. Potential suppliers (vendors) are carefully selected through a dedicated process to ensure the delivery of high-quality and reliable services. Most of the published work in the context of how to evaluate and select the right supplier is based on a normal process with plain steps, nevertheless, no literature was reported to evaluate suppliers in a measurable way and select the potentials depending on a quantitative model. The purpose of this paper is to propose a practical quantitative model called the Supplier Qualification Model that enables the organizations to easily evaluate and select the potential suppliers through a measurable approach depends on monitoring and executing the SLAs of the SAM. The proposed model has been verified by implementing it through building an extension for one of the worldwide leading Agile management platforms according to Gartner (Microsoft Team Foundation Server). Multiple versions of the extension were implemented to target the major versions of Microsoft Team Foundation Server and validated by using them in 426 worldwide companies. This proves the suitability of the model to be used.

Keywords—Agile practices; vendor selection; CMMI; outsourcing; software acquisition; supplier agreement management; supplier selection; supplier evaluation

I. Introduction

In the last decade, the software development markets became more competitive and demanding, where it came to be crucial for organizations to invest more effort to improve their software processes to meet specific requirements. As a result, they had to follow a quality model to improve their software development process, increase their capability and maturity level, and become a benchmark for their competitors [1].

The Capability Maturity Model Integration (CMMI) Product Suite that was resealed in January 2002 paved the way for the organizations to improve their processes for acquisition, development, and sustainability [2]. Therefore, a large number of software development organizations (according to the CMMI Institute, 5000 businesses in 49 countries all over the world, and 1900 appraisal were conducted [3]) embark on it to improve their performance, develop higher-quality software, meet stakeholders satisfaction and achieve external validation [4].

As the software market mandate not only a product of high quality but also time and cost control [1], the organizations found that working Agile based on the CMMI model would allow continuous improvement and help them to reach the required maturity level [5].

Agile is a traditional software development approach that was created in 2001[6], and its methodologies have gained widespread acceptance among a large number of organizations as a quality-focused and highly collaborative mechanism to manage software development and improve the delivery process [7]. Agility is based on the idea that high-quality software can be developed by following a set of rules that allows continuous product improvement and testing depending on rapid feedback and testing [8].

Williams and Cockburn [9] defined the Agile team at the beginning to be as small as 50 people or fewer. However, Duka [10] showed that the larger teams had adopted Agile methodologies over the waterfall methods due to the great benefits of Agile practices. As well, in 2010 studies in the Agile journal [10] showed that 88% of companies with some of them have more than 10.000 employees are adopting Agile.

Different studies [11–13] were conducted to display the great benefits of working agile from the perspective of productivity and time management which leads at the end to customer satisfaction. They showed that agility can increase productivity to about 88%, improve cost efficiency by about 26%, and 41% time to market. Also, the 12th annual state of the Agile report [14] revealed that working Agile, 71 % manage change priorities, 66% improve the project visibility, gives 65% better business and IT alignment, speed the project delivery 62%, and 61% increase the team productivity. Besides, the Standish Group Chaos Report 2018 [15] results exhibited that working agile gives higher success and lower failure rates when compared to projects adopting the waterfall method (see Fig. 1 and Fig. 2).

Another reason that many organizations adopt Agile practices is that it permits the so-called distributed development (software outsourcing) [16], where an organization selects a third-party service provider (supplier) to execute a part of a software development project. The strategy of software outsourcing brought enormous benefits over the in-house development, where it reduced the operating costs, saved time, and gave immediate access to talented and higher-level IT professionals [17,18]. Watts Humphrey [19] implied that the quality of the software development process determines the quality of the produced software, and in the case of outsourced products the quality of software acquired from suppliers. This means that the selection process of the suppliers should be done according to specific and carefully calculated criteria.

Since the process to manage and control outsourced development projects is complicated and faces obstacles as in-house projects. Agile organizations had to adopt a specific process to manage the relationship between them and the service provider to be able to identify and select the potential suppliers. For an Agile organization that needs to be compliant with the CMMI supplier agreement management (SAM) process area, it must demonstrate an explicit commitment to establish a process-based confirmation through people and should verify the execution of the process and validate the outcome of process execution through a measured result [2]. EM Soares et al. [20] mapped the relationship between the supplier agreement management process area and agile practices, where they developed a catalog of the best practices using the concepts of the agile methodologies to manage the software acquisition process in the Supplier Agreement Management (SAM) process area of CMMI-DEV. Neither EM Soares nor other studies in the literature to the best of our knowledge was reported to introduce a quantitative method that helps the organizations to differentiate between software development suppliers, to select the potential ones based on measured data, and to monitor the execution of service level agreements established between the stakeholders and service providers according to CMMI-DEV v.2 specific goals of SAM process area.

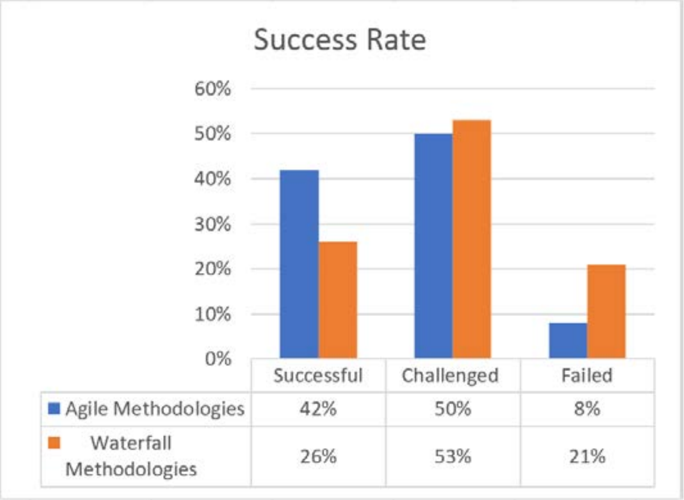


Fig. 1. Projects Success Rate.

