Volume 6, Issue 4, April 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

CMMI Key Process Areas and FDD Practices

Rituraj Deka, Nomi Baruah
Department of Computer Science & Engineering, Dibrugarh University, Dibrugarh, Assam, India

Abstract: ***The development of Information Technology during the past few years resulted in designing of more and more complex software. The outsourcing of software development makes a higher requirement for the management of software development project. Various software enterprises follow various paths in their pursuit of excellence, applying various principles, methods and techniques along the way. The new research is proving that CMMI and Agile Methodologies can benefit from using both methods within organizations with the potential to dramatically improve business performance. The paper describes a mapping between CMMI key process areas (KPAs) and Feature-Driven Development (FDD) communication perspective, so as to increase the understanding of how improvements can be made in the software development process.***

General Terms: Requirement Management, Project Planning, Project Monitoring and Control.
Keywords: Agile, CMMI, FDD, KPAs

I. INTRODUCTION

It has been seen that during early days, the adopters of CMMI were developers of large-scale, risk-averse, mission critical systems, whereas the adopters of Agile Methodologies focuses on smaller, single-team development projects with volatile requirements, the inaccurate information about CMMI and Agile results in misperceptions which arises the following factors: misuse of CMMI practices and overlaid on development activities that may have already been perceived by software developers, lack of information between CMMI community and Agile community, the difficulty of terminologies i.e. in CMMI(e.g. discipline, quality assurance and predictability) and Agile methods(e.g. continuous integration and test-driven development and collective ownership),etc.Each of these frameworks might look at odds with each other, making it difficult to use two or more. The difficulty occurs because much of the information shared regarding these frameworks is from success and failure stories, rather than understanding the specifics of each framework [1].Projects combining agile methods with CMMI combines adaptability with predictability to better serve large customer needs[2]. .

II. CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

CMMI,which is a integration of three models.It integrates Capability Maturity Model for Software(SW-CMM) v2.0 draft C,the Systems Engineering Capability Model(SECM),and the Integrated Product Development Capability Maturity Model(IPD-CMM) v0.98.These three models were selected because of their successful adoption[3].The representation of CMMI can be staged and continuous[4].In the staged representation of CMMI,it is divided into 5 maturity levels(ML).CMMI consist of 22 key process areas(KPAs).Maturity Level 1 does not contain any key process areas.

III. FEATURE-DRIVEN DEVELOPMENT (FDD)

The "Agile Movement" in software industry saw the light of day with the Agile Software Development Manifesto published by a group of software practitioners and consultants in 2001[5].For many people the appeal of these agile methods is their reaction to the bureaucracy of the monumental methods. These new methods attempt a useful compromise between no process and too much process, providing just enough process to gain a reasonable payoff [5].FDD is one of the Agile Software Development approach.FDD is a model-driven short iteration process.FDD is built around a core set of "best-practices". The chosen practices are not new but this particular blend of ingredients is new [6].FDD does not cover the entire software development phase. The development consist of two main stages i.e. discovering the list of features to implement and feature-by-feature implementation.FDD includes the roles,artifacts,goals and timelines needed in a project[7].FDD defines six key roles and implies a number of roles. The key roles are Project Manager, Chief Architect, Development Manager, Chief Programmers, Class Owners and Domain Expert. They are five processes within FDD.They are shown in Figure 3.Process 4 and Process 5 i.e. Design by Feature and Build by Feature are iterated.

IV. MAPPING CMMI KEY PROCESS AREAS AND FDD PRACTICES

The paper discusses about those key process areas(KPAs) of CMMI which the FDD practices satisfies.The other process areas are related to organization and FDD practices are concerned to software development and independent of

organization.The key process areas of CMMI described below are satisfied by FDD and it explains how the KPAs are satisfied by the FDD practices.

IV.1 Requirement Management

FDD employs development of a feature-list to manage functional requirements and development tasks. Solution requirement analysis begins with a high-level examination of the scope of the system and its context. Features in this respect are small pieces of client-valued functions expressed in the form<action><result><object>[8].So, Requirement Management is satisfied by FDD.

