The Role of Software Process Simulation Modeling in Software Risk Management: a Systematic Review

Dapeng Liu[1,2], Qing Wang[1], Junchao Xiao[1]
[1]*Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*
[2]*Graduate University of Chinese Academy of Sciences, Beijing 100039, China*
{liudapeng, wq, xiaojunchao}@itechs.iscas.ac.cn

Abstract

Nowadays software projects are still suffering from many problems due to various kinds of software risks. Software risk management is a crucial part of successful project management, but it is often not well implemented in real-world software projects. One reason is that project managers lack effective and practical tools to manage software risks. Software Process Simulation Modeling (SPSM) has been emerging as a promising approach to address a variety of issues in software engineering area, including risk management. However, the current state of how SPSM supports software risk management is not yet clear. This paper presents a systematic literature review which purpose is to obtain the state of the art of the applications of SPSM in software risk management. We drew the following conclusions from the review results: (1) The number of SPSM studies on software risk management is relatively small, but increasing gradually in recent years. (2) SPSM is mainly applied in risk analysis and risk management planning activities. (3) Software risks related to requirements, development process and management process are the ones most studied by SPSM. (4) Discrete-Event Simulation and System Dynamics are two most popular simulation paradigms, while Hybrid simulation methods are more and more widely used. (5) Extend, iThink and Vensim are the most popular simulation tools in SPSM. (6) Most of SPSM approaches and models have not been well applied into real-world risk management practices.

1. Introduction

Software projects have been suffering from many kinds of problems, such as cost/schedule overruns and poor product quality, for quite a long time. One of the most important factors for these project problems is unmanaged risks [1].

Software risk management is a crucial part of successful software project management. Software engineering researchers and practitioners have proposed lots of approaches and techniques for managing software risks systematically and effectively. One of the most important works is the software risk management framework proposed by Barry Boehm [2]. In this framework, the practice of software risk management involves two primary steps each with three subsidiary activities: risk assessment step involves risk identification, risk analysis and risk prioritization activities, while risk control step involves risk management planning, risk resolution and risk monitoring activities [2]. This framework defined the basic activities for software risk management, which have been adopted by many other risk management models and processes.

Although there are many approaches and techniques, software risk management is often neglected in real-world project management [3]. A study by the Project Management Institute showed that risk management is the least practiced of all project management disciplines in the IT industry [1]. In actual software projects, risks are often managed by intuition of project managers, and the complete risk management process is rarely followed [4]. One of the main reasons for this phenomenon is that project managers lack practical techniques and tools to effectively manage software risks. Existing approaches and models for software risk management are rarely applied in actual software projects, because either they are too general to guide the operational risk management activities, or their applicability is limited to some special scenarios.

Software Process Simulation Modeling (SPSM) is a promising approach which has been used to address a variety of issues in software engineering [5, 7] since it

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

was introduced into this field in the late 1980s [6]. Zhang et al. [7] carried out a systematic review on the SPSM research in the recent decade. The review results showed that risk management is one of ten purposes for SPSM. However, the details of how SPSM has been used to support software risk management are unclear. Zhang's systematic review [7] did not provide further information on this topic.

In order to make clear the role of SPSM in software risk management at present, we conducted a systematic literature review on this topic. The objective of this review is to obtain a global vision of current applications of SPSM in software risk management, which further indicates future directions for software engineering researchers and practitioners.

This paper reports the process and results of this systematic review. It is organized as follows. Section 2 describes the detailed process of this systematic review. The results of the review and discussions of the findings are presented in Section 3 and 4 respectively. Finally, in Section 5, we present limitations of the systematic review, and summarize this research.

2. Systematic review process

Systematic literature review, which is also referred to as systematic review, is one of the most important methods in Evidence-Based Software Engineering (EBSE) proposed by Kitchenham et al. [13]. It is an unbiased and repeatable means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest [8].

Our systematic review followed the guidelines proposed by Kitchenham [8]. Three researchers participated in the review process. One PhD student acted as the principal reviewer, who was responsible for developing the review protocol, searching and selecting primary studies, assessing the quality of primary studies, extracting and synthesizing data, and reporting the review results. One senior researcher was responsible for validating the review protocol, monitoring the study searching and selecting process, and checking the data of study quality assessment and information extraction. The other researcher acted as a mediator who dealt with divergences between the two researchers' decisions.

2.1. The research questions

The research question (RQ) of this systematic review is "*What is the state of the art of the applications of SPSM in software risk management?*" This question can be split into the following five more specific research questions:

- RQ1: Which risk management activities have employed SPSM?
- RQ2: Which kinds of software risks SPSM has been applied to manage?
- RQ3: Which SPSM paradigms have been used in software risk management?
- RQ4: Which SPSM tools have been used to support software risk management?
- RQ5: How practical are the existing SPSM approaches and models in real-world software risk management?

2.2. Search strategy

In order to perform an exhaustive search for primary studies, our search strategy consisted of manual search in major SPSM sources, and online search in relevant digital libraries.

