Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 2, No.3

An Improved & Adaptive Software Development Methodology

Mohammad Naderuzzaman (Corresponding author)
Faculty of Computer Science and Engineering
Atish Dipankar University of Science and Technology
Banani, Dhaka, Bangladesh
nader_u@yahoo.com

Fazley Rabbi
Faculty of Computer Systems and Software Engineering
University Malaysia Pahang
Pahang, Malaysia
fazley.rabbi@ymail.com

Abul Hashem Beg
Faculty of Computer Systems and Software Engineering
University Malaysia Pahang
Pahang, Malaysia
ahbeg_diu@yahoo.com

Abstract:
The methods of software development have increased a lot from the beginning. From the first waterfall to current agile methodology there still have some drawbacks. For this reason, the software delivery is still a very challenging and heavy-duty work. In this paper, we proposed a new software development methodology which is easy to implement and will help software development companies a secure and robust software releases. The proposed SDLC process is known as 4A. The empirical result shows that the proposed methodology is more adaptive and flexible for developers and project managers.

Keywords: development method, software development, SDLC Process.

1. Introduction

Software development methodology is a framework that is used to structure, plan, and control the process of developing an information system - this includes the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application [1]. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One software-development methodology framework is not necessarily suitable for use by all projects. Each of the available methodology frameworks is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations [2]. A software-development method is said to be an agile software development when a method focuses on people, communication-oriented, flexible (ready to adapt to changes in expected or unexpected at any time), speedy (to support rapid and iterative development of the product in small versions), lean (focuses to shorten the time and cost and quality improvement), responsive (responding to correctly anticipated and unexpected changes), and learning (focus on product improvement during and after development [5].

In all the software-development methodology there have some advantages and some disadvantages. Different methodology is good for different sort of projects. As the technology changing by the day, therefore, always have the chance to come up with the better methodologies or modify existing methodologies. In this section, it has been shown some short coming of different software-development methodologies:

1.1 Waterfall:

i. Project is divided into sequential phases, with some overlap and splash back acceptable between phases [3]
ii. Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time [2]

1.2 Prototype:

i. Not a standalone, complete development methodology approach, but rather an approach to handling selected portions of a larger, more traditional development methodology (i.e. Incremental, Spiral, or Rapid Application Development (RAD)) approaches.

ii. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process [2]

1.3 Incremental:

i. It doesn't just say that requirement's definition is hard; INCREMENTAL says it shouldn't even be attempted. INCREMENTAL views requirement changes as inevitable and expected. Everybody seems to agree that customers often don't know what they need, but to say that requirement's definition is futile is equivalent to saying those not only are customers incapable of articulating their needs, developers are incapable of meeting them. Do you want to work under a methodology that believes you cannot successfully communicate with your customer?

ii. Allowing customers to "change their minds" means that software-development outfits are expected to incur the cost of the unlimited wants of consumers. If customers cannot articulate their needs, they probably cannot distinguish between a "need" and a "want", but will expect you to foot the bill just the same. Unlimited wants combined with limited means is the classic "economizing problem" - and the resource that gets used up first is "time" how much "time" will it take", how much "time" do we have, how much "time" until we demo.

iii. And most importantly, INCREMENTAL is micromanagement. It does not respect programmers as professionals - it treats them like assembly line workers [2]

1.4 Spiral:

Too much focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments [4]

1.5 RAD:

i. Attempts to reduce inherent project risk by breaking a project into smaller segments
ii. Key emphasis is on fulfilling the business need, while technological or engineering excellence is of lesser importance [2]

In this paper, the SDLC process has been changed and proposed a new SDLC process known as 4A. The proposed 4A process has been compared with the existing SDLC process.

2. Related Work

The framework of a software-development methodology consists of:

i. A software development philosophy, with the approach or approaches of the software development process
ii. Multiple tools, models and methods, to assist in the software development process

These frameworks are often bound to some kind of organization, which further develops, supports the use, and promotes the methodology. The methodology is sometimes documented in some kind of formal documentation. One of the oldest software-development tools is flowcharting, which has its roots in the 1920s. The software-development methodology didn't emerge until the 1960s. According to Elliott (2004), the Systems development life cycle (SDLC) can be considered to be the oldest formalized methodology for building information systems. The main idea of the SDLC has been "to pursue the development of information systems in a very deliberate, structured and methodical way, requiring each stage of the life cycle from inception of the idea to delivery of the final system, to be carried out in rigidly and sequentially"[2]. The main target of this methodology in the 1960s has been to develop a large scale functional business system in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines [2].