IV.2 Project Planning

The planning team in FDD consists of a Project Manager, Development Manager and Chief Programmer. They planned the order in which features will be developed. The planning is based on dependencies, risk, complexity, workload balancing, client repaired milestone and checkpoints [9].So, Project Planning is satisfied by FDD.

IV.3 Project Monitoring and Control

A project progress is tracked and made visible during the Design by Feature/Build by Feature phases. Each feature in FDD has six milestones, three from Design by Feature phase (Domain, Walkthrough, Design and Design Inspection) and three from Build by Feature phase (Code, Code Inspection, Promote to Build).Once these milestones are completed, the date is placed on the Track by Feature Chart which is prominently displayed for the team. When all the six milestones of a feature are completed, the completion is reflected on the "Burn-Up" chart. All features are scoped to be completed within a maximum period of two weeks which includes all the six milestones [9].So, Project Monitoring and Control is satisfied by FDD.

IV.4 Measurement and Analysis

In FDD, any function that is too complex to be implemented within two weeks is further decomposed into smaller functions until each sub-problem is small enough to be called a feature. These features do the analysis and measurement part[10].So, Measurement and Analysis is satisfied by FDD.

IV.5 Process and Product Quality Assurance

In FDD, quality assurance activities are done during the iteration. It focuses on the use of the established quality assurance practices with the agile development method. These activities are unit testing, regular builds, design inspection, code inspection and individual code ownership. At the end, the team does a separate system testing [11].So, Process and Product Quality Assurance is satisfied by FDD.

IV.6 Configuration Measurement

The FDD practices are all driven from a client valued feature perspective. Configuration Management practice of FDD helps with identifying the source code for all features that have been completed to date and to maintain a history of changes to classes as feature team enhance them[12].So, Configuration Management is satisfied by FDD.

IV.7 Requirement Development

The FDD does not address the initial creation of management of documented requirements and assumes that requirements development has been accomplished in a supporting process [13].So, Requirement Development is satisfied by FDD.

IV.8 Technical Solution

The FDD is more focused on the technical architecture of the solution. The scoping phase of a FDD SDPM strategy consist of a collaborative sessions with the customer to model the solution and the solution approach[14].So, Technical Solution is satisfied by FDD.

IV.9 Product Integration

In FDD, control occurs not by conformance to concept-driven plans but by constant integration and testing of the evolving feature sets as they emerge during the product development process. Having product architecture is important but having good technology integration is important. For this reason architects need to be heavily involved in product integration [15].So, Product Integration is satisfied by FDD.

IV.10 Verification

The FDD uses six sharply defined milestones to track progress of each feature through design by feature and build by feature. A milestone is reported complete only when all the work for that task has been finished and verified to be so. They promote to build milestone which is attained when all the code for a feature has been verified into the version control system used to generate "the build" [10].So, Verification is satisfied by FDD.

IV.11 Validation

The FDD includes the deployment into their iteration cycle. Validation involves the validation teams which will not validate until the end of the project when all the features are implemented. The intermediate releases cannot be validated and cannot be used against production data[16].So, Validation is satisfied by FDD.

IV.12 Integrated Project Management

From a project management perspective, FDD provides a framework for managing the big issues that arise in most projects at a time and in a manner that will have the greatest impact. The FDD approach is focused on people and people management as that the key to successful project management. The project manager facilitates rather than dictates the process. Another project manager responsibility is stakeholder coordination [17].So, Integrated project management is satisfied by FDD.

IV.13 Risk Management

The FDD developers know how to plan and how to establish meaningful milestones. They get their risk reduction that comes from managing a project that delivers frequent, tangible, working results. They see frequent results that they understand. And they exactly how far the project is at point in time[18].So, Risk Management is satisfied by FDD.

