Investigating the Benefits of Combining PSP with Agile Software Development

Wenrong Yang, Mengjiao Shen, Han Su, Guoping Rong and Dong Shao

Software Institute of Nanjing University, 210093 Nanjing, P.R. China

Abstract. Agile software development is getting popular due to chaotic and changing environments of modern software projects. But there are cases of industrial teams experiencing failure with agile software development. One of the main reasons may be the inadequate capability of involved team members. The Personal Software Process (PSP) is a plan-based software process intending to improve individual software engineering's competence. Therefore, the integration of PSP with agile methods will probably help to give full play to the advantages of agile method. This paper aims to summarize the existing evidence of combination of PSP with agile software development, so as to identify the benefits.

1 Introduction

Since 1990's agile methodologies have captured widespread interest. They adopt four value preferences put forward by the Agile Manifesto [1]: Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan. Agile Methodologies are gaining popularity in both industry and academia although there are still controversy over some parts of them and their successfully applied to real projects.

There have been many cases of industrial projects applying agile methodologies with great success during the past few years [2] [3]. But also many agile development projects and agile methodologies experiments failed because developers are unwilling or unable to apply the agile practices in a disciplined and professional manner [4] in which case no process will repair their impropriety. In fact, there are 49.9999 percent of the world's software developers below average suggested by statistics [7]. The incapability aspects are mainly related to making unreasonable estimations of individual work and failing to schedule a reasonable plan [1]of the tasks to be performed.

Personal Software Process (PSP) provides a framework of forms, guidelines and procedures for the improvement of individual software developers and help to produce a defined process [5]. Individual software developers can use PSP to gather historical data to measure their performance, work patterns and practices in order to make better

Yang W., Shen M., Su H., Rong G. and Shao D..
Investigating the Benefits of Combining PSP with Agile Software Development.
DOI: 10.5220/0003579500360043
In *Proceeding of the 1st International Workshop on Evidential Assessment of Software Technologies* (EAST-2011), pages 36-43
ISBN: 978-989-8425-58-4

plans and more accurate estimates, improve productivity and product quality [6]. It includes seven PSP processes, grouped into four process levels [7]:

- PSP0 - establishing a measured performance baseline
- PSP1 - size, resource, and schedule plans
- PSP2 - defect and quality management
- PSP3 - scaling up PSP methods to larger projects

The individual developers gradually improve competency and applies higher levels of PSP, as the software development process evolves.

Since PSP is a framework for improvement of individual developers, it seems that complementing agile software development with PSP is a good way for resolving the problem. Furthermore, PSP could also conduce to manage the problem of reluctance. In another perspective, agile development methodologies promise to better meet customer's needs, improve product quality, shorten development times and offer a solution to rapidly changing requirements [5]. PSP as a plan-driven approach promises predictability, stability, maintainability and high assurance. It is meaningful to combine PSP with agile methods to meet the needs of Modern software projects which require both adaptability and predictability.

In this paper, we perform an extensive literature search for articles related to integration of PSP and agile methods and then analyze and synthesize them to identify the benefits achieved by incorporating PSP into agile software development.

The article is structured as follows: Section 2 discusses the research method. Section 3 lists the benefits achieved by combining PSP with agile software development. Finally, section 4 shows the main conclusions and describes our future work.

2 Method

2.1 Search Strategy

We conducted an extensive literature search for the articles related to integration of PSP and agile software development. Search terms were firstly chosen as '(PSP AND agile)'. But the search results turn out to be not good. So we changed 'agile' to the main agile software development methods jointing with OR. We consider the main agile software development methods including: Scrum, Feature-Driven Development (FDD), Adaptive Software Development (ASD), Crystal, Dynamic Systems Development Method (DSDM), XP, Lean Development (LD) and Rational Unified Process (RUP). RUP is sometimes classified as plan-driven method. But as it can be as agile as it is wanted to be [8], it's included in this study. Then the search terms turned out to be '(PSP AND (Scrum OR FDD OR ASD OR Crystal OR DSDM OR XP OR LD OR RUP))'. The search was then run in 13 academic databases: ACM Digital library, EBSCOhost, Wiley Inter Science Journal Finder, CiteseerX Library, CALIS, SpringerLink, ISI Web of Science, ProQuest, INSPEC, SAGE, IEEEXplore, ACM Digital Library and ScienceDirect. In addition, the Google Scholar search engine was used.

2.2 Study Selection

In order to assess the potentially relevant primary studies for their actual relevance after obtained, the inclusion and exclusion criteria were defined.

