International Journal of Computer Applications Technology and Research
Volume 5– Issue 7, 465 - 471, 2016, ISSN:- 2319–8656

Suitability of Agile Methods for Safety-Critical Systems Development: A Survey of Literature

Mary Walowe Mwadulo
Department of Information Technology
Meru University of Science and Technology
P.O BOX 972-60200 Meru, Kenya.

Abstract: Lately, agile methods have widely been used in large organizations. This contrasts to previous practice, where they were mainly used for small projects. However, developers of safety critical systems have shied away from using these methods for the right and wrong reasons. Adoption of agile methods for safety critical system development is low and there is need to find out why this is so especially since agile methods allow a more relaxed approach towards documentation, flexible development lifecycle based on short iterations and accommodates changing requirements. This paper presents a report of a detailed analysis of literature and aims to shed light on the suitability of agile methods for developing safety critical systems .The findings indicate that many organizations are relying on traditional methods to develop safety critical systems because they are familiar with them and have been thoroughly tested over time. However with the advent of agile methods there is a paradigm shift by non safety critical system developers, nevertheless this is not happening with the safety critical system developers and there is need to find out why.

Keywords: Agile methods, agile methodology, safety-critical systems, Suitability

1. INTRODUCTION

A safety critical system sometimes referred to as life critical system is a system whose failure or malfunction can cause harm or is responsible for preventing harm [1]. The high costs of failure of safety critical systems means that trusted methods and techniques must be used for development. Consequently, safety critical systems are usually developed using well-tried techniques rather than embrace new techniques and methods such as agile. [2],[3] indicate that about 80% of respondent organizations were following an agile approach because researchers and practitioners wanted a method which would replace the bureaucratic traditional methods. This sharply contrasts to previous practice where agile methods were used for small organizations developing small applications. The interpretation of the word agile methods varies from one researcher to another. In [4] it is an umbrella for well defined methods which also vary in practice. Notably, Safety critical system developers have shied away from applying agile methods and very little rigorous research existing in this area [2]. Developers have continued to apply traditional methods such as Waterfall, V-model and Iterative and Incremental approach [1], [5].

Adoption of agile method for safety critical system development is low. In [5] only 25% of organizations develop software in accordance with agile practices, thus there is need to find out why this is so especially because agile methods allow a more relaxed approach towards documentation, flexible development lifecycle based on short iterations and accommodates changing requirements which are a great concern for the traditional methods.

The paper aims to shed light on suitability of agile methods for developing safety critical system.

In [1] agile methods will be suitable for safety critical systems when they are tailored and customized to ensure that safety

objectives are met. However, there is limited support for developing safety critical software because quality control mechanisms supported by current agile processes have not proven to be adequate to assure users that the product is safe. Also, agile practices such as code refactoring, minimal documentation, iterative development, conservative nature of critical systems developers and not wanting to make internal operations public have been identified to be unsuitable for safety critical systems development and have deter agile adoption [1], [5].

The remaining part of this paper is structured as follows: section 2 presents safety critical systems, section 3 presents agile methods, section 4 presents the discussion and section 5 presents conclusion.

2. SAFETY CRITICAL SYSTEMS

The main concern with safety critical systems is the consequence of failure. If the failure of a system could lead to consequences that are determined to be unacceptable, then the system is safety-critical. In essence, a system is safety-critical when we depend on it for our well being. An example of safety-critical system failure which was reported by Ben-ari M. is Ariane V launch failure caused by a software error. As such, safety critical systems must be certified by a regulatory agency to ensure that they are fit-for-purpose. This ensures proper development practices have been applied to promote system correctness as the final outcome. It is important that adherence to the objectives of the relevant standards can be demonstrated [1]. These standards identify the objectives that a system or project must meet before it is allowed to be deployed in its operational environment. Because it is virtually impossible to demonstrate deterministic correctness for any significant piece of software, most of these standards have concentrated on specifying process objectives and requirements for evidence that the processes have been followed [1]. Some of the standards include: DO-178C (Software Considerations in Airborne Systems and Equipment Certification), DO-331 (Model-Based Development and Verification) and DO-332 (Object Oriented Technology and Related Techniques). Although these standards specify the objectives that any process must meet if it is used to develop a safety-critical system, the standards do not specify the processes themselves [1]. As long as a process can be demonstrated to meet the needs of the relevant standard, the development team is free to use whatever processes they desire. This leaves the option available to use agile methods, with their accompanying advantages, provided that the safety objectives can be achieved.

