American Journal of Economics and Business Administration 3 (2): 347-351, 2011
ISSN 1945-5488

Agent Technology Application Strategies in Personal and Team Software Process Environment

Rusli Abdullah, Amir Mohamed Talib and Erizal Kamal Misran
Department of Information System, Faculty of Computer Science and Information Technology,
University Putra Malaysia, 43400, Serdang, Selangor, Malaysia

Abstract: Problem statement: Modern system development emphasize on the use of standard and procedures to ensure the system develop meets user needs and of high quality. Standards such as Capability Maturity Model Integration (CMMI), Software Process Improvement and Capability determination (SPICE) and ISO/IEC 90003:2004 are introduced. Personal Software Process (PSP) focusing on improving the individual engineer tasks in software development activities and can be extended to Team development (TSP) of large-scale software systems. **Approach:** To ensure the quality of system, defect must be managed or organized properly. Knowledge Management (KM) provides a solution to manage and promoting shared defect-information as knowledge. Agents technology offer a way to deal with complex systems that have multiple and distinct components and are often used as a metaphor for autonomous, intelligent entities. **Results:** Agent technology is also used to enhance a Knowledge Management System (KMS) based on software agent application strategies to increase the performance of knowledge sharing and its utilization. **Conclusion:** This study presents an agent-based application strategy in a KMS that manages defect-information as knowledge and shares it amongst software engineers who are majoring involved in PSP and TSP system development environment.

Key words: Agent technology, software process, personal and team software process, knowledge management

INTRODUCTION

In the early 1990s, (Humphrey, 1995) developed Personal Software Process (PSP) focusing on improving the individual engineer tasks in software development activities. PSP is further extended to team development of large-scale software system, known as Team Software Process (TSP). PSP is a structural framework of forms, guidelines and procedures for developing software, resulting in a collection of historical data. These historical data can be measured in terms of effort, size and defects, which later can be transformed into useful knowledge.

Knowledge has widely been acknowledged as one of the determining factors for corporate competitiveness and advantage (Nonaka and Takeuchi, 1995). In the past, we have seen the rapid increase of approaches to Knowledge Management (KM). KM technology has been recognized as an enabling and often even a leading, factor for connecting (e.g., people to other people or knowledge) and converting (e.g., data into knowledge). Software agents offer a way to deal with complex systems that have multiple and distinct components and are often used as a metaphor for autonomous, intelligent entities. Therefore, agents are perfectly qualified to model collaboration management systems (Wooldridge, 2009).

By adopting PSP in software development activities, software engineers will be able to improve their personal performance. For some software engineering project, teams are required and the effectiveness of the teams largely determines the quality of the software engineering. However, since PSP is applied on individual level, defect knowledge is only known to that individual. In other word, there is a lack of knowledge sharing regarding defect found during software development (Basili and Boehm, 2001).

Defect in PSP: PSP practices are designed to assist software engineers with quality and predictability issues. The quality component of the PSP strategy focuses on managing the defects in the software being produced. It is assumed that by improving defect

Corresponding Author: Rusli Abdullah, Department of Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400, Serdang, Selangor, Malaysia

management, engineers can produce more consistently reliable components of their software (Hirmanpour and Schofield, 2003).

Although some perceive PSP and TSP as a tedious and too much hassle, it provides benefits to software engineers. The most significant benefit of TSP is improved productivity and/or schedule reduction resulting from early defect removal and improved planning and tracking. Early defect removal:

- Reduces average time to remove a defect from hours to minutes
- Reduces testing costs/schedules by 20-48%
- Increases the non-test costs/schedules by 5-10%
- Yields net average savings of about 25%

Another benefit of PSP and TSP is it improved planning and tracking which increases task hours per week by up to 50%. Task hour increases translate directly to productivity increases (Basili and Boehm, 2001).