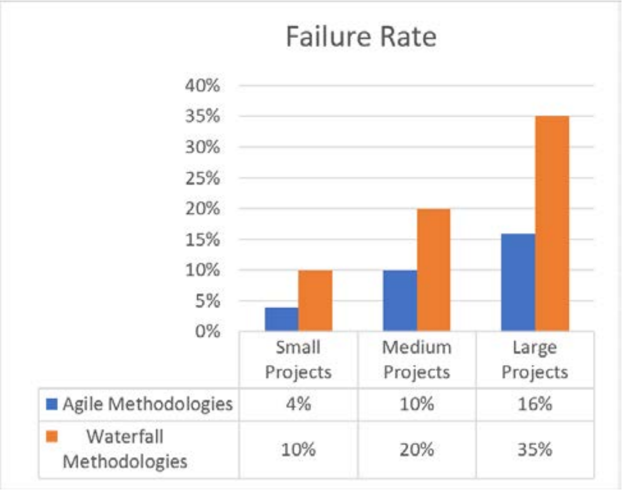


Fig. 2. Projects Failure Rate.

The main goal of this work is to propose a reference model that can be used to evaluate the supplier agreements quantitatively. And help the organizations to identify and select the potential suppliers. In the coming sections of this paper, a background about the supplier agreement management (SAM) process area and the selection process of potential suppliers is presented in a detailed manner. Then an introduction to the proposed model, its mathematical profile, and how it is in the evaluation process of suppliers is introduced. As well as, clarification of why the SQM model will be helpful for organizations to select the potential suppliers is demonstrated. Afterward, the verification and validation of the SQM model were discussed. Finally, a vision for further work in the future is proposed.

II. Background and Related Work

Supplier agreement management (SAM) is a CMMI process area at maturity level 2, it falls in the project management category and aims to manage the acquisition process of products and services from suppliers [21]. SAM is applicable only when organizations deal with a third party (supplier) to acquire products or services for software development. The specific goals of this process area were formulated to determine the acquisition type, to select suppliers, to establish and maintain supplier agreements, to execute the service level agreements (SLAs), to monitor the processes of the selected supplier, to accept the acquired product, and to ensure the successful delivery of products.

The supplier selection process comes directly after the identification of the acquisition type required from the vendor/supplier. Suppliers are selected based on an evaluation of their ability to meet specific predefined requirements and founded criteria. In the evaluation criteria, it is crucial to define the critical factors for the project such as costs, the geographical location of the supplier, quality services, supplier’s performance record, prior experience, etc. to be able to identify and select the potential suppliers, then to evaluate their proposals [20,22,23]. Afterward, the risks associated with each proposed supplier and his ability to perform work must be assessed through the evaluation of their prior experience and prior performance [22].

Selecting the right supplier who can deliver a high-quality and reliable product at the desired time and within the defined budget is not an easy task, and unfortunately, not much literature was dedicated to define a measurable method with quantitative terms to select potential suppliers. Nevertheless, authors always report the evaluation and selection process of suppliers in a plain way as mentioned above, for example,

G.O'regan [24] reported that in the identification process of suppliers, organizations may search or depend on recommendations from colleagues and previous work relationships to select the candidate suppliers. Then an evaluation team will be responsible to rate each one of them according to the founded criteria. Afterward, a shortlist of potential suppliers will present their proposals followed by a Q & A session, where the final decision will be made depending on this discussion.

Chaudhary et al. [25] stated that candidate suppliers can be chosen based on their capability or experience in delivering the desired product or services. After selecting the potential candidates, the final selection will be performed depending on their proposals, and the advantages and the disadvantages of each in the light of the factors of the predefined criteria.

EM Soares et al. [20] developed a catalog of the best practices to manage the software acquisition process by employing agile methodologies. They build up their work depending on the framework activities of Furtado [26], where they showed that these activities are compatible with the specific goals of the SAM process area in CMMI-DEV. They used this catalog to define finely the steps to evaluate and select suppliers based on the agile practices using the formal alternate evaluation method. Where they made a survey and collected information about suppliers to have a list, then they analyzed these data to limit this list to the most suitable candidates. Afterward, the potentials were selected depending on a checklist of the project requirements and the features of the supplier. Finally, the supplier who meets all the requirements or is closest to them will be selected.

Although these references give detailed and precise ways to evaluate and select suppliers, to the best of our knowledge, no study was reported in the literature to propose a method or model that facilitates the differentiation and evaluation of suppliers and the risk associated with each one of them in a measured way.

Since the identification of potential suppliers and the risk associated with each one of them is considered a crucial role in the success of a software acquisition process, the present study proposes a model that facilitates the evaluation and estimates the risk associated with each proposed supplier.

The developed model calculates the supplier's qualification through a quantitative score (SQScore) based on the established service level agreements (SLAs), enables organizations to use this quantitative score (SQScore) to distinguish between suppliers depending on their historical accumulative SQScores.

III. Introducing Supplier Qualification Model (SQM)

A. The Need for a Qualification Model for Selecting the Right Supplier

According to the CMMI SAM process area reference guide, many factors control the selection process of suppliers such as budget (money), quality, project business value, supplier reliability, compliance, the risk of failure, supplier experience, and previous work [21].

In the light of the above factors, when an organization must select a supplier from a list of professional suppliers, before the final selection, a set of questions should be answered to define the potential (right) supplier:

- Which supplier could deliver better business value when delivering the project?
- Does the supplier have experience working with the organization?
- What is the rate of associated failure of each supplier?
- What is the percentage of failure of each supplier?
- What is the risk of assigning the project to each supplier?

The supplier qualification model (SQM) will help the organizations to answer these questions, not only to select the supplier but also to monitor the selected supplier during the project execution for a better future qualification process.

B. Idea behind the SQM Model

According to SAM specific goal (SAM.SG.1), Establish Supplier Agreements [27], organizations established some service level agreements (SLAs). Each particular SLA mentioned in the supplier agreement has terms and conditions that suppliers should fulfill to satisfy this SLA. For example, an organization can establish a service level agreement to determine the level of quality for a specific number of works (Wn), the maximum amount of budget for a set of specified features, and/or the level of skilled resources that can handle certain types of work.