In [1],[5] 50% of the organizations are developing software in accordance with the V-Model, 25% in accordance with agile practices and remaining 25% in accordance with other development lifecycles such as the Waterfall, and Iterative & Incremental approaches. This implies 75% of organizations are using traditional methods which involve huge effort in planning and documenting yet little time is spent in actual development. A lot of time is spent in early planning, more than is justifiable given that little information is available at initial stages of a project. Agile holds that this work should be done in a way that discovers and repairs defects immediately so that the emphasis is on defect avoidance in the product itself, rather than production of documentation.

3. AGILE METHODS

Agile methods were introduced as an alternative to traditional methodologies, which had a lot of documentation and were too restrictive when dealing with changing requirements. In response to these concerns agile methods offer a more relaxed approach towards documentation and provide a flexible development lifecycle based on short iterations.

Plan-driven methodologies have proven to be valuable and useful in safety critical projects, the evolving market of software products of the last few years puts this approach to the test. A growing competition, ever changing technologies and more diverse groups of clients have changed the expectations towards software development methods [3]. The need to deliver systems of acceptable quality, faster and at lower cost in comparison to competitors evoked seeking an alternative.

Table 1: Summarized comparative study of the different agile methodologies:

No.	Agile Methodology	Team size	Iteration Length	Support distributed team	System Criticality	Application area
1	Extreme Programming (XP)	Small to medium size	2-3 Weeks	No	Not geared for one system	Projects that require 2 -10 programmers
2	SCRUM	7-10 people	4 Weeks	Yes	Not Addressed	Software and non software projects
3	Feature Driven Development (FDD)	4- 20 people	1 – 4 Weeks	Yes	Not addressed	Large complex banking projects
4	Dynamic Software Development Methodology (DSDM)	2-6 People	Not addressed	Not addressed	Not addressed	Used in Europe
5	Adaptive Software Development	Determined by scope and size of the project	Determined by project schedule and the degree of uncertainty	Yes	Addresses: Risk analysis and aversion techniques	Has not be used as a methodology to develop a system
6	Crystal methodologies	Any team size, highly skilled and experienced	4 Months for large, highly critical projects	Yes	Addresses: Failure resulting in loss of money and life	Used in internet banking
7	Agile Modelling (AM)	Depends on the development process being used	Depends on the development process being used	Depends on development process being used	Depends on the development process being used	No record of current or previous use of the methodology

3.1 Agile methods used for developing safety critical systems

Several agile methods exist, but Extreme programming and SCRUM have been applied in the development of safety critical systems.

3.1.1 Extreme Programming (XP)

Extreme Programming targeted small co-located teams developing non critical products. It guides the developer through planning, coding, designing and testing phases. The purpose is to deliver what the customer needs at the time it is needed, emphasize on team work and accept changes anytime [6]. XP is suitable when requirements are unclear or dynamic and the project has high risks. However, it is unsuitable when the team is large, low cooperation between the developer and the customer and testability is not done throughout the project.

In [7] used a select number of XP practices during the development of safety critical systems and reported to have had a 53% improvement in average quality compared to the plan-driven software development projects.In [8] describes how XP practices are used in a large company developing safety critical system. He suggests that some XP practices, such as simple designs integrated

with test first development and refactoring work quite well in the safety critical area.

3.1.2 SCRUM

SCRUM methodology was initiated by Ken Swaber in 1995 [6]. It has project management as part of its practices with the aim of simplifying project control through simple processes. The team does not move to a new phase unless the current one is complete. It is not suited for products where the focus is on usability [6].

In [9] presented a novel idea to integrate SCRUM into safety cycle to enable iterative incremental development in safety critical systems.

3.2 Merits of using agile methods for safety critical systems development

Agile methods have a reputation for being fast and adaptive but undisciplined and lacking in robustness. However, agile methods require a great deal of discipline, and these practices enhance both quality and team productivity. Because of this, agile development practices should be applied to the development of safety-critical systems.

Douglass [1] identified secondary benefits to using agile methods which include improved productivity, improved time to market, improved customer satisfaction and decreased development costs.. This is achieved by getting the development right the first time, and by doing things in such a way that you can verify them as you are doing them.

Verifiable artifacts can be incrementally created using agile methods which makes analysis, simulation and testing easy. The construction and verification of components during system development is a key benefit that agile brings; this iterative process offers the sort of quality improvement needed in safety-critical systems at a (relatively) low price.

Vouri [10] analyzed general agile values, principles and practices against general principles of safety-critical software development and concluded that many features of agile development can be beneficial for creating truly safe systems. He identified incremental release, reduced documentation and increased customer participation as some of the benefits of agile methods. In [11] identified benefits such as early return on investment, short time to market, improved quality, enhanced client relationships and better team morale.

