

# Incorporating Innovative Practices in Software Engineering Education

Ghazi Alkhatib

Princess Sumaya University for Technology  
Amman - Jordan

Ghassan Issa

Petra University  
Amman – Jordan

Aiman Turani

Applied Science University  
Amman - Jordan

Daoud Daoud

Princess Sumaya University for Technology  
Amman - Jordan

M. Ibrahim Zaroor

Petra University  
Amman - Jordan

**Abstract**—Software Engineering (SE) practices deals with business requirements that continue to operate in dynamically changing and turbulent environments. SE practitioners must confront the business need for persistent innovation and build appropriate future workforce culture. Agile software development approaches view change from a perspective that reflects today's tumultuous business and technology environments. Introducing these innovative methodologies, along with the more traditional CMMI method, in IT department is a challenging task. Such task requires not only technology knowledge, but also the human side of organization, team, and individual readiness to accepting such a challenge. To deal with this side, the paper employs creative thinking processes and the four brain concept. In such environment, linking agile methods and CMMI is imperative to improving software quality.

**Keywords**—Software engineering education, Agile methods, CMMI, HBDI, creative thinking, visual studio team system.

## I. INTRODUCTION

This paper integrates creative thinking processes, the four-brain concept known as Herrmann Brain Dominance Instrument (HBDI), and SE practices to create an environment that is conducive to fruitful adaptation to these new practices as well as comparing and linking different approaches to software development to model business requirements. [1] The paper specifically highlights the debate between advocates of model-based SE Capability Maturity Model Integration (CMMI) of the Software Engineering Institute (SEI) at Carnegie Mellon University and agile methods, such as extreme programming, Crystal methods, lean development, Scrum, RUP, and adaptive software development (ASD). [2] Finally, the paper presents an overview of the software developed by SEI and marketed by Microsoft as Visual Studio Team System (VSTS). The software supports both CMMI and agile methods software development practices. The paper includes a plan to

incorporate this software in an introduction to software engineering course at the undergraduate level.

## II. RESEACH FRAMEWORK

This section contains discussions on three dimensions: the human dimension, the technical dimension, and the education dimension. CMMI for Services Version 1.3 contains a framework for organizations adapting CMMI that has three dimensions linked together by processes: people with skills, training, and motivation; tools and equipment; and procedures and methods defining the relationship of tasks.

### A. Introduction to creative thinking and the four brain quadrants: the human dimension

Motivated people have to be creative and innovative, and this is necessary to keep moving the organization from one level of CMMI to the next.

Creative thinking steps include: interest, preparation, incubation, illumination, verification, application, validation, and storage. The paper added the verification and validation steps as used in software engineering: verification will ensure that the correct procedure is followed in the adaptation of new methods, while validation will ensure that new methods do achieve intended outcomes and provide the justification to continue the use of such new methods. The definitions of the four brain quadrants are as follow: quadrant A (QA): Logical, factual, rational, critical, analytical, quantitative, authoritarian, and mathematical, quadrant B (QB): Technical reader, data collector, conservative, controlled, sequential, articulate, dominant, and detailed, quadrant C (QC): Musical, spiritual, symbolic, talkative, emotional, intuitive, (regarding people), and reader (personal), and quadrant D (QD): Intuitive (regarding solutions), simultaneous, imaginative, synthesizer, holistic, artistic, and spatial. Measure of HBDI may be applied to individuals, teams, and organizations. Any of these three measurements may show strength in one or more of these

quadrants. Each individual, team, or/and organization has a profile that depicts the strength in one or more of these quadrants. The figure below depicts sample team profiles.

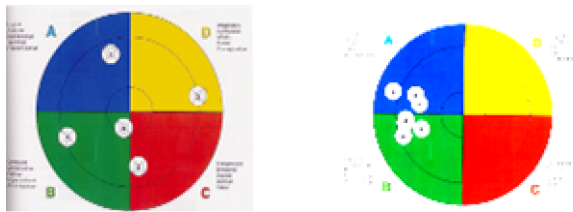


Figure 1. Team with balance profiles on the left and team with strong profiles in QA and QB on the right.