The proposed SQM is considering these factors depending on a set of SLAs execution measures. When the supplier fails to satisfy the terms and conditions of an agreed SLA, this is termed as a violation, hence the total number of violations is given as (Vn), while the percentile of deviation from the original goal of the SLA refers to the violation percentage (Vp). Since the degree of importance of an established SLA varies depending on the nature of the SLA and the outstanding project, in the present model, we were concerned to measure two SLA weights, (i) the compliance weight (ComplianceWt) which identifies how much compliant is a supplier with all the founded SLAs, and (ii) the risk weight (RiskWt) to estimate the associated risk of the supplier.

Organizations can control the weights of the SLAs, where in some cases they can decide to give a low score for any violation of a certain SLA regardless of the percentage of the violation, while for other SLAs the percentage of failure is more important. Since the two weights are mutually exclusive, the summation of both weights per SLA should be 100 points of weight.

C. Mathematical Profile of the SQM Model

The values of the ComplianceWt and the RiskWt are defined according to specific rules related to the project. This could differ from one SLA to another. As an example, for some SLAs, if the cost of failure to comply with a given SLA is very high, the compliance weight (CompilanceWt) should have a high value. Nevertheless, for other SLAs, the cost of failure will not be as high as long as the percentage of failure is low, Hence the SLA should have a high-risk weight (RiskWt) and low-Compliance weight (ComplianceWt) as will be explained in the following example.

If we assume that an organization is assigning a project to Supplier 1 and that 8 SLAs were established with each SLA has a number of work items (Wn). Each SLA has designated specific RiskWt points and ComplianceWt points (please note that the summation of each pair is 100). Each SLA has been violated a number of times (Vn), each time the violation is incurred with a specific percentage of violation (Vp). These values are depicted in the following set of matrices.

$$\begin{array}{cccccc} RiskWt & ComplianceWt & Vn & Wn & Vp \\ := \begin{bmatrix}20\\40\\15\\80\\50\\90\\30\\23\end{bmatrix} & := \begin{bmatrix}80\\60\\85\\20\\50\\10\\70\\77\end{bmatrix} & := \begin{bmatrix}6\\2\\1\\2\\2\\0\\4\\1\end{bmatrix} & := \begin{bmatrix}6\\7\\1\\4\\6\\5\\3\\2\end{bmatrix} & := \begin{bmatrix}10\ 40\ 20\ 5\ 12\ 7\\20\ 20\ 0\ 0\ 0\ 0\\30\ 0\ 0\ 0\ 0\ 0\\10\ 20\ 0\ 0\ 0\ 0\\3\ 10\ 0\ 0\ 0\ 0\\0\ 0\ 0\ 0\ 0\ 0\\3\ 10\ 20\ 30\ 0\ 0\\10\ 0\ 0\ 0\ 0\ 0\end{bmatrix} \end{array}$$

Hence, if the number of SLAs (n) is given as,

$$n := \mathrm{rows}(\mathrm{Wt}) = 8 \tag{1}$$

moreover, the index (i) is,

$$i := 0..n-1 \tag{2}$$

The SLA Risk Weight Index (SlaRiskWI) and the SLA Compliance Weight Index (SLACWI) that measure the importance of each SLA relative to the whole project can be calculated using equations (3) and (4). For both weight indices, the total value of each is 100.

$$\boldsymbol{SlaCWI}_i := \boldsymbol{ComplianceWt}_i \Big/ \sum_{i=0}^{n-1} \boldsymbol{ComplianceWt}_i * 100 = \begin{bmatrix}17.699\\13.274\\18.805\\4.425\\11.062\\2.212\\15.487\\17.035\end{bmatrix} \tag{3}$$

$$\boldsymbol{SlaRiskWI}_i := \boldsymbol{RiskWt}_i \Big/ \sum_{i=0}^{n-1} \boldsymbol{RiskWt}_i * 100 = \begin{bmatrix}5.747\\11.494\\4.31\\22.989\\14.368\\25.862\\8.621\\6.609\end{bmatrix} \tag{4}$$

Using the above equations (3 & 4), for (I = 3) the SlaRiskWI$_3$, which is the SLA risk percentage for the fourth SLA out of all risk weights for the whole project) will be 22.989%, and the SlaCWI$_3$, which is the SLA compliance percentage of the fourth SLA out of all compliance weights for the whole project is 18.805%. Then using the violation percentage (Vp), the Violation Severity (VS) can be calculated as follows:

$$\boldsymbol{VS} = \sum_{\boldsymbol{r=0}}^{\boldsymbol{cols(Vp)-1}} \frac{\boldsymbol{Vp}^{\langle r \rangle}}{\boldsymbol{100}} = \begin{bmatrix}0.68\\0.4\\0.3\\0.3\\0.13\\0\\0.63\\0.1\end{bmatrix} \tag{5}$$

Dividing the violation severity (VS) by the number of work gives the average violation percentage (Average Compliance) for all work assigned to the supplier (see equation (6)).

$$\boldsymbol{AverageCompliance}_i := \frac{\boldsymbol{VS_i}}{\begin{array}{c}\boldsymbol{if(Wn_i>0)}\\ \boldsymbol{||Wn_i}\\ \boldsymbol{else}\\ \boldsymbol{||1}\end{array}} = \begin{bmatrix}0.113\\0.057\\0.3\\0.075\\0.022\\0\\0.21\\0.05\end{bmatrix} \tag{6}$$

1) Evaluating Risk vs. Compliance: While using Average Compliance is helpful in many cases, using this measure only for qualifying suppliers, could be misleading. That is why we need in some cases to have an additional measure that considers risk associated with each supplier. For instance, If we assumed a case study of two suppliers, supplier A and supplier B, have the same violation severity, and supplier B has fewer number of violations than supplier A., In this case, the number of violations of supplier B (who has fewer violations) will indicate how much risk is associated with this supplier. For more clarification let us assume that both suppliers are assigning five projects and their violation percentages were as follows:

Vp Supplier A = [30,30,30,30,30] &

Vp Supplier B = [0,0,0,75,75].