The inclusion criteria were:

(a) The study analyzed and proposed a specific way of incorporating PSP elements into an agile method.

(b) The study proposed a brand new process model that is made up by combining PSP with agile software development.

(c) A case study assessing the integration of PSP with agile software development methods.

Not all of these criteria must be present for every study, however, at least one of them. (c) is supposed to be a necessary conditions for high confidence , but as the related studies are limit they are included for reference.

The exclusion criteria were:

(a) As English is the most popular language in scientific world, results in other languages were discarded.

(b) Studies that discussed only on an abstract level about PSP and agile but not practice level were discarded.

Finally there are eight relevant studies left: [9] presents the combination of PSP practices with DSDM; [10] involves combination of PSP with RUP; [15] proposed a new process model mainly made up by PSP and Scrum; [11], [12], [13], [14] and [4] are about combination of PSP with XP. [12], [14] and [4], meanwhile, are about the same method named eXPERT. As we can see, other agile methodologies' combination with PSP may be less studied by far, including Feature-Driven Development (FDD), Adaptive Software Development (ASD), Crystal and Lean Development (LD).

2.3 Data Extraction

After the primary studies were obtained, data extraction was performed to get information to address the research questions. There are foundation information including data extractor, date of data extraction, identifier (unique identifier for the study), bibliographic data (author, year, tile, source) .and core information which related to research questions. The data extraction form used is shown in Table 1. The main information extracted for each relevant primary study is listed below:

- Which specific agile methodology was chosen to be integrated with PSP in this study?
- Which PSP elements are used to complement the agile methodology and what is the benefit?
- Are there any challenges of combing PSP with agile methodologies presented by the author? If is, then what are they?
- Is the proposed solution validated?

As it is the primary studies that we were interested in, only one data extraction form was completed for each study instead of each publication. For example, [12], [14] and

[4] were considered achievement of the same study, so only one data extraction form was finished.

Table 1. Data extraction form.

Data item	Description	
Data extractor	The reviewer that perform the data extraction	
Date of data extraction	The date of data extraction	
Identifier	Unique identifier for the study	
Bibliographic data	Author, year, tile, source	
Agile methodology	The agile methodology was chosen to be integrated with PSP	
PSP practices	PSP elements that used to complement the agile methodology	Benefits
Challenges	Challenges of combing PSP with agile methodologies	
The validated technology	The technology that was used to validate the proposed solution	

3 Benefits of Combining PSP with Agile Software Development

We identify three kinds of relations between PSP practices and agile practices: conflict, match and complementary. Complementary relation is the focus of this work.

- Conflict: incompatible parts between PSP and agile methods. PSP belongs to the plan-driven process model camp which consists of conflict with agile software development especially at high abstract level like value perspective. For example, agile software development are more likely to value people while PSP is more likely to value process.

- Match: similar parts between PSP and agile software development which may be expressed in different terms or same ones. For example, PSP3 (Cyclic Personal Process) can offer a close match with short iterations of DSDM. The requirements and planning stage of PSP3, carries out similar functions to DSDM's Business Study and the Cyclic Development section closely matches the Design and Build Iteration in DSDM [9]. Coding Standard is present in both PSP and XP, and so on.

- Complementary: supports that PSP (agile software development) provides for agile software development (PSP). PSP offers supports for agile software development in several ways listed below. Although stated in numerical order, they are actually closely interconnected.

(1) PSP offers metrics and data collecting mechanisms.Agile software development does not assure to have predefined metrics or specific metrics collecting approach. PSP provides metrics concerning project, product and resource attributes to provide adequate support to monitor and control the software development process at an individual level [10]. Developers may be unaware of the state of their own process without metrics. Collecting even basic defect metrics will at least allow them a rudimentary evaluation of their own software capability. Coleman and Verbruggen [9] suggested adopting defect metrics, productivity metrics, size metrics, time/schedule metrics, extended metrics and maintenance metrics in a DSDM and PSP integration case to support objective data based decision. Svensson [10] disseminated metrics referring to specific software engineering tasks to relevant workers and others which

can be applied generally for improving the software process to all workers in the case of integration of PSP and RUP. For example, metric test Defects/KLOC is only disseminate to test personnel, while metric like Time in Phase is disseminate to all workers.

(2) PSP supports to make more reasonable estimation.Estimations of some agile methods like RUP are based on data collected throughout the organization and others like XP are done in the way of planning game which mainly depends on the capability of engineers. PSP helps to make more reasonable estimation based on personal data which provides the team a possibility to make more realistic commitments. Coleman and Verbruggen [9] used a PSP size estimating technique named proxies which bases on historical data to estimate the work to be developed within the timebox of DSDM [9]. Bozheva [12] applied an adapted Probe method instead of LOC to reflect complexity of customer requirement in the case of integration of PSP and XP to gradually close the difference between the estimated and the actual implementation duration.

