

Correlating and Tabulating Form of Agile Development Methodologies

Debdatta Chatterjee
M.Tech in computer science
(software engineering)
KIIT University, BBSR, India

Susmita Saha
M.Tech in computer science
(software engineering)
KIIT University, BBSR, India

Mrs. Leena Das
Assistant Professor
KIIT University, BBSR,
India

Abstract

The main intent of agile development methodologies is to develop the project in a systematic manner. This development process systematically controls the task that performs to gain the end product and objective of the project. The iterative nature of this methodology makes this process more flexible. Its compact time cycle improves the project in small release. The main characteristics of agile development methodologies are: iterative incremental, people oriented, light weight, test driven development. In this paper we are considering mainly four agile development methodologies, they are: Extreme Programming (XP), Scrum, Crystal and Feature Driven Development (FDD). A short survey is identified on strength and weakness of the development methodologies and also a comparison in tabular format is introduced.

Keyword: Agile Methodologies, XP, Scrum, Crystal, FDD, Strength and Weakness.

1. Introduction

Agile methodologies appeared as a reaction to traditional ways of developing software and acknowledged the need for an alternative to documentation driven, heavyweight software development processes. Different agile methodologies are in use such as Extreme Programming, Scrum, Crystal and Feature Driven Development. These different methodologies have less documentation and more code oriented features that are common among them [9]. Extreme

Programming (XP) tends to be best accepted by the developers [5].

Extreme Programming is most popular of the various types of agile methodologies. It takes many of the best wellknown software development practices and applied them during system development. XP is a set of values, principals and practices for rapidly developing high quality software by preaching the values of community, simplicity, feedback and courage [1,9]. According to [1] XP has 12 core practices:

Table 1. Extreme Programming Core Practices

Core Practice	Description
Planning Game	The customer and development teams decide on what features will be implemented.
Simple Design	The simplest design that meets the business requirements is selected.
Small Releases	Release a useful set of features into production and update them frequently in short cycles.
Metaphor	It helps us understand the parts of the system and helps us communicate more effectively by choosing system name.
40- Hour Week	Maintain productivity and avoid burn out of the development team.
Onsite Customer	Onsite customer ensures effective communication as a output minimum paper work will be required.
Coding Standard	Coding standard is a way of communication and documentation.

Continuous Integration	Frequent code integration means less chances that there will be diversion and it will result in more testing.
Refactoring	The risk of introducing error is reduced by applying small updates in small steps.
Pair Programming	Two Programmers works in a single machine. This helps to review the code constantly.
Collective code ownership	The code is the property to every member of the team.
Testing	Test driven development. Justified the code at all times and before new features are added.

In the *Scrum model*, projects are divided into small parts of for that can be incrementally developed and delivered over time boxes that are called Sprints. The products therefore get developed over a series of manageable chunks. Each Sprint typically takes only a couple of week. At the end of each Sprint, stakeholder and team members meet to assess the progress and the stakeholder suggests to the development team any changes an improvement they feel necessary[4]. Scrum introduces the idea of "empirical process control" that is Scrum uses the real world progress of a project - to plan and schedule releases. Scrum has three fundamental roles: Product Owner, ScrumMaster, and team member.

Product Owner: In Scrum, the Product Owner is responsible for communicating the vision of the product to the development team.
Scrum Master: Scrum Master works to remove any impediments that are interrupting the team from achieving its sprint goals.
Team Member: For software projects, a typical team includes a mix of software engineers, architects, programmers, analysts, QA experts, testers, and UI designers.

Crystal Methods help the developer to know the characteristics of the project and to address the variability of the environment. Crystal inventor *Cock Burn* thinks the basic methodology should be "barely sufficient". He competes "You need one less notch control than you expect and less is better when it comes to delivery quickly"[2]. This method is the most lightweight, adaptable approaches to software development based on problem characteristics. Several of the key features of crystal are full of team work, simplicity and communication. It depends on

size of team and criticality of project, it also emphasize on face to face communication. Presently two types of Crystal are introduced: *Crystal Clear* and *Crystal Orange*. Crystal model has mainly three elements: Documents and artifacts, Roles, Process.

Feature Driven Development (FDD) was approached by Jeff De Luca and Peter Coad in 1997. This methodology is a highly and short iterative approach which emphasizes quality at all steps. It provides accurate and meaningful progress and status information with the minimum of overhead and disruption for the developers. FDD is mainly structured with eight best practices: Individual class ownership, DOM (domain object modelling), Feature teams, Developing by Feature, Inspections, and Regular builds, Configuration Management, Reporting or Visibility of results.

2. Overview of Agile Methodologies

- **Extreme Programming:**

Extreme Programming emphasizes teamwork. Managers, customers, and developers are all equal partners in a collaborative team. Extreme Programming implements a simple, yet effective environment enabling teams to become highly productive. The team self-organizes around the problem to solve it as efficiently as possible. XP includes practices that are a fresh to development team. Open work spaces, Pair Programming, and the 40 hour work week may lead to resistance from developers and management[7].

