

The Study of International Standard Implementation for Supporting Software Process in Startup Business Enterprise

Waraporn Jirapanthong¹⁺

¹ College of Creative Design and Entertainment Technology, Dhurakij Pundit University,
110/1-4 Prachachuen Rd., Laksi, Bangkok, Thailand

Abstract. This research focuses on the study of ecosystems and international standards for software process that support startups. Startups can compete others or make its own unique by having a smart and very up-to-date software or application. It is believed that the activity flow for development process in the successful corporate entrepreneur needs to be thoroughly facilitated. It is expected to enable business startups including software tech startups therein to be more effective and generate a competitive advantage via providing ideal conditions for startups to move faster and more effectively than others. The work applies Personal Software Process (PSP) for software development activities, and uses PSP scripts to follow the activities in software development. Moreover, we study the literature regarding ecosystems' overall performance. We have studied the literature for startup globally. The scale of quantifying is applied with score across the success factors. The list of factors and metrics is identified. The life cycle of startups is also drawn and discussed.

Keywords: startups, ecosystems, PSP, software process, international standards.

1. Introduction

The paper presents the study of ecosystems and international standards for software process that support startups. It appears in common that software or applications are being developed on purpose to support and facilitate the business models of Startups. Each Startup can compete others or make its own unique by having a smart and very up-to-date software or application. We also believe that the activity flow for development process in the successful corporate entrepreneur needs to be thoroughly facilitated. Startups should define start and end point of a development process clearly and disciplined enough on the process. One of our focus is to enable business startups including software tech startups therein to be more effective and generate a competitive advantage via providing ideal conditions for startups to move faster and more effectively than others. The work applies Personal Software Process (PSP) [1] for software development activities, and uses adapted PSP scripts to follow the activities in soft-ware development. In particular, we have adapted a development script in order to enable automatic traceability. Moreover, we study the literature regarding ecosystems' overall performance. The scale of quantifying is applied with score across the success factors. The list of factors and metrics is identified. The set of success factors are performance, funding, market research, global connectedness, resource attraction, startup experience, talent, corporate involvement, and founder. We have studied the literature for startup globally. Particularly, we observed the life cycle of startups. The age of startups influence the success of startups. At the early stage startups are almost completely dependent on the resources and skills present in their local, so the ranking and other metrics essentially measure their access to resources.

2. Background

⁺ Corresponding author.

E-mail address: waraporn.jir@dpu.ac.th

2.1. Startups

Typically, a startup will begin by building a first minimum viable product (MVP), a prototype, to validate, assess and develop the new ideas or business concepts. In addition, startups founders do research to deepen their understanding of the ideas, technologies or business concepts and their commercial potential [2]. Startups have several options for funding. Venture capital firms and angel investors may help startup companies begin operations, exchanging seed money for an equity stake in the firm. Venture capitalists and angel investors provide financing to a range of startups (a portfolio), with the expectation that a very small number of the startups will become viable and make money.

2.2. Personal Software Process (PSP)

Personal Software Process (PSP) is a self-improvement process that drives a software developer to control, manage, and improve his/her work [1]. It is a structured framework of forms, guidelines, and procedures for developing software. The purpose of PSP is to assist a software developer to improve software engineering skills.

The baseline process, PSP0, is to provide a framework for writing the first program and for gathering data on work. The PSP0 process is driven by scripts which guide the work. The scripts guide software developers through the process steps, the logs are recorded for process data, and the plan summary provides summary record and reports. In the planning step, a software developer plans to do the work. In the development steps which include design, code, compile, and test. At the end, in the postmortem step, a software developer compares his/her actual performance with the plan and produces a summary report. There are three main process elements in PSP0 i.e. the planning, development, and postmortem phases. For the postmortem phase, three main activities are defined in the scripts. Those are project review, defect recording, and time recording as appeared in the following activities during the postmortem phase: i) the activity called defect recording aims to review the project plan summary to verify that all of the defects found in each phase were recorded, and to use recollection, record any omitted defects, ii) the activity called defect data consistency aims to check that the data on every defect in the defect recording log are accurate and complete, to verify that the number of defects injected and removed per phase are reasonable and correct, and to use recollection, correct any missing or incorrect defect data, and the activity called time aims to re-view the completed Time Recording log for errors or omissions and to use recollection, correct any missing or incomplete time data.

According to [1], the authors have proposed a template of defect recording log. The log document contains information, for example, a software developer's name, program's name. The log document shows a list of defect log which consists of: i) project identifier, ii) fixing date, iii) a unique defect number, iv) defect type, v) injected phase, vi) remove phase, vii) fix time, viii) fix reference, and ix) description. The defect type is classified as *documentation*, *syntax*, *build/package*, *assignment*, *interface*, *checking*, *data*, *function*, *system*, and *environment* defects.