(3) PSP helps to optimize plans. PSP helps to make more appropriate schedule plan, quality plan, etc. PSP provides task and schedule planning templates and the schedule plan which is based on the empirical estimation mentioned above could be practical and realistic. Quality plan can offer support to track and control product quality basing on quality goals. Coleman and Verbruggen [9] included PSP process scripts, checklist and Process Improvement Proposal (PIP) in the quality plan in the PSP and DSDM integration case. Rong, Dong and He [15] recommended tracking core PSP quality indices such as phase yield, review rate, PQI and A/FR in the quality plan of the PSP and scrum integration case.

(4) PSP offers support to produce software products up to quality goals.PSP helps engineers to plan, measure, and track product quality from the initial development phase with help of quality measures and quality plan mentioned above. Furthermore, there are other practices offered by PSP to increase quality of the product. PSP3 emphasizes the use of reviews to ensure that the input to subsequent system increments is ‘clean’ versions of software. Though some agile methods recommend to do review too, the review does not use a checklist based on the engineer’s defect profile as is the case with PSP. Coleman and Verbruggen [9] suggested each individual developer to conduct a design and code review with testing then executed by a technical peer to ensure more error-free product for the user in the DSDM and PSP integration case. Svensson [10] believed it’s a better way to include a checklist based on the engineer’s defect profile when performing the review in the RUP and PSP integration case.

(5) PSP provides additional documentation for agile methods. As being lightweight, agile methods recommend usage of a small amount of documentation. However adoption of appropriate documentation from PSP may be cost-effective. Except for the documentation mentioned above, there are other examples of PSP documentations employed in agile projects. According to Mihaylov, Ivanov etc. [5], PSP provides scripts which support each engineering activity and facilitate its correct completion as complementary to XP. Coleman and Verbruggen [9] use PIP to document process shortcomings and suggested solutions in the DSDM and PSP integration case.

(6) PSP helps individual engineers to improve their personal performance gradually.PSP enhances agile methods with concrete practices which will provide more useful and effective guidance to software developers. It helps them to better estimate, plan and track their daily work. And as PSP is organized in levels, engineers and companies could adopt a level in accord with their current state and maturity level, and then to improve them step by step. Bozheva [12] stated that the combination of XP with the selected set of PSP practices makes the developers more aware about their abilities to finish a particular task at a particular level of quality during a certain time. Dzhurov, Krasteva and Ilieva [13] also argued that developers can learn from their performance variations and improve the process.

When combining PSP with agile methods, some challenges derive from the fact that they belong to different cultures. Agile methodologies value people to be the critical factor while PSP value process. It's somehow difficult to introduce some changes in people's culture and the way of their thinking. For example, to convince the clients and higher management that already feel comfortable with the traditional software engineering approaches to adopt agile approaches [4].

PSP and agile methodologies seem as two extremes concerning documentation. On one hand, agile methods recommend creation of only necessary documents. On the other hand, PSP requires almost every activity to be documented. This indicates the adoption of only needful PSP principles which are most important and cost-effective for reaching the goal and with additional modifications instead of usage of all PSP principles blindly. Ilieva and Stefanova [14] suggested that those modified principles could be used mainly for measuring defects and effectiveness of activities in order the problems to be identified in time and to be eliminated in the future. The eXPERT approach [13] incorporate all the XP practices with three from PSP: time and effort recording, defect recording and task estimation for the purpose of measuring defects and effectiveness of activities in order the problems to be identified in time and to be eliminated in the future. PXP [13] keeps the basic principles of PSP and replace the formal and complex methods for planning, system design and its verification with XP practices.

The detailed form filling for data collecting may lead to efficiency reduction of development in the initial phases. In order to prevent this, either the recording approaches should be simplified or the software tools should be used for data recording [9]. Code reviewing and inspection could also benefit from usage of software assistance tools.

4 Conclusions and Future Work

In this paper, we studied the existing evidence of integrating PSP with agile methods and identify the supports that PSP can offer for agile methods. PSP enhances agile methods with concrete practices to provide more useful and effective guidance to individual developers, including:

- quality control mechanisms
- more accurate plan and estimation
- data collecting mechanisms

- framework for gradually improve the individual competency

When incorporating PSP into agile methods, the PSP process should be adapted to the project's own needs and applied accordingly. And software tools for assistance should also be used to improve efficiency.