Firstly, in order to ensure the quality of this review, we searched three major sources of SPSM manually:

- The proceedings of the *ProSim* [1] workshop (1998-2006) and *International Conference on Software Process (ICSP)* [2] (2007-2008): The *ProSim* workshop initiated in 1998 was the most important symposium on SPSM, which has become a special track of *ICSP* since 2007.
- *Software Process: Improvement and Practices (SPIP)* [3] (1996-2007): Authors of good papers from *ProSim* and *ICSP* were often invited to submit an extended version of their papers to the journal *SPIP*, so it is another important source of SPSM.
- *Journal of Systems and Software (JSS)* [4] : *JSS* published two special issues on SPSM: Volume 46, Issue 2-3, 1999, and Volume 59, Issue 3, 2001.

Secondly, in order to improve the completeness of this review, we searched four major digital libraries with high relevance to software engineering:

- *IEEEXplore* [5]
- *ACM Digital Library* [6]
- *ScienceDirect* [7]
- *SpringerLink* [8]

[1] http://www.prosim.pdx.edu/
[2] http://www.icsp-conferences.org/
[3] http://www3.interscience.wiley.com/journal/15482/home
[4] http://www.elsevier.com/wps/find/journaldescription.cws_home/505732/description#description
[5] http://ieeexplore.ieee.org/
[6] http://portal.acm.org/
[7] http://www.sciencedirect.com/
[8] http://www.springerlink.com/

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

For the online search in digital libraries, search keywords are very important for the quality of retrieved results, so they must be chosen carefully. The basic search string we used was *("process simulation" AND software)*. The reason why *"risk"* is not included in the search string was that it cannot help improving the searching accuracy. On one hand, *"risk"* is a common word which cannot effectively exclude false positives from retrieved results. On the other hand, some studies which can be regarded as software risk management did not mention *"risk"* explicitly. Although there was a lot of noise in the retrieved results, we thought it is worth doing to ensure the quality of the systematic review.

As to the phrase *"process simulation"*, there are many other kinds of "process simulation" besides "software process simulation", such as business process simulation, chemical process simulation and so on, so we used the keyword *"software"* to limit the search scope. In order to verify the effectivity of this search string, we did a pilot experiment among about 20 SPSM studies. The results showed that all these studies contain the keywords *"process simulation"* and *"software"*.

Due to the different functions and features of search engines, the search strings for the four digital libraries were similar but not the same. Table 1 shows the specific search strings for the four digital libraries.

Table 1. Search strings for digital libraries

Digital libraries	Search string
IEEEXplore	'process simulation' and software
ACM Digital Library	("process simulation") and (software)
ScienceDirect	Subject: Computer Science FULL-TEXT("process simulation") and FULL-TEXT(software) AND EXCLUDE(smi, "41850,5748,5657","Computer Aided Chemical Engineering, Microelectronics Journal, Microelectronic Engineering") AND EXCLUDE-BEF(pubyr, "1999")
SpringerLink	"process simulation" and software subject > Software Engineering

2.3. Study selection

The inclusion and exclusion criteria were defined explicitly in the review protocol we developed for this systematic review. The included primary studies should belong to one of the following two categories:

- Category A: Studies which proposed new approaches or models of SPSM which could be applied in software risk management.
- Category B: Studies which applied existing approaches or models of SPSM to the real-world project risk management.

The following studies would be excluded during study selection:

- Studies on SPSM but not regarded as software risk management.
- Studies focusing on simulation modeling methodologies or languages.
- Studies on education and training.
- Studies which introduced simulation tools.
- Tutorials, editorials, posters, position papers, panel papers, keynotes, abstract, and so on.

With regard to multiple papers on the same work or continued research, only the latest or the most comprehensive one would be included.

Our study selection process was organized into two separate steps, described as follows:

Step 1: For searched studies, determine whether these studies were relevant studies based on their titles, abstracts and keywords. Relevant studies are the potential candidates for primary studies. The full papers of relevant studies were kept for further inspection.

Step 2: For every relevant study, review the full paper carefully to determine whether it was a primary study.

The second step of the study selection process was necessary because in the software engineering field, title, abstract and keywords are usually not enough to determine the content of a paper [9]. Therefore, we needed to review the full paper before making a final decision to include or exclude a relevant study.

2.4. Study quality assessment

We developed a checklist to assess the quality of primary studies. The contents and scoring rule of this checklist were adapted from the one in Zhang's systematic review [7]. For every question in the checklist, the answer is "yes", "partial" or "no". For the sake of comparison, the scores were translated to numerical values. For "common questions" and "questions for primary studies in Category A", "yes", "partial" and "no" were translated to 0.1, 0.05 and 0 respectively. For "questions for primary studies in Category B", "yes", "partial" and "no" were translated to 0.12, 0.06 and 0 respectively. By translating, quality scores of primary studies in both categories are all between 0 and 1. Table 2 presents this study quality assessment checklist.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

2.5. Data extraction

We designed a data collection form for the data extraction process. The contents of the form were mainly derived from the five research questions of the systematic review. The data items in the form and their descriptions are presented in Table 3.