Every software-development methodology framework acts as a basis for applying specific approaches to develop and maintain software. There are a number of software-development approaches that have been used since the origin of information technology. These software-development approaches are: [2]

i. Waterfall Methodology: linear framework type
ii. Prototyping Methodology: iterative framework type
iii. Incremental Methodology: combination of linear and iterative framework type
iv. Spiral Methodology: combination of linear and iterative framework type
v. Rapid Application Development (RAD) Methodology: Iterative Framework Type
vi. Extreme Programming Methodology

For a long time, many people have ruined the method and model of software development. And they also tended to mix the method of software development to develop processes for improvement. For example, they assume that the RUP (Rational Unified Process) and CMMI (Capability Maturity Model Integration) are the method of software development. Another opinion is that the method is the traditional, process-oriented and object-oriented is modern, and it is assumed that they are not related. It is a common phenomenon that people are confused with the concept of a method of software, which is a fundamental concept in the field of software engineering.

[Definition1] "Method" or "methodology" refers to a series of related "methods and technologies," that follows some "principles and strategies". They can be "development methodology" and "the methodology of the process." "Principals and strategies" may be the development strategies, including the famous "functional decomposition" strategy and strategies iterative process as a model. Method development process of developing methods to contain the method and object-oriented development, and also includes the method of formalization and clean room software engineering [6, 7]. To improve software reuse, component-based method has attracted attention both in the search area [8].

[Definition 2] "Development model" refers to strategies and how the whole process should continue. This is the strategy or model for the process. Forsberg, K. noted in its "Perceptible Project Management" that "the model is to clarify the relationship and determines the essential elements and thus reduces the possible misunderstanding," which is one of the reasons to build several models. Development model includes a waterfall, incremental model, evolutionary model, spiral model, the prototype model and iterative model RUP, etc. They all have their own development paths and define the specific steps used in the development. CMMI is a process improvement model. The process has to respect his methods and approach guides the process. Therefore, they are closely related and cannot be separated.

In recent years, many methods have been developed. The literature [9] discusses software engineering methods and other targeted agents such as Aspect-Oriented (AOP), Agile, SOA, etc., Until now, the methods have been proposed are varied and very prosperous. However, the metadata-oriented method is rarely mentioned. Metadata is a point of research in software engineering [10] and many information systems were built based on metadata. The metadata-based approach has been applied in e-government systems [11], light ORM framework design [12], the management system [13], etc.

Wolfgang Ketter et al. [14], proposed a new agile method for EMU component development. The Mash | match method for agile development of EMU, the components are defined based on the comparison of twelve methods of agile development. They have shown that certain aspects of the business agile methods tested can be used in

the case of Mash | find a methodological approach. Four existing agile methods are adapted to the new methodological approach - Mash | Match is XP, PP, FDD, and DSDM. The study notes that XP has the greatest influence on Mash | Match (9 out of 12 factors), followed by PP FDD and DSDM. Mash | Match method can be applied in organizations with business development organization rigid and smaller organizations that have an ad-hoc approach to development. For the first time, the method can be applied to the evolution of front-end, and second, to provide a methodology, while maintaining the flexibility of ad-hoc development.

3. Proposed 4A Method

It is not always possible to solve all the problems in a single methodology. However, it is possible to improve existing methodologies or create a new methodology from the scratch. In this paper, we have changed the SDLC process. Our proposed SDLC process is known as 4A. Figure 1 shows the structure of the 4A process.

3.1 Analysis phase:

1. Analyze the whole requirements and system.
2. Divide the requirement into small tasks which can be accomplished within 3 hour or 6 hour.

3.2 Architecture phase:

1. Design the whole system, create a different diagrams
2. Make the target date of final release
3. Create coding plan
4. Create the testing plan
5. Create Documentation plan
6. Create the delivery plan
7. Create the support plan

3.3 Acquaintance phase:

1. Build the coding team, testing team
2. Mentor and help the team understand the whole system and expected output.

3.4 Apply:

1. Implement Coding
2. Implement Documentation
3. Implement Testing
4. Delivery

The Entire diagram will be looked like in Figure 2:

From Figure 2, in the analysis phrase, the system analyst or project development manager will analyze the whole system and break down into smaller tasks. The tasks will be sliced in such a way that one single task can do within a day or half day. The reason behind it is, the development manager can keep track of the number of tasks and number of days easily and calculate the tentative delivery date. In the architectural phrase, the system architect, will create and plan everything for the system. S/he will do everything with the team leader, as the team leader will have a better idea about the team. In an acquaintance phrase, the project manager/development manager will create the development team, testing team, GUI team, database team, etc. And finally, The tasks will be distributed among the team. In apply the phrase, the sprint starts and the team lead with the team members will start developing each task concerning the deadline. They can use the white board or excel sheet to keep track of their time. After each sprint there will have discussion (retrospective), code review among the teams and other activities such as team outing.

4. Empirical Result and Discussion

Software professional's satisfaction is important to secure high-quality software development. Keep in mind this is the case has been contacted several software and web development companies, which has an offshore development center in Bangladesh. The survey among all developers, designers and QA team members has been

done. Table 1 shows the number of professionals and their designation, with whom the mapping has been accomplished.

From Table 2 and Table 3, it can be clearly seen that 4A has outperform in each case.

5. Conclusion

This paper proposed a new software-development methodology known as 4A. 4A is easy to implement for software-development companies. The empirical survey shows that 4A is useful for two projects comparing waterfall and scrum. More surveys can be done throughout other software-development companies.

