Towards Agile Implementation of Test Maturity Model Integration (TMMI) Level 2 using Scrum Practices

Ahmed B. Farid
Information Systems Dept,
Faculty of Computers and
Information, Helwan University,
Cairo, Egypt

Enas M. Fathy
Information Systems Dept,
Faculty of Computers and
Information, Helwan University,
Cairo, Egypt

Mahmoud Abd Ellatif
Information Systems Dept,
Faculty of Computers and
Information, Helwan University,
Cairo, Egypt

***Abstract*—The software industry has invested the substantial effort to improve the quality of its products like ISO, CMMI and TMMI. Although applying of TMMI maturity criteria has a positive impact on product quality, test engineering productivity, and cycle-time effort. But it requires heavy software development processes and large investments in terms of cost and time that medium/small companies do not deal with it. At the same time, agile methods deal with changing requirements and continuous delivery of valuable software by short time iterations. So that the aim of this paper is to improve the testing process by applying a detailed mapping between TMMI and Scrum practices and verifying this mapping with a study providing empirical evidence of the obtained results. The research has been evaluated on a sample of 2 large-sized TMMI certified companies. In conclusion, the experimental results show the effectiveness of using this integrated approach compared to other existing approaches.**

Keywords—Agile software development; Scrum; TMMI; Software Testing; CMMI

I. Introduction

As the software industry is evolving rapidly but it still far away from zero defects software, so the software industry focused on developing the development process [1].

"Despite the fact that testing often accounts for at least 30-40% of the total project costs" [1], most of the widely used software industry standards like Capability Maturity Model Integration (CMMI) only give limited attention to the testing process improvement, so that the testing community created a TMMI [1].

"The TMMI is a detailed model for test process improvement and is positioned as being complementary to the CMMI." [1].

TMMI aims to help software organization to improve the software quality by preventing defects and save time and cost [2].

more and more software organizations moving from the waterfall model to more agile models like Scrum, as the benefits of agility including faster delivery to customer, flexibility with requirement changes and higher software quality [3].

The paper is organized as follows: Section 2 presents the background and related research of agile software development and TMMI. Section 3 describes the research problem. Section 4 and 5 describe the research goals, approach, and assumptions. Section 6 shows the theoretical mapping of the two approaches. Section 7 shows overall results of the theoretical mapping. Section 8 describes the background of the analyzed study. Section 9 summarizes the assessment results from the study. Section 10 the results of this study are discussed. The last section concludes the paper with the key findings, research limitations and future work related to the TMMI and agile integration.

II. Overview

A. TMMI

Capability Maturity Model Integration (CMMI) includes few testing process areas like verification and validation with lack of details so Test Maturity Model Integration (TMMI) has been created to cover most of testing processes [4].

TMMI includes best practices of testing that helps in process improvement and assessment [5].

There are five levels in the TMMI, each level contains a number of process areas, to move to the higher level it should complete all process areas in the lower level [6].

"At TMMI level 2, testing becomes a managed process and is clearly separated from debugging. The main objective of testing in a TMMI level 2 organization is to verify that the product satisfies the specified requirements" [1].

B. Scrum

Scrum is not a defined process or standard it is a framework that enables the organizations to set their own customized processes and techniques for their software development projects [7].

"The main aim of using Scrum is to enable a project team to react quickly, simply and appropriately instead of wasting time and energy creating, implementing and updating outdated plans" [8].

"The Scrum team consists of a Product Owner, Scrum Master, and a Development Team. Scrum teams are self-organizing and cross-functional, they are not directed by others outside the team and have all the competencies to accomplish their work" [9].

There are 3 roles in Scrum process: Scrum Master who audits and monitors the scrum process, Product owner who responsible for product requirements and the team who develop the products [10].

Scrum begins with creating the product backlog that contains customer requirements. Then the team prioritizes the requirements. The iteration is a Sprint that consists of a period of one month's duration. Every sprint starts with sprint planning meeting that clarify the details of tasks and they discuss the tasks and the progress in daily scrum meeting and at the end of each sprint, sprint review meeting is held to evaluate the sprint and check the current status against the sprint goals determined in sprint planning meeting [11].