In each of the creative thinking steps, specific type of people who are strong in one or more of these quadrants is required. To start with, all four types of quadrants are required for interest and preparation steps. For incubation and illumination, however, people strong in QD and QC are required to carry on the task of adopting new ideas. People who are strong in QC are needed to convince people who are strong in AQ and QB, who normally will object to change unless they are gently brought into the wagon, perhaps through information meeting at lunch or sports activity. Overall, applying CMMI requires individuals, teams, and organization that are strong in quadrants A and B, while applying agile methods would require strength in quadrants C and D. Individuals who are extremely strong in one quadrant should not be put at top management or as team leaders, since such HBDD characteristic would hamper progress towards adapting new SE methodologies. The paper suggests that the team leader should be alternated according to the step involved in creative thinking: steps 1 and 2 for QA and QB, steps 3 and 4 for QC and QD, steps 5 for QA and QB, step 6 for QB and QA, step 7 QA and QD, step 8 for QB.

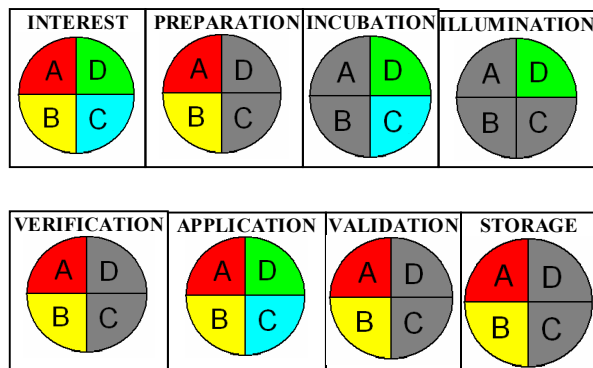


Figure 2. Steps in creative thinking and role of the four quadrants for each step (colors represented importance)

In one reported case, for example, a software development team consisted of member profiles that are mainly strong in QA and QB and Weak in QC and QD, while the team leader

had a strong QD. Team members were waiting for instructions from the team leader, while as the team leader was expecting members to perform independently. The team was dissolved and the project failed. Therefore, HBDD and creative thinking should assist in building effective communication channels among the following groups: team members, team members and team leader, teams and stakeholders, and in the case of agile methods, end users who are part of the development team.

The following figure shows these interactions.

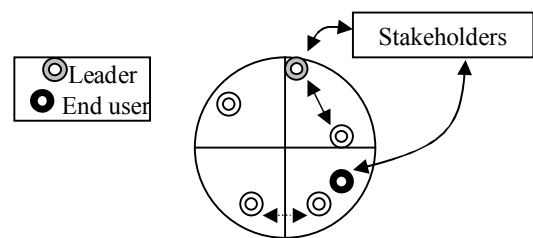


Figure 3. Interactions among team members, team leader, end user, and stakeholders

Once organizations establish their readiness level of creative thinking, they can embark on achieving the adaptation of newer technologies in software engineering.

### B. CMMI vs. Agile methods (AM): the technology dimension

CMMI evolved from the multiple models of earlier versions into one comprehensive integrated continuous model. It is depicted in two dimension array: Vertical dimension represents the previous levels and termed as generic goal 0 through 5 as follows: incomplete, performed, managed, defined, quantitatively, and optimized. The horizontal dimension depicts process area categories and termed as: process management, project management, engineering, and support. Version 1.3 consists of three components: CMMI for Service, CMMI for Acquisition, and CMMI for Development. The following table provides comparison of continuous and stage implementations of CMMI.

TABLE I. CMMI CONTINUOUS AND STAGED REPRESENTATIONS [3]

Level	Continuous	Staged
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined

Level 4		Quantitative
Level 5		Optimized

Here is a quote the paper opted to include without paraphrasing [3]:

“The continuous representation is concerned with selecting both a particular process area to improve and the desired capability level for that process area. In this context, whether a process is performed or incomplete is important. Therefore, the name “Incomplete” is given to the continuous representation starting point.

The staged representation is concerned with selecting multiple process areas to improve within a maturity level; whether individual processes are performed or incomplete is not the primary focus. Therefore, the name “Initial” is given to the staged representation starting point.”

Naturally, selecting a particular process for improvement encourages organizations to achieve agility at the selected process rather than wait to uplift the whole organization to the next level.