Table 2. Study quality assessment checklist

Common questions	**Score**
Did the study clearly state the aims/research questions?	yes/partial/no
Did the study explicitly state the role of this research in software risk management?	yes/partial/no
Did the study review the related work for the problem?	yes/partial/no
Did the study recommend the further continuous research?	yes/partial/no
Questions for primary studies in Category A	
Are the model's assumptions explained explicitly?	yes/partial/no
Is the model construction fully described?	yes/partial/no
Did the study explain why choosing the applied simulation paradigm?	yes/partial/no
Has the model been validated or demonstrated in real projects context?	yes/partial/no
Did the study carry out a sensitivity or residual analysis?	yes/partial/no
Did the study compare the results with other models/simulators?	yes/partial/no
Questions for primary studies in Category B	
Did the study fully describe the background or context of the problem?	yes/partial/no
Did the study explain why choosing the applied model or simulator?	yes/partial/no
Did the study fully describe the process of applying the model or simulator to risk management?	yes/partial/no
Did the study analyze or evaluate the effects of applying the model or simulator?	yes/partial/no
Did the study summarize the lessons learned from the application?	yes/partial/no

Table 3. Items and descriptions of the data extraction form

Common items	**Description**
Year	The year when the primary study was published.
Source	The conference, journal or book where the primary study was published.
Category	The category of the primary study: A or B
Risk Management Activities	The activities which SPSM was applied to, in the Boehm's risk management framework [2].
Software Risks	The software risks involved in the primary study. The software risks were categorized according to the SEI's risk taxonomy [10].
Number of Software Risks	The number of software risks involved in the primary study.
Simulation Paradigm	The simulation paradigm used to build the simulation model.
Simulation Tool	The simulation tool used to build and execute the simulation model.
Applied in Practice	Whether the primary study has been applied to risk management of real-world software projects. *Yes*: The primary study has been applied to actual software projects. *Partial*: The primary study has not been applied in practice, but it was validated or evaluated by actual project data. *No*: The primary study has not been applied in practice at all.

3. Results of the systematic review

Following the review process described in Section 2, 124 papers were identified as relevant studies from 1419 search results, and finally 27 papers from relevant studies were determined as primary studies after carefully reviewing. The details of the study search and selection are shown in Table 4. It is noted that the sources in manual search and digital libraries are partly overlapped. The proceedings of *ProSim* in 2006 and *ICSP* are included in the *SpringerLink*, and *JSS* is a

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

journal of *ScienceDirect*. Duplicated studies were excluded during online search in digital libraries. Appendix in Section 8 at the end of this paper lists all 27 primary studies.

Table 4. Details of study search and selection

Source	Search Date	Search Results	Relevant Studies	Primary Studies
Proceedings of *ProSim* and *ICSP*	2008.9.1	146	50	**11**
SPIP	2008.9.20	318	16	**2**
JSS Special Issues	2008.10.8	23	6	**2**
IEEEXplore	2008.10.11	115	15	**2**
ACM Digital Library	2008.10.12	364	7	**1**
ScienceDirect	2008.10.13	191	12	**4**
SpringerLink	2008.10.12	262	18	**3**
Other known papers				**2**
Total		1419	124	**27**

Among these primary studies, 22 studies fall into the Category A, and the rest belongs to the Category B. The average quality score of all primary studies is 0.620, ranging from 0.4 to 0.8. In the Category A, the average quality score is 0.616, while in the Category B, the average quality score is 0.636. It is similar in study quality between these two categories.

These primary studies were published from the year 1994 to 2008[9]. The distribution of studies by year is shown in Figure 1. We note that there are only 2 primary studies published before 2000. One possible reason is that SPSM had not gained interests of software engineering researchers and practitioners at that time. Another explanation may be that the most important symposium on SPSM *ProSim* was hold since 1998. After 2000, there are more studies on applying SPSM to risk management. There is a pulse in the year 2001 because of the special issue of *JSS* on SPSM (Volume 59, Issue 3). From 2005, the number of primary studies is gradually increasing, which may indicate that it is becoming an active research topic, gaining more attentions from software engineering researchers and practitioners. It is noted that the number in 2008 is not representative because the year has not ended when we performed this review.