C. Related work

There have been a number of publications focusing on the relationship between CMMI project management process areas and Scrum practices, showing the gaps and strengths between them [12, 13] but unfortunately their studies did not include an empirical evidences and Others analyze the feasibility of the CMMI process improvement in combination with the agile software development and project management area [14] however this study provide only a theoretical survey at the project management level . Neil Potter and Mary Sakry [15] listed only the main practices of CMMI that map cleanly to Scrum process steps. Few publications intend to increase the understanding of the relationship between Scrum and CMMI reporting empirical results that confirm theoretical comparisons between Scrum practices and CMMI level 2 [16]. Unlike these researches, As known software does not work correctly can lead to many problems, including loss of money, time or business reputation, and could even cause injury or death so this study is concentrated on testing process and presents the details of the mapping between Scrum practices and TMMI, and illustrates this mapping with a study providing empirical evidences of the obtained results which will increase the understanding of the mapping in theoretical and practical study. This research covers Level 2 process areas in TMMI.

III. PROBLEM DEFINITION

Organizations seek to improve testing process by applying new methodologies and models. When talking about TMMI**,** it requires a considerable amount of time and effort for implementation, it may require additional resources and knowledge required in smaller organizations and also requires a major shift in organizational culture and attitude so the number of TMMI certified companies is limited to 23 only all over the world. Trying to merge TMMI with another methodology will be a good choice to avoid those disadvantages. So it is needed to know if using TMMI will be compatible with the Agility concept or not. And if the two approaches are compatible the next step is to identify how the integration will be analyzed between them and how the organizations will apply this mapping perfectly.

IV. RESEARCH APPROACH

The goal of this research was to study how TMMI model could be used in assessing Scrum practices and to help TMMI certified companies to shift toward agility. The approach was developed in Three-phase model. The first phase was to discover an appropriate possible theoretical mapping between TMMI and Scrum practices. The second phase was to verify the applicability of this mapping. An empirical study has been conducted, the mapping was applied to a sample of TMMI certified companies to make sure that merging TMMI, and Scrum processes within those organizations is possible. The Third phase has been conducted in order to measure the effect of this merge.

V. RESEARCH ASSUMPTIONS

- Using TMMI and Scrum practices together will be useful for the organizations to achieve better team productivity.
- TMMI needs a heavy documentation in contradiction with agile software development which focuses on small projects so organizations can reduce this problem and improve their test process areas by applying both of them together.

VI. STUDY PHASE 1: DESIGNING A MAPPING MATRIX BETWEEN TMMI AND SCRUM PRACTICES

After a deep study of TMMI Practices, and scrum recommended practices. A Mapping matrix has been produced to show how scrum practices could be mapped to the TMMI practices.

The tables below show several TMMI practices (using TMMI text taken from the model definition) and how Scrum can implement each practice. To appraise Level 2, it is assumed that the Scrum implementation is robust and shows evidence of the TMMI practice being performed.

For each process area, a mapping between its specific practices and the Scrum practices was carried out. Several considerations were identified in order to establish this mapping, identifying gaps and strengths. After that, a coverage rating for each practice was established considering the following criteria in table I. The data is filled based on conducting interviews, questionnaires and practical work with experts in testing field working in Vodafone Group, ITWORX, HP Enterprise Services and MCS and the results are combined in "Rating" column as Covered, Uncovered or Partially Covered.

TABLE I. CRITERIA FOR SCRUM PRACTICES RATING

Rating		**Criteria**
U	Uncovered	The practice is not covered by Scrum.
P	Partially Covered	The practice is partially covered by Scrum.
C	Covered	The practice is fully covered.

PA 2.1 Test Policy and Strategy

“The purpose of the Test Policy and Strategy (TPS) is to develop and establish a test policy, and an organization-wide or program-wide test strategy in which the test levels are unambiguously defined. To measure test performance, test performance indicators are introduced [1]”. The mapping of TPS process area and its related scrum practices is shown in table II.