3.3 Demerits of using agile Method for critical system development

There is some doubt that agile methods alone cannot be sufficient to handle development of safety critical systems .The quality control mechanisms supported by current agile processes have not proven to be adequate to assure users that the product is safe

Agile practice of having minimal documentation would have a detrimental impact on the development of safety critical systems because it is crucial to ensure traceability. Traceability helps to establish compliance to standards and regulations [2]. McHugh [5] attempted to solve the traceability issue by suggesting a tool “echo” that provides a mechanism to maintain traceability between the requirements and each stage of development while developing software in accordance with agile practices. However, [1], [2] suggested adding traceability links. Links are automatically established as developers check in code that implements a certain task.

Refactoring is another agile practice that would not fit naturally with the development of safety critical systems. If code is refactored on a critical system, it has the potential to invalidate previous certification or security analysis. This would cause extensive rework and would need to be avoided, whenever possible.

For safety critical system development there is need to have up front planning so that certification and safety analysis can be carried out early in the project. However, up-front planning can be difficult to perform following agile practices as requirements are volatile changes are welcomed and expected in an agile project. McHugh [5] has recommended before a project begins agile practices it can use techniques such as user stories.

Iteration is an agile practice that allows a project to be released in piecemeal which helps handle complexity. These iterative,

incremental ways of working are significantly better at producing software with fewer defects in less time than serial waterfall approaches [1]. However, regulatory standards prohibit developers from releasing software to a live environment without been fully tested [1].

3.4 Adopting agile methods for safety- critical systems

There is a growing body of evidence supporting the theory that incorporating agile practices into safety-critical projects is not only feasible but also potentially profitable. Lukasiewicz [3] made a literature survey on applying agile methods in regulated environments and found only a small number of publications which, they think "could indicate a very low level of adoption of agile methods in regulated safety critical domain; however it may indicate a reluctance of companies in these domains to make their internal practices public". In their paper, they reported some issues with agile methods and suggested solutions to make agile work.

Paige [12] studied agile in the development of high-integrity systems, including the analysis of elements in agile and the adaption of agile processes. Their main finding was that agile methods can be adapted to safety-critical development by not replacing plan driven processes, but applying them in appropriate tasks.

Gary [13] suggested agile methods are suitable for open source safety critical software, because these methods are synergistic with safety principles, not orthogonal to them. Agile methods bring strong practices in the area of process management and software construction, while having a philosophy that allows for traditional safety-oriented practices to the extent they are warranted. This reinforces Boehm [14] argument comprehensive project management is required for safety-critical software development.

In [15] analyzed application of agile methods in aerospace industry, however there results cannot be generalized due to specific requirement in aerospace development. They concluded that agile methods can be applied but more cooperation is needed within the aerospace community.

In [16] presented an approach on how incremental methods can be used in safety critical development. They claimed that agile methods can provide benefits, but the methods are not directly applicable in regulated areas. They suggested an upfront design in the process that at least produces information for a hazard analysis, before the agile portion of the process begins. The iterations of the software that the agile process produces also needs to include sufficient arguments that the software releases are sufficiently safe. For large scale development they propose a modular system where the modules are dependent on each other by arguments.

Pikkaraine [17] found that Agile Assessment is an efficient method to clarify what agile practices are suitable for the organization's product development and customer co-operation. Another finding was that the use of the best suitable agile practices would improve incremental development monitoring and traceability of requirements.

Douglass [1] discussed six steps to successful agile adoption which correlates with Sidky [11] three main stages. They agree, there is need to know what makes the agile software suitable for them.

4. DISCUSSIONS

In this review I presented what is currently known on using agile methods for developing safety critical systems. Results show that there has been little research in this area and the research available is subjective with no empirical evidence to support the findings. Most studies conducted were based on case studies with no rigorous controlled experiments. Therefore, it seems that evidence on determining suitability of agile methods for safety critical systems development needs more research.

Researchers attempted to determine suitability through identification of agile practices or developing framework that points out how agility should be handled. Sidky [11] looked at suitability in terms of agile practices. They discussed three stages (Making the Go/No-go decision, discarding inappropriate

practices and determining the right practices to adopt) that will enhance applicability of agile practices to mission and life critical systems. They recommended that minimal documentation and evolvability of requirements is unsuitable for safety critical systems although it is emphasized a lot in agile software development. McHugh [5] agrees in addition to regulatory compliance, lack of upfront planning and the process of managing multiple releases. He makes recommendation on how these barriers can be overcome and points out a research gap in identification of critical success factors for using agile practices when developing medical device software.

Lukasiewicz [3] discussed risks posed by introducing agile practices to safety critical software's. He concluded that agile methodologies should be regarded as complementary to plan driven practices instead of being the replacement. However, his point of view was from a software engineer perspective and the research was not conclusive since the data collected during the experiment was not yet finally processed. Therefore what was presented reported on the scope of the raw data collected than the final conclusion derived from the data.