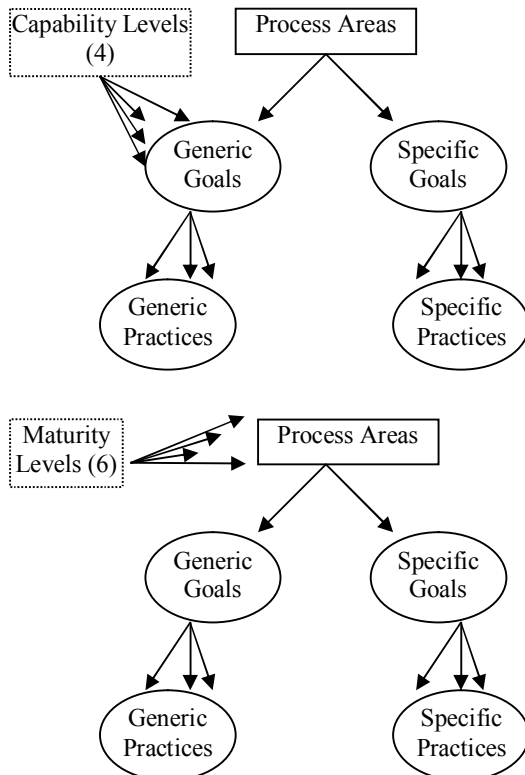


Figure 4. Continuous representation of CMMI above figure and staged representation of CMMI below figure.

In the first figure, capability levels are linked to Generic Goals, while in the second figure Maturity Levels are linked to Process Areas. This organization makes the staged representation more complex to follow than the continuous representation.

Explaining all components of CMMI is beyond the scope of any research paper. The research will deal with CMMI as a complete and complex process definition that requires significant resources and time. In this regard, CMMI maps into the third dimension of procedures and methods defining the relationship of tasks.

CMMI mentions agility in the last common goal (level), but a small percentage of IT departments achieves that goal. Therefore, this paper devices approaches to include agile principles within CMMI practices to achieve agility at all levels. Agile methods encompass many different methodologies, such as RUP, SCRUM, and ADS. Discussing these methodologies is also beyond the scope of this paper. Even a list of references may span several pages. For employing individual agile methods in CMMI, readers should refer to [4, 5], and for a complete references on agile methods refer to [6]. Normally, agile methods are discussed in relationship to level 2 in CMMI. To understand the fundamental thrust of agile methods, the agile manifesto is listed below. [7]

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

The authors of the agile manifesto do not eliminate the need for the right items: documentation, processes and tools, contract negotiations, and planning. Therefore, organization using CMMI and wanting to adapt AM should deemphasize these items; while organizations using AM who plan to get certified according to CMMI practices should re-emphasize these items.

As to integration strategies of AM in CMMI, this paper envisions three approaches: front-end, back-end, and from within CMMI.

The first two approaches are used by companies who employ the full CMMI process improvement. The front end is applied when developing initial contacts with clients with unclear or unknown requirements as a mean of developing a throwaway prototype. The back end is applied for maintenance updates and repair and minor versioning. A full scale CMMI processes need not to be deployed in these cases.

The figure below depicts the third approach of linking CMMI and AM from within CMMI using the staged representation. The figure suggests horizontal and vertical movement to agility: the first one within each level and the latter one between lower and higher levels. Achieving horizontal agility should assist IT organization to move seamlessly to a higher level agility. In the horizontal movement, basic processes at each level will be eliminated when more experiences are gained from continuous practices, while concentrating more on the advance processes.

the waterfall model, while AM is linked to other models, such as prototyping, water fountain, spiral, and incremental.

A list of topics is provided below:

- History of the waterfall model
- History of CMMI: from CMM to CMMI and the three parts of services, acquisition, and development.
- Structure of CMMI components
- Achieving different levels of CMMI certification
- History of client-led design and end-user computing
- Practices of RAD and user participation
- The evolution of agile methods
- Overview of agile methods: RUP, SCRUM, XP, etc.
- Linking agile methods and CMMI
- Factors that affect the use of a methodology are: contracted vs. in-house development, size of software, team member experience, IT Department experience, software risk factor (such as air traffic controller system), and type of software developed (Transaction Processing (TPS), such as ERP implementation vs. Management support systems (MSS), such as DSS and ES). The table below lists factors and recommended methodology.

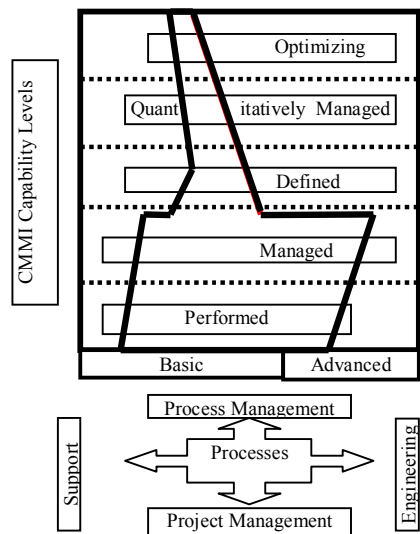


Figure 4. Continuous improvement in CMMI to reach Agility at each level.

