

# The Need for Novel Development Methodologies for Software Projects in Universities: A Sri Lankan Case Study

D. M. P. Dias, N. D. Kodikara, and Mahen Jayawardena

**Abstract**—Software engineering requires coordination across a number of disciplines and is concerned with theories, methods and tools for professional software development. Project management is considered as the main function which facilitates better utilization of knowledge, skills, resources, tools and techniques in any such project to achieve its intended objectives. But, there is no such standard method that can be used in every project. Project Management practices mainly depend on the industry it is been used; i.e. construction, manufacturing, marketing, auditing, software engineering, etc. For software engineering there are number of standard Project Management methods used in different circumstances. Nevertheless, it was difficult to find a specific methodology that can be used in a university context, which is more focused on innovations, research and learning. The paper aims to motivate why a novel software product development methodology should be developed for used within universities, such methodology may even be possible to be later implemented within the software industry, where innovation is a major ingredient.

**Index Terms**—Software engineering, project management, commercial product development, innovations.

## I. INTRODUCTION

Software engineering (SE) involves wider responsibilities than simply the application of technical skills. Software, the end result of such a process must be designed to satisfy both present and future needs and should ideally last many years in a changing environment. The systematic approach utilized can be different due to several factors such as; development environment, client requirements, developer capabilities, resources available, complexity, size, reliability, criticality requirements, etc. Different Project Management (PM) methodologies have been defined for various software development projects, after considering these factors. Hence, there cannot be a single PM process which works ideally for each and every project.

There is a difference between PM practices in software development at universities and in industry. The success of a university project is mainly depends on interest of the project team, innovative ideas and inventions. When we look at several well-known software projects, unlike other industries, most of them have started with an innovation or limitless interest of an individual or a group of colleagues [1], [2], [3].

Manuscript received January 20, 2013; revised March 19, 2013. This work was supported in part by the National Research Council, Sri Lanka under Grant 12-074.

The authors are with the University of Colombo School of Computing, Colombo 00700, Sri Lanka (e-mail: dmp@ucsc.cmb.ac.lk, ndk@ucsc.cmb.ac.lk, mcj@ucsc.cmb.ac.lk).

Therefore, it may be difficult to use a standard methodology for a new software product until it stabilizes up to some extent.

The Sri Lankan software industry is detailed in the next section. A review of existing project management methodologies are discussed in section III and the reasons that prevent the use of formal PM methods for research and development projects will be discussed with case studies in section IV of the paper.

We try to motivate also why a single best project management methodology may not suite the full life cycle of a software product and provide future directions and conclusion in Sections V and VI.

## II. SRI LANKAN SOFTWARE INDUSTRY AND UNIVERSITY EDUCATION

In the study of Adaptation analysis of Agile Project Management (APM) for managing IT projects in Sri Lanka [4] they have stated that the traditional PM methods are no longer applicable and effective to manage complex IT projects in Sri Lanka as they are unable to cope with rapid changes. According to the literature, most of the IT projects fail not due to technical issues but due to people and PM issues. Applying PM practices which are designed for other disciplines is another main reason to failure of IT projects in Sri Lanka. As given in the results of their study by [4], awareness level of APM practices among Sri Lankan project managers is not satisfactory. It was found that only 25% of the project managers are aware of APM, others are outdated. It was also found that the expertise available in Sri Lanka for APM is not satisfactory and there are no facilities available to provide the required training.

According to Munasinghe et.al. [5] Sri Lanka has the opportunity to take advantage of the shift of the software industry from USA towards Asia. India has been able to take the best advantage due to the availability of trained human capital, infrastructure support and appropriate policy framework [5]. In Sri Lanka only a few universities/institutions facilitates high-end IT education and training. Most local universities/institutes do not have adequate connections with industry. There is a lack of industry-based training in the curriculums. Universities give high technical theoretical knowledge, but less experience about how they could be applicable in the industry. Therefore, most of the graduated students do not meet industry standards. As identified by [5], a major drawback in Sri Lanka is the nonexistence of a proper policy to keep up the standards of

computer education and examinations. Another main issue is the minimal number of research carried out in this area in Sri Lanka.