This paper is still part of on-going research of software process integration. Each method has its own home ground, shortcomings, and typical pitfalls besides PSP and agile methodologies. There is no individual method that can solve all problems gracefully. Sometimes, we need to integrate models to meet sophisticated needs. We will conduct a systematic review to summarize the existing evidence of software process model integration then position and provides recommendations for further research on software process model integration. Moreover, we identify 3 critical dimensions that can be used to help investigating the integration of different software process models (see Figure 1): practice and process; organization, team and personal; agile and plan-driven. The cube can serve as the foundation for exploring different software process integration and the relationship between them. Each model can be mapped into a block of the cube. For instance, PSP belongs to the block that stands for process, personal and plan-driven and Scrum process, team and agile. After mapping the model, it will be more intuitional to study different integrations for different purposes with different integrating methods like combination of models in the same block, sharing one same dimension or cross different dimensions and so forth. And then develop patterns or a framework for software process integration and identify some significant combinations that have not been studied yet.

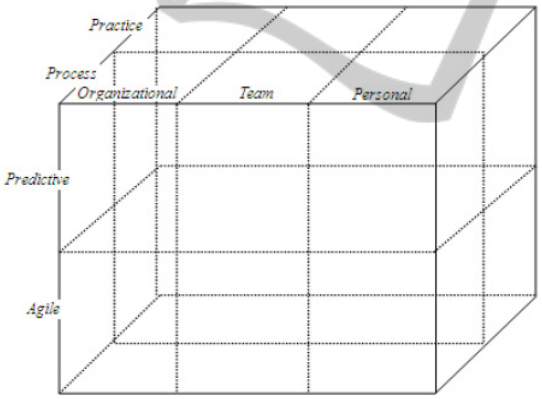

Fig. 1. The cube for software process integration.

References

1. Beck, K., et al., 2001. The Agile Manifesto. http://www.agileAlliance.org.
2. Highsmith, J., 2002. Does Agility Work? Software Development, 2002. 10(6): p. 30-37.
3. Beck, K., 2000. The XP Series: Extreme Programming Explained. Addison-Wesley.
4. Mihaylov, I., Ivanov, P., Stefanova, E., Eskenazi, A. and Ilieva, S., 2003. The expert approach – a case study. In Proceedings of the International Conference on Computer Systems and Technologies, 19-20 June 2003, Sofia, Bulgaria,CompSysTech'2003.

5. Boehm, B. and Turner, R., 2004. Balancing Agility and Discipline : Evaluating and Integrating Agile and Plan-Driven Methods. In Proceedings of the 26th International Conference on Software Engineering, Edinburgh, Scotland, May 18-28, 2004,ICSE'04, (2004), IEEE ,3-4.
6. Humphrey, W., 1995. A discipline for software engineering. Addison-Wesley.
7. Boehm, B.2002. Get Ready for Agile Methods , with Care. , IEEE Computer 35 (1) (2002) 64–69.
8. Scott Ambler. Overcoming the Myths of IBM Rational Unified Process (RUP) and Agile Development. http://www-01.ibm.com/software/info/television/html/M649306B47502P68.html
9. Coleman, G. and Verbruggen, R., 1998. A quality software process for rapid application development. Software Quality Journal 117, (1998), 107-117.
10. Svensson, H., 2005. Developing Support for Agile and Plan-Driven Methods. Royal Institute of Technology Department of Computer and Systems Sciences.
11. Agarwal, R. and Umphress, D., 2008. Extreme Programming for a Single Person Team. In Proceedings of the 46th Annual Southeast Regional Conference on XX,ACM-SE '08, March 28-29,2008,Auburn,AL,USA, 82-87.
12. Bozheva, T., 2003.Practical Aspects of XP Practices. XP 2003, Lecture Notes in Computer Science, vol. 2675, pp. 360-362. Springer, 2003.
13. Dzhurov, Y., Krasteva, I., and Ilieva, S.,2009. Personal Extreme Programming – An Agile Process for Autonomous Developers. In Proceedings of International Conference on Software, Services & Semantic Technologies, October 28-29, 2009, Sofia, Bulgaria.
14. Ilieva, S. and Stefanova,E., 2002. Expert approach for e-business software development. In Proceedings of International conference Basic Technologies for E-business'2002, Albena, 16-18 September.
15. Rong, G., Dong, S. and He, Z., 2010. SCRUM-PSP: Embracing Process Agility and Discipline. In Proceedings of 2010 Asia Pacific Software Engineering Conference,30 November – 3 December 2010, Sydney, Australia,APSEC2010.