References

Geoffrey Elliott. (2004). Global Business Information Technology. p.87.

Centers for Medicare & Medicaid Services (CMS) Office of Information Service (2008). Selecting a development approach. Webarticle. United States Department of Health and Human Services (HHS).

Markus Rerych. (2007).Wasserfallmodell Entstehungskontext, Institut für Gestaltungs- und Wirkungsforschung, TU-Wien.

Barry Boehm .(1996). A Spiral Model of Software Development and Enhancement. In: ACM SIGSOFT Software Engineering Notes (ACM) 11(4):14-24.

Roy morien, Pornpit, W. (2008). Supporting Agility in Software Development Projects Defining Project Ontology. IEEE conference on Digital Ecosystems and Technologies, 2008. DEST 2008, on pp. 229 – 234.

Roger S. P. (2000). Software Engineering, A Practitioner's Approach, Fifth ed., McGraw-Hill.

Chilong, Z.; Lin, Y.; Sun W. (2006). Practical Software Engineering, 2nd ed., Electronic Industry Publishing House.

Yang Fuqing. (2005). Deliberation on the development of software engineering method", Journal of Software, vol.16, no.1, pp.1-7.

Qiang, H.; Xiaomei, Z.; Ming, R.; Lou, X. (2007). A light-weight ORM frame design based on type meta-data", Computer Engineering, vol.33, no.22, pp. 54-59.

Zhong, L.; Xingming, Z. (2007). The meta-data management method based on the directory path", Journal of Software, vol.18, no.2, pp. 236-245.

Yongtao, X.; Yanzhang,W.; Xuelong, C.. (2007). Meta-data research on the E-Government office systems", Computer Engineering and Design, vol.28, no.4, pp. 907-913.

Qiang, H.; Xiaomei, Z.; Ming, R.; Xinyuan, L. (2007). A light-weight ORM frame design based on type meta-data", Computer Engineering, vol.33, no.22, pp. 54-59.

Hao, H.; Xiaowen, L.; Suhong, L.(2007). Meta-data management system design and implementation based on J2EE", Computer Application Research, vol.24, no.5, pp.198-203, May 2007.

Ketter, W.; Banjanin, M.; Guikers, R.; Kayser, A. (2009). Introducing an Agile Method for Enterprise Mash-Up Component Development. IEEE Conference on Commerce and Enterprise Computing,IEEE Xplore, pp. 293-300.

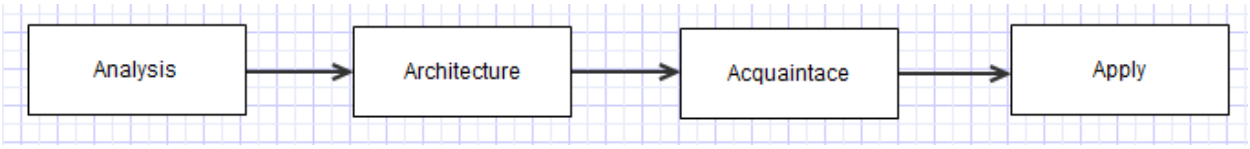

Figure 1: 4A structure

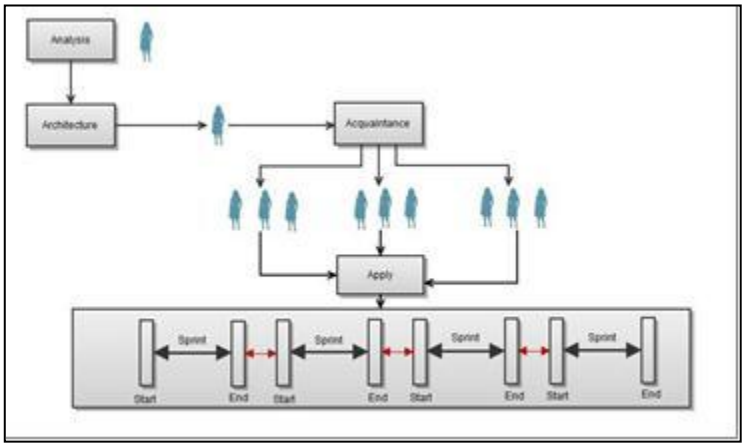

Figure 2: Architectural process

Table 1: Survey report on Improved Agile and Current Agile

Profession	No of Professionals
Manager	3
Architect	2
Sr. Developers	20
Developers	30
Jr. Developers	40
QA	20
Artist/Designer	20

Table 2: Scrum and 4A

Project Name: Obachegou, Brazil		
Number of sprint: 20		
Experiment	**Scrum**	**4A**
Successful Delivery/Sprint	15	18
Mitigated Risks	2	4

Table 3: Waterfall and 4A

Project Name: **iNetFoods, UK**		
Number of sprint: 25		
Experiment	**Waterfall**	**4A**
Successful Delivery/Sprint	12	21
Mitigated Risks	6	6