Certain Sri Lankan universities are involved in commercial software development projects. UCSC's<sup>1</sup> Advance Digital Multimedia Technology Center (ADMTC), Wireless Ad-Hoc and Sensor Network (WASN) Laboratory, e-Learning center (eLc) and Language technology research laboratory (LTRL) function both as research and development (R&D) units. Undergraduate and postgraduate students at UCSC are given the opportunity to involve in the software development projects conducted by these centers (e.g.: Vidusayura at ADMTC). Sri Lanka institute of information technology (SLIIT) operates the Sri Lanka Technology Incubator also known as conceptnurs-ery.com by giving the opportunity to their students to work in this facility by engaging themselves in industry owned R&D projects. Information technology research unit (ITRU) at University of Moratuwa (UOM) is their center for IT R&D. Information technology center at University of Peradeniya (UOP) involves in Information and Communication Technology education, research, development and associated services for their students.

### III. ANALYSIS OF FORMAL PROJECT MANAGEMENT METHODOLOGIES

A project can be defined as, a temporary endeavor undertaken to create a unique product, service or a result. Irrespective of the size and complexity, all projects can be mapped to a common life cycle structure; i.e. project starting, project planning, project executing and closing [6]. A software project should be closed on time, within budget and according to agreed specifications. Therefore, funds, staff, ideas/plans, experts, materials, workspace, equipment and customer are considered as the interrelated parts of a project [7]. Smooth coordination of the above factors is even more essential. Further, external influences such as; client deadlines, liability issues, public and ethical concerns, government and societal issues and laws could make a significant impact on project life cycle. Some of the known PM practices are PMBOK® [8], PRINCE 2 [9] (Table I), Scrum [10], Extreme Programming [11] (Table II), MSF - Microsoft Solution Framework [12] and ITIL - Information Technology Infrastructure Library [13] (Table III).

TABLE I: PMBOK® [8] AND PRINCE 2 [9] METHODS

PM method	Process
PMBOK® - Project Management Body of Knowledge	<ul style="list-style-type: none"> <li>↳ Project initiation</li> <li>↳ Project planning</li> <li>↳ Project execution</li> <li>↳ Project monitoring and control</li> <li>↳ Project closure</li> </ul>
PRINCE 2 - Projects IN Controlled Environment	<ul style="list-style-type: none"> <li>↳ Identify the activities and dependencies</li> <li>↳ Prepare estimates and the schedule</li> <li>↳ Analyze the risks</li> <li>↳ Document the plan</li> <li>↳ Project product description</li> <li>↳ Product breakdown structure</li> <li>↳ Product flow diagram</li> </ul>

Referring to Table I both PMBOK® and PRINCE 2 methodologies have a similar set of structured processes. Therefore, constraints such as scope, time, communications, cost, quality, procurement, human resources and integration of all those aspects are strictly considered throughout the project life cycle. Furthermore, in PMBOK® there is a special unit called project management office for managing all the projects in a large organization. Formal designations such as project manager, software engineer, etc. are well-defined in both of the methodologies.

Referring to Table II Scrum [10] and Extreme programming (XP) [11] are two agile methodologies that were designed for small development teams. In agile methods marketing team prioritize the features, while product development team provides estimates for the effort. They estimate for a short period and a small piece of software is completed within the given time and budget. Again the structured manner and the constraints are there as previous methods.

TABLE II: SCRUM [10] AND EXTREME PROGRAMMING [11]

PM method	Process
Scrum	<ul style="list-style-type: none"> <li>↳ Project initiation</li> <li>↳ Get your backlog in order</li> <li>↳ How to estimate your product backlog</li> <li>↳ Sprint Planning/clarify requirements</li> <li>↳ Sprint Planning/estimate tasks</li> <li>↳ Create a collaborative workspace</li> <li>↳ Sprint</li> <li>↳ Stand Up and Be Counted</li> <li>↳ Track Progress / Daily Burndown Chart</li> <li>↳ Finish When You Said You Would</li> <li>↳ Review, Reflect, Repeat</li> </ul>
Extreme programming (XP)	<ul style="list-style-type: none"> <li>↳ Planning game</li> <li>↳ Small releases</li> <li>↳ Metaphor</li> <li>↳ Simple design</li> <li>↳ Testing (test-driven development)</li> <li>↳ Re-factoring</li> <li>↳ Pair programming</li> <li>↳ Collective ownership</li> <li>↳ Continuous integration</li> <li>↳ 40-hour work week</li> <li>↳ On-site customer</li> <li>↳ Just rules</li> <li>↳ Open workspace</li> </ul>