The injected phase is the one in which a defect was injected. The remove phase is the one in which a software developer found and fixed the defect. The fix time is the time a software developer took to find and fix the defect. The fix reference is used to note the number of the defective fix which refers a mistake fixing one defect and later find and fix that new defect. The description section is a note that described the reason or location that the defect was fixed. Also, the authors proposed a template of time recording log. The log document contains information, for example, a software developer's name, program's name. The log document shows a list of time log which consists of: i) project identifier, ii) phase, iii) start date and time, iv) interruption time, v) stop date and time, vi) delta time, and vii) comments.

3. Research Methodology

We have surveyed the success rate of startups globally. One of our research objectives focused on quantifying the impact of internal success and future factors. Particularly, we focused on factors under the control of a startup founder. We then turned to external success factors to quantify how they affect a startup's ability to succeed.

We found that early-stage startups are highly dependent on their surrounding startup ecosystem, and so if we can create healthier startup ecosystems, we can generate more successful startups.

Our research involves two steps. First, it requires an assessment of an ecosystem against our lifecycle model, using metrics to identify its phase of development. This assessment provides and what policies and practices its leader decided to focus on.

Second, we studied an ecosystem's overall performance. The scale of quantifying is applied with score across success factors, and the breakdown of each factor score into sub-factors. We also studied how each ecosystem compares to other ecosystem at similar stages of development, and what are the most important gaps to address.

3.1. Startups and Software Projects

For the first part, startups are asked to contributed in this study. The study was conducted to identify the practices of startups and how clearly contribute to software project success. It investigated team knowledge, allocated resources, and deployed software processes. The software projects are developed in order to support the startup teams. The study involved four startup teams with the equivalent skills but worked on the same projects and used different processes.

The startup development teams that have experience in software development were participated in the study. Each team was established and included co-founders, team members, and mentor. The team member are responsible to be different roles in software projects i.e. system analyst, project manager, and software developer. Each startup team has their own business model and market targets. In particular, the software projects are conducted based on each business model, customers, and market targets. There are similar and different requirements. Four software projects were developed in similar timeline. The projects have finished within 4 months with an effort of 2.5 person-months. The development team was required to achieve the software development projects by following Personal Software Process (PSP) [1] and ISO29110 [2] to complete the software projects.

3.2. PSP Scripts for Developing Startups Software

According to Personal Software Process (PSP), the process drives a software developer improving his own performance by controlling and managing his work. It is a structured framework of forms, guidelines, and procedures for developing software. The process is driven by scripts through the process steps i.e. design, code, compile, test, and postmortem steps. Additionally, traceability is included as activities during the process. However, there are still difficulties to use traceability records or relations in order to improve software process.

When startups has been establishing, software for startups need to be developed rapidly. As described in Section 2, startup consists of main elements. One of those is individuals. It is commonly found that developers are working alone to develop software for startups. According to [7], our work is applying PSP to support the individual process for software developing. One of our contributions is to explain the presumed causal relationships between software artefacts under PSP-based development. We have employed several forms to follow the activities in PSP [1]. The development script [7] is applied for programming. The process is designed for developing a large program in a sequence of small incremental steps.

3.3. Quantifying Startup Performance with Success Factors and Metric

Second, we studied an ecosystem's overall performance. The scale of quantifying is applied with score across the success factors. The list of factors and metrics is identified. The set of success factors are performance, funding, market research, global connectedness, resource attraction, startup experience, talent, corporate involvement, and founder. For example, global connectedness: this factor quantifies the valuable international relationships that exist between startup leaders, and how they were developed. It translates into a startup's ability to reach out outside its own ecosystem and highly correlates with the ability to attract foreign customers. For this, it is categorized within market targets reach inside the ranking model.

We have studied the literature for startup globally. Particularly, we observed the life cycle of startups. The age of startups influence the success of startups. At the early stage startups are almost completely

dependent on the resources and skills present in their local, so the ranking and other metrics essentially measure their access to resources.

4. Results and Future Work

According to the first part of our research, the startups explored and experienced our proposed approach for software projects. In order to take into consideration the PSP-based software developing including the standard of ISO 29110 [3], we have created a scenario in our testing: changes to functional requirements. In particular, a participant who takes a role of software developer was asked to perform software development activities under PSP. In particular, the participant was asked to follow the development script (as shown in Table 1). The scenario involved many types of documents. Therefore, the traceability relations were expected to be captured among various types of documents. Particularly, during the development script, we asked the participant to perform following tasks:

- (i) Apply the tool to generate traceability relations after finishing Requirements and Planning activities.
- (ii) Apply the tool to generate traceability relations after finishing High-level Design Review (HLDR) activities.

Those tasks aimed to generate traceability relations between:

- (a) functional requirements;
- (b) functional requirements and class diagram; and
- (c) functional requirements and use cases.

To measure the project effort for startups establishing, we evaluated the number of entities that were created when the startup software products were developed and time. The entities are such as software design, software design specification, software code, and startup development team. Attributes are such as defects discovered in design review, number of pages, number of line of code, number of operations, and team size, average team experience. In particular the size measure of software products involves two ratios: (a) line of code (LOC) and (b) function point (FP). As shown in Tables 3, the number of LOC created for each startup teams (ST1, ST2, ST3, and ST4) are 3689, 1251, 4512, and 2280. The average numbers of LOC created for each individual software product are 6830, 5420, 9025, and 8845. However, the numbers of function point between same startup software products using different software processes are the same.