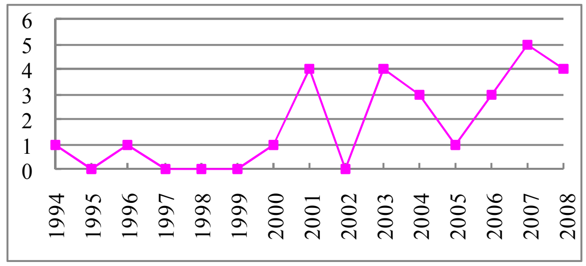

Figure 1. Distribution of primary studies by year

Figure 2 shows the distribution of primary studies by source. *ProSim*/*ICSP* are the most important symposium on SPSM, which contributed the most of the primary studies. Other important sources for primary studies are *JSS*, *International Conference on Product Focused Software Process Improvement (PROFES)*, and *Information and Software Technology (IST)*. It is noted that the two studies in *SPIP* and two of the four studies in *JSS* firstly appeared on *ProSim*, so actually *ProSim*/*ICSP* contributed more than half of primary studies (15 of 27). The other 6 primary studies were respectively published in *Annual NASA Goddard Software Engineering Workshop*, *ACM SIGSOFT Software Engineering Notes*, *Value-Based Software Engineering* (book), *Journal of Systems Architecture*, *Computer* and a PhD dissertation.

The following five subsections will respectively answer the five research questions of this systematic review.

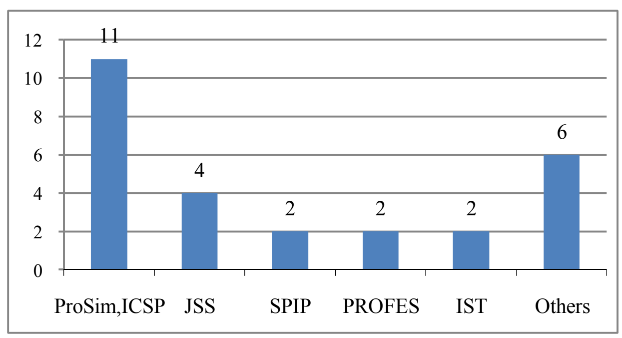

Figure 2. Distribution of primary studies by source

3.1. Risk management activities (RQ1)

The summary of risk management activities in Boehm's risk management framework involved in primary studies is shown in Figure 3. Among 27 primary studies, most of them (18) involved one risk management activity, while 7 studies involved two

[9] One primary study [PS17] was formally published in 2009. But it was still counted as in 2008 because it had been accepted when we performed this review.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

activities, 1 study involved three activities, and 1 study involved four activities.

Figure 3 shows that most of the SPSM approaches and models focused on risk analysis and risk management planning activities. In risk analysis activity, the typical application of SPSM is analyzing the variation of key project factors due to software risks to assess the impact or predict the loss caused by risks. In risk management planning activity, the typical application of SPSM is predicting the effects of various actions or changes on software projects for mitigating or resolving software risks, to help project managers choose the most suitable one for risk management plan.

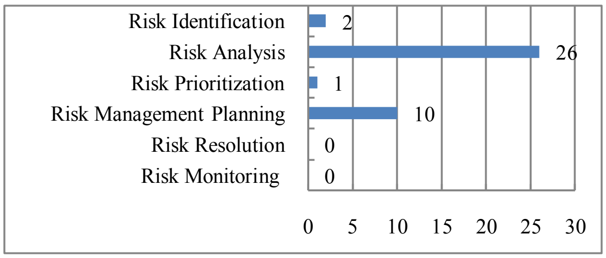

Figure 3. Risk management activities

An interesting phenomenon is that most primary studies which were applied in risk management planning (9 of 10) were also applied in risk analysis activity. In fact, we found that many other SPSM approaches and models applied in risk analysis were possible to be extended to support risk management planning activity.

Although there are two studies involving risk identification activity, the capabilities of identifying risks were all provided by other parts of these studies, not by SPSM. The capabilities of SPSM in risk identification, prioritization, resolution and monitoring activities have not been well explored, and more research is needed on it

3.2. Software risks (RQ2)

SEI's widely-accepted software risk taxonomy has three levels: class, element and attribute [10]. We categorized software risks involved in primary studies on the element level. It is noted that although one software risk is possible to be put into more than one category, we put every risk in exact one category the most suitable for it. The distribution of software risks among categories is shown in Figure 4. Among the 27 primary studies, 3 studies proposed general approaches or models for software risk management, not for specific software risks, which will be discussed later in Section 4.3. Most of the other primary studies (14 of 24) involved one software risk, while 6 studies involved two risks, 2 studies involved three risks, 1 study involved four risks, 1 study involved six risks, and 1study involved twelve risks.