Using equations (5) and (6), the violation severity of both suppliers is 150, and their average compliance is 30 respectively. This means that both suppliers have the same failure percentage for the whole assigned work.

For supplier A, since the violation percentage of each project is small, the risk is therefore low. On the other hand, for supplier B, nevertheless, he assigned three projects successfully, the associated risk would be high because of the high violation percentage of the last two elements of violated projects array (75,75).

Based on the numbers of the original example of (5) & (6), the average violation percentage of violated work (which is the AverageRisk) will be calculated by dividing the violation severity (VS) by the number of violations (see equation 7).

$$\boldsymbol{AverageRisk_i} \coloneqq \frac{\boldsymbol{VS_i}}{\begin{matrix}\boldsymbol{if(Vn_i>0)}\\ \boldsymbol{||Vn_i}\\ \boldsymbol{else}\\ \boldsymbol{||1}\end{matrix}} = \begin{bmatrix}0.113\\0.2\\0.3\\0.15\\0.065\\0\\0.158\\0.1\end{bmatrix} \qquad (7)$$

Equations 6 and 7 calculate the average violation percentage for the whole assigned and violated work. Nevertheless, the success factor of not making compliance failures, and not incurring risk factors should be rewarded for each supplier. Therefore, the success value must be considered by calculating the average compliance success, and the average risk success as follows:

$$AverageComplianceSuccess_i \coloneqq 1 - AverageCompliance_i = \begin{bmatrix}0.887\\0.943\\0.7\\0.925\\0.978\\1\\0.79\\0.95\end{bmatrix} \qquad (8)$$

$$AverageRiskSuccess_i \coloneqq 1 - \ AverageRisk_i = \begin{bmatrix}0.887\\0.8\\0.7\\0.85\\0.935\\1\\0.843\\0.9\end{bmatrix} \qquad (9)$$

2) Calculating SQScore for each SLA: The Compliance Score for each SLA (SlaComSc) is calculated by subtracting the product of the *AverageCompliance$_i$* and *SlaCWI$_i$* of the violations, from the product of *AverageComplianceSuccess$_i$* and *SlaCWI$_i$* (see equation (10)). Based on this equation, when the supplier achieves successes more than failures, the SlaComSc$_i$ will be a matrix of positive numbers otherwise, it will be negative numbers.

$$SlaComSc_i \coloneqq (AverageComplianceSuccess_i * SlaCWI_i) - (SlaCWI_i * AverageCompliance_i) = \begin{bmatrix}13.687\\11.757\\7.522\\3.761\\10.583\\2.212\\8.982\\15.332\end{bmatrix} \qquad (10)$$

Similarly, the Risk Score for each SLA (SlaComSc) is calculated using equation 11.

$$SlaRiskSc_i \coloneqq (SlaRiskWI_i * AverageRiskSuccess_i) - \ (SlaRiskWI_I * AverageRisk_i) = \begin{bmatrix}4\\6\\1\\1\\ \\2\\5\\5\end{bmatrix} \qquad (11)$$

3) Calculating SQScore: Therefore, the Supplier Compliance Score for all SLAs and the Supplier Risk Score for all SLAs, per each supplier, will be calculated from the sum of the SlaComSc and the sum of the SlaRiskSc respectively as seen in equations 12 and 13.

$$\sum_{i=0}^{n-1} SlaRiskSc_i = 78.712 \qquad (12)$$

$$\sum_{i=0}^{n-1} SlaComSc_i = 73.837 \qquad (13)$$

Finally, the SQScore of the supplier will be the average of both SlaRiskSc and SlaComSc.

$$QScore \coloneqq \left(\sum_{i=0}^{n-1} SlaRiskSc_i + \sum_{i=0}^{n-1} SlaComSc_i\right)\Big/2 = 76.27 \qquad (14)$$

Equation 14 shows that the supplier scored a success rate of 76.274. Now we can say that using this model organizations will be able to differentiate easily between suppliers based on their historical projects.

4) Evaluating suppliers: If we assume that an organization has the following historical SQScore for six different suppliers:

Supplier 1	***Supplier 2***	***Supplier 3***	***Supplier 4***	***Supplier 5***	***Supplier 6***
90	76	50	33	55	80

The organization needs to categorize these suppliers into three ranges a high, medium, and low depending on their SQScores. Hence we have to calculate the standard deviation. To do so, firstly the mean is calculated by dividing the sum of suppliers score by the number of suppliers:

If n = index of the supplier:

$$meanSQScore\ (Average) \coloneqq \frac{\sum_{i=0}^{n-1} x_i}{n} = 64 \qquad (15)$$

Then calculate the difference from the mean for each supplier,

Supplier 1	***Supplier 2***	***Supplier 3***	***Supplier 4***	***Supplier 5***	***Supplier 6***
26	12	-14	-31	-9	16

Afterward, we get the variance from the sum of the square of each difference, then dividing by the number of suppliers.

$$Variance \coloneqq \frac{\sum_{i=0}^{n-1} difference_i^2}{n-1} = 462.8 \qquad (16)$$

Now the Standard Deviation (**σ**) is the square root of the variance.

$$\sigma \coloneqq \sqrt{Variance} = \mp 21.513 \qquad (17)$$

Adding equations 15, 16, and 17 we get:

$$\boldsymbol{\sigma} \coloneqq \sqrt{\frac{1}{n-1}\sum_{i=0}^{n-1}(\boldsymbol{x_i}-\boldsymbol{\bar{x}})^2} \qquad (18)$$

Using the standard deviation and the mean, suppliers 2, 3, 5, and 6 can be categorized as average suppliers (the period of values between the $meanSQScore \mp \boldsymbol{\sigma}$), while supplier 1 is above average, and supplier 4 is below average as illustrated in Fig. 3.