Knowledge Management System (KMS): Knowledge Management (KM) is a system for managing, gathering, organizing, refining, analyzing and disseminating of knowledge in all of its forms within an organization for certain purposes. Knowledge Management System (KMS) can be defined as an IT-based system developed to support/enhance the processes of knowledge creation, storage/retrieval, transfer and application (Alavi and Leidner, 2001).

Meso and Smith (2000) proposed a KMS architecture that combine all aspects as well as component, which consist of technology, functions and knowledge, in such it shows the connectivity between components of the KMS implementation shown in Fig. 1. In terms of technology, KM should have features such as computer-mediated collaboration, intelligent agents, search engine and web browsing. In terms of functions, KM involves the processes of acquiring, organizing, disseminating or sharing and using knowledge among users. Lastly, in terms of knowledge, KM should look at the know-how, know-what, know-why and so on.

Agent technology: Agent technology is a software programs or tool that could be developed and as strategies that used to facilitate and support the Community of Practice (CoP). It is one of the most promising technologies for dealing with distributed collaborative environment and social interaction in KM (Abdullah, 2008). Definition by Wooldridge and Jennings, 1995; as well as by Abood *et al.* (2008) "An agent is a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives".

Applications of agent technology to KM start from the realization that KM and software agent systems have several similarities. Agent-based technology and KM services include (Klusch, 1999; Muthuchelvi *et al*., 2009; Avakian *et al*., 2006):

- Search for, acquire, analyze, integrate and archive information from multiple heterogeneous sources
- Inform users when new information of special interest becomes available
- Negotiate for, purchase and receive information, goods or services
- Explain the relevance, quality and reliability of that information
- Learn, adapt and evolve to changing conditions

PSP and TSP defect management and KMS: Defects that an individual software engineer discovers during his Personal (PSP) development activities might be similar and important to his team member and vice versa. Thus defect information, along with the solution to the defect, should be shared amongst the software engineers. These defects knowledge can be stored in a data repository, where it can be accessed by other software engineers. Whenever software engineer who study in a Team (TSP) develop new software finds a defect, he can search the data repository for the similar defect. If similar defect is found, the knowledge regarding the defect will be display along with the solution. This will help the software engineer to reduce time to fix the defect found. Through the sharing of these defects knowledge, software engineer can improve not just the productivity of study, but also the quality of products (Basili and Boehm, 2001).

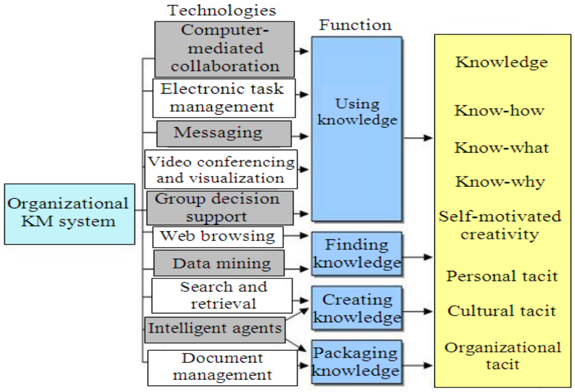

Fig. 1: KMS Architecture (Meso and Smith, 2000)

The use of agents in KM in managing defect can be seen in two perspectives. Firstly, agents are often used to model the organizational environment where the KM system will operate and on the other hand, software agents can be used to implement the services, or functionalities, of KM systems. Secondly, multi-agent models are used as the virtual counterpart of real-life societies and organizations, which facilitates the design process since it reduces the conceptual distance between the system and the real-world application it has to model (Sushil *et al*., 2008; Zhang *et al*., 2008).

MATERIALS AND METHODS

Respondents for this research were selected from the 12 officers at Malaysian Qualification Agency (MQA) Information Technology Department, who are involved in software development activities. Respondents were involved during preliminary analysis as well as during system evaluation process. Two sets of questionnaires were used in this research. The first questionnaire, which is the preliminary survey, was used to gather information regarding knowledge on defect management and KMS. The second questionnaire was used to evaluate the defect KMS based on functionality and usability factors.

In performing this research, we use the following phase shown in Fig. 2:

Phase 1: Analysis of Literature Review-the analysis and reviews on defect management in PSP and TSP, KMS and agent technology were carried out and discussed.

Phase 2: Analysis of existing KMS methods-study on how defects knowledge was managed in software development. Facts gathering were also performed at this stage to determine the preliminary condition of respondents.

Phase 3: Proposed defect KMS-a model for defect KMS using PSP approach is proposed, incorporating agent technology into the system.

Phase 4: Defect KMS implementation-the system is realized based on preliminary survey.

Phase 5: Defect KMS testing-to test the system so that it is defect free and test the functionalities of the software agent in this system.

Phase 6: Evaluations, analysis and results-system is evaluated based on functionality and usability. Data gathered from the questionnaire would then be analyzed and the results would determine whether the objectives of this research have been achieved.

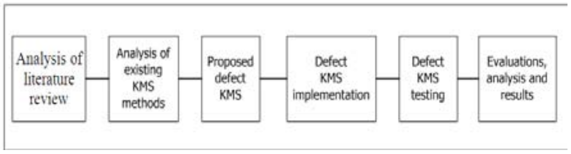
Analysis of literature review
Analysis of existing KMS methods
Proposed defect KMS
Defect KMS implementation
Defect KMS testing
Evaluations, analysis and results

Fig. 2: Research approach

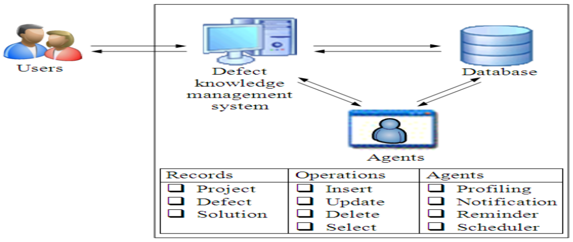

Fig. 3: The proposed architecture of defect KMS

Table 1: Software agents and their task

No.	Name	Task description
1.	Profiling agent	Record knowledge according to its categories (e.g., defect type) Keep track of which defect knowledge is relevance (most common error) and which is popular (most accessed)
2.	Notification agent	Notify system user regarding new knowledge added into the system
3.	Reminder agent	Remind system user regarding defect that has not yet been provided with the solution
4.	Scheduling agent	Determine the number of project currently in hand and disable the developer option to accept new project if more than 3 projects

Defect knowledge base proposed: Defect information will be gathered in a data repository, which can be shared and access by authorized users who are also involved in software development (Fig. 3). The proposed system will be accessible via intranet. Agent-based KMS is ideal for this purpose where KM allows the managing, gathering, organizing, refining, analyzing and disseminating of knowledge in all kind of forms within an organization for certain purposes (Davenport and Prusak, 2000).

There are four software agents applied to this system using simple reflex agent, which uses condition-action rules. Table 1 describes the agents and its task.

RESULTS

Based on preliminary survey that has been conducted, we have found that only 25% of the respondents are aware of KMS existence as shown in Fig. 4.

Another finding is that only 25% of the respondents records defect they found during their software development activities shown in Fig. 5. Furthermore, to solve the defect problem faced, respondents opted for the following sources of solution, which are colleagues, internet, books and online forums as supported by Fig. 6.

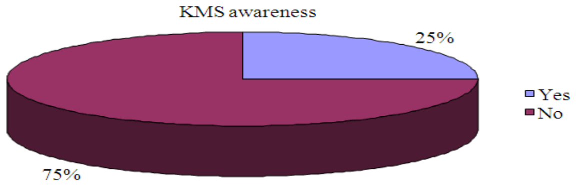

Fig. 4: KMS awareness in MQA IT Dept

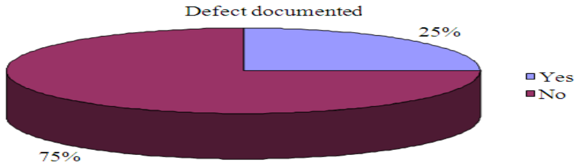

Fig. 5: Respondent document defect found during software development engineers by the Notification

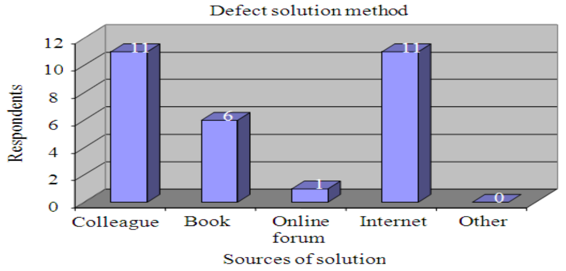

Fig. 6: Sources of knowledge for solving defects

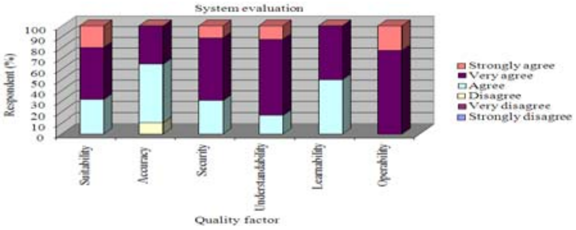

Fig. 7: Evaluation of system based on sub-characteristic of functionality and usability factors

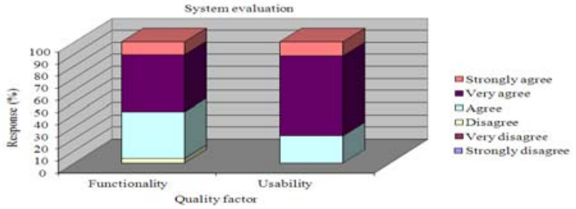

Fig. 8: Evaluation of system based on functionality and usability factors

Besides that, we did perform evaluation towards the system based on the functionality (suitability, accuracy and security) and usability (understandability, learn ability and operability) factors. Respondents were given a period of time to use the system and at the end, they are required to complete a questionnaire. The results are shown in Fig. 7 and 8. 10% of the respondents rated accuracy as being poor due to system unable to provide with sufficient information when they do defect searching. This is mainly due to little knowledge managed to be gathered in the given period of time. Should the time be extended, this problem may not arise.

However, based on the architecture of the system, respondents gave feedback that the information inserted was immediately disseminated to other software Agents. New knowledge were categorized according to defect types used in PSP by the Profiling Agent, which also determined defect knowledge that commonly occurs during software development process. Software engineers were also reminded by the system once they login into the system regarding the defects that has not been solved yet by the Reminder Agent. Once this knowledge is updated, the system will send notification on new defect knowledge insertion. Software engineers were also well balanced in terms of workload, due to the management of project distribution by Scheduler Agent. Overall, the agents applied into the defect KMS have effectively encourage software engineers to share defect knowledge among them, thus help improves software development process in terms of fast defect solution and reducing defect earlier in the development process.

DISCUSSION

Based on the above survey analysis, it is considered that the most relevant factors for the successful implementation of momentum for the KM initiatives and agent technology are:

- Knowledge sharing needs to be nurtured enabled within and aligned with organizational objectives. The underlying concern is employees do not want to share knowledge using agent-based strategy of KMS based on PSP and TSP practices. Successful organizations empower employees to want to share and contribute intellectual knowledge, by rewarding them for such actions. And with organizational leaders role models of knowledge sharing and interface regularly with staff, teams and stakeholders in review sessions and openly talk about successes and failures
- The first important variable is leadership with a vision, strategy and ability to promote change of the management to a compelling KM based on PSP and TSP practices actively promoted by agent technology that clearly articulates how KM contributes to achieving organizational objectives

- The interview guide was used during all the interviews. However, separate interviews were used. Before conducting the interviews, the agent technologies were contacted to assign the responsible or grant permission to contact the recognized expert in the concerned area or field of activity for the topic in software development activities and described the agent technology and how the demonstrated in the system and how the agents called the other agents and the ability of agents to passed their parameters

CONCLUSION

In this work, we have analyzed and proposed a defect agent-based strategy of KMS based on PSP and TSP practices. By proposing this defect KMS, we are able to promote knowledge sharing in software development process. With the ability to widely share this knowledge, software engineers are able to reduce defects, improve product quality and development time, as well as having a more proper way to manage defects knowledge.

For future study, we could include effort and size measure (e.g., effort measure-man-hour of each software engineer per project. We could also implement push technology such as short messaging services (sms) and electronic-mail for better knowledge dissemination.

ACKNOWLEDGEMENT

The researchers would like to thank the anonymous reviewers of American Journal of Applied Sciences for their valuable remarks and comments.

REFERENCES

Abdullah, R., 2008. Knowledge Management System in a Collaborative Environment. 1st Edn., Penerbit Universiti Putra Malaysia, Malaysia, ISBN: 10: 9675026243, pp: 94.

Abood, A.A., A.N. Abdalla and S.K. Avakian, 2008. The application of multi-agent technology on transient stability assessment of Iraqi super grid network. Am. J. Applied Sci., 5: 1494-1498. DOI: 10.3844/ajassp.2008.1494.1498

Alavi, M. and D.E. Leidner, 2001. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. MIS Quart., 25: 107-136.

Avakian, S.K., A. Anwar and A.N. Abdalla, 2006. The application of agent technology on the power flow of Iraqi supper grid network. Am. J. Applied Sci., 3: 2174-2177. DOI: 10.3844/ajassp.2006.2174.2177

Basili, V. and B. Boehm, 2001. CeBASE software defect reduction top 10-List. IEEE Comput., 34: 135-137.

Davenport, T.H. and L. Prusak, 2000. Working Knowledge: How Organizations Manage what they Know. 1st Edn., Harvard Business Press, USA., ISBN: 10: 1578513014, pp: 199.

Hirmanpour, I. and J. Schofield, 2003. Defect management through the personal software process. http://www.crosstalkonline.org/storage/issue-archives/2003/200309/200309-Hirmanpour.pdf

Humphrey, W.S., 1995. Introducing the personal software process. Annals Software Eng. 1: 311-325.

Klusch, M., 1999. Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet. 1st Edn., Springer-Verlag New York, Inc. Secaucus, NJ, USA., ISBN: 3540651128, pp: 498.

Meso, P. and R. Smith, 2000. A resources-based view of organizational knowledge management systems. J. Knowl. Manage., 4: 224-234. DOI: 10.1108/13673270010350020

Muthuchelvi, P., G.S.A. Mala and P. Ramachandran, 2009. Agent based grid resource discovery with negotiated alternate solution and non-functional requirement preferences. J. Comput. Sci., 5: 191-198. DOI: 10.3844/jcssp.2009.191.198

Nonaka, I. and H. Takeuchi, 1995. The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation. 1st Edn., Oxford University Press, USA., ISBN-10: 9780195092691, pp: 304.

Sushil, R., K. Garg and R. Bhargava, 2008. Design, validation, simulation and parametric evaluation of a novel protocol for locating mobile agents in multiregion environment. J. Comput. Sci., 4: 256-271. DOI: 10.3844/jcssp.2008.256.271

Wooldridge, M., 2009. An Introduction to Multiagent Systems. 1st Edn., Wiley, ISBN: 0470519460, pp: 405.

Wooldridge, M.J. and N.R. Jennings. 1995. Intelligent agents: Theory and practice. Knowl. Eng. Rev., 10: 115-152. DOI: 10.1017/S0269888900008122

Zhang, C., D.Tang, Y. Liu and J. You, 2008. A multi-agent architecture for knowledge management system. Proceedings of the Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference, Oct. 18-20, Jinan Shandong, pp: 433-437. DOI: 10.1109/FSKD.2008.361