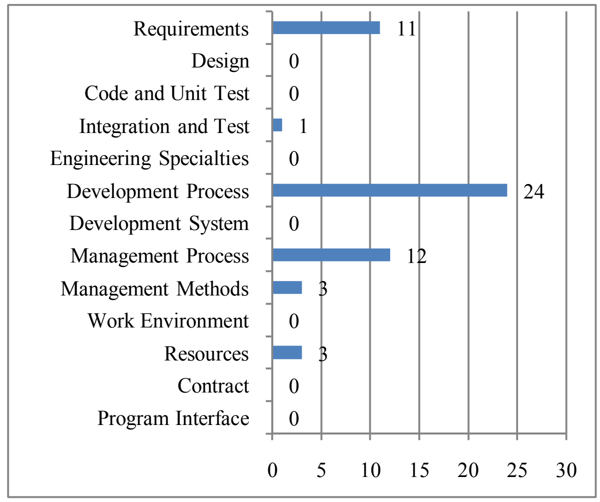

Figure 4. Distribution of software risks

In the "Product Engineering" class of the risk taxonomy [10], the most studied software risks are the "Requirements" related ones. It is consistent with the data in Software Engineering Risk Repository [11]. Many researches also showed that requirements-related risks are one of the most important ones in software projects [2, PS6]. We think the reason is that software requirements are a critical factor for software projects' success, and risks on requirements more likely lead to severe damage to software projects [2, 4].

In the "Development Environment" class, the most studied software risks are the "Development Process" and "Management Process" related risks. It is reasonable because SPSM is suitable for analyzing processes. For "Development Process", SPSM was mainly used to predict the project performance after changing the development process. For "Management Process", the major problem SPSM was applied to address was project planning. The 3 risks in "Management Methods" are all about personnel management.

In the "Program Constraints" class, 3 risks in the "Resources" are all about staff availability and continuity. Human resource is another important source of software risks.

In brief, SPSM is good at analyzing and managing software risks related to requirements and processes.

3.3. SPSM paradigms (RQ3)

The distribution of SPSM paradigms used in primary studies is presented in Figure 5.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

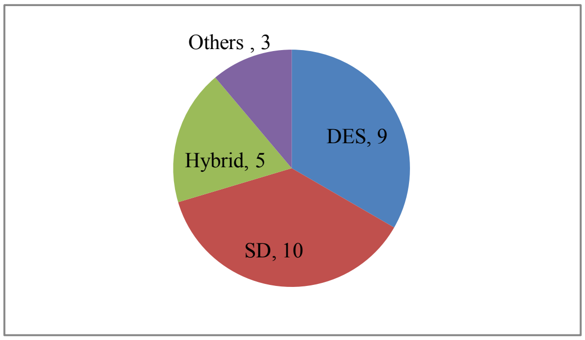

Figure 5. Distribution of simulation paradigms

The results show that Discrete-Event Simulation (DES) and System Dynamics (SD) were the most widely used simulation paradigms. Hybrid simulation was more and more frequently applied in SPSM in recent years. Most of the hybrid simulation models (4 of 5) were combinations of DES and SD, where SD was used in higher level and DES in lower level of the models. The other hybrid simulation model incorporated three paradigms: DES, continuous and analytical methods [PS26]. Another difference between this model and other hybrid ones was that the model applied DES method in the higher abstract level and continuous and analytical methods in the lower level.

3.4. SPSM tools (RQ4)

Figure 6 presents the simulation tools distribution used in primary studies. The results show that *Extend*[10], *Vensim* [11] and *iThink* [12] were the most popular simulation tools in software risk management.

Extend is primarily used for building DES models. 7 of 9 studies using DES paradigm chose *Extend* as their modeling tool. Moreover, *Extend* is capable of building continuous models, thus it can be used in hybrid simulation modeling. In the 5 studies which proposed hybrid simulation models, 3 of them used *Extend*.

Vensim and *iThink* were major tools for building continuous simulation models. 8 of 10 studies which adopted SD paradigm used *Vensim* or *iThink* as their modeling tool.

2 studies developed their own simulation tools using *Smalltalk* and *Java* respectively. 1 study used *Crystal Ball*[13]. 6 studies did not mention the simulation tools they used.

[10] http://www.extendsim.com/
[11] http://www.vensim.com/
[12] http://www.iseesystems.com/
[13] http://www.oracle.com/crystalball/

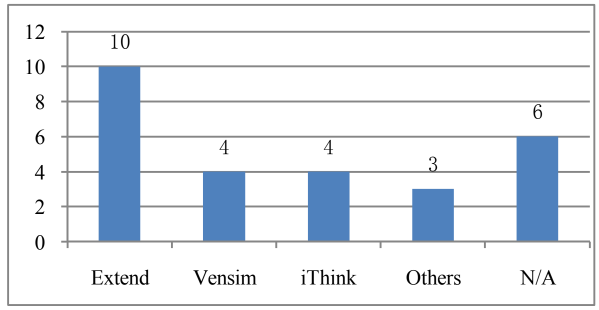

Figure 6. Distribution of simulation tools

3.5. Practical applications of SPSM (RQ5)

The state of applications in practice of the primary studies is shown in Figure 7. It is disappointing that only 22.2% (6 of 27) of the primary studies have been applied in the real-world software projects risk management. Nearly half of studies (12 of 27) were validated or evaluated using actual project data, while the others were still in laboratory. We think the results are reasonable for such a rapidly developing approach. It also indicates that more work is needed for software engineering researchers and practitioners in applying SPSM into software risk management practices.