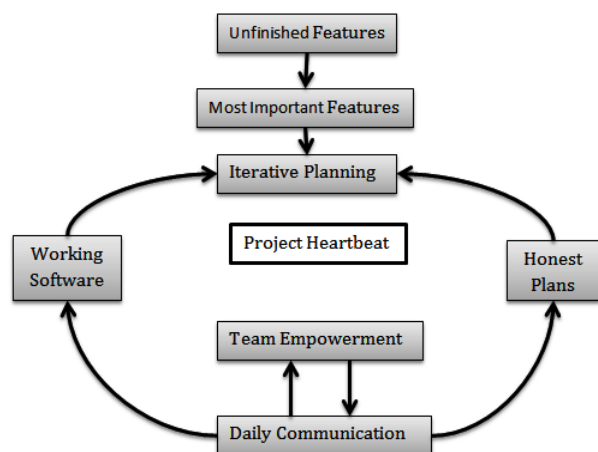


Figure1. Extreme Programming

Table2. Strength and Weakness of Extreme Programming

Strength	Weakness
Customer holds feature priority where developer holds estimation.	Requires onsite customer
It allows us to focus on coding and avoid needless paperwork and meeting	Architectural and designing concerns are difficult to determine by new adopter
Frequent feedback opportunities	As it takes a lot of discipline to keep its difficult to get many developers to accept the practices.
It leverages the power of simplicity.	Reduces the importance of a well thought out architecture.

• **Scrum:**

Scrum is mainly used in project which changes business requirements rapidly. In the scrum methodology the same technique is used which is established in the sport in Rugby[8]. To improve communications a project is forwarded to team members and break the operation into cycles which is known as "Sprint". The arrangement of the development method is more focused than the coding technology in SCRUM.[6].

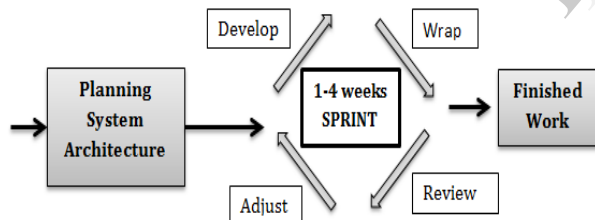


Figure2. SCRUM

Table3. Strength and Weakness of SCRUM

Strength	Weakness
Business value helps us to prioritize.	Technical practices are unspecified.
It encourages teamwork and transparency. It helps breaking down hierarchies.	It focuses totally on features.

The overhead cost in terms of process and management is minimal thus leading to a quicker cheaper result.	It is good for small fast moving projects.
Customer participation and steering.	Only provide project management support, other disciplines are out of scope.

• **Crystal:**

Crystal Method is an agile Software Development Methodology. Tools or techniques affect people more than software development based on this argument [2]. As their business needs demand crystal development methodology develop methods. Methods family has common strategy criteria, such as incremental delivery, user direct participation, automatic regression testing, etc.[10]. Two types of Crystal are introduced: *Crystal Clear* and *Crystal Orange*. Crystal model has mainly three elements: Documents and artifacts, Roles, Process. Crystal Clear targeted at a D6 project and could be applied to a C6 or a E6 project and possibly to a D10 project. Crystal Orange is targeted at a D40 project. Crystal Orange is for 20-40 programmers, working together in one building on a project.

Table4. Strength and Weakness of Crystal

Strength	Weakness
As project size grows, cross functional teams are utilised to ensure consistency	May not work when for distributed teams.
The human component has been considered for every aspect of the project support structure.	Expects all team members to be co-located.
Only methodology that specifically accounts for life critical project.	Moving from one flavour of crystal to another in mid project doesn't work.
Testing is highly prioritized that at least one tester should present on each project team.	Adjustments are required from one project size/ structure to another in order to follow the described flavour of crystal for that project size / criticality.

• Feature Driven Development (FDD):

FDD consists of five steps process [3]. The five steps are as follows:

- i. Develop an overall model of the required operation.
- ii. Build features list.
- iii. Prioritization and implementation of plan.
- iv. Design the prioritized listed features.
- v. Build the features.

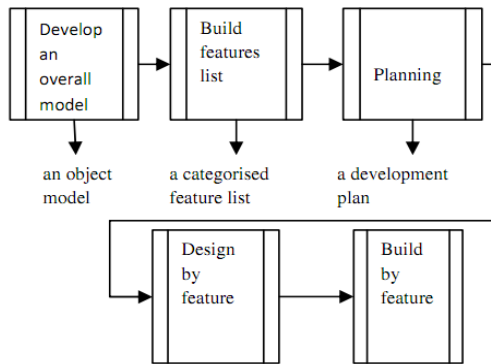


Figure3.FDD

Table5. Strength & Weakness of FDD

Strength	Weakness
Scales to large team and Project value.	Promotes individual code ownership
Supports multiple teams working in parallel.	It opposed to share team property.
Features track all aspect of a project.	Iterations are not well defined.
Designing and building features understandable and adoptable.	Model centric aspect effects on existing systems that have no models.

3. Tabularization form of Agile Methodologies

Table6. Key Concerns of agile methods

Extreme Programming (XP)	
Phases	Exploration, Iteration Planning, Iteration to release, Production, Maintenance, Death phase.
Techniques	Pair Programming, Test Driven Development,

Distinguishing Factors	Continuous Integration. <ul style="list-style-type: none"> • 10-12 programmers are expected to co-locate. • 12 core practices • Minimal previous paper works. • Feedbacks from programmers and developers.
SCRUM	
Phases	Review Release Plan, Sprint, Sprint release, Closure
Techniques	Team Development, List uncompleted works, Separation of project, Scrum Meetings, Burn down charts
Distinguishing Factors	<ul style="list-style-type: none"> • Developer practices are wrapped around the project management. • Priorities are unchanged in 1 month Sprints • Scrum Team conducts daily Scrum meeting. • Display Progress is viewed in Burn down chart.
Crystal	
Phases	Usable code is delivered frequently, Deliberate advancement, Osmotic Interaction.
Techniques	Daily meetings, Layout programming, Burn charts.
Distinguishing Factors	<ul style="list-style-type: none"> • It is a modifiable development methodologies which leads for atomic to larger teams • This depends on size of team and necessity of project • Highlights on Confronting communication • Deal with people, communication, group, skills, talents, and conversation as Prime-order effects • Beginning with minimum techniques and construct as absolutely necessary
Feature Driven Development (FDD)	
Phases	Develop overall model, Design feature list, Planning, configuration, Construct feature.
Techniques	Designing objects, Speculated by feature, Isolated code Controlling, Gaining progress report.
Distinguishing Factors	<ul style="list-style-type: none"> • It consists of five sub-processes, which helps us to view In and Out criteria. • Development are object

	models, structural shape and sequence diagrams. <ul style="list-style-type: none"> • 14 days features • Scalable to larger teams • Highly-specified development practices.
--	---

4. Conclusion and future work:

This paper explores the co-relation among the Agile development methodologies that are Extreme Programming, Scrum, Crystal & Feature Driven Development. The various factors that handles the implementation of an agile development method are: Team size, training, the involvement and support of management authorization of external resources, and company size all significantly impact the implementation of an agile Software Development Method. In this paper we have collected four different methodology of agile software development. A brief comparison in tabular format is given in section 3. From the configuration we can conclude that Extreme programming (XP) is the best known and widely used agile development method. It takes an extreme approach to iterative development. Whereas The Scrum approach is a general agile method but its focus is on managing iterative development rather than specific agile practices. In respect to XP and Scrum, Crystal is a simple and effective method, it is suitable for developing in the scene and in FDD whole the processes are divided into small iterative features, which helps to stay the process simple and works in an efficient manner.

From this paper we can conclude that which type of projects can be used in which respective agile development methodology. Though it is a basic survey work as well as comparison among the different methodologies, we can get a clear idea about the strength and weakness of XP, Scrum, Crystal, FDD, which are easily approachable for novice learners.

5. Reference

- [1] Beck, Kent, *Extreme Programming explained: Embracechange*. 2000, MA, Addison Wesley.
- [2] Highsmith J. *Agile Software Development Ecosystems*, 2002 MA: Addison Wesley.
- [3] Palmer, S. & Felsing, J. *A practical guide to feature-driven development*. Prentice Hall. Upper Saddle Hill River, NJ. 2002.
- [4] Bob Hughes, Mike Cotterell, Rajib Mall, *'Software Project Management'*. P. 92.
- [5] Ilieva, S..P. Ivanov, and E. Stefanova. *Analysis of an agile methodology implementation*. in 30th EUROMICRO conference 2004. Rennes, France, IEEE computer society.
- [6] Mann, C. & Maurer, F. *A case study on the impact of scrum on over time & customer satisfaction*. Proceedings of the agile development conference (ADC '05). Denver, CO. 72-79. 2005.
- [7] Jenkins, S.B. *Musings of an old school programmer*. Communications of the ACM. 49(5), 124-126, 2006.
- [8] Jeffrey A. Livermore, *Factors that Impact Implementing an Agile Software Development Methodology*, IEEE., 2007 1-4244-1029-0/07.
- [9] Agarwal, R. and D. Umphress, *Extreme Programming for a single person team*, in proceedings of the 46th Annual South East Regional Conference on XX. 2008, ACM: Auburn, Alabama. P. 82-87.
- [10] DaYong Sang. *Requirement Analysis of Agile Development Process* programmer, 2009, 2, P. 70-75.