IV.14 Quantitative Project Management

In FDD, guidelines are given to each of the process for the amount of time that should be spent. There are five incremental, iterative processes in FDD.Processes 1 to 3 are done at the start of a project and then updated throughout the development cycle. Processes 4 and 5 are done incrementally on two week cycles. Each of these processes has specific entry and exit criteria [9].FDD has a set of best practices. Inspections is one of the practice adopted in FDD.Inspection are carried out to ensure good quality design and code, primarily by detection of defects [12].So, Quantitative Project Management is satisfied by FDD.

V. CONCLUSION

The goal of this paper was to compare the feature-driven development (FDD) which is an agile methodology in relation to the key process areas of the CMMI model. It shows the gaps and strengths existence between the FDD software development methodology and CMMI software process improvement model. It is seen that FDD practices maps mostly with key process areas of maturity level 2 and maturity level 3 of CMMI.Therefore, FDD is a good implementation for those software organizations which are in CMMI maturity level 2 and maturity level 3.any function that any function that run across the full width of the page – one column wide. We also recommend e-mail address (Helvetica 12-point). See the top of this page for three addresses. If only one address is needed, center all address text. For two addresses, use two centered tabs, and so on. For three authors, you may have to improvise.

REFERENCES

[1] Potter, M. and Sakry, M. 2009.Implementing Scrum(Agile) and CMMI® Together Helping You Improve Your Engineering Process, Vol. 16, No. 2. www.processgroup.com/newsletter.htm

[2] Jakobsen, C.R. 2009.Scrum and CMMI Going from Good to Great. In Proceedings of the Agile Conference (Chicago, IL 24-28, Aug, 2009).IEEE DOI: 10.1109/ AGILE.2009.31.

[3] Cursive Handwriting (CMMI® for Development, Version 1.3), CMMI-DEV V1.3, CMMI Product Team, Technical Report. Software Engineering Institute at Carnegie Mellon University, 2010.

[4] Jones, L.G. 2002.Software Process Improvement and Product Line Practice:CMMI and the Framework for Software Product Line Practice.US:SEI,Carnegie Mellon University.

[5] Fowler, M. 2001.The New Methodology. http://www. martinfowler.com/articles/newMethodology.html.

[6] http://www.step-10.com/SoftwareProcess/FeatureDriven Development/FDDPractices.html.(accessed on December 2014)

[7] Palmer, S. R. and Felsing, J. M. 2002.A Practical Guide to Feature-Driven Development. Upper Saddle River, NJ, Prentice Hall.

[8] Lane, D.2011.The Chief Information Officer's Body of Knowledge: People, Process and Technology. Copyright© 2011 by John Wiley & Sons Inc. All Rights Reserved.books.google.co.in/books?isbn=1118113780,pp.162.

[9] Williams, L.2007.A Survey of Agile Development Methodologies.agile.csc.ncsu.edu/SEMaterials/AgileMethods.pdf,pp.16.

[10] Goyal,S.2007/08.FDD Agile Techniques for Project Management and Software Engineering, pp.9.Technical University,Munich.csis. pace.edu/~marchese/cs616/Agile /FDD/fdd.pdf.

[11] Nawaz, A. and Malik, K.M. 2008.Software Testing Process in Agile Development, pp.32.

[12] en.wikipedia.org/wiki/Feature-driven development,(ace ssed on December 2014)

[13] Westfall, L.2008.The Certified Software Quality Engineer Handbook, ASQ Quality Press Pub.ISBN-13:978-0873897303, pp.138.

[14] Wysocki, R.K.2006.Effective Software Project Management.Indianpolis, IN:Wiley Pub.ISBN:97807645 96360.

[15] Puri, C.P.2009.Agile Management: Feature Driven Development.Global India Pub, pp.223.books.google. co.in/books?isbn=9380228260.

[16] Khramtchenko, S.2004.Comparing eXtreme Progra mming and Feature Driven Development in academic and regulated environments.www.featuredrivendevelop pment.com/files/FDD-vs-XP.pdf.

[17] Bauer,M. 2004.FDD and Project Management.Cutter IT Journal.

[18] www.petercoad.com/download/bookpdfs/jmcuch06.pdf.(accessed on September,2014)