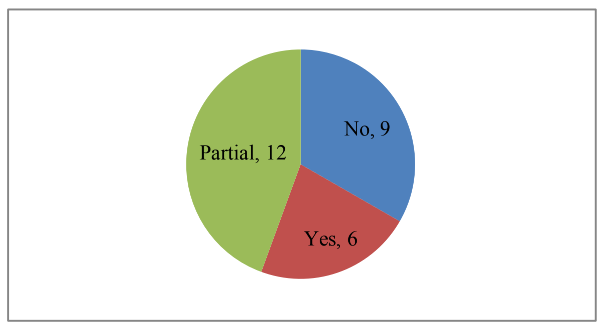

Figure 7. Applications of SPSM in practice

4. Discussions

4.1. Some findings in the review process

Although “risk management” is one of the main purposes for SPSM research [7], the number of primary studies for this purpose is relatively small. Moreover, among the 27 primary studies of this systematic review, only 7 studies explicitly stated the role of SPSM in the software risk management, while another 3 studies mentioned risk management activities implicitly. The other studies did not relate themselves to risk management, although their approaches and models could be applied to one or more risk management activities. We think it may be because software risk management has not been given

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

sufficient attention in project management. However, the increasing number of studies on this topic in recent years may indicate software risk management is getting more and more interests from software engineering researchers and practitioners.

SPSM has been introduced into software engineering area for two decades since Abdel-Hamid's pioneering work [6]. During the review process, we found the focus and simulation paradigms are different between studies in the first decade and the last decade. At the early stage, researchers primarily focused on proposing new simulation modeling methodologies and building simulation models for various software processes. SD was the predominant simulation paradigm at that time because it is suitable to model software processes in the high and macro level. In contrast, studies of SPSM in recent years paid more attention to addressing specific issues in software projects, such as requirements creep and technology adoption. DES was increasingly used in SPSM research. DES models have explicit representations of factors of software projects in great detail, supporting analysis in low and micro level. It is another possible explanation why the number of primary studies of the review is relatively small before 2000, and increasing in recent years.

As to authors of the primary studies, David Raffo [PS14, PS21, PS23, PS25, PS27] and Dietmar Pfahl [PS5, PS9, PS10, PS11, PS19] contributed the most primary studies. Raffo proposed Generalized Process Simulation Model (GPSM), and built a tailored IEEE12207software process model with IV&V process being used at NASA [12]. Three of Raffo's primary studies used this model to investigate technology adoption and process change [PS14, PS21, PS23]. Three of Pfahl's primary studies focused on risks in operational release planning [PS9, PS10, PS11]. He and his colleagues built DES model [PS9, PS10] and SD model [PS11] to address this issue.

In fact, both Raffo and Pfahl are leading researchers in SPSM community. In addition, they are important organizers of *ProSim/ICSP*, which is the main symposium on SPSM.

4.2. Research problems of primary studies

The research problems addressed in the primary studies were various across the software engineering field. Some research problems gained interests from more than one primary studies. As one of the most frequent and severe risks in software projects, requirements creep/volatility was investigated in 11 primary studies. Other common research problems included software release planning (3 studies), technology adoption and substitution (5 studies), XP practices (3 studies), and rework (2 studies). These problems are all important sources of risks in software projects.

Another kind of research problems widely investigated by primary studies was staff-related risks, including staff turnover, lack of staff commitment, low morale, lack of staff continuity, instability of personal productivity, and so on. Human resource is one of the most important factors for software projects, in the meanwhile it is also one of the most difficult parts in project management due to uncertainties of human behaviors.

4.3. General risk management approaches using SPSM

Among the primary studies, there are three ones which proposed general approaches or models for software risk management [PS18, PS19, PS20].

Barros et al. [PS18] proposed scenario based risk management, which introduced a very important kind of artifacts: risk archetypes. Every risk archetype contained all information of one software risk needed by project managers. SPSM was used to analyze the impact caused by the risk, predict the effects of strategies mitigating and resolving the risk in the risk archetype. Risk archetypes were reusable, so they were valuable for improving the risk management of software organizations.

Pfahl proposed a SPSM-based risk analysis procedure ProSim/RA [PS19]. It was a general guideline for analyzing software risks using SPSM. Although the case study in [PS19] used SD paradigm, the procedure itself was independent of simulation paradigms. This procedure was adopted by Al-Emran et al. [PS9] to analyze several risks in operational release planning.

Madachy developed a SD model for an inspection-based software process which could support quantitative evaluation of the process, cost estimation and project risk assessment [PS20]. This model incorporated a knowledge-based component which used expert heuristics to assess project risks based on cost factors. It could analyze, rank and prioritize individual risks, and give advice to help manage these risks.

5. Limitations and summary

The main limitation of this research is that the review process recommended for PhD students is less rigorous than the one used by multi-researchers. However, in our review process there were two senior researchers who monitored all activities and checked

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

the main documents produced in the systematic review. We believe our review process can assure the quality of the systematic review.

