

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/277022110>

Determining The Best Agile SDLC for Bangladesh's Software Industry

Article · April 2015

CITATIONS

3

READS

1,088

5 authors, including:



Debajyoti Karmaker

Australian National University

25 PUBLICATIONS 126 CITATIONS

[SEE PROFILE](#)



M. Saef Ullah Miah

Universiti Malaysia Pahang

44 PUBLICATIONS 146 CITATIONS

[SEE PROFILE](#)



Hafizur Rahman

American International University-Bangladesh

8 PUBLICATIONS 49 CITATIONS

[SEE PROFILE](#)



Al Imran

American International University-Bangladesh

9 PUBLICATIONS 77 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Ethereum based voting distributed app (dApp) [View project](#)



Public Health [View project](#)

Determining The Best Agile SDLC for Bangladesh's Software Industry

D Karmaker[†], M S U Miah^{*}, H Rahman^{*}, M A Imran^{*} and M Islam[†]

Department of Computer Science

American International University-Bangladesh

{d.karmaker, saef, hafizur}@aiub.edu and mazhar01123@gmail.com,

Abstract— Agile methodologies start possessing popularity in the last decade. In Bangladesh this was the time when software development starts to emerge as a promising industry. But most of the Software firms in Bangladesh follow waterfall method to develop software even the small outsourcing oriented firms. Now some firms have started following agile methodologies in testing basis but they are not very willing to transform their whole development process to follow agile methodologies from following Waterfall method. This study has focused on scopes and benefits of following agile methodologies instead of Waterfall method. This study has determined SCRUM as the best methodology to follow for the emerging software development industry of Bangladesh.

Keywords: *Agile methodologies, SCRUM, extreme programming, software development life cycle.*

I. INTRODUCTION

The idea of software development life cycle starts when Dr. Winston Royce described a model to manage large software development projects in 1970. In his model he suggested to do the design and related research before starting programming. After some days his model started to be called Waterfall model and becomes the most followed SDLC. But to cope with the real constantly changing business world, some new methodologies has been introduced. These methodologies having some differences have one thing in common, all of these suggest to break the requirement into small stories so that a change cannot affect the whole system. And this was the starting of module based software development. Agile methodologies let the followers to break down a big project to parts and make it more manageable. These methodologies ensure that every member of the team is doing his best and keep everyone of the team informed about the status of others. In Bangladesh Waterfall method is followed by most of the software development firms. But these practice causes a lack of flexibility in requirement changing. Clients don't feel free to change requirement whenever he wants to and the developer team are not willing to entertain client's changes most of the times. Following Agile method can increase flexibility in changing requirement, can decrease development time and complexity. But the most important and fabulous thing about Agile methodologies is it increases the spirit and mutual understanding among team members. This study comes out with the result of following Agile methodologies specially the SCRUM method instead of following Waterfall method is far more beneficiary for Bangladeshi software firms.

II. BACKGROUND STUDY

A. Agile Methods

Agile methods' main idea is to break projects into small requirements and completing those requirements one after one. Small requirements are divided into short stories and completion of all the stories completes a requirement. SCRUM and XP are two most practiced Agile methodologies. This section will highlight SCRUM and XP methods briefly.

1) *SCRUM Methodology*: When following SCRUM, a project requirement is divided into small stories and with some stories a sprint is organized. If any change in requirement occurs then it possibly hits one or two stories. So, the development team faces less complexity. Every story is designed in such a way that when implementing a story, a developer does not need to think about other stories. The process of Sprint management is shown in Fig.1. SCRUM is thought to be a framework to manage projects but later it has been adopted as a software development process. Difference between SCRUM and other methods is that when following SCRUM it is assumed that analysis, design, and development processes in a Sprint are not stable. A process is used to manage the risk and unpredictability and ensuring flexibility, responsiveness, and reliability. SCRUM is fully responsive to changes and adjusts the project in every sprint. SCRUM methodology needs the teams working on a project to sit together every day which is known as 'Daily Meeting' and at the starting of the sprint 'Sprint Planning Meeting' and at the end of a sprint 'Sprint Review Meeting'. These meetings keep every member of project teams and the client updated about the project status. The SCRUM practices are listed in Table

TABLE I. SCRUM PRACTICES

Product Backlog
Effort Estimation
Sprint
Daily Meeting
Sprint Planning Meeting
Spring Backlog
Sprint Review Meeting
Sprint Burn Down Chart

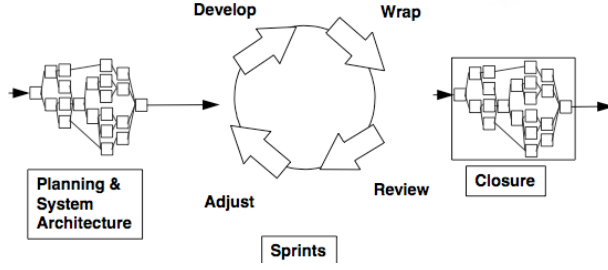


Fig. 1. SCRUM Methodology [10].