TABLE III: MSF [12] AND ITIL [13]

PM method	Process
MSF - Microsoft Solution Framework	<ul style="list-style-type: none"> <li>↳ Envisioning: Vision/scope approved</li> <li>↳ Planning: Project plans approved</li> <li>↳ Developing: Scope complete</li> <li>↳ Stabilizing: Release readiness approved</li> <li>↳ Deploying: Deployment complete</li> </ul>
ITIL - Information Technology Infrastructure Library	<ul style="list-style-type: none"> <li>↳ Identify the activities and dependencies</li> <li>↳ Prepare estimates and the schedule</li> <li>↳ Analyze the risks</li> <li>↳ Document the plan</li> <li>↳ Project product description</li> <li>↳ Product breakdown structure</li> <li>↳ Product flow diagram</li> </ul>

The Rational Unified Process [14] is a software engineering process which consists of a PM aspect. It also provides a disciplined approach to assigning tasks and responsibilities within a software development organization? Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable

<sup>1</sup> University of Colombo School of Computing, www.ucsc.lk

schedule and budget.

Referring to Table III MSF - Microsoft Solution Framework [12] and ITIL - Information Technology Infrastructure Library [13] methodologies are focused on different aspects in the software industry. While MSF is used inside Microsoft Corporation for management of their own products, ITIL is supported for implementation of IT services within organizations. Similar to above mentioned methods MSF and ITIL also have the formal structured nature and constraints.

#### A. Overall Discussion of Formal PM Practices

When considering all the above methodologies almost all of them have their own characteristics and specific purposes. Most of them are focused on industry and large organizations. Throughout the development phases, strict constraints are followed such as schedules, budgets, etc. in order to perform the tasks and project activities in a systematic manner. Project manager, project team and other roles are considered as important.

During our study we could not find such methodology which is specifically designed to be used in a university context for learning and research oriented projects. Therefore, next step was to find out if these standard methodologies can be used for software development projects within universities where innovations considered as a major outcome.

### IV. SUCCESSFUL SOFTWARE PROJECTS WHICH WERE STARTED WITHIN UNIVERSITIES

If we look at most success stories in IT projects (some of which are detailed later), at the inception they did not use any formal project management methodologies. We can take as examples; Google Inc. started by Larry Page and Sergey Brin students at Stanford University when they first invented Google [1], Mark Zuckerberg co-founders of the social networking site Facebook®, a founder of Microsoft corporation Bill Gates, the Linux open source operating system [2], Apache and Mozilla open source projects [3]. The local case study we have used is Vidusayura a real-time perception enhanced virtual environment for maritime education which was developed by a group of academics at UCSC.

#### A. Google Inc.

Page and Brin founders of Google®, had grown up in homes that valued scholarly research published [1]. When they invented the Google search engine, they were students working on student projects at Stanford University. They didn't want to build another search engine, they didn't want to build another company, but they tried to solve interesting problems and collected ideas to build a full-scale search engine [1]. They had found the path toward a Ph.D. thesis through this academic research. One of their professors has said that it would soon become clear that what they had created together was more than just a way to further their academic research. They first implemented a prototype of a comprehensive search engine for internal use at Stanford: Based on conventional search engine technology. Google.stanford.edu became popular around the campus by

word of mouth. They did not have funds to hire a graphic designer to make the interface more elegant, thus they kept it clean and simple. As the database and number of users increased very fast they needed more computers and more cash. They were funded by the Stanford Digital Libraries project. When they were trying to sell the product, it didn't seem to matter that they had something better. While Larry and Sergey saw the search engine as the most important part of the Internet experience for computer users hunting for information, others saw it only as a helping tool [1]. They were rejected by many popular investors including Yahoo.

Finally, Andy Bechtolsheim, cofounder of Sun Microsystems liked their idea and agreed to fund the project. They came up with a revenue model for advertisers who are interested in reaching out to the online users. Google helps them create text-based Ads through Google Adwords a self-serve auction-based advertising program. These ads appear next to the search results. Most advertisers pay Google on a Cost per Click (CPC) basis, which means advertisers pay when users click their Ads.