In this paper, we present the results of a systematic review on the topic of the role of SPSM in software risk management. The aim of this systematic review is to obtain the state of the art of the applications of SPSM to software risk management up to the present day. The results of this research can also serve to indicate future directions for software engineering researchers and practitioners.

Based on the results obtained from the systematic review, we summarize the following conclusions on this topic.

- Although the number of primary studies is relatively small, the tendency of gradually increasing publications during recent years may indicate applying SPSM to software risk management is becoming an active research topic.
- Currently SPSM is mainly applied in risk analysis and risk management planning activities. The capabilities of SPSM in other risk activities have not been well explored. More research is needed on it.
- Software risks related to requirements, development process and management process are the ones most studied by SPSM.
- Discrete-Event Simulation and System Dynamics are two most widely used simulation paradigms. Hybrid simulation is increasingly adopted in software risk management.
- *Extend* is the most popular simulation tool for DES and Hybrid simulation modeling, while *Vensim* and *iThink* are two widely used tools for continuous simulation.
- Most of the existing SPSM approaches and models have not been applied into actual risk management practices. Software engineering researchers and practitioners should cooperate more closely in the future.

6. Acknowledgement

This research is supported by the National Natural Science Foundation of China under grant Nos. 60573082, 90718042, the Hi-Tech Research and Development Program (863 Program) of China under grant No. 2006AA01Z185, 2007AA010303, 2007AA01Z186, as well as the National Basic Research Program (973 program) under grant No. 2007CB310802.

7. References

[1] Charette, R.N., “Why software fails”, *IEEE Spectrum*, 2005. 42(9): pp. 42-49.

[2] Boehm, B.W., “Software risk management: principles and practices”, *IEEE Software*, 1991. 8(1): pp. 32-41.

[3] Dedolph, F.M., “The Neglected Management Activity: Software Risk Management”, *Bell Labs Technical Journal*, 2003. 8(3): p. 91-95.

[4] Pandian, C.R., *Applied Software Risk Management: A Guide for Software Project Managers*, Auerbach Publications, 2006.

[5] Kellner, M.I., Madachy, R.J., and Raffo, D.M., “Software process simulation modeling: Why? What? How?”, *Journal of Systems and Software*, 1999. 46(2-3): pp. 91-105.

[6] Abdel-Hamid, T.K., “The dynamics of software project staffing: a system dynamics based simulation approach”, *IEEE Transactions on Software Engineering*, 1989. 15(2): pp. 109-119.

[7] Zhang, H., Kitchenham, B., and Pfahl, D., “Reflections on 10 Years of Software Process Simulation Modeling: A Systematic Review”, in *Making Globally Distributed Software Development a Success Story*, 2008. pp. 345-356.

[8] Kitchenham, B.A., “Guidelines for Performing Systematic Literature Reviews in Software Engineering”, Software Engineering Group, School of Computer Sciences and Mathematics, Keele University, and Department of Computer Science, University of Durham, 2007.

[9] Brereton, P., et al., “Lessons from applying the systematic literature review process within the software engineering domain”, *Journal of Systems and Software*, 2007. 80(4): pp. 571-583.

[10] Carr, M.J., et al., “Taxonomy-Based Risk Identification (CMU/SEI-93-TR-6, ESC-TR-93-183)”, Software Engineering Institute, Carnegie Mellon University: Pittsburgh, Pa., 1993.

[11] Higuera, R.P. and Haimes, Y.Y., “Software Risk Management (CMU/SEI-96-TR-012 ESC-TR-96-012)”, Software Engineering Institute, Carnegie Mellon University: Pittsburgh, Pa., 1996.

[12] Raffo, D., Nayak, U., Wakeland, W., “Implementing Generalized Process Simulation Models”, the 6th International Workshop on Software Process Simulation and Modeling (ProSim), St. Louis, Missouri, 2005.

[13] Kitchenham, B.A., Dybå, T., and Jørgensen. M., “Evidence-Based Software Engineering”, in Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society, 2004.

8. Appendix: primary studies

[PS1] Hansen, G. A., “Simulating software development processes”, *Computer*, 1996. 29(1): 73-77.

[PS2] Eickelmann, N. S., “Empirical studies to identify defect prevention opportunities using process simulation technologies”, in Proceedings of 26th Annual NASA Goddard Software Engineering Workshop, 2001.

[PS3] Kuppuswami, S., Vivekanandan, K., Ramaswamy, P., Rodrigues, P., “The effects of individual XP practices on

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

software development effort", *SIGSOFT Software Engineering Notes*, 2003.28(6): 6-6.

[PS4] Turnu, I., Melis, M., Cau, A., Setzu, A., Concas, G., Mannaro, K., "Modeling and simulation of open source development using an agile practice", *Journal of Systems Architecture*, 2006.52(11): 610-618.

[PS5] Pfahl, D., Lebsanft, K., "Using simulation to analyse the impact of software requirement volatility on project performance", *Information and Software Technology*, 2000.42(14): 1001-1008.