Vuori [10] analyzed the agile principles and processes and gave guidance on how organizations could change their processes to a more agile way without risking the safety or marketability of the products or causing increased product and liability risks. However the unanswered question how an organization will know it needs to be agile

Djik[18] designed frameworks which help software practioners determine whether a software project is suitable for an agile method. He discussed two contingency factors; influence of limitation of agile methods and organization capability to handle agility which is determined by the culture values of the organization and the individual capabilities of the team. However suitability is not a term which is expressed in observable, quantifiable factors, but rather a scale where complete suitability only exist in an ideal situation. Then the big question is how to measure suitability? Yet there is no empirical research that has been conducted which links observable values of contingency factors to methodology selection. Though he concluded the model was suitable for the particular environment, the model was not validated. In [9] proposed a novel idea for developing safety-critical software-intensive systems by the use of Scrum into the safety lifecycle to enable iterative incremental development in safety-critical systems. While these models of adapting agile practices to suit safety-critical projects are valuable sources of knowledge, there is still a need to develop a more easy to use and thorough set of guidelines for safety-critical software companies that would like to adapt agile practices into their project development.

Attempts made to replace traditional methods with agile methods fail and [3] suggested that agile methodologies should be regarded as complementary to plan driven practices instead of being the replacement.

5. CONCLUSIONS

The low adoption of agile methods for safety critical systems development is as a result of developers of these systems being too conservative and wanting to use the traditional methods because they have been tested and they are familiar with. This cannot be entirely blamed on them given the fact that the consequence of failure of such systems can be catastrophic. There is also the reason of an organization not wanting to make its internal operations public and as such would want to use a method that they already know. Also agile practices such as minimal documentation, refactoring of code, upfront planning and iterative release of project contradicts safety requirement standards of safety critical systems. However, agile methods can help improve both quality and productivity and can be employed in the development of safety-critical systems. In safety-critical development, the key concern is safety, and in agile methods, the paramount concern is quality thus, there is no contradiction. Success stories of using Extreme programming and SCRUM agile methods have been documented though most of them were based on case studies. Safety-critical systems are difficult to develop. In addition to normal concerns about quality and time-to-market, safety critical systems must also meet the demanding objectives of relevant safety standards and are subject to rigorous certification. However, if an agile method is chosen carefully, the benefits would be tangible and, despite the concerns, actually increase the

chance of developing a stable safety critical system. Importantly, agile practices should be tailored to the needs of safety-critical systems development.

6. REFERENCES

[1] Douglass, B.P., and Ekas, L.. Adopting agile methods for safety-critical systems development. IBM, 2012. http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&htmlfid=RAW14313USEN

[2] Fitzgerald B and Stol K. and Sullivan R. and Donal O. Scaling agile methods to regulated environments: An industry case study, 2013.

[3] Lukasiewicz K. assessment of risks introduced to safety critical software by agile practice: a software engineer's perspective, 2012.

[4] Pathak K. and Saha A. , 2013. Review of agile software development methodologies

[5] McHugh M. Integrating Agile Practices with a Medical Device Software Development Lifecycle, 2012.

[6] Hneif M. and Ow S. Review of agile methodologies in software development. International Journal of Research and Reviews in Applied Sciences, 2009.

[7] Drobka,. Piloting XP on Four Mission-Critical Projects. IEEE Software, 2004.

[8] Greening, J. Launching XP at a Process-Intensive Company. IEEE Software, 2001.

[9] Guo Z. and Hirschmann . An Integrated Process for Developing Safety-critical Systems using Agile Development Methods, 2012.

[10] Vouri M. Agile development of safety critical software, 2011.

[11] Sidky, A. and Arthur, J. Determining the applicability of agile practices to mission and life-critical systems. In *Proceedings of the 31st IEEE Software Engineering Workshop*, SEW '07, pages 3–12, Washington, DC, USA. IEEE Computer Society,2007.

[12] Paige R. et al Towards Agile Engineering of High-Integrity Systems. Proc. of 27th International Conference on Computer Safety, Reliability and Security (SAFECOMP) 2008.

[13] Gary et al. Agile methods for open source safety critical software, 2012.

[14] Boehm B. Get ready for agile methods with care.,2002.

[15] VanderLeest and Butler (2009)

[16] Ge, X., Paige, R. F., and McDermid, J. A. (2010). An iterative approach for development of safety-critical software and safety arguments. In *Proceedings of the 2010 Agile Conference*, AGILE '10, pages 35–43, Washington, DC, USA. IEEE Computer Society.

[17] Pikkarainen M.An approach for assessing suitability of agile solutions: A case study, 2005.

[18] Djik V. Determining the suitability of agile methods for a software project, 2011.