That is how Google has become the most popular and used search engine throughout the world. Currently, Google is a structured company which has its own Google code of conduct. Google has started in a very informal way and now carried out its operations as a well-structured world class organization. There was a tipping point which it has transformed from informal to formal. In Google's case, it may be the point that Andy Bechtolsheim funded them.

#### B. Linux Open Source Operating System

Most successful IT projects are driven initially by a personal motivating factor. Every good work of software starts by scratching a developer's personal itch [2]. Linus Torvalds did not write Linux® from scratch. Instead he has re-used the code from Minix, a tiny Unix-like operating system.

Raymond [2] says that we have to do something at least once to know the real problems, so that second time we can do it right. The most important thing in an open source community developed project is to managing the peoples' interest rather than a project's constraints. When you lose interest in a program you should not keep on doing it, best thing is to hand it over to another competent successor [2].

At present, Linux uses more structural development methods. Although anyone can download and customize Linux platform, nobody can release it to the general public without acceptance of the top level of the hierarchy. Linux may not have an exact tipping point as Google. Probably it could be a series of activities which have transformed from unstructured to structured methods.

#### C. Apache

The development of Apache server follows traditional development methods [15]. Apache was begun by a group of geographically distributed volunteers, without any standard procedure. At the beginning there was no such development process in order to make decisions. None of the Apache Group (AG) members were full time employed at Apache server project. The people who have worked more than six months continuously are nominated for the Apache

membership. As founders they had 8 members and it grew rapidly. Sometime later, all of them followed a simple method in developments; first they discovered a problem, determining whether a volunteer will work on it, identifying a solution, developing and testing the code within their local copy of the source, presenting the code changes to the AG for review, and committing the code and documentation to the repository [15].

Likewise, Apache development has become more systematic eventually unlike at the start. Here also similar to Linux open source project, the tipping point can be identified as a set of events that has happened for some time. They may have become methodical by experience.

#### D. Vidusayura: A Sri Lankan Success Story

Computer based ship simulators are popular, cost effective and safe method which can be used for research and training of navel marine engineers and officers. *Vidusayura* is being developed by a group of academics at University of Colombo (UOC) since 2008. It was initiated as a consequence of a project proposal given to Sri Lanka Navy for a tender. The tender was cancelled due to the criticality of war at that time.

However, a member of the initial project team has started an MPhil research based on the submitted proposal. Early objectives were to develop a commercial product and deliver Navy requirements for revenue in mind. Outcomes of the project were changed to academic purposes, importantly, completion of the undergraduate and postgraduate projects, conference publications/posters, exhibitions, awards, etc. Aligned with above objectives, product development has been happening constantly according to the requirements stated in the initial proposal. The scope of the project has re-defined time to time because of the student projects. Priority is given to the eye catching features, thus product could take greater attention in the exhibitions. Followed by the literature, international maritime software standards have been considered for the product development.

*Vidusayura* was funded by several grant organizations and the lab facilities were provided by the university. Schedule and the deadlines are mostly depends on the students projects, conferences and exhibitions. There was no formal project manager or any other role. Undergraduate and postgraduate students from University of Colombo were involved in development effort under senior lecturers supervision. Some of the Navy officers helped them informally to refine and validate the project requirements. M.Phil. degrees, final year projects, conference publications/posters, awards and exhibitions are considered as the success factors or the measurements of quality of the product.

In *Vidusayura* the tipping point may not have been reached yet, since it is not fully implemented in the real user environment. Nevertheless, they also have had its own unique practices within the project when dealing with events such as conference deadlines, fundraising through research grants, etc.

#### V. THE TIPPING POINT FROM 'INFORMAL' TO 'STANDARD'

Considering above case studies; it may not be possible to use standard existing project management methodologies in

projects especially within Universities where *innovation* is a key ingredient. Without having pre-defined clear objectives, a client, specific requirements and scope, it may not be possible to anticipate schedule, resources, budget, quality, etc. of a project.

The main outcome of this research is that there is uniqueness at the beginning of each of those projects. Later on that unique nature has become standardized in most of the open source projects such as Linux and Apache. Whereas, in Google and Vidusayura type of projects they have developed some formal procedures, after their product become more stabilized and known by the users. It may be possible to have a tipping point for all student projects which are carried out within universities. It would be beneficial to identify this point in order to apply the practices and processes more effectively.

#### VI. CONCLUSION

It may not be possible to define a *standard methodology* for project management for software projects that involve universities as a major stakeholder, due of their innovative, diverse and dynamic nature. At the beginning of the paper we have discussed the standard project management practices. It is understood that such methodologies have not used at the *beginning* of the discussed case studies in section IV.

In the future, we hope to extend the research in identify with some accuracy the *tipping point* of a project where it moves from an unstructured (or non-existent) project management methodology to a more structured methodology. We also hope to define a more suitable project management methodology for use within universities for innovative software product development which can be useful at the initial phase. Ultimately, seek out the possibilities of implementing such methodology within the software industry, where innovation is a major ingredient.

#### ACKNOWLEDGMENT

The first author would like to express her gratitude to the University of Colombo School of Computing, Sri Lanka for facilitating to conduct this research work. The authors wish to convey their sincere thanks to the National Research Council, Sri Lanka for the research grant.

#### REFERENCES

- [1] D. Vise and M. Malseed, *The Google Story: for Google's 10th Birthday*. Delacorte Press, 2008.
- [2] E. Raymond, "The cathedral and the bazaar," *Knowledge, Technology & Policy*, vol. 12, no. 3, pp. 23–49, 1999.
- [3] A. Mockus, R. Fielding, and J. Herbsleb, "Two case studies of open source software development: Apache and mozilla," in *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002.
- [4] D. Jayawardena and L. Ekanayake, "Adaptation analysis of agile project management for managing it projects in sri lanka," in *Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on*. IEEE, 2010, pp. 1–4.
- [5] L. Munasinghe, P. Peter, and T. Perera, "Growth Prospects for the Software Industry in Sri Lanka and an Appropriate Policy Framework," 2002.
- [6] C. Hewagamage and K. Hewagamage, "Redesigned framework and approach for it project management," in *Proc. International Journal of*

*Software Engineering and Its Applications*, vol. 5, no. 3, pp. 89–106, 2011.

- [7] R. Angus, N. Gundersen, and T. Cullinane, *Planning, performing, and controlling projects*, Prentice Hall, 2002.
- [8] P. Guide, “*A guide to the project management body of knowledge*,” Project Management Institute, Edition, 2000.
- [9] C. Bentley, *Prince2: a practical handbook*, Routledge, 2012.
- [10] L. Rising and N. Janoff, “The scrum software development process for small teams,” *IEEE trans. Software*, vol. 17, no. 4, pp. 26–32, 2000.
- [11] P. Abrahamsson and J. Koskela, “Extreme programming: A survey of empirical data from a controlled case study,” in *Empirical Software Engineering, 2004. ISESE’04. Proc. 2004 International Symposium on. IEEE*, pp. 73–82, 2004.
- [12] G. Lory, G. Director, J. Carter, U. MSFmentor, J. Rief, and T. West, “Microsoft solutions framework version 3.0 overview,” Microsoft, Jun, 2003.
- [13] V. Arraj, “*ITIL @: The Basics*,” Buckinghamshire, UK, 2010.
- [14] P. Eeles, K. Houston, and W. Kozaczynski, *Building J2EE applications with the rational unified process*, Addison-Wesley Professional, 2002.
- [15] A. Mockus, R. Fielding, and J. Herbsleb, “A case study of open source software development: the apache server,” in *Proc. of the 2000 International Conference on Software Engineering, IEEE*, pp. 263–272, 2000.



**D. M. P. Dias** is currently following a MPhil degree at University of Colombo School of Computing, Sri Lanka. Ms. Dias has completed B.Sc. (ICT) degree in 2011 at the same university. Her research interests include IT project management, software engineering and human computer interaction.



**N. D. Kodikara** is a senior lecturer at University of Colombo School of Computing, Sri Lanka. Prof. Kodikara has received his PhD (Computer Science) from Manchester University, England in 1989. His research interests include computer graphics, virtual reality, image processing and computer vision.

**Mahen Jayawardena** is a lecturer at University of Colombo School of Computing, Sri Lanka. Dr. Jayawardena has received his PhD (Scientific Computing) from Uppsala University, Sweden in 2010. His research interests include bioinformatics, high performance computing and computer science education.