Additionally, the quality measure of software product involves maintainability measurement such as coding effort, design effort, percentage of modules changes, classes changes, classes added. In addition, we measured the maintainability metrics in external view such as mean time to implement the changes. Based on the same set of changes, it is found that the mean time to implement the changes on startup products, ST1, which is created by using startup software team is 15.5 days. Comparatively, it is found that the mean time to implement the changes on software product, ST1, which is created by using a single software development process is 24 days. It is thus believed that well- and proper-implementation of startup software product will be effective to maintainability.

Table 1: The details of LOC and FP created for startup software products using PSP

By using PSP			By without using PSP		
Startup Team	LOC created	FP created	Startup Team	LOC created	FP created
ST1	3689	10	ST1	6830	10
ST2	1251	5	ST2	5420	5
ST3	4512	15	ST3	9025	15
ST4	2280	4	ST4	8845	4

Moreover, we also compare the development team’s satisfaction. We conducted the survey and interview. The developers are observed for the satisfaction regarding the process of startup establishing with PSP. It is found that the developers are satisfied the process that emphasis the software more than the documentation. However, the process would be difficult to inexperienced developers and some experience developers tend to

resist some software product line practices. According to the survey, it is found that 33% of developers tend to resist software product line practices with the above reasons, whereas 70% of developers are positive to using software product line practices. Particularly, 82% of developers are satisfied when performed the maintenance phase with startup process. Some of startup software artefacts are used during the maintenance phase. And it is satisfied by the developers. However, application engineering process depends on developer's skill. Moreover, some developers are unsatisfied to frequently update the documentation.

For the second part of our research, we have studied startups in Thailand which have been established during 2015 – 2017. We analyzed the Business Data Warehouse of the Department of Business Development, Ministry of Commerce, Thailand. The profile of startups ecosystems is then narrowed down to be four phases. Each phase is related to difference size, strengths, and challenges. Firstly, the initial phase is the phase that startups are active and creating new ideas. There are number of startups having low output, local experience, and having a challenge of resource leakages. The main objectives are to grow and build a larger and more connected community by activating local entrepreneurs, talent, and investors.

Secondly, the realization phase is the phase that startups exit placed the ecosystem to build a startup that attracts resource and realized by nearby regions. At this phase, the main objectives are to foster connections with global ecosystems so local startups can develop world-leading startups and unicorns. Thirdly, the enlargement is the phase that successful startups exits and unicorns have elevated the ecosystem to the global stage. It however has some gaps in funding and global connectedness, and ability to reproduce a pattern of startups. The main objectives are to expand, fill remaining resource gaps, and increase global connectedness by enabling and taking advantage of global resource. Lastly, the union phase is the phase that startups are growing and their resources are balanced and competitive with others. The main objectives are to unite the ecosystems with the global, national, and local flows of resources and knowledge inside and outside.

The research focuses on the study of startups how to scale up from initial to union phase. We also focus on the study of the international standards supporting tech startups in Thailand. We apply Personal Software Process (PSP) for software development activities, and uses PSP scripts to follow the activities in software development. The work focus on the objective to enable business startups including software tech startups therein to be more effective and generate a competitive advantage via providing ideal conditions for startups to move faster and more effectively than others. However, a number of possible directions for further investigations have been identified. We plan to continue the future work of the research what needs to be done to improve and increase the benefits of the findings. Firstly, in startup ecosystems, experience matters – it is the pool of knowledge and networks that startups can draw on. The analysis relies on two factors i.e. team experience and ecosystem experience. It is therefore believed that the analysis can be deep driven to support a better way of scaling the startups and to better identify the startup profiles. We also plan to study the standard of ISO 29110 and focus on the two main processes i.e. Project Management (PM) and Implementation (I) Processes. Also, the software projects for Startups are convincingly formed as Scrum. It is believed that the standard would drive the process more effectively and efficiently, in which can reflect a quick respond to dynamics in ecosystems.

5. References

- [1] Humphrey, W. S. (2005). *PSP: A Self-Improvement Process for Software Engineers*. Addison Wesley. ISBN: 0-321-30549-3.
- [2] Blake, M. and S. Wijetilaka. (2015). 5 tips to grow your startup using SWOT analysis. Sydney. Retrieved 13 August 2015.
- [3] International Organization for Standardization (ISO). (2016). *Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 1: Overview. ISO/IEC TR 29110-1*. Geneva.
- [4] Business Data Warehouse. 2013. Department of Business Development, Ministry of Commerce. Retrieved 15 January 2018.
- [5] Startup Genome. 2017. *The Global Startup Ecosystem Ranking 2017*. Retrieved 15 January 2018.
- [6] Oecd Organisation for Economic Co-operation and Development (OECD), *Small and Medium Enterprise (SME) Outlook Report, 2005*
- [7] Jirapanthong, W. (2017). Personal Software Process with Automatic Requirements Traceability to Support Startups. *Journal of Reviews on Global Economics*. Vol.6, pp. 367-374.