TABLE II. MAPPING BETWEEN TPS SPECIFIC PRACTICES AND SCRUM PRACTICES

TPS	TMMI Practice	Scrum Practice	Rating
SP 1.1	Define test goals	• Specify the test goals based on business needs, this occurs in release planning meeting and sprint planning meeting.	C
SP 1.2	Define the test policy	• Define test policy in release planning meeting.	C
SP 1.3	Distribute the test policy to stakeholders	• Share the test policy with involved stakeholders in release planning meeting.	C
SP 2.1	Perform a generic product risk assessment	• Create Generic Risk Profiler which highlight commonly identified risks and how to reduce them, this occurs in release planning meeting and sprint planning meeting.	C
SP 2.2	Define test strategy	• Define the test strategy in release planning meeting.	C
SP 2.3	Distribute the test strategy to the stakeholders	• Share the test strategy with involved stakeholders in release planning meeting.	C
SP 3.1	Define test performance indicators	• Define sprint metrics to enhance the predictability for examples: ○ Measuring progress ▪ Burn down charts. ○ Defect metrics ▪ Root cause analysis. ▪ Results from defect tracking system. ○ Traceability metrics ▪ Test coverage (story vs. test cases). ▪ Code coverage.	C
SP 3.2	Deploy test performance indicators	• Discuss the results with involved stakeholders in daily Scrum meeting.	C

PA 2.2 Test Planning

“The purpose of Test Planning (TP) is to define a test approach based on the identified risks and the defined test strategy, and to establish and maintain well-founded plans for performing and managing the testing activities [1]”.

The mapping of TP process area and its related scrum practices is shown in table III.

TABLE III. MAPPING BETWEEN TP SPECIFIC PRACTICES AND SCRUM PRACTICES

TP	TMMI Practice	Scrum Practice	Rating
SP 1.1	Define product risk categories and parameters	• Create Product Risk Profiler which addresses the functional and non-functional risks that most concern the involved stakeholders.	C
SP 1.2	Identify product risks	• Identify product risks, assess the associated level of risk, estimate the effort required to reduce those risks sufficiently, and then mitigate those risks through test design, implementation, and execution.	C
SP 1.3	Analyze product risks	• In Release planning: business representatives who know the features in the release provide a high-level overview of the risks, and the whole team, including the tester(s), may assist in the risk identification and assessment. • In Sprint planning: the whole team identifies and assesses the quality risks.	C
SP 2.1	Identify items and features to be tested	• Select, allocate, and prioritize test conditions to maximize effectiveness and efficiency according to product (Quality) risk analysis, this occurs in release planning meeting and sprint planning meeting.	C
SP 2.2	Define the test approach	• Identify the relevant test design techniques, this occurs in release planning meeting and sprint planning meeting.	C
SP 2.3	Define entry criteria	• Identify sprint entry criteria in sprint planning meeting.	C
SP 2.4	Define exit criteria	• Identify sprint exit criteria in sprint planning meeting.	C
SP 2.5	Define suspension and resumption criteria	• Specify suspension and resumption criteria in sprint planning meeting.	C
SP 3.1	Establish a top-level work breakdown structure	• Identify user stories and write them down on the cards at the prepared Sprint Task board.	C
SP 3.2	Define test lifecycle	• Specify test phases in sprint planning meeting.	C

SP 3.3	Determine estimates for test effort and cost	• tester detect how many hours should have to complete testing for each of selected user stories In sprint planning meeting for examples: o Using PERT technique. o Using User Story points (USP) technique.	C
SP 4.1	Establish the test schedule	• Specify the testing effort based on the user stories sizes in sprint planning meeting. • Specify the testing phases in the sprint planning meeting.	C
SP 4.2	Plan for test staffing	• In sprint planning meeting, the testing team clarifies open questions and for each item the team discusses if they have enough capacity, the required know-how and if everything else needed is available.	C
SP 4.3	Plan stakeholder involvement	• In sprint planning meeting, the testing team discusses if they have the required resources based on the sprint backlog.	C
SP 4.4	Identify test project risks	• In Release planning: business representatives who know the features in the release provide a high-level overview of the risks, and the whole team, including the tester(s), may assist in the risk identification and assessment.	C
SP 4.5	Establish the test plan	• Create test plan by the team based on the sprint planning meeting.	C
SP 5.1	Review test plan	• Review test plan in daily Scrum meeting.	C
SP 5.2	Reconcile work and resource levels	• Review resources allocation in daily Scrum meeting.	C
SP 5.3	Obtain test plan commitments	• After the sprint planning meeting the Scrum master of the team -calls the team to define the details of how the committed items are going to be implemented.	C

PA 2.3 Test Monitoring and Control

“The purpose of Test Monitoring and Control (TMC) is to provide an understanding of test progress and product quality so that appropriate corrective actions can be taken when test progress deviates significantly from plan or product quality deviates significantly from expectations [1]”. The mapping of TMC process area and its related scrum practices is shown in table IV.

