การนำมาตรฐานสากลไปปฏิบัติเพื่อสนับสนุนกระบวนการซอฟต์แวร์ในองค์กรธุรกิจแบบ startup

THE STUDY OF INTERNATIONAL STANDARD IMPLEMENTATION FOR SUPPORTING SOFTWARE PROCESS IN STARTUP BUSINESS ENTERPRISE

วราพร จิระพันธุ์ทอง*
Waraporn Jirapanthong*

* รองศาสตราจารย์และปริญญาเอก วิทยาลัยครีเอทีฟดีไซน์แอนด์เอ็นเตอร์เทนเมนต์เทคโนโลยี มหาวิทยาลัยธุรกิจบัณฑิตย์

* Associate Professor and Ph.D., College of Creative Design and Entertainment Technology, Dhurakij Pundit University

* Email: waraporn.jir@dpu.ac.th

บทคัดย่อ

    งานวิจัยนี้มุ่งเน้นการศึกษาระบบนิเวศน์และมาตรฐานสากลสำหรับกระบวนการซอฟต์แวร์ที่สนับสนุนสำหรับองค์กรแบบสตาร์ทอัพ ปัจจุบันสตาร์ทอัพสามารถสร้างอำนาจการแข่งขันทางธุรกิจหรือสร้างความเป็นเอกลักษณ์เฉพาะของตนเองได้โดยการใช้ซอฟต์แวร์หรือแอพพลิเคชั่นที่ทันสมัยและอัจฉริยะ งานวิจัยนี้มีสมมติฐานที่เชื่อว่าการดำเนินกิจกรรมเพื่อการพัฒนาของผู้ประกอบการที่ประสบความสำเร็จนั้นองค์กรจำเป็นต้องได้รับการอำนวยความสะดวกอย่างทั่วถึง จากการศึกษาพบว่าซอฟต์แวร์หรือแอพพลิเคชั่นที่ทันสมัยและอัจฉริยะจะช่วยให้ธุรกิจที่เพิ่งเริ่มต้นซึ่งรวมถึงองค์กรแบบซอฟต์แวร์เทคสตาร์ทอัพจะมีประสิทธิภาพและสร้างความได้เปรียบในการแข่งขัน โดยการจัดหาสภาพและข้อได้เปรียบที่เหมาะสมสำหรับการเริ่มต้นทำงานให้เร็วขึ้นและมีประสิทธิภาพมากขึ้นกว่าองค์กรอื่น การศึกษานี้ประยุกต์ใช้กระบวนการซอฟต์แวร์ส่วนบุคคล (PSP) สำหรับกิจกรรมการพัฒนาซอฟต์แวร์และพัฒนาการปรับใช้สคริปต์สำหรับทำตามขั้นตอนในกระบวนการซอฟต์แวร์ส่วนบุคคล นอกจากนี้มีการศึกษาวรรณกรรมที่เกี่ยวข้องกับระบบนิเวศน์นี้โดยรวมและสตาร์ทอัพโดยทั่วไป มีการศึกษาการวัดระดับเพื่อประเมินความสำเร็จผ่านปัจจัยและเมตริกต่างๆและวงจรชีวิตของสตาร์ทอัพ

คำสำคัญ: สตาร์ทอัพ ระบบนิเวศน์ กระบวนการพัฒนาซอฟต์แวร์แบบบุคคล กระบวนการพัฒนาซอฟต์แวร์
     มาตรฐานสากล

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 uniqueness 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. This study applies Personal Software Process (PSP) for software development activities, and uses PSP scripts to follow the activities in software development. Moreover, researchers study the literature regarding ecosystems' overall performance. We have studied the literature for ecosystems and startups globally. The scale for evaluating the success by applying success factors and matrix, including the life cycle of startups, are studied.

Keywords: Startups, Ecosystems, Personal Software Process (PSP), Software Process, International
     Standards

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 uniqueness by having a smart and very up-to-date software or application (Myler, 2017). 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 study applies Personal Software Process (PSP) (Humphrey, 2005) for software development activities, and uses PSP scripts to follow the activities in software 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 (Kask & Linton, 2013; Katila, Chen & Piezunka, 2012; Robehmed, 2013; Startup Genome, 2017; Teece, 2010). Particularly, we observed the life cycle of startups. The age of startups influence the success of startups (Startup Genome, 2017). 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.

Research Objectives

1. To evaluate the implementation of the international standard ISO/IEC 29110 for supporting software process for startups in the ecosystems

2. To apply Personal Software Process to support software development activities for individuals

Background

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 (Blake & Wijetilaka, 2015). Nowsaday, in the ecosystems, 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. 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 (Humphrey, 2005). 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, there are three main process elements in PSP0 namely: the planning, development, and postmortem phases. 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.

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 if data in a defect recording log is accurate and complete, to verify if the number of defects per phase is reasonable, and to correct any missing or incorrect defect data.