2) *XP Methodology*: XP is a software development methodology that depends on five values to complete a project communication, simplicity, feedback, courage and respect. XP has some unique practices to ensure quality software development such as test-first programming, pair programming, incremental design, continuous integration etc. This practices and some principles such as humanity, quality, mutual benefit, improvement, diversity, failure, flow, opportunity, redundancy, etc. to bridge the gap between values. XP divides requirement into stories like SCRUM does and it instead of sprint chooses stories to complete within a week and it summarizes the whole project at first weekday. Application definition and requirement have no rigid form in XP. These are redefined continuously every week. As a result XP deliver rapid deliveries with low rate of bugs.

TABLE II.: XP PRACTICES

Sit Together	Real Customer Involvement
Whole Team	Incremental Deployment
Informative Workspace	Team Continuity
Energized Work	Shrinking Teams
Pair Programming	Root Cause Analysis
Stories (User Stories)	Shared Code
Weekly Cycle	Code And Tests
Quarterly Cycle	Single Code Base
Slack	Daily Deployment
Ten Minute Build	Negotiated Scope Contract
Continuous Integration	Pay Per Use
Test First Programming	
Incremental Design	

III. COMPARISON

Scrum and Extreme Programming (XP) are definitely very aligned. In fact, if someone walked in on a team doing one of these processes he/she might have hard time quickly deciding whether he/she had walked in on a Scrum team or an XP team. The differences are often quite subtle, but they are important. Our study has found four main differences between Scrum and XP.

Scrum teams typically work in iterations (called sprints) that are from two weeks to one month long. XP teams typically work in iterations that are one or two weeks long.

Scrum teams do not allow changes into their sprints. Once the sprint planning meeting is completed and a commitment made to delivering a set of product backlog items, that set of items remains unchanged through the end of the sprint. XP teams are much more amenable to change within their iterations. As long as the team hasn't started work on a particular feature, a new feature of equivalent size can be swapped into the XP team's iteration in exchange for the un-started feature [8]

Extreme Programming teams work in a strict priority order. Features to be developed are prioritized by the customer (Scrum's Product Owner) and the team is required to work on them in that order. By contrast, the Scrum product owner prioritizes the product backlog but the team determines the sequence in which they will develop the backlog items. Nobody wrote about a Scrum team not choose to work on the highest-priority item. And a Scrum team will very likely choose to work on the second most important. However, at some point one of the high priority items may not be a good fit for the sprint being planned—maybe a key person who should work on it will be swamped by work on higher priority items. Or maybe it makes sense to work on a slightly lower priority item (let's say #10 on the product backlog instead of #6) because the team will be working in the code where #10 would be implemented.

Scrum doesn't prescribe any engineering practices; XP does. I love the XP practices, particularly things like test-driven development, the focus on automated testing, pair programming, simple design, refactoring, and so on. However, I think it's a mistake to say to the team "you're self-organizing, we trust you, but you must do these specific engineering practices...." This sends a mixed message to the team that causes confusion. I love the XP practices but don't like mandating them. Nobody wrote about the value on their own.

These are small and often subtle differences between Scrum and XP. However, they can have a profound impact on the team. Our study has proved a thing that "starting with Scrum and then adopting some practices of XP provides best results."

IV. DETERMINING THE BEST METHODOLOGY

The main goal of our study is to determine the best agile SDLC for Bangladesh's software industry. Most of the software firms of Bangladesh works for foreign clients and most of the clients doesn't provide the whole requirement at the starting. So these firms works on projects without knowing the projects' nuts and bolts. So, the situation requires a system that can help to develop a system properly based on provided requirement and keeping scopes to change the system easily. Moreover the new and small scale firms work on small and low budget projects. At most of the cases the estimated time of a project is less than 3 months and resources are very limited. So these firms need a proficient and well managed software development life cycle that can help the team to develop the system within the time and resource and deliver a efficient and satisfactory system to please the client to ensure winning the next project of the client.