TABLE IV. MAPPING BETWEEN TMC SPECIFIC PRACTICES AND SCRUM PRACTICES

TMC	TMMI Practice	Scrum Practice	Rating
SP 1.1	Monitor test planning parameters	• Monitor the progress by using specific tools for examples: o Sprint Burn down chart. o Sprint Task board o Progress chart. o Release Burn down chart.	C
SP 1.2	Monitor test environment resources provided and used	• This is Scrum master responsibility.	C
SP 1.3	Monitor test commitments	• Monitor the team commitments in Daily Scrum meeting and Sprint review meeting. • Sprint burn down chart that tracks effort remaining. • Release burn down chart that tracks story points that have been completed.	C
SP 1.4	Monitor test project risks	• This is Scrum master responsibility.	C
SP 1.5	Monitor stakeholder involvement	• Monitor stakeholder involvement in: o Daily Scrum meeting. o Sprint review meeting.	C
SP 1.6	Conduct test progress reviews	• Review the progress in: o Daily Scrum meeting. o Sprint review meeting.	C
SP 1.7	Conduct test progress milestone reviews	• This occurs in sprint review meeting.	C
SP 2.1	Check against entry criteria	• This occurs in daily Scrum meeting.	C
SP 2.2	Monitor defects	• Discuss defects in daily Scrum meeting. • Defects per iteration metric: o This metric is calculated either as a simple count or count weighted by the severity of defects that was introduced during a Sprint.	C
SP 2.3	Monitor product risks	• This occurs in daily Scrum meeting.	C
SP 2.4	Monitor exit criteria	• This occurs in daily Scrum meeting.	C
SP 2.5	Monitor suspension and resumption criteria	• This occurs in daily Scrum meeting.	C

SP 2.6	Conduct product quality reviews	• This occurs in daily Scrum meeting.	C
SP 2.7	Conduct product quality milestone reviews	• This occurs in sprint review meeting.	C
SP 3.1	Analyze issues	• Discuss the defects in daily Scrum meeting • Analyze the appeared defects in the shared test management tool.	C
SP 3.2	Take corrective action	• This occurs in: o Daily Scrum meeting. o Sprint review meeting.	C
SP 3.3	Manage corrective action	• This is Scrum master responsibility.	C

PA 2.4 Test Design and Execution

"The purpose of Test Design and Execution (TDE) is to improve the test process capability during test design and execution by establishing test design specifications, using test design techniques, performing a structured test execution process and managing test incidents to closure [1]". The mapping of TDE process area and its related scrum practices is shown in table V.

TABLE V. MAPPING BETWEEN TDE SPECIFIC PRACTICES AND SCRUM PRACTICES

TDE	TMMI Practice	Scrum Practice	Rating
SP 1.1	Identify and prioritize test conditions.	• Create test conditions and scenarios based on the selected user stories. • Prioritize the test conditions according to the size given to each user story.	C
SP 1.2	Identify and prioritize test cases	• Write the required test cases which needed for each user story. • Sort the test cases according to the importance of each one of them.	C
SP 1.3	Identify necessary specific test data	• Prepare the test data required to fulfill the test cases.	C
SP 1.4	Maintain horizontal traceability with requirements	• Use Traceability metrics to measure test coverage (story vs. test cases).	C
SP 2.1	Develop and prioritize test procedures	• Develop the test procedures and determine if test scripts will be executed automatically or manually.	C
SP 2.2	Create specific test data	• Create the required test data as identified in SP 1.3.	C
SP 2.3	Specify intake test procedure	• Specify the main areas needed to be tested. • Detect the positive paths to decide the start of testing.	C
SP 2.4	Develop test execution schedule	• This occurs in sprint planning meeting.	C
SP 3.1	Perform intake test	• Execute the smoke test to decide if the current build will be tested or rejected.	C
SP 3.2	Execute test cases	• Execute test cases manually or by using automatic test scripts.	C
SP 3.3	Report test incidents	• Record the appeared defects in the shared test management tool. • Discuss the defects in daily Scrum meeting.	C
SP 3.4	Write test log	• Track the test execution in: o Sprint Task board. o Daily Scrum meeting.	C
SP 4.1	Decide disposition of incidents in configuration control board	• This occurs in daily Scrum meeting.	C
SP 4.2	Perform appropriate action to fix the test incidents	• This occurs in daily Scrum meeting.	C
SP 4.3	Track the status of test incidents	• Review the issues in the shared test management tool. • Discuss the issues status in daily Scrum meeting.	C

PA 2.5 Test Environment

"The purpose of Test Environment (TE) is to establish and maintain an adequate environment, including test data, in which it is possible to execute the tests in a manageable and repeatable way [1]".The mapping of TE process area and its related scrum practices is shown in table VI.

TABLE VI. MAPPING BETWEEN TE SPECIFIC PRACTICES AND SCRUM PRACTICES

TE	TMMI Practice	Scrum Practice	Rating
SP 1.1	Elicit test environment needs	• Define requirements for test environment in sprint backlog. • This occurs in sprint planning meeting.	C
SP 1.2	Develop the test environment requirements	• Set requirements for test environment in sprint backlog. • This occurs in sprint planning meeting.	C
SP 1.3	Analyze the test environment requirements	• This occurs in sprint planning meeting.	C

SP 2.1	Implement the test environment	• Implement the test environment as specified and according to the sprint backlog.	C
SP 2.2	Create generic test data	• Create the required test data.	C
SP 2.3	Specify test environment intake test procedure	• Specify environment sanity check to decide if the environment is implemented correctly.	C
SP 2.4	Perform test environment intake test	• Run the environment sanity check.	C
SP 3.1	Perform systems management	• This is operation (support) resources responsibility according to the SLA (service level agreement) identified in the sprint planning meeting.	C
SP 3.2	Perform test data management	• This is the responsibility of the tester with the operation (support) resources identified in sprint planning meeting.	C
SP 3.3	Coordinate the availability and usage of the test environments	• This occurs in sprint planning meeting.	C
SP 3.4	Report and manage test environment incidents.	• Report environment incidents in: ○ Daily Scrum meeting. ○ The shared test management tool.	C

PA 2.6 Configuration Management

“The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits [18]”. “CM is not specifically called out in Scrum. However, in an agile environment it is pretty easy to add a layer of CM to protect your work. Even for groups that like to use white boards, you can be creative and at least establish some basic protection by labeling items (e.g. “V1.1,” or “Story dated 1/2/YY”) and taking a photo [15]”.

The changes in requirements are managed implicitly in product backlog and sprint backlog [17]. This practice is classified partially covered.

PA 2.7 Process and Product Quality Assurance

“The purpose of Process and Product Quality Assurance (PPQA) is to provide staff and management with objective insight into processes and associated work products [18]”. The mapping of PPQA process area and its related scrum practices is shown in table VII.

TABLE VII. MAPPING BETWEEN PPQA SPECIFIC PRACTICES AND SCRUM PRACTICES

PPQA	TMMI Practice	Scrum Practice	Rating
SP 1.1	Objectively Evaluate Processes	• Scrum master periodically checks that the scrum process is being followed. • Evaluate the scrum process in retrospective meeting.	C
SP 1.2	Objectively Evaluate Work Products and Services	• Scrum master periodically checks that the scrum process is being followed. • Evaluate the scrum process in retrospective meeting.	C
SP 2.1	Communicate and Ensure Resolution of Noncompliance Issues	• This is Scrum master responsibility.	C
SP 2.2	Establish Records	• This is Scrum master responsibility.	C

VII. OVERALL RESULTS OF THE MAPPING

General results of the mapping are shown in the *Fig. 1*, a consolidated view of the coverage of TMMI Level 2 Process areas by Scrum practices.

This result shows that most of specific practices of TMMI Level 2 Process areas are Covered and 90% of Configuration management process area is Uncovered. This means that TMMI Level 2 process areas are mostly satisfied for the organizations that need to improve their test processes by applying TMMI within Scrum.

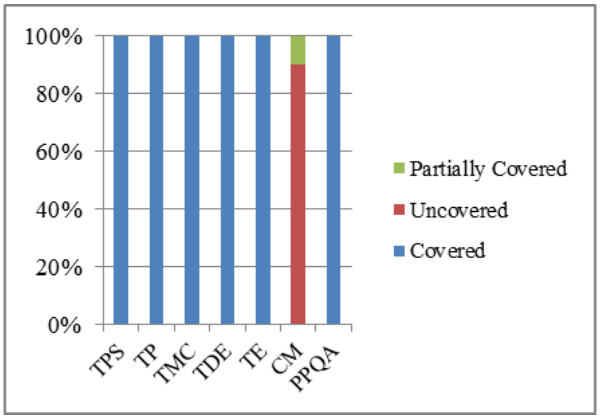

Fig. 1. TMMI Level 2 Process areas covered by Scrum

VIII. Study Phase 2: Empirical Study

Once theoretical comparison between TMMI and Scrum were established, a sample of the certified companies was selected to apply the approach and find the strengths and weakness in using the provided study.