According to (Humphrey, 2005), 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 showsa 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: a) documentation that refers a defect on comments or messages, b) syntax that refers a defect on spelling, and punctuation, typos, and instruction formats, c) build,package that refers a defect on change management, library, and instruction formats, d) assignment that refers a defect on declaration, duplicate names, scope, and limits, e) interface that refers a defect on procedure calls and references, I/O, and user formats, f) checking that refers error messages, and inadequate checks, g) data that refers a defect on structure, and content, h) function that refers a defect on logic, pointers, loops, recursion, computation, and function defects, i) system that a defect on configuration, timing, and memory, and j) environment that a defect on design, compile, test, or other support system problems.

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 tool 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) omments.

The phase is the one in which a software develop worked on i.e. planning, design, test. The start data and time is the date and time when a software developer working on a process

activity. The interruption-time is duration time that was not spent on the process activity. The stop date and time is the date and time when a software developer stop working on that process activity. The delta-time is the clock time that a software developer actually spent working on the process activity, less the interruption time. The comments section is used to remind a software developer any unusual circumstances regarding an activity.

      3. ISO/IEC29110(ISO 2016)

      ISO/IEC 29110 introduces the major concepts for software and systems engineering profiles for very small enterprise (VSE), and defines the terms common to the set of documents associated with VSE profiles. ISO/IEC 29110 is the international standard which the main purpose is to support the software development life-cycle. In particular, the standard is composed of two main processes i.e. Project Management and implementation. Each process includes a set of activities, and an activity is composed of a set of tasks. The two main processes that are described at the task level in the ISO/IEC 29110 are mainly supported VSEs.

Research Methodology

      We have surveyed the success rate if startups globally. One of our research objectives focused on quantifying the impact of internal success and future factors. Particularly, we focused ones 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 highly depend on their surrounding startup ecosystem (Startup Genome, 2017). 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. The assessment includes the policies and practices that are applied by team leaders.

      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.

      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 and worked on the same projects but used different processes.

      The startups are chosen based on a set of criteria i.e. the levels of maturity, professional experts, professional community and society, including their related business domain. Particularly, the startups have ICT literacy and their business has identified a need to develop a digitally enhanced, customer-driven business strategy, but execution is on a project basic. Moreover, their progress is not predictable nor repeatable. Each team was established and included co-founders, team members, and mentor. The team member are responsible to be in different roles in software projects i.e. system analyst, project manager, and software developer. Each startup team has their own business model and marketing targets. In particular, the software projects are conducted

based on each business model, customers, and marketing 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) and ISO29110 (International Organizational for Standardization (ISO), 2016) to complete the software projects.

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 previous section, startup consists of main elements. One of those is individuals. It is commonly found that developers are working alone to develop software for startups. Our work is applying PSP to support the individual process for software developing. In this section, the development script for individuals are presented.

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. We have developed a development script as shown in Table 1. The script is adapted from Humphrey (2005). It is the cyclic process that we applied for programming. The process is designed for developing a large program in a sequence of small incremental steps.

Table 1 A development script

Purpose	To guide development of programs	
Entry Criteria	Problem description or component	
Step	Activities	Description
1	Requirements and Planning	- Obtain the requirements and produce the development plan. - requirements document - design concept - size, quality, resource, and schedule plans - Produce a master Issue Tracking log
2	High-level Design (HLD)	- Produce the design and implementation strategy - Functional Specifications - State Specifications - Operational Specifications - Development Strategy - Test Strategy and Plan

Table 1 A development script (continued)

Purpose		To guide development of programs
Entry Criteria		Problem description or component
Step	Activities	Description
3	High-level Design Review (HLDR)	- Review the high-level design - Review the development and test strategy - Fix and log all defects found - Note outstanding issues on the Issue Tracking log - Log all defects found
4	Development	- Design the program and document the design in the PSP Design templates - Review the design and fix and log all defects - Implement the design - Review the code and fix and log all defects - Compile the program and fix and log all defects - Test the program and fix and log all defects - Complete the Time Recording log. - Reassess and recycle as needed
5	Postmortem	- Complete the Project Plan Summary form with the actual time, defect, and size data
Exit Criteria		- A thoroughly tested program - Completed Project Plan Summary with estimated and actual data - Completed Estimating and Planning templates - Completed Design templates - Completed Design Review checklist and Code Review checklist - Completed Test Report template - Complete Issue Tracking log - Completed PIP forms - Completed Time and Defect Recording logs

3. Quantifying Startup Performance with Success Factors and Metric

For the second part, 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 marketing 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 influences the success of startups. At the early stage startups almost completely depend on the resources and skills present in their local, so the ranking and other metrics essentially measure their access to resources.We found that the factors and metrics for Startups Assessmentas shown in Table 2.