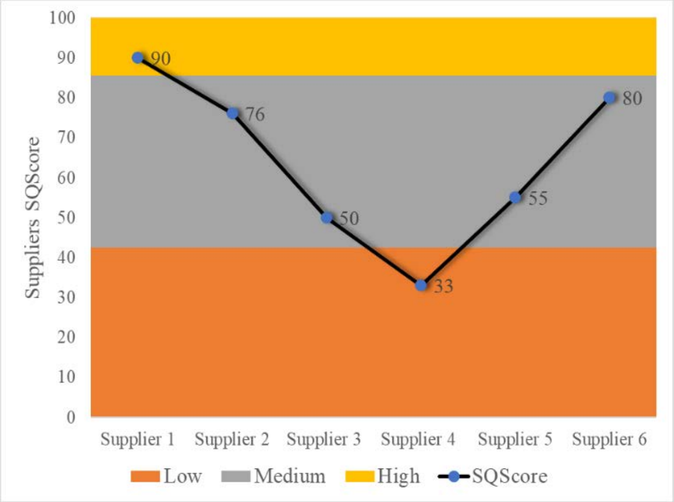


Fig. 3. The High, Medium, and Low Ranges of Suppliers SQscores.

IV. SQM Verification and Validation

A. Model Verification

To verify the applicability of the proposed SQ Model, an extension tool has been built to automate applying the proposed SQM for software teams. In this study, the extension is based on the Microsoft Team Foundation Server (TFS), which is one of the worldwide leading software engineering platforms according to Gartner [28].

The verification process was executed in two steps, (i) check the possibility of extending an agile tool (in this study is the TFS) to calculate the total SQscore of a supplier and of each established SLA using the proposed model, (ii) verify the possibility of extending the TFS client (Visual Studio) that allows organizations to create SLAs based on the SQ model, and monitor and track the SLA execution and the supplier SQ score.

The TFS Scrum process template provides organizations with the artifacts, processes, and workflows which are necessary to adopt the scrum method. The proposed TFS SLA Server Extension adds the following artifacts to the TFS Scrum process (i) SLA Configuration, (ii) SLA Violation Work-Items. The SLA Configuration will store the Service Level Agreements information such as SLA Risk Weight, SLA compliance weight, SLA threshold, and SLA percentage (see Fig. 4). Nevertheless, the SLA violation will store the violation information such as the violation percentage. Organizations can establish and maintain Service Level Agreements by creating as many SLA configuration items as needed to define flexible rules such as work deadline information and relation with other work. The SLA configurations are implemented dynamically to allow customizations that fit many different business scenarios. For example:

- Escalating work that in-progress for more than n number of hours/days.
- Escalating work with deadline configured on fields.
- Escalating all work related to a specific feature, epic, or product backlog items.
- Escalating work that takes more than n% of parents.
- Escalating assignment of specific work to a specific resource with a specific persona.

Once the organization created and activated SLAs for a particular project, the SLA is executed, and the SQscore is calculated by the TFS extension. Hence, the organization can monitor and track the execution and satisfaction of the SLAs.

The TFS SLA Client extension introduces a new capability to Visual Studio which allows organizations to monitor and track SLAs and measures the SQScore of a specific Supplier.

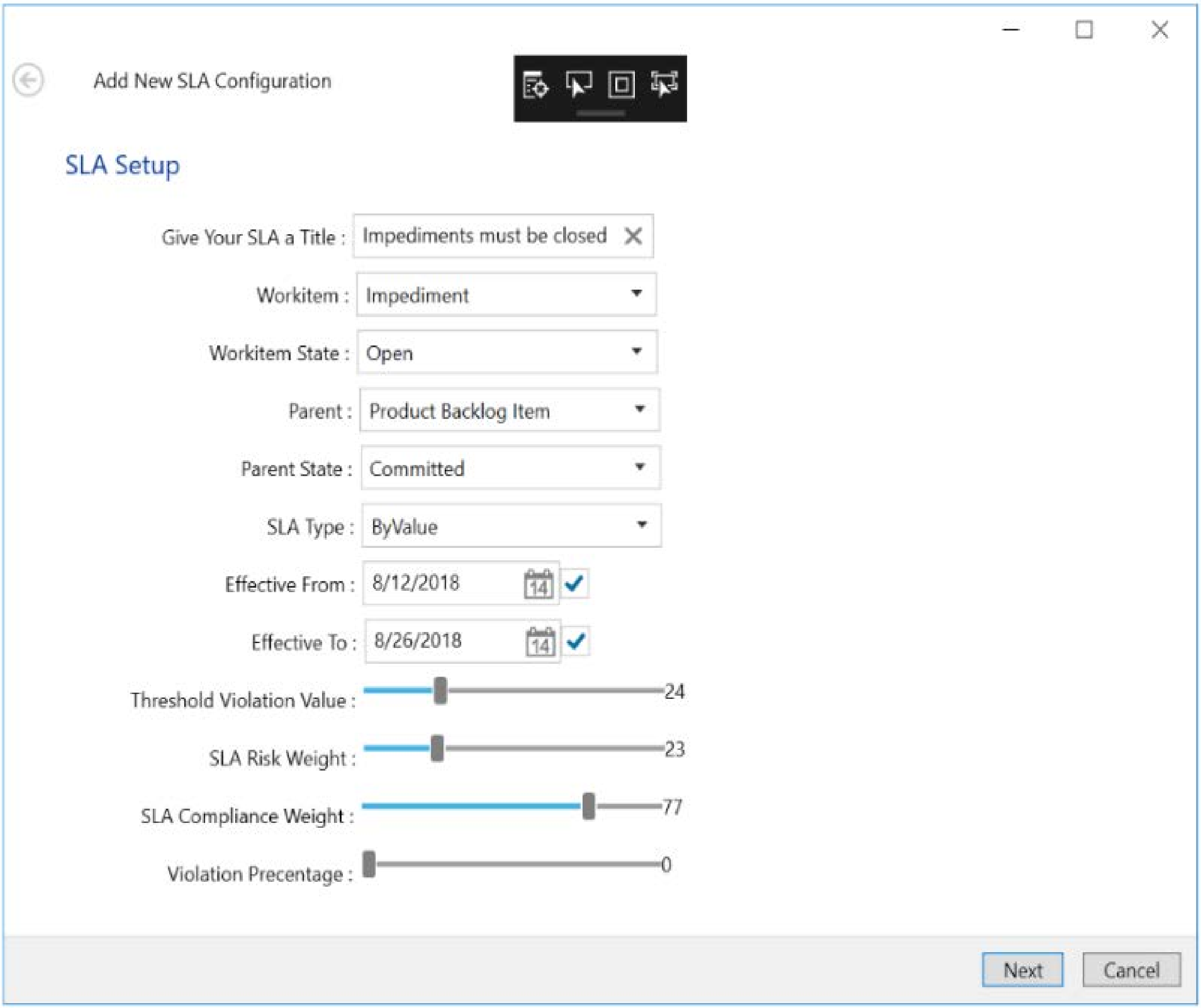
If Project X (see Fig. 5) with 8 SLAs has been established and executed by the supplier; each SLA has been violated several times, the extension calculates the Supplier Qualification Model Score by calculating the SQScore for each SLA using the model described above.