A. *Study description*

The number of TMMI certified companies from level 2 to level 5 is 23 all over the world like Vodafone UK, Tech Mahindra Global Test Factory, Wipro Limited, Virgin Media, etc. A sample of 2 large companies was taken which is 8.7% from the certified companies. Working on 4 projects; the average of testing team members in each project was 6 members. The process used in all the projects was Agile/Scrum. This study was limited to TMMI Level 2 and all the involved teams were less than 10 members.

IX. Study Phase 3: Verifying Applicability of Research Work

The objective of the study was to discuss the effects of testing projects before and after using Agile/Scrum practices inside the TMMI process and to show an empirical evidence of understanding the mapping between TMMI and Scrum practices. Scrum practices widely used as they enhance the testing process in specific KPIs that mentioned in this study. The study was applied on a telecom projects which using Scrum process and the projects' data is extracted from HP ALM test management tool (represented in table VIII). In this project the KPIs are calculated based on the following equations and (P) stands for defect priority and the result is shown in table IX:

Test Case Effectiveness = Number of valid defects mapped to test cases / Total number of valid defects.

Defect Quality = Total number of valid defects / Total number of defects.

Defect Density = Number of valid defects / Total number of executed test cases.

Defect Removal Efficiency = Valid defects / (Valid defects + Valid Production defects).

Total Number of defects = Number of defects created (P1+P2+P3+P4).

Total number of valid defects mapped to test cases = Number of defects created (P1+P2+P3+P4) Mapped to test cases.

Total Number of cancelled defects= Number of defects cancelled (P1+P2+P3+P4).

Total Number of valid defects = Total Number of defects - Total Number of cancelled defects.

TABLE VIII. Calculated Test Execution Factors before and after using Scrum

Test execution factor	**Before Scrum**	**After Scrum**				
		Iteration 1	***Iteration 2***	***Iteration 3***	***Iteration 4***	***Iteration Average***
Number of test cases executed	200	100	120	120	100	110
Number of priority 1 defects created (P1)	8	5	7	6	4	5.5
Number of priority 2 defects created (P2)	20	10	12	9	5	9
Number of priority 3 defects created (P3)	25	15	12	10	6	10.75
Number of priority 4 defects created (P4)	19	20	15	17	12	16
Number of defects created P1 Mapped to test cases	3	4	6	5	4	4.75
Number of defects created P2 Mapped to test cases	11	8	10	8	3	7.25
Number of defects created P3 Mapped to test cases	13	12	11	7	6	9
Number of defects created P4 Mapped to test cases	10	16	12	15	11	13.5
Number of defects cancelled P1	3	1	1	2	1	1.25
Number of defects cancelled P2	8	2	1	2	0	1.25
Number of defects cancelled P3	7	3	2	1	0	1.5
Number of defects cancelled P4	4	1	0	1	1	0.75
Valid Production defects	19	5	7	5	4	5.25
Total Number of defects	72	50	46	42	27	41.25
Total Number of valid defects	50	43	42	36	25	36.5
Total number of valid defects mapped to test cases	37	40	39	35	24	34.5
Total Number of cancelled defects	22	7	4	6	2	4.75

TABLE IX. CALCULATED KPIS BEFORE AND AFTER USING SCRUM

KPIs Values	Before Scrum	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration Average
Test Cases effectiveness	51%	80%	85%	83%	89%	84%
Defect Quality	69%	86%	91%	86%	93%	89%
Defect Density	25%	43%	35%	30%	25%	33%
Defect Removal Efficiency	72%	90%	86%	88%	86%	87%

The following figure (*Fig. 2*) displays the percentages of test cases effectiveness before and after using Scrum and it illustrates that the study helped in enhancing this KPI by [100*(84-51)] /51= 64.71%.

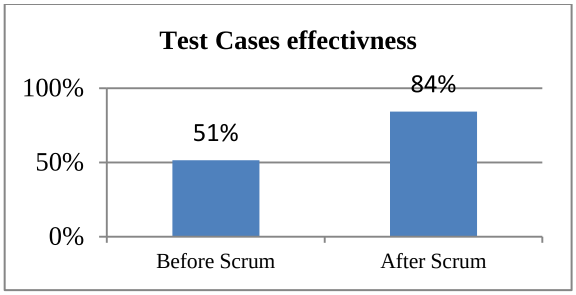

Fig. 2. A summary of test Case effectiveness after using Scrum

The following figure (*Fig.3*) displays the percentages of defect quality before and after using Scrum and it illustrates that the study helped in enhancing this KPI by [100*(89-69)] /69= 28.99%.

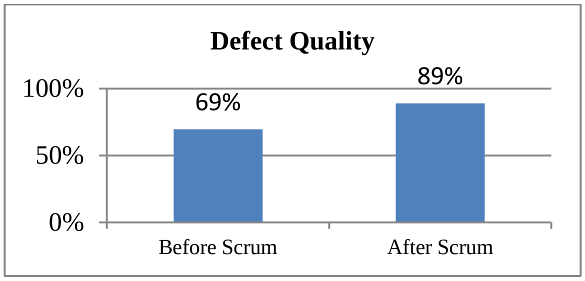

Fig. 3. A summary of Defect Quality after using Scrum

The following figure (*Fig. 4*) displays the percentages of defect density before and after using Scrum and it illustrates that the study helped in enhancing this KPI by [100*(33-25)] /25= 32%.

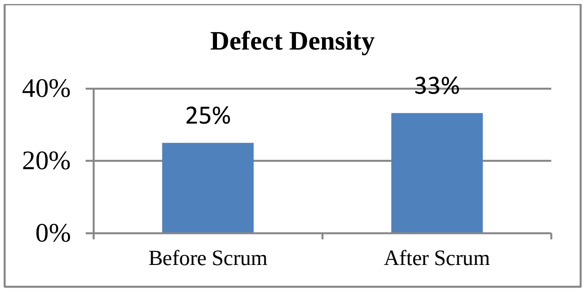

Fig. 4. A summary of Defect Density after using Scrum

The following figure (*Fig. 5*) displays the percentages of defect removal efficiency before and after using Scrum and it illustrates that the study helped in enhancing this KPI by [100*(87-72)] /72= 20.83%.

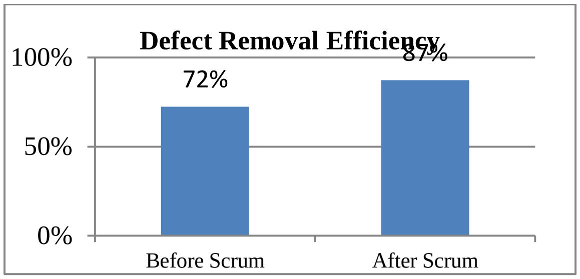

Fig. 5. A summary of Defect Removal Efficiency after using Scrum

The following questionnaire is distributed among testing experts to display their opinions about how was the communication between involved stakeholders in the projects using legacy and others using Scrum practices. The result of this survey is shown in table x and the summary is represented in *Fig. 6.*

TABLE X. TESTING KPIS BEFORE / AFTER USING SCRUM

KPI	Before Scrum	After Scrum
Quality Delivered to Customers (k1)	75%	95%
Team Enthusiasm (k2)	70%	80%
Retrospective Process Improvement (k3)	40%	80%
Communication among stakeholders (k4)	50%	90%
Customer Satisfaction (k5)	60%	85%
Employee Satisfaction (k6)	70%	80%
Meet the project deadlines (k7)	70%	95%

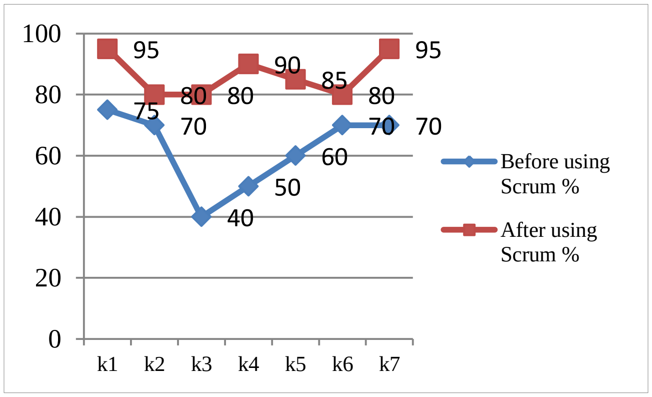

Fig. 6. A summary of testing KPIs before / after using Scrum

X. DISCUSSION

From the previous study the results show that using the mapping helped in enhancing and improving test cases effectiveness to reach 89% in some iteration and also defects quality was increased to reach 93% in some iteration.

The research helped the companies to make improvements in the quality delivered to the customers as project requirements were discussed and refined during sprint planning and daily scrum meetings. It helped in increasing the customer and employee satisfaction. Also applying the research helped the companies in the following achievements:

- Business assurance - consistent delivery adherence and test effectiveness exceeding customer targets and improved business assurance.
- Step change in cost - significant reduction in test costs given to the client.
- Tester error is reduced from iteration to iteration based on root cause analysis of defects.
- Small iterations helped in identifying issues early and made it easier to respond to change.

XI. CONCLUSION AND FUTURE WORK

The goal of this paper was to improve testing process by comparing Scrum practices with TMMI Level 2 process areas, showing the major strengths and gaps between them and identifying how organization are adopting complementary practices in their projects to make these two approaches more compliant.

The study results showed that most of the specific practices of TMMI Level 2 Process areas are covered and 90% of configuration management process area is uncovered and also the organizations that their process based on TMMI model can be improved by adding some Scrum practices.

In the light of that, it is suggested as future works: The field of software engineering is changing very quickly so more studies will help the companies to find a detailed reference containing the best practices regarding TMMI maturity levels and another point to map the TMMI with other agile methods for example XP, FDD, and DSDM.

REFRENCES

[1] Erik van Veenendaal, “Test Maturity Model Integration (TMMI)”, release 0.1, pages [1- 219], 2012

[2] David Consulting Group, “What is TMMi® – and Why Should You Care?” , pages [1- 2], May 2013

[3] Dean Leffingwell, “Scaling Software Agility: Best Practices for Large Enterprises”, pages [5- 15], 2007

[4] Erik van Veenendaal, Richard Grooff and Rob Hendriks,”Test Process Improvement using TMM(i)”, STAR Tester, Issue 19, pages [1- 7], October 2007

[5] David Consulting Group, “How does TMMI work and what does it contain”, pages [1- 5], June 2013

[6] Cognizant 20-20 insights, “Adopting the Right Software Test Maturity Assessment Model”, pages [1- 5], June 2014

[7] Ken Schwaber and Jeff Sutherland, “The Definitive Guide to Scrum: The Rules of the Game”, pages [1- 16], July 2013

[8] Linz, Tilo,” Testing in Scrum: A Guide for Software Quality Assurance in the Agile World”, pages [11-32], March 2014

[9] Sander Bannink, “Challenges in the Transition from Waterfall to Scrum – a Casestudy at Portbase, Twente Student Conference on IT”, pages [1-10], January 2014

[10] Deepa Vijay and Gopinath Ganapathy, “GUIDELINES TO MINIMIZE THE COST OF SOFTWARE QUALITY IN AGILE Scrum PROCESS, IJSEA, Vol.5, No.3”, pages [61- 69], May 2014

[11] Ahsan Nawaz & Kashif Masood Malik, “SOFTWARE TESTING PROCESS IN AGILE DEVELOPMENT”, pages [3-9], June 2008

[12] Ana Sofia C. Marçal, Felipe S. Furtado Soares, Teresa M. Maciel, “Blending Scrum practices and CMMI project management process areas”, pages [17-29], January 2008

[13] Ana Sofia C. Marçal, Bruno Celso C. de Freitas, Felipe S. Furtado Soares, Arnaldo D. Belchior, “Mapping CMMI Project Management Process Areas to SCRUM Practices”, pages [1- 9], 2007

[14] Armin Preis, “Integration Evaluation of Scrum and CMMI - Theoretical Survey at the Project-Management Level”, pages [1- 28], January 2012

[15] Neil Potter and Mary Sakry, “IMPLEMENTING SCRUM (AGILE) AND CMMI® TOGETHER”, pages [1- 6], March 2009

[16] Jessica Diaz, Juan Garbajosa, and Jose A. Calvo-Manzano,” Mapping CMMI Level 2 to Scrum Practices: An Experience Report”, pages [93-104], 2009

[17] Juha Koskela, “Software configuration management in agile methods”, pages [33- 54], ESPOO 2003

[18] CMMI Product Team, “CMMI® for Development”, Version 1.2, pages [114- 130, 353- 363], August 2006