Table 2 Startups Assessment Factors/Metrics

Factors/Metrics	Description
Performance of Ecosystem	Refers the performance of the ecosystems that startups belong to.
Ecosystem Value	Refers the growth of ecosystems that startup belong to.
Exit Value	Refers the strategy of pitching and selling out to a larger company. The values include Exit Value Growth Index, and Exit Growth Rate.
Startup Valuations	Refers the growth of money in startups.
Output	Refers the growth of outcome that startups produced. It is measured by Output Growth Index.
Funding Access	Refers accessibility in funding including Early-stage Funding and Early-stage Growth Index
Quality	Refers Experienced VD Firm Index
Market Reach	Refers a ratio of market share
Global Reach	Refers customer targets i.e. Rest of the World Customers, Foreign Customer
Global Connectedness	Refers the connection to global enterprises
Founder Relationships	Refers values i.e. International Outbound, International Inbound, Locally-Made Global Connections
Resource Attraction	Refers an outstanding resource of enterprise
Startup	Refers a classification of startups
Entrepreneur	Refers a region of startups i.e. Global Attraction, National Attraction, Leakage
Startup Experience	Refers an experience of startups
Ecosystem	Refers number of Exits
Team Advisors with Equity	Refers Stock Options to all employees, Founder Hypergrowth, and Experience
Talent Access	Refers Experience of Engineers/ Developers, Time to High Developers, and Experienced Growth Employees
Corporate Involvement Interest	Refers Levels of Positive Involvement, and Negative Involvement
Founder Ambition	Refers Globally Leading Product
Go-Global Strategy	Refers Targeting Global Market First, Growth Leader Global Experience, and Growth Team Global Focus

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 (International Organizational for Standardization (ISO), 2016), 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. The scenario was created based on an environment of startups in the ecosystem. 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. And as shown in Table 4, 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 3 The details of LOC and FP created for startup software products using PSP

Startup Team	LOC created	FP created
ST1	3689	10
ST2	1251	5
ST3	4512	15
ST4	2280	4

Table 4 The details of LOC and FP created for startup software products without the use of PSP

Startup Team	LOC created	FP created
ST1	6830	10
ST2	5420	5
ST3	9025	15
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 startups 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 inexperience 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' 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 profiles data from theDepartment 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. However, it has some gaps in funding and globally connecting between startups and larger companies. Start need to identify a pattern of processes to be repeatable. 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 unicorn phase. We also continue the further 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. It is found that the experience is significant tostartup ecosystems. 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.

References

Blake, M. and S. Wijetilaka. (2015). *5 tips to grow your startup using SWOT analysis*. Sydney. Retrieved August 13, 2015, from http://www.afr.com/it-pro/5-tips-to-grow-your-startup-using-swot-analysis-20150226-13pkj5.

Business Data Warehouse. (2013). Department of Business Development, Ministry of Commerce. Retrieved January 15, 2018, from http://datawarehouse.dbd.go.th.

Humphrey, W. S. (2005). *PSP: A Self-Improvement Process for Software Engineers.* Addison Wesley. ISBN: 0-321-30549-3.

International Organizational for Standardization (ISO). (2016). *Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 1: Overview.* ISO/IEC TR 29110-1. Geneva. Retrieved January 15, 2018, from https://www.iso.org/standard/51150.html.

Kask, J. & Linton, G. (2013). Business mating: When start-ups get it right. *Journal of Small Business and Entrepreneurship, 26* (5), 511. doi:10.1080/08276331.2013.876765.

Katila, R., Chen, E. L. & H. Piezunka. (2012). *All the right moves: How entrepreneurial firms compete effectively.* Strategic Entrepreneurship Journal. 6. doi:10.1002/sej.1130. Retrieved May 15, 2017, from https://onlinelibrary.wiley.com/doi/abs/10.1002/sej.1130.

Myler, L. (2017). 7 Ways To Make Your Business Stand Out In A Crowd Of Competitors. *Forbes Entrepreneurs.* Retrieved February 23, 2018, from https://www.forbes.com/sites/larrymyler/2017/02/23/7-ways-to-make-your-business-stand-out-in-a-crowd-of-competitors/#38550f8862fa.

Organisation for Economic Co-operation and Development. (2005). *Small and Medium Enterprise (SME) Outlook Report, 2005.* OECD. Retrieved December 4, 2017, from http://www.oecd.org/cfe/oecdsmeandentrepreneurshipoutlook-2005edition.htm.

Rachleff, A. (2017). *To Get Big, You´ve Got to Start Small.* Tech Crunch. Retrieved January 28, 2017, from www.inc.com.

Robehmed, N. (2013). What Is A Startup?. *Forbes.* Retrieved April30, 2016, from https://www.forbes.com/sites/natalierobehmed/2013/12/16/what-is-a-startup/#66f9676a4044

Startup Genome. (2017). *The Global Startup Ecosystem Ranking 2017.* Retrieved January 15, 2018, from https://startupgenome.com/thank-you-enjoy-reading/

Teece, D. J. (2010). Business Models, Business Strategy and Innovation. *Long Range Planning, 43* (2–3), 172. doi:10.1016/j.lrp.2009.07.003.