This may lead to a more lean and agile IT organization before reaching the highest level. New IT employee should be tested for HBDDI conformance and given proper position in teams, as well as adequate training in creative thinking. Throughout this process, organizations should strike a balance between the right hand and the left side items: AM should generate more documentation artifacts to meet CMMI requirements for verification and certification, and CMMI on the other hand should de-emphasize documentation artifacts when possible.

### C. Building the Technology Culture: the education dimension

#### 1) Undergraduate curriculum component

Many undergraduate curriculums in computer science and software engineering as well IT-based Colleges include an introductory course in software engineering. The paper proposed that such course should include a module for comparing and contrasting agile methods and CMMI. The presentation of AM and CMMI should be in relationship to software development life cycle models: CMMI is linked to

TABLE 1. FACTORS AND RECOMMENDED METHODOLOGY

Factor	CMMI	Agile
Contracted vs. in house	contracted	In-house
Team member experience	Low to medium	High to med.
IT Department experience	Low to medium	High to med.
Risk	Low to med.	High
Type of system	TPS	MSS

#### 2) Overview of MSTs

The discussions below present overview of the Microsoft VSTS. With Visual Studio Team System and the integrated process templates, teams can deliver predictable results, continuously improve and adapt, and effectively collaborate and communicate with team members and stakeholders. Visual Studio Team System includes Microsoft Solutions Framework (MSF) for Agile Software Development and MSF for CMMI Process Improvement. In addition, partner organizations offer processes including SCRUM, and RUP for Visual Studio Team system. These templates can be used as is, or customized to enact individual development processes. The list below include suggested outline for a presentation on MS VSTS: [8]

##### a) Foundational Principles

The following are the eight foundational principles, which form the backbone for the other models and disciplines of MSF:

- Foster open communication
- Work towards a shared vision
- Empower team members

- Establish clear accountability and shared responsibility
- Focus on delivering business value
- Stay agile, expect change
- Invest in quality
- Learn from all experiences

*b) MSF Models*

MSF consists of two models:

- MSF Team Model. This describes the role of various team members in a software development project. The members of this team would be:

- Product Management: Mainly deals with customers and define project requirements, also ensures customer expectations are met.

- Program Management: Maintains project development and delivery to the customer

- Architecture: Responsible for solution design, making sure the solution design optimally satisfies all needs and expectations

- Development: Develops according to the specifications.

- Test: Tests and assures product quality

- Release/Operations: Ensures smooth deployment and operations of the software

- User Experience: Supports issues of the users.

One person may be assigned to perform multiple roles. MSF also has suggestions on how to combine responsibilities such as the developer should not be assigned to any other role.

- MSF Governance Model.

This describes the different stages in processing for a project. The MSF Governance Model has five overlapping tracks of activity (see below), each with a defined quality goal. These tracks of activity define what needs to be accomplished and leave how they are accomplished to the team selected methodology. For instance, these tracks can be small in scope and performed quickly to be consistent with an agile methodology, or can be serialized and elongated to be consistent with a Waterfall methodology.

- Envision - think about what needs to be accomplished and identify constraints

- Plan - plan and design a solution to meet the needs and expectations within those constraints

- Build - build the solution

- Stabilize - validate that the solution meets the needs and expectations... "sync and stabilize"

- Deploy - deploy the solution

*c) MSF Project Management Process*

- Integrate planning and conduct change control

- Define and manage the scope of the project

- Prepare a budget and manage costs

- Prepare and track schedules

- Ensure that right resources are allocated to the project

- Manage contracts and vendors and procure project resources

- Facilitate team and external communications

- Facilitate the risk management process

- Document and monitor the team's quality management process

*d) MSF for Agile Software Development methodology*

The MSF for Agile Software Development (MSF4ASD) is intended to be a light weight, iterative and adaptable process.

The MSF4ASD uses the principles of the agile development approach formulated by the Agile Alliance. The MSF4ASD provides a process guidance which focuses on the people and changes. It includes learning opportunities by using iterations and evaluations in each iteration.

*e) MSF for Capability Maturity Model Integration Process Improvement methodology*

The MSF for Capability Maturity Model Integration Process Improvement (MSF4CMMI) has more artifacts, more processes, more signoffs on milestones, more planning, and is intended for projects that require a higher degree of formality and ceremony.