[PS6] Houston, D. X., Mackulak, G. T., Collofello, J. S., "Stochastic simulation of risk factor potential effects for software development risk management", *Journal of Systems and Software*, 2001.59(3): 247-257.

[PS7] Neu, H., Hanne, T., Munch, J., Nickel, S., Wirsen, A., "Creating a Code Inspection Model for Simulation-based Decision Support", in Proceedings of 4th International Workshop on Software Process Simulation and Modeling (ProSim), 2003.

[PS8] Melis, M., Turnu, I., Cau, A., Concas, G., "Evaluating the Impact of Test-First Programming and Pair Programming through Software Process Simulation", *Software Process: Improvement and Practice*, 2006, 11(4).

[PS9] Al-Emran, A., Kapur, P., Pfahl, D., Ruhe, G., "Simulating Worst Case Scenarios and Analyzing Their Combined Effect in Operational Release Planning", in *Making Globally Distributed Software Development a Success Story*, 2008. p. 269-281.

[PS10] Al-Emran, A., Pfahl, D. and Ruhe, G., DynaReP: A Discrete Event Simulation Model for Re-planning of Software Releases, in Software Process Dynamics and Agility. 2007. p. 246-258.

[PS11] Al-Emran, A. and Pfahl, D., Operational Planning, "Re-planning and Risk Analysis for Software Releases", in Product-Focused Software Process Improvement (PROFES). 2007. p. 315-329.

[PS12] Nonaka, M., Zhu, L., Ali Babar, M., Staples, M., "Project Cost Overrun Simulation in Software Product Line Development", in Product-Focused Software Process Improvement (PROFES). 2007. p. 330-344.

[PS13] Powell, A., Murdoch, J. and Tudor, N., "Modeling Risk-Benefit Assumptions in Technology Substitution", in *Software Process Dynamics and Agility*. 2007. p. 295-306.

[PS14] Raffo, D., Ferguson, R., Setamanit, S., Sethanandha, B. D., "Evaluating the Impact of the QuARS Requirements Analysis Tool Using Simulation", in *Software Process Dynamics and Agility*. 2007. p. 307-319.

[PS15] Aranha, E. and Borba, P., "Using Process Simulation to Assess the Test Design Effort Reduction of a Model-Based Testing Approach", in *Making Globally Distributed Software Development a Success Story*. 2008. p. 282-293.

[PS16] Liu, D., Wang, Q., Xiao, J., Li, J., Li, H., "RVSim: A Simulation Approach to Predict the Impact of Requirements Volatility on Software Project Plans", in *Making Globally Distributed Software Development a Success Story*. 2008. p. 307-319.

[PS17] Choi, K. and Bae, D.-H., "Dynamic project performance estimation by combining static estimation models with system dynamics". *Information and Software Technology*, 2009. 51(1): p. 162-172..

[PS18] Barros, M.d.O., Werner, C.M.L. and Travassos, G.H., "Supporting risks in software project management", *Journal of Systems and Software*, 2004. 70(1-2): p. 21-35.

[PS19] Pfahl, D., "ProSim/RA — Software Process Simulation in Support of Risk Assessment", in *Value-Based Software Engineering*. 2006. p. 263-286.

[PS20] Madachy, R., "A software project dynamics model for process cost, schedule and risk assessment", Department of Industrial and Systems Engineering, USC, Ph.D Dissertation, 1994.

[PS21] Raffo, D. M., Menzies, T., "Evaluating the Impact of a New Technology Using Simulation: The Case for Mining Software Repositories", the 6th International Workshop on Software Process Simulation and Modeling (ProSim), 2005.

[PS22] Martin, R. H., "What Makes Software Management Hard?", in 5th International Workshop on Software Process Simulation and Modeling (ProSim), 2004.

[PS23] Raffo, D., Nayak, U., Setamanit, S., Sullivan, P., Wakeland, W., "Using Software Process Simulation to Assess the Impact of IV&V Activities", in 5th International Workshop on Software Process Simulation and Modeling (ProSim), 2004.

[PS24] Ferreira, S., Collofello, J., Shunk, D., Mackulak, G., Wolfe, P., "Utilization of Process Modeling and Simulation in Understanding the E□ects of Requirements Volatility in Software Development", in 4th International Workshop on Software Process Simulation and Modeling (ProSim), 2003.

[PS25] Wakeland, W., Martin, R. H., Raffo, D., "Using design of experiments, sensitivity analysis, and hybrid simulation to evaluate changes to a software development process: a case study". *Software Process: Improvement and Practice*, 9(2-3), 2004.

[PS26] Donzelli, P., Iazeolla, G., "Hybrid simulation modelling of the software process", *Journal of Systems and Software*, 59(3), 2001.

[PS27] Martin, R., Raffo, D., "Application of a hybrid process simulation model to a software development project", *Journal of Systems and Software*, 59(3), 2001.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE