A Technical Perspective of Knowledge Management in Collaborative Software Maintenance Environment

Mohd Zali Mohd Nor, Rusli Hj. Abdullah

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, Serdang, Selangor
E-mails : zali@itstar.com.my, rusli@fsktm.upm.edu.my

ABSTRACT

Knowledge management (KM) is critical in Software Maintenance (SM) organizations to provide an environment for creating and sharing knowledge. SM environment is complex, knowledge-driven and highly collaborative. Issue such as inadequate knowledge is still regarded as a major challenge in SM. To find out more on the above limitation in term of technical perspective, we study the activities of a maintenance organization and conducted a survey and discussion. Based on the results, we conclude that domain and technical knowledge are important to maintainers, but are often not available. Also, usefulness of tools to acquire and share knowledge are limited, mainly due to lack of integration between tools.

Keywords:
Knowledge Management, Software Maintenance, Collaborative Technology

1.0 INTRODUCTION

In recent years, many organizations consider knowledge management (KM) to be strategically important to their business (KPMG, 2000; Lawton,2001; Rus & Lindval, 2001). With spending estimated at USD12 billion in 2012, KM is envisaged to contribute to the organizations in the following manners (KPMG,2003):

- Bring synergies among different teams, units or departments
- Accelerate innovation and boosting revenues for market development.
- Improve quality in operational and functional processes
- Reduce costs and exposure to business risks

With the above 'promises', KM are also enticing to I.T. departments and organizations, especially in managing software engineering activities. However. within software engineering activity cycle, SM has yet to receive proper attention although it has been identified as costlier process (SWEBOK,2004). Previous works (Fjelstad & Hamlen, 1998; Lientz et al.,1978; Pigoski,1997;Schach et al.,2003) estimated SM costs of between 60% to 90% of total software life cycle costs.

Before the problems and issues in managing knowledge in collaborative SM are discussed, it is imperative to describe the technical perspective, based on the tools, environments and technologies, as follows:

- Software maintenance and it's activities
- Knowledge required in software maintenance activities, and problems associated with the knowledge
- Tools and technologies to support KM in collaborative SM.

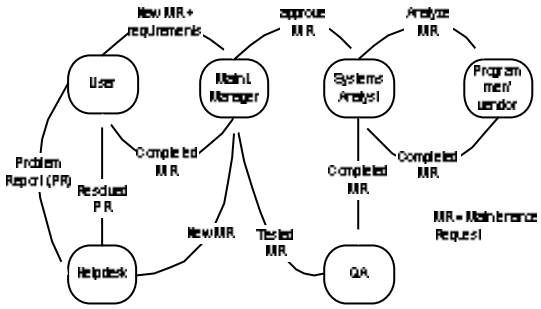
1.1 Software Maintenance Activities

As an overview, software maintenance is defined as "The totality of activities required to provide cost-effective support to software system. Activities are performed during the pre-delivery stage as well as the post-delivery stage."(IEEE 14764,2006). In term of differences in types of maintenance, The four widely known are (IEEE 14764,2006; Lientz & Swanson, 1981):

- Corrective – to fix discovered program faults.
- Adaptive – to cater for changes in business requirements and external environment.
- Perfective – to improve performance and maintainability.
- Preventive – to detect and correct latency faults before it becomes faulty.

A recent study (Schach et al., 2003) concluded that the corrective maintenance nowadays may consumes up to 50%-68% of all maintenance.

Software maintenance processes and activities have been largely standardized. Standard organizations such as ISO, IEEE, and CMMI have detailed the activities to be carried-out by software maintainers (April et al.,2005; IEEE 14764,2006). However, SM organizations may opt to have their own variations. As an example, in the studied organization, the post-delivery maintenance activities for in-house applications are based on waterfall approach, per figure 1 below:



		TIME	
		Same Time (*Synchronous*)	Different Time (*Asynchronous*)
SPACE	Same Space	***1st Quadrant*** Spontaneous collaborations, formal meetings, classrooms	***2nd Quadrant*** Design rooms, Project scheduling
	Distributed	***3rd Quadrant*** Video conferencing, net meetings, phone calls	***4th Quadrant*** Emails, blogging, authoring, voice mails, fax

knowledge, but more are hidden as tacit knowledge due to scarcity of documentation (Viscaino et al.,2003).

- The issues of inadequate knowledge available to maintainers are common. Maintainers, especially novice, spends a lot of time searching, discovering and understanding software knowledge. Earlier studies indicates around 50% of efforts are spent on this activity and rely more on source code than any other source of information (Das et al., 2007; Fjelstad & Hamlen, 1998).
- Software maintenance, by itself, is complex and is a constantly changing process, with different phases, activities and projects. Organizations often have problems identifying resources and use of knowledge (Rodriguez et al., 2004).
- More and more organizations are having distributed maintenance teams to gain advantages of different time zones, lower personnel costs, and rapid growth of internet (Sengupta,2006). As SM organizations grow and becomes more distributed, members need to collaborate and share ideas and knowledge efficiently.

To overcome the above issues, it is critical that the KMS be properly studied and formulated for software maintenance organization. As a start, this study shall first examine the usefulness of the various tools used by maintainers to store and extract information and knowledge to assist them in their daily activities.

This paper shall be structured as follows: Section 2 reviews the related works on the subject issue. Then, section 3 briefly discuss the research design and section 4 discusses the SM aspects and the technological perspective.

2.0 RELATED WORKS

Lindval et al. (2003) and Lawton (2001) elaborate the above various tools, and group the tools into document and content management, collaboration services, data and knowledge discovery, expert networks, CRM, expertise/competence management and e-learning.

However, usefulness of the abovementioned tools to support wide range of knowledge acquisition and sharing in SM is suspect. Without proper and seamless integration between tools, it is hypothetically difficult to search for exact information/knowledge required.

3.0 RESEARCH DESIGN

This study is a part of a longitudal study, with the aim to develop the framework, system architecture and tools to support KM in collaborative SM. To evaluate the usefulness of tools in knowledge acquisition and sharing in collaborative SM environment, a study was conducted in an application maintenance unit in an international shipping carrier company.

First, the SM activities and tools used by the organization were identified. Then an opinion survey was conducted among selected eight key maintainers within the organization to gauge their opinions on the important knowledge required to support maintenance activities and usefulness of the current tools to collaborate, acquire and share knowledge. The eight subjects represent the systems analysts, programmers and database administrator, who performs the bulk of major tasks in post-delivery stage of maintenance. Hence, the scope of the knowledge required shall be slightly different from other supporting roles of SM.

Based on 5-Likert's scale measurement, subjects were asked to rate their opinion on the followings:

- Importance of the following knowledge in carrying out SM activities:
  - Organizational knowledge
  - Managerial knowledge
  - Technical knowledge
  - Domain knowledge
  - Knowledge on the source or knowledge
- Usefulness of current tools to obtain information during analysis or coding:
  - Helpdesk tools
  - Knowledge portal
  - Software configuration management (SCM)
  - Source Code management
  - Collaboration tools

Due to ordinality of the data, simple descriptive analyses based on bar-charts are performed. A discussion was held after data collection, to find out the reasons behind the data.

4.0 DISCUSSIONS AND FINDINGS

In the studied SM organization, the SM activities tools, and knowledge required are summarized in table 1. In brief, different maintainers use different tools to perform different activities. Most of these tools are not integrated, as such that the user may have to open different tools to retrieve different information. In some instances, duplicate information are stored in different tools. For example, a problem ticket keyed-in into Helpdesk application is replicated in the SCM application, in order to generate the MR for bug-fixing. In other instances, to determine the cause of a problem, maintainers have to retrieve information from various sources. This is time consuming.

Based on the survey on importance of knowledge areas, technical and domain knowledge are deemed more important (see figure 2). Further discussion reveals the following issues, which makes acquiring knowledge challenging:

- Documentation are most of the time not up-to-date. Although previous designs are stored in SCM system, they are in form on MS-Word documents which are not efficient for knowledge retrieval.
- Domain knowledge are often not available, Explicit domain knowledge exists in form of Best Practice high level guidelines. Often, maintainers have to rely on users and also codes to understand the details.
- Human experts still hold most of the tacit knowledge that are not readily available to others. However, there are still reservation toward knowledge sharing, especially when job security is concerned.
- Other maintenance teams are located in Europe, USA, and Hongkong. As such, face-to-face meetings are seldom and tacit knowledge transfer is difficult.

Table 1: SM Activities, Tools and Information required/shared

Areas	**Activities**	**Tools**	**Information required/shared**
Helpdesk	log user complaints	BMC Remedy	other similar logs
	Check domain best practice	MS-Sharepoint portal	High level business rules
	Check known issues	SCM - Serena TeamTrack	known release issues
	Communicate with users	email, phone	more info on problems, inform solution
	Communicate with maintainers	email, phone, MSN Messenger	More info on source of problems
	Log MR	SCM - Serena TeamTrack	Create MR for maintenance team to fix bugs or enhance application
Maintenance Manager	Track MRs	SCM - Serena TeamTrack	MR requirements
	Plan and approve MRs	SCM - Serena TeamTrack	View open versions and datelines
		MS-Project	Resources availability
	Communicate with users	email, phone, MSN Messenger	More info on problems and requests
	Assign MRs for development	SCM - Serena TeamTrack	Organizational knowledge...Who to assign to
Analyst	Communicate with users	email, phone, MSN Messenger	More info on problems and requests
	Prepare designs and proposals	MS-Word, Serena TeamTrack	Previous similar designs in SCM repository
	Check domain best practice	MS-Sharepoint portal	business rules
	Communicate with maintainers	email, phone, MSN Messenger	design information
Programmer/QA testers	Amend codes	PVCS	version control info.
	Update progress and completion	SCM - Serena TeamTrack	MR status and priorities
	Test codes	Mercury Winrunner	Test Plans
	Communicate with analysts and manager	email, phone, MSN Messenger	design information, deadlines, business rules
	Communicate with other maintainers	email, phone, MSN Messenger	technical knowledge

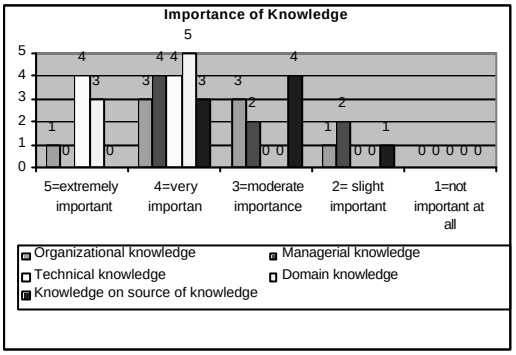


Figure 3: Importance of knowledge

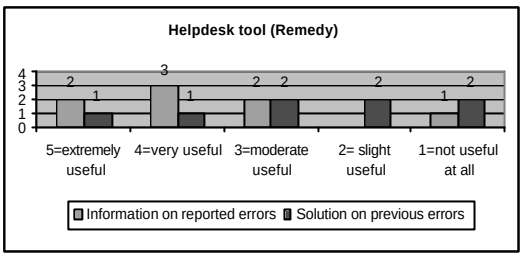


Figure 4: Usefulness of Remedy helpdesk tool

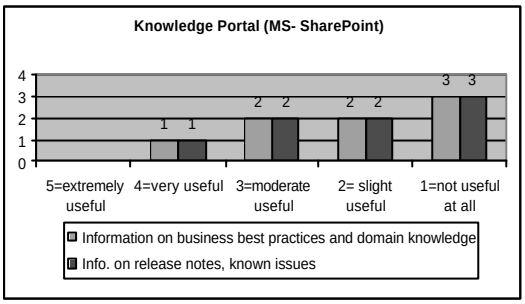


Figure 5: Usefulness of Knowledge Portal

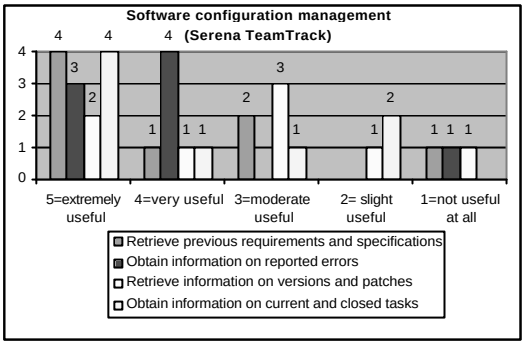


Figure 6: Usefulness of SCM tool (TeamTrack)

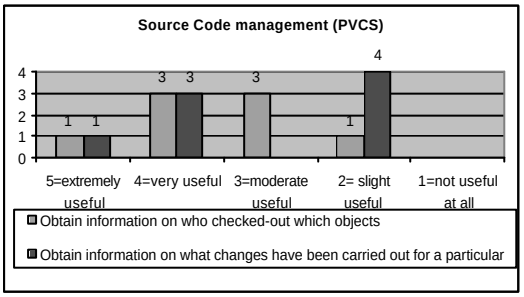


Figure 7: Usefulness of Source Code Management tool (PVCS)

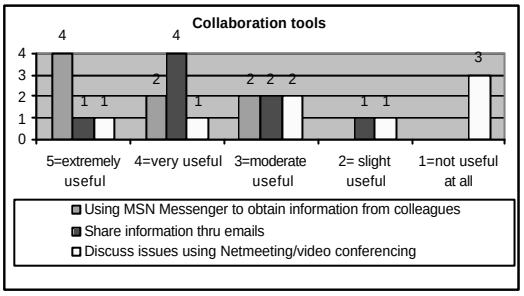


Figure 8: Usefulness of Collaboration Tools

In the studied organization, usefulness of the tools in acquiring knowledge are still suspect, as depicted in the survey results below:

- Use of helpdesk tool (BMC Remedy) is not very helpful for maintainers in finding solutions for earlier reported problem. This is due to limited search capability and the fact that maintainers prefer to go straight to the codes to identify the problems
- Maintainers seldom visits knowledge portal to seek domain information. This is due to slow responses, and lack of details on domain knowledge.
- SCM (Serena TeamTrack) is proven to be a very useful tool as it keeps majority of the explicit information, including the domain information.
- Source Code management tools does not provide detail information on what changes were carried out on the objects. This is because the information is stored in SCM. Maintainers prefer to use SCM for this purpose.
- For multi-site collaboration, MSN Messenger and email are preferred over video/netmeeting conferencing. According to the subjects, Messenger and email allows them to continue working rather than having to allocate full time for conferencing.
- The main challenge for information/ knowledge acquisition and sharing is lack of integration between tools, and the way the information is stored. Although kept in SCM and KM portal, specifications and domain knowledge are still in MS documents format, and searching for specific knowledge is still challenging.

5.0 CONCLUSION AND FUTURE WORKS

Software maintenance is complex, knowledge-intensive and highly collaborative. Managing knowledge in this area is therefore critical to ensure that maintainers can perform software maintenance activities properly and timely, by sharing and obtaining vital knowledge. However, there are several major problems exists, which hampers effective knowledge sharing.

Based on the SM activities in an SM organization and literatures and tools review, this paper presented the

technical perspective of KM in Software Maintenance area. This include the results of survey on knowledge required and usefulness of SM tools in an in-house application maintenance organization. Two major problems are identified; first, detail domain knowledge are often not available, and often, maintainers have to rely on experts and also codes to understand the details. Second, many tools are still not integrated to allow seamless knowledge combination, which hampers knowledge acquisition and sharing.

This paper represents a part of a longitudal study, with the aim to develop the framework, system architecture and tools to support KM in collaborative SM. Our next course of action is to conduct a survey on SM tools in a specific organization domain, to gain insight on the problems and barriers to adoption of SM tools in order to propose an appropriate framework for KM in collaborative SM.

REFERENCES

Anquetil n. and de Oliveira K.M., (2006). Software Maintenance Ontology, in Chapter 6 of *Ontologies for Software Engineering and Software Technology*, Springer Berlin Heidelberg.

April A. et al., (2005). Software Maintenance Maturity Model (SMmm): The Software Maintenance Process Model, *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 17 No. 3.

Das, S., Lutters, W., Seaman, C., (2007). Understanding Documentation Value in Software Maintenance, *Proceedings of the 2007 Symposium on Computer human interaction for the management of information technology*.

De Sousa K., Anquetil N., de Oliveira K., (2004). Learning Software Maintenance Organizations, *Workshop on Learning Software Organizations*, Banff, Canada.

Fjeldstad R. and Hamlen. W., (1998). Application Program Maintenance Study: Report to Our Respondents, 1979, in S.L. Pfleeger, *Software Engineering- Theory and Practices*, Prentice Hall.

Ghali N., (1994). *Managing Software Development Knowledge: A Conceptually-Oriented Software Engineering Environment*, PhD. Thesis, University of Ottawa, Canada.

Guide to the Software Engineering Body of Knowledge (SWEBOK), (2004). *The Institute of Electrical and Electronics Engineers*, Inc.:New York, NY.

IEEE 14764 (2006). Standard for Software Engineering - Software Life Cycle Process – Maintenance. *The Institute of Electrical and Electronics Engineers*, Inc.:New York, NY.

Jarke, M. and Rose, T., (1988). Managing Knowledge about Information System Evolution, Proceedings of the 1988 ACM SIGMOD International Conference.

Kitchenham B. et al., (1999). Towards an ontology of software maintenance, *Journal of Software Maintenance: Research and Practice*, Vol. 11 No.6.

KPMG *European KM Survey 2003*, retrieved from www.knowledgeboard.com on 11/1/2008.

KPMG *Knowledge Management Research Report 2000*) retrieved from www.knowledgeboard.com on 11/1/2008.

Lawton G., (2001) "Knowledge Management: Ready for Prime Time?," *IEEE Computer*, Vol.34, No. 2.

Lientz B.P. and Swanson E., (1981). “Problems in application software maintenance”. *Communications of the ACM*, Vol. 24 No.11.

Lientz, B. P., Swanson, E. B., and Tompkins, G. E., (1978). Characteristics of Application Software Maintenance. *Communications of the ACM*.

Lindvall M. et al., (2003). Software Systems Support for Knowledge Management, *Journal of Knowledge Management*, Vol.7 No.5.

Nonaka, I. & Takeuchi, H., (1995). *The Knowledge-Creating Company*. New York: Oxford University Press, Inc.

Pigoski T.M., (1997) *Practical Software Maintenance:Best Practices for Managing your Software Investment*, John Wiley & Sons.

Rama, J. and Bishop, J., (2006). A survey and Comparison of CSCW Groupware Applications, *Proceedings of the SAICSIT*.

Rodriquez O.M., et al., (2004) Understanding and Supporting Knowledge Flows in a Community of Software Developers, *Lecture Notes in Computer Science*, Vol.2198.

Rus, I., and Lindvall, M., (2001). Knowledge Management in Software Engineering, *IEEE Software*, Vol.19 No.3.

Santos G. et al., (2005). Knowledge Management in a Software Development Environment to Suport Software Process Deployment, *Lecture Notes in Computer Science*, Vol.3782.

Schach S. R., Jin B.O., Yu L., Heller G.Z. and Offutt J., (2003). Determining the Distribution of Maintenance Categories: Survey versus Measurement, *Journal of Empirical Software Engineering*, Vol 8.

Sengupta B., Chandra S., Sinha V., (2006). A Research Agenda for Distributed Software Development, *Communications of the ACM*.

Viscaino A. et al., (2003). Supporting Software Maintenance in Web Repository through a Multi-Agent System, *Lecture Notes in Computer Science* , Vol. 2663.

Wenger E, (1998). *Communities of Practice: Learning, Meaning, and Identity*, Cambridge University Press.

Wilson, T.D., (2002). The nonsense of knowledge management, *Journal of Information Research*, Vol.8 No.1, paper no. 144.