INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Systematic Review of Agile Methodologies for Software Development

Mr. Anuj Fotariya[1], Mr. Omprakash Mandge[2]
[1]MCA Student, MET Institute Of Computer Science
[2]Assistant Professor, MET Institute Of Computer Science

Abstract: With the increase in complexity and innovation in IT products and services, developing them, that to by the shared efforts of team members brings in the need of some standardized development model or approach. Due to several advantages of agile methodologies over the traditional waterfall based models, it is the preferred choice. Many a times, a single agile method is also not enough to meet the current scenario of product development. We therefore propose a Competitor Driven Development (CDD) model, a hybrid agile process model for IT product development by abstracting practices from Extreme Programming (XP) and Feature Driven Reuse Development (FDRD) agile methods. This model is based on self-realizing requirements generation for Product development by keeping an eye watch on competitor's upcoming launch of the product and market response to it rather than customer explicitly specifying the requirements. This development model can be generally used by the organizations who intend to develop IT product for mass targeted customers rather than an individual or an organization.

I. INTRODUCTION

As compared to the traditional waterfall based models agile methods are more suitable, especially in the scenarios when the complete requirements of the development is not known in advance, or when the chances of requirements getting change are high, or when quick product launch with some basic functionalities is essential to meet the market needs and competition. The adaptive planning with respect to the requirement changes, continuous customer involvement and flexibility in documentation are some of the reasons that make agile a preferred choice for IT product development.

"Single agile model doesn't fits in all the real time situations we face. Let's take an example when the procedure or the algorithm of implementation is unclear and we are aware of only the final output in such scenario's we prefer performing Pair Programming, Test-First or Test-Driven Development (TDD) practices of Extreme

Programming (XP) for developing the software solution, wherein we write the test cases and then write the code to pass those test cases. Test-First practice of XP ensures that the code is properly tested for functioning correctly. It also ensures that no bug is getting introduced due to any new changes, i.e. it welcomes change. Also the amount of coding done is just enough to pass all the test cases, no extra code is present.

We perform Feature Driven Development (FDD) especially when the domain is new and when the client talks in terms of the features to be implemented and when there is less interdependency among the features i.e. the features can be developed separately and can be merged later.

The customer involvement is continuous via phone calls, emails, Watsap and we even meet client on-site once or twice a week to discuss progress and gather new requirements if any. Market is highly competitive, if we fail to satisfy client in terms of on-time product delivery, specified changes in requirements then it becomes very difficult to sustain them." The above response clearly gives the idea about the scenarios when it is preferable to use XP and when it is preferable to use FDD.

As changes are frequently required to any IT product under development due market competition, a reengineered model of FDD called Feature Driven Reuse Development (FDRD) process model [7] is developed which integrates reuse concept with the FDD process model. Effective reuse of software artifacts results in high productivity and less cost of development thereby increases the profits of organization and helps in achieving improved status in the market in terms of quality product and less time to market.

II. PROBLEM STATEMENT

To design a development model that helps developing an IT product in accordance with the upcoming IT products in the market, thereby helping the manufacturing firm stay competitive and survive in the market.

III. PROPOSED MODEL

In today's competitive market, Developers should not only focus on their product but also on the products what their competitor's are offering. Competitor's emerge for every product. Instead facing the one, become the one. Every manufacturing firm is not an inventor, but can make money by developing similar product at competitive price. CDD process model complements this approach for manufacturing IT products. CDD process model, Is based on self-realizing requirements generation by keeping an eye watch on competitor's launch of the product rather than customer explicitly specifying the requirements.

Can be generally used by the organizations who intend to develop IT product for mass targeted customers rather than an individual or an organization.

1) *Identify the next upcoming launch of the Competitor:* In this phase, research and development team of the organization gathers the requirement for product development by keeping an eagle's eye watch on the product that the competitor is planning to launch in the market as appears outside by means of advertisements that the competitor does through mediums like social media, hoardings and several others informative resources for making the market aware of their upcoming product. (Note: The product of the competitor is not yet started getting sold only its publicity is observed in the market, possibly beta release may get available.) Therefore, in this phase the customer involvement in requirement management is of the type "Design for Customers", wherein the customer is not explicitly specifying the requirements rather requirements are gathered through market research.

2) *Forecast Selling Point:* In this phase research and development team tries to identify the "selling mantra" of the Competitor i.e. the strategy that the competitor uses to make people feel of buying their product. From this it can be predicted that whether the product will be successful or not.

3) *Conduct Feasibility Study:* If the product seems to be interesting and profit making, the organization conducts various types of feasibility study for developing the similar product. Depending on the result of feasibility study the organization makes the decision of whether to move ahead for development or not.

4) *Develop a identical/better looking Prototype:* If the output of the feasibility study is positive then the organization makes the decision of developing the product and arranges all the needful resources for its development. Looking at the appearance/GUI in advertisements/video promotions the organization develops an identical product prototype, even though with less number of features but with some uniqueness and some additional features different from the Competitor if possible. As the requirement understood by research team may not be clear enough and there are high end chances of the requirements getting changed or more requirements may be recognized by the research team after the start of development and also the algorithm/procedure for implementation are not clear enough. In such scenario, XP is the best choice, as the design is implicitly generated when following the Test-First practice of XP. So, the efforts are not wasted in changing the design as the requirement changes. This Test-First/TDD practice of XP is accompanied by other practices of XP like Pair Programming, Continuous integration, Sit together, Whole team, Energized work.