In order to compare two methodologies, we first must define a set of attributes that will serve as benchmark. Fernandes and Almida in [3] performed a comparison between agile methods using a set of four attribute

extracted from the IEEE's Software Engineering Body of Knowledge (SWEBOK) Knowledge Areas (KA) and all the principles defined in the agile manifesto we encapsulated as the 5th attribute. Their objective was analyzing the practices proposed by each method to evaluate the coverage degree for the KAs and the agility of the methods.

The KAs selected were

- i. Software Requirements
- ii. Software Construction
- iii. Software Testing
- iv. Software Engineering Management

For each attribute, the authors utilized a qualitative classification system, the criteria labels were: Not Satisfied, Partially Satisfied and Adequately Satisfied see table-3

TABLE III. FERNANDES AND ALMEIDA'S COMPARISON CRITERIA [3]

NS	Not Satisfied	None of the proposed practices or concepts of the method support the attribute or principle.
PS	Partially Satisfied	The proposed practices or concepts of the method support the sub-attribute or principle, although some of its aspects are not considered.
AS	Adequately Satisfied	The proposed practices or concepts of the method entirely support the sub-attribute or principle.

Based on Fernandes and Almeida's criterions we've studied the Bangladeshi Software industry to determine the best methodology for this industry's most of the firms.

Attributes	XP			SCRUM		
	AS	PN	NS	AS	PN	NS
Software Requirements	60%	40%	0	60%	20%	20%
Construction of Software	75%	25%	0	0	0	100%
Software Testing	100%	0	0	0	0	100%
Software Eng. Mgt.	80%	0	20%	80%	20%	0
Agile Principles – Proposed Practices Relation	75%	8%	17%	50%	17%	33%

Our results show that XP performed better on the attributes Software Requirement, Construction of Software and Software Testing while Scrum improved on Software Engineering Management. . The practice of XP that suggests software teams should have a close

relationship with the customer had a positive impact in the Software Requirements attribute; other practices such as Test First and Pair programming reflected good scores as well. Regarding the "agility" of the methodologies the authors are not surprised to find that XP had a better score than Scrum since they recognize that XP has influenced the principles that form the Agile Manifesto.

Finally we conclude different projects demand different requirements and the election of a methodology depends on software project under consideration in a given situation.

V. CONCLUSION

It is evident that agile methodologies have been put to the industry, including in development projects. Industry and research have been shown that agile methodologies significantly increase quality in software. However all software projects are different and a careful evolution should be taken in order to determine the best fit methodology.

For the software firms of Bangladesh XP could be the best agile methodology to be used at most of the projects because most of the firms work on outsourced projects and they need to keep clients informed about everything that's project related. Most of the Bangladeshi firms could not employ enough employee for a project so that they need to best utilize their resources like time and man power and obviously they need to ensure the software quality. So XP could be the best methodology to follow by ensuring some changes based on project needs and resource limitation.

REFERENCE

- [1] Ulises Reyes Blog: "Agile Methods: Scrum and Extreme Programming," Ulises Reyes Blog: Agile Methods: Scrum and Extreme Programming. N.p., n.d. Web. 15 Jan. 2014.
- [2] Shen Bin "Agile Methods (Scrum, XP) Applying into Small (Micro) Enterprise Business Website development", P 13-17.
- [3] Fernandes, JoãoM, Almeida, Mauro, "A Technique to Classify and Compare Agile Methods", P 385-386.
- [4] Mohammad Shahidul Islam, Sentayehu Tura, "Exploring the difference between Agile and Lean: A stakeholder perspective, Scrum + Engineering Practices: Experiences of Three Microsoft Teams, Laurie Williams, Gabe Brown, Adam Meltzer, Nachiappan Nagappan", p 9-11
- [5] Akif R. H. Majeed , "Issues and Challenges in Scrum Implementation" International Journal of Scientific & Engineering Research, Volume 3, Issue 8, August-2012 ISSN 2229-5518.
- [6] Jeff Sutherland, Ph.D. Ken Schwaber, Co-Creators of Scrum "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework" 2007.
- [7] Mark C. Paulk, "Extreme Programming from a CMM Perspective". (2009) 21-27. Print
- [8] Lindstrom, Lowell, and Ron Jeffries. "Extreme Programming and Agile Software Development Methodologies." Information Systems Management 21.3 (2004): 41-52. Print.
- [9] "Scrum & Agile Training and Consulting That Helps You Succeed." Scrum Training Agile Training from ScrumMaster Mike Cohn. N.p., n.d. Web. 6 Apr. 2007.
- [10] ADRIAN (DECEMBER 28, 2006) SCRUM SOFTWARE DEVELOPMENT PROCESS, Available at: <http://kosmaczewski.net/scrum-software-development-process/> (Accessed: 6th July 2014).