For example, SLA no. 320 has been violated 6 times. Therefore the Risk score is 4.44, the Compliance score is 13.69, and the SQScore is 9.1, and the supplier SQScore for the project is 76, as shown in (Fig. 5).

B. *Model Validation*

The TFS SLA Server and TFS SLA Client extensions have been published to Microsoft Visual Studio Gallery [29] and have scored 426 (for both versions) usages for the TFS SLA Server and 614 (for both versions) usages for the TFS SLA Client (check Ref. [29]) as illustrated in Fig. 6.

The SLA has been used by many organizations in the public and government sectors in multiple countries, which proofs the validity of the proposed SQM Model.



Add New SLA Configuration

SLA Setup

Give Your SLA a Title : Impediments must be closed

Workitem : Impediment

Workitem State : Open

Parent : Product Backlog Item

Parent State : Committed

SLA Type : ByValue

Effective From : 8/12/2018 ✓

Effective To : 8/26/2018 ✓

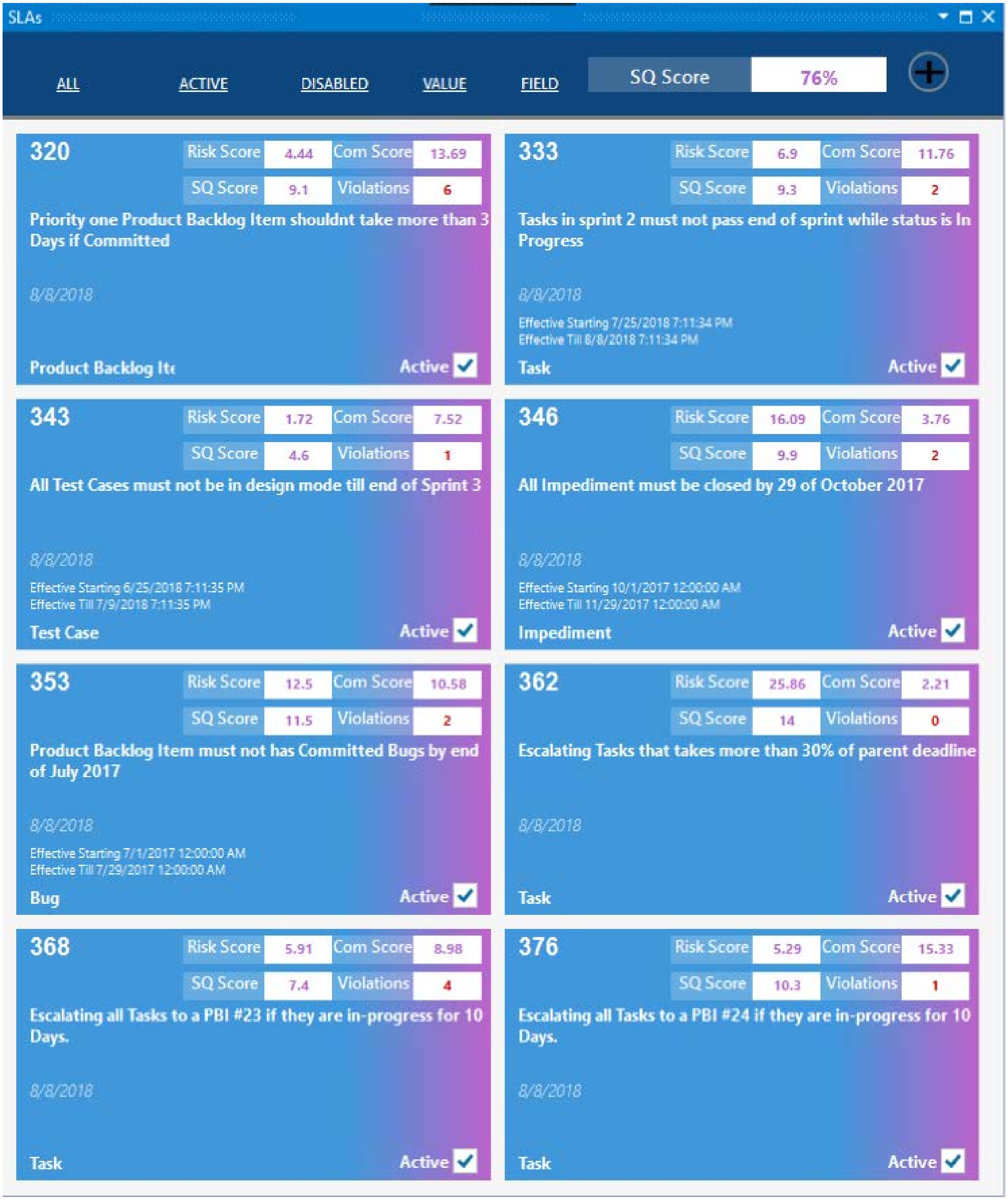
Threshold Violation Value : 24

SLA Risk Weight : 23

SLA Compliance Weight : 77

Violation Precentage : 0

Fig. 4. SLA Configuration.