The MSF4CMMI is a formal methodology for software engineering. Capability Maturity Model was created at the Software Engineering Institute of Carnegie Mellon University, and is a process improvement approach that provides organizations with the essential elements of continuous process improvement resulting in a reduced SDLC, improved ability to meet the cost and schedule targets, building products of high quality. The MSF4CMMI has extended the MSF4ASD guidance with additional formality, reviews, verification and audit. This results in a SEP that relies on process and conformance to process rather than relying purely on trust and the ability of the individual team members. The MSF4CMMI has more mandatory documents and reports than the agile version, and this more formal development process reduces risk on large software projects and provides a measurable status. One of the benefits of using the CMMI process is the standard evaluation by which one can compare the ability to develop software in other organizations.

*f) Lab components for MSTs*

Of course, to complete students' and trainers' exposure to CMMI and AM development practices, they should be exposed to actual experiences in development small projects in a laboratory set up. The list below contains the components for establishing a laboratory with VSTS.

- Visual Studio Team System 2010 Team Foundation Server Workgroup Edition

- Microsoft Visual Studio Team Suite 2010, includes:  
Microsoft Visual Studio Team Architecture 2010  
Microsoft Visual Studio Team Development 2010  
Microsoft Visual Studio Team Test 2010 (includes Microsoft Test and Lab Manager)  
Microsoft Visual Studio Team Architecture 2010  
Microsoft Visual Studio Team Development 2010  
Microsoft Visual Studio Team Test 2010 (includes Microsoft Test and Lab Manager)  
Microsoft Visual Studio Team Test 2010 Essentials (includes Microsoft Test and Lab Manager)  
Microsoft Visual Studio Team Lab Management 2010  
Microsoft Visual Studio Team Foundation Server 2010

- Visual Studio Team System 2010 Test Edition
- Visual Studio Team System 2010 Test Load Agent
- MSF for Capability Maturity Model Integration Process Improvement methodology
- MSF for Agile Software Development methodology

For more technical and detailed information on MSF and VSTS, readers may visit the Microsoft Web site.

### III. SUMMARY AND CONCLUSIONS

This paper presented a framework for integrating innovative practices in software engineering education, namely agile methods and CMMI. The paper highlighted the importance of the human side by linking creative thinking steps and HBDI to the process of adaptation of innovative practices in software development. CMMI for Development states that AM could be used in CMMI: [9, page IV]

"Informative material was improved, including revising the engineering practices to reflect industry best practice and adding guidance for organizations that use Agile methods."

It also lists the following characteristics of AM: [9, page 58]

- Direct involvement of the customer in product development
- Use of multiple development iterations to learn about and evolve the product

- Customer willingness to share in the responsibility for decisions and risk

The resulting mix between CMMI and AM should improve software processes quality. Further research will include the following topics:

- Agile methods and VSTS
- Capability Maturity Model Integration (CMMI) and VSTS
- Global software development in VSTS
- ISO implementation in VSTS
- Other model-based and agile methods and VSTS
- Comparative analysis of agile vs. CMMI software development: a case study
- Evaluating Usability of VSTS
- Developing distributed service oriented architecture in VSTS
- Testing approaching in VSTS
- Customization of VSTS for different methodologies
- Business intelligent and VSTS
- Comparing VSTS and UML standards

### REFERENCES

- [1] B. Boehm, "Get ready for agile methods , with care," Computer, vol. 35, iss. 1, January 2002., pp. 64-69.
- [2] J. Highsmith, and A. Cockran, "Agile software development: the business of innovation," Computer, vol. 34, iss. 9, Sept. 2001, pp. 120-127.
- [3] CMMI-SVC, V1.3, Improving processes for providing better service, November 2010, TECHNICAL REPORT.
- [4] J. A. H. Alegre'ia, and M. C. Bastarrica, "Implementing CMMI using a Combination of Agile Methods," CLEI ELECTRONIC JOURNAL, vol. 9, iss. 1, PAPER 7, JUNE 2006, pp. 1-15.
- [5] N. Potter and M. Sakry, "Implementing SCRUM (agile) and CMMI together," The Process Group-Post newsletter, vol. 16. iss. 2, March 2009, pp. 1-6. [www.processgroup.com/newsletter.htm](http://www.processgroup.com/newsletter.htm).
- [6] <http://www.semq.eu/leng/proimpam.htm>
- [7] <http://agilemanifesto.org>
- [8] [http://en.wikipedia.org/wiki/Microsoft\\_Solutions\\_Framework](http://en.wikipedia.org/wiki/Microsoft_Solutions_Framework), accessed November, 29 2010.
- [9] CMMI-DEV, V1.3, Improving processes for providing better service, November 2010, TECHNICAL REPORT.