5) *Launch the Product in the Market:* Once the similar product is developed, launch the product in the market at very low price as compared to the competitor as there are comparatively limited features. Launch the product if possible nearby the same time when the competitor's product starts getting sold in the market.

6) *Reverse Engineer:* Once the competitor's product starts getting sold in the market, purchase it, analyze all its features in detail and thereby do step by step reverse engineering of the competitor's product.

7) *FDRD:* After the reverse engineering of the competitor's product is done, list down all the better and attractive features present in it to incorporate them in our product. Since now, the requirements in terms of features are clearer enough, use FDRD process model for its implementation. Reuse concept merged with FDD in FDRD helps in achieving high productivity in less time thereby fulfills the requirement of market competition, as in market early bird catches the prey.

8) *Release:* Once the better and attractive features from the competitor's product are incorporated in our product, launch it in the market as a new release, at a competitive price.

9) *Customer Feedback:* Market survey is conducted and customer feedback is taken on our and competitor's product, for their likes and dislikes. Now the type of customer involvement for requirement management is "Design with Customers", where the requirement change is to improve over the disliked features/quality of our product and incorporate liked features/quality of competitor's product. These requirements changes are then sent to the appropriate phase of FDRD and the cycle continues

Parameter	Scrum	XP	FDD
Process Phases	1. Pre-game phase 2. Development phase 3. Post-game phase	1. Exploration phase (User stories) 2. Planning phase (Stories for next iteration) 3. TDD 4. Continuous Integration and Refactoring	1. Develop an Overall Model 2. Build a Feature List 3. Plan by Feature 4. Design by Feature 5. Build by Feature
Core/Driven by	Priority driven	Test driven	Model driven
Training Program Availability	Certified Scrum Master Program is available @ scrumalliance.org	No such program available	No such program available

Parameter	Scrum	XP	FDD
Developers skills	Moderately skilled, Scrum trained	Highly skilled, experienced	Moderately skilled
Unit Testing	Simultaneously, in the end of each sprint	Test-First	Simultaneously during the build of feature
Design	High level at beginning, some detailing during each Sprint planning	Design activities are implicit, on the fly, incremental	Detailed designing before implementation
Location of Team Members	Co-location is preferred	Co-location is a necessity	Allows distributed teams
Scalability of Project team	Suitable for team size < 10. If more members, create sub-teams	Small team of 3-20 members, internally perform pair programming	Up to 50-60 members, 3-5 co-located at feature level.

Parameter	Scrum	XP	FDD
Documentation	Minimum, increases when multiple, distributed teams.	Minimum, most of communication is done verbally	Spend sufficient amount of time in documentation
End user involvement	Product Owner represent end-users in Scrum	End users are actively involved, requires highest customer involvement	End-users participate through feedback reports and requirements documents
Reuse	-	Rejects the idea of code for reuse. Focuses only on today's need following the YAGNI (You aren't going to need it) approach	FDRD the enhance model of FDD specially makes at attempt to merge reuse concept with traditional FDD

IV.   PROPOSED HYBRID MODEL
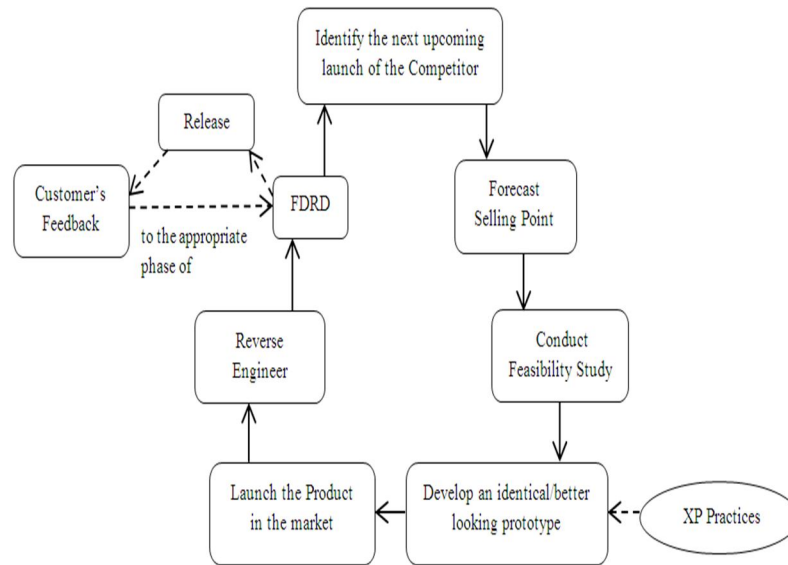
V.   CONCLUSION

The paper proposes CDD, a hybrid agile process model of XP and FDRD for IT product development.   CDD process model helps in developing an IT product in accordance with the upcoming IT products in the market. CDD not only combines the strengths of XP and FDRD but also takes into consideration market orientation which is very important for a product to get successful in the market

REFERENCES

[1]   R. Steven Wingo, Murat M. Tanik, "Using an Agile Software Development Methodology for a Complex Problem Domain" Proceedings of the IEEE SoutheastCon 2015, Fort Lauderdale, Florida April 9-12, 2015.
[2]   Wikipedia

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)