SLAs

SQ Score 76%

320
Risk Score 4.44 Com Score 13.69
SQ Score 9.1 Violations 6
Priority one Product Backlog Item shouldnt take more than 3 Days if Committed
8/8/2018
Product Backlog Ite
Active ☑

333
Risk Score 6.9 Com Score 11.76
SQ Score 9.3 Violations 2
Tasks in sprint 2 must not pass end of sprint while status is In Progress
8/8/2018
Effective Starting 7/25/2018 7:11:34 PM
Effective Till 8/8/2018 7:11:34 PM
Task
Active ☑

343
Risk Score 1.72 Com Score 7.52
SQ Score 4.6 Violations 1
All Test Cases must not be in design mode till end of Sprint 3
8/8/2018
Effective Starting 6/25/2018 7:11:35 PM
Effective Till 7/9/2018 7:11:35 PM
Test Case
Active ☑

346
Risk Score 16.09 Com Score 3.76
SQ Score 9.9 Violations 2
All Impediment must be closed by 29 of October 2017
8/8/2018
Effective Starting 10/1/2017 12:00:00 AM
Effective Till 11/29/2017 12:00:00 AM
Impediment
Active ☑

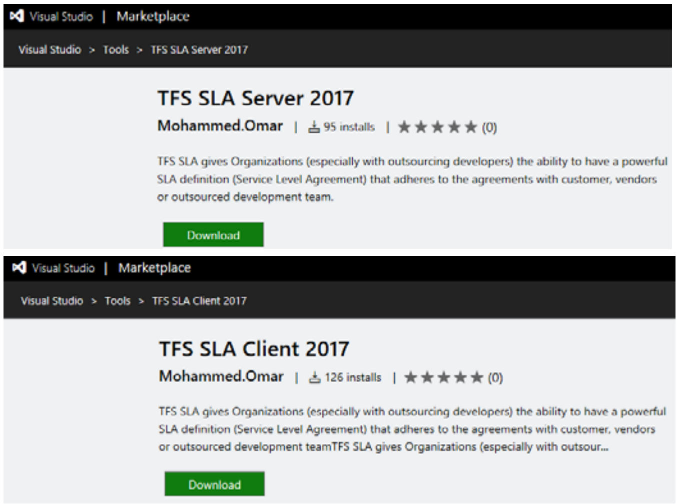
353
Risk Score 12.5 Com Score 10.58
SQ Score 11.5 Violations 2
Product Backlog Item must not has Committed Bugs by end of July 2017
8/8/2018
Effective Starting 7/1/2017 12:00:00 AM
Effective Till 7/29/2017 12:00:00 AM
Bug
Active ☑

362
Risk Score 25.86 Com Score 2.21
SQ Score 14 Violations 0
Escalating Tasks that takes more than 30% of parent deadline
8/8/2018
Task
Active ☑

368
Risk Score 5.91 Com Score 8.98
SQ Score 7.4 Violations 4
Escalating all Tasks to a PBI #23 if they are in-progress for 10 Days.
8/8/2018
Task
Active ☑

376
Risk Score 5.29 Com Score 15.33
SQ Score 10.3 Violations 1
Escalating all Tasks to a PBI #24 if they are in-progress for 10 Days.
8/8/2018
Task
Active ☑

Fig. 5. TFS SLA Client Extension.


Visual Studio | Marketplace
Visual Studio > Tools > TFS SLA Server 2017
Download
Visual Studio | Marketplace
Visual Studio > Tools > TFS SLA Client 2017
Download

TFS SLA Server 2017

Mohammed.Omar | 95 installs | ★★★★★ (0)

TFS SLA gives Organizations (especially with outsourcing developers) the ability to have a powerful SLA definition (Service Level Agreement) that adheres to the agreements with customer, vendors or outsourced development team.

TFS SLA Client 2017

Mohammed.Omar | 126 installs | ★★★★★ (0)

TFS SLA gives Organizations (especially with outsourcing developers) the ability to have a powerful SLA definition (Service Level Agreement) that adheres to the agreements with customer, vendors or outsourced development teamTFS SLA gives Organizations (especially with outsour...

Fig. 6. TFS SLA usages [29].

V. Future Work

The proposed model is an efficient suit for an organization that wants to qualify their vendors and categorize them into categories. It works effectively for the organizations that have vendors with previous data, nevertheless, in the case of new vendors, organizations will not have previous data, every organization will have its own SLAs. To help overcome this problem, future research can define and unify a set of common SLAs for vendors who want to get appraised and qualified by the proposed SQM. A global system that uses predefined SLAs should be created to give appraisal and SQScore for each vendor, therefore organizations could use this global SQScore to validate vendors.

VI. Conclusion

Evaluating and selecting service suppliers is a hard task especially that up to now, no method or model was reported to evaluate suppliers in a quantified manner according to the CMMI's SAM process area and its corresponding specific goals. The present work has proposed a quantitative model to help teams and organizations to evaluate and select potential suppliers based on the previous history of each one of them. The proposed model has introduced some measures to evaluate the success and failure of each supplier of being compliant with the agreed SLAs. Besides, it presented some other measures to evaluate the risk of dealing with each supplier. Taking these two factors in consideration, a set of mathematical equations are used to come up with the SQScore which could help in categorizing the suppliers into three categories (low, medium, and high performing suppliers) based on the average value of the SQScores of all suppliers and the corresponding Standard Deviation. This quantitative model has been verified by implementing it in a practical extension tool. Furthermore, it has been validated by using this tool in about 600 companies which proves it is suitable.

Acknowledgment

We would like to acknowledge Eng. Hosam Kamel from Microsoft for his help during building the first version of the TFS SLA extension.

References

[1] F. Soares, S. Meira, An Agile Maturity Model for Software Development Organizations, in IARIA XPS Press, Venice, Italy,2013. https://www.iaria.org/conferences2013/ICSEA13.html.

[2] D. O'Neill, CMMI The Agile Way, CROSS TALK J. Def. Softw. Eng. 29 (2016) 1–9.

[3] M. Chaudhary, A. Chopra, CMMI for Development: Implementation Guide, Apress, 2017. //www.springer.com/de/book/9781484225288 (accessed July 31, 2018).

[4] L.S. Plácido, L. Araújo, S. Sampaio, M. Marinho, R. Godoi, A. Luna, Q. B. da Silva, Organizational Motivations for Adopting CMMI-based Software Process Improvement: An Extended and Updated Systematic Review, in: Fortaleza, Brasil, n.d.

[5] P. Mcmahon, P. Pem Systems, Extending agile methods: A distributed project and organizational improvement perspective, (2005).

[6] History: The Agile Manifesto, (n.d.). http://agilemanifesto.org/history.html (accessed July 17, 2018).

[7] Scott W. Ambler, Scaling Agile: An Executive Guide, (2010). ftp://public.dhe.ibm.com/software/emea/de/rational/agility/Scaling-Agile--An-executive-guide.pdf .

[8] T. Dingsøyr, S. Nerur, V. Balijepally, N.B. Moe, A decade of agile methodologies: Towards explaining agile software development, J. Syst. Softw. 85 (2012) 1213–1221. doi:10.1016/j.jss.2012.02.033.

[9] L. Williams, A. Cockburn, Guest Editors' Introduction: Agile Software Development: It's About Feedback and Change, Computer. 36 (2003) 39–43. doi:10.1109/MC.2003.1204373.

[10] D. Duka, Agile Experiences in Software Development, in: Opatija, Croatia, 2012. https://www.balcon-project.eu/events/mipro2012.

[11] K. Beck, Manifesto for Agile Software Development, (2001). http://agilemanifesto.org/principles.html.

[12] David F. Rico, What is the ROI of agile vs. traditional methods? An analysis of XP, TDD, Pair Programming, and Scrum, (2008). http://davidfrico.com/rico08b.pdf.

[13] Version One ,3rd Annual 'State of Agile Development' Survey results., (2008). http://www.versionone.com.

[14] VersionOne, 12th Annual State of Agile Report, VersionOne 12th Annu. State Agile Rep. (2018). https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report.

[15] CHAOS Report: Decision Latency Theory (2018) Package - The Standish Group, (n.d.). https://www.standishgroup.com/store/services/10-chaos-report-decision-latency-theory-2018-package.html (accessed July 22, 2018).

[16] D. Batra, Modified Agile Practices for Outsourced Software Projects, Commun ACM. 52 (2009) 143–148. doi:10.1145/1562164.1562200.

[17] 10 reasons why outsourcing software development works!, (n.d.). https://www.orientsoftware.net/software-outsourcing/why-outsourcing/ (accessed July 31, 2018).

[18] S. Gantman, IT Outsourcing in the Public Sector: A Literature Analysis, in: Glob. Sourc. Serv., World Scientific/NOW Publishers, 2016: pp. 83–134. doi:10.1142/9789813109315_0003.

[19] W. Humphrey, Managing the Software Process, Addison-Wesley Publishing Company,Inc, 1989.

[20] E.M. Soares, S.R.B. Oliveira, M.C. dos Santos, A.M.L. Vasconcelos, A Catalog of Best Practices about Supplier Agreement Management and Agile Practices, in: IARIA XPS Press, Rome, Italy, 2016. https://www.iaria.org/conferences2016/ICSEA16.html.

[21] CMMI for Development, Version 1.3, Software Engineering Institute (SEI), 2010. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf.

[22] CMMI - Supplier Agreement Management (SAM) Process Area, CMMi - Supplier Agreem. Manag. SAM. (n.d.). http://www.software-quality-assurance.org/cmmi-supplier-agreement-management.html (accessed August 10, 2018).

[23] Guidelines for Supplier Agreement Management in CMMI, CMMI Consult. Blog. (2013). https://www.cmmiconsultantblog.com/cmmi-faqs/guidelines-for-supplier-agreement-management-in-cmmi/ (accessed August 10, 2018).

[24] G. O’Regan, Supplier Selection and Management, in: G. O’Regan (Ed.), Concise Guide Softw. Eng. Fundam. Appl. Methods, Springer International Publishing, Cham, 2017: pp. 123–130. doi:10.1007/978-3-319-57750-0_8.

[25] M. Chaudhary, A. Chopra, CMMI DESIGN, in: CMMI Dev. Implement. Guide, Apress, 2017: pp. 9–69. www.springer.com/de/book/9781484225288 (accessed October 1, 2018).

[26] J. C. C. Furtado, Spider-ACQ: An Approach to the systematization of Products and Services Acquisition Process Based Quality Multimodels., Masters Dissertation, PPGCC in Federal University of Pará (UFPA), 2011.

[27] Supplier Agreement Management (SAM) (CMMI-DEV), (n.d.). https://www.wibas.com/cmmi/supplier-agreement-management-sam-cmmi-dev (accessed July 24, 2018).

[28] Magic Quadrant for Enterprise Agile Planning Tools, (2017). https://www.gartner.com/doc/reprints?id=1-3Z6ZBTD&ct=170501&st=sb.

[29] TFS SLA Server 2017 - Visual Studio Marketplace, (n.d.). https://marketplace.visualstudio.com/items?itemName=MohammedOmar.TFSSLAClient2017 (accessed August 16, 2018).