Business Process Modeling for Software Acquisition

- A Literature Review

Hussain Alfaraj
School of Computer Science, Engineering and Mathematics
Flinders University
Adelaide, SA 5001, Australia
Email: hussain.alfaraj@csem.flinders.edu.au

Shaowen Qin
School of Computer Science, Engineering and Mathematics
Flinders University
Adelaide, SA 5001, Australia
Email: shaowen.qin@flinders.edu.au

***Abstract* — Business Process Modeling can be used as an effective tool to understand requirements of software acquisition. This paper examines the current literature on the practices and their associated benefits and challenges in making software acquisition decisions with business process modeling, and demonstrates the need for further research in this area.**

Keywords - business process modeling; software acquisition; software choice; outsourcing.

I. INTRODUCTION

Organizations that are considering software acquisition have three main choices: commercial off-the-shelf (COTS) software, in-house development of the software they need, or contracting or outsourcing the development of the software from another company. All three choices have advantages and disadvantages, and are appropriate for some applications. However, before determining the appropriate sourcing choice for software, the requirements for the software must be determined. Business process modeling (BPM) is one way in which these software requirements may be determined; by examining the environment into which the software will be integrated and what it is expected to accomplish by the software, it is possible for the requirements determination of the software to be completed. There are a number of techniques that may be used to accomplish this task. This paper explores the issue of software choice and the application of BPM specifically to the outsourcing path of software acquisition, and discusses further research that could be conducted in this area.

II. SOFTWARE CHOICE

Software choice is simply the process of determining requirements for a given piece of software and making the appropriate selection among COTS or custom software. Software requirements fall into both functional (performance and task oriented) categories and nonfunctional (cost, ease of use and administration, etc) categories. Although this is an activity that is engaged in almost routinely by businesses, there is relatively little recent research on the process of software choice. One of the most commonly seen researches on software choice is the enterprise resource planning (ERP) system. These software systems are large, complex, and expensive, and also require a substantial amount of organizational change to implement, making them an ideal candidate for the use of BPM and cost modeling during the selection process [1]. Reference [1] identified several ERP process models that were used during the selection of an ERP system, including a number of themes that were different between processes at different organizations. The author's research indicated that the process of selection and implementation was highly individualized between organizations. Examining specifically from the inter-organizational perspective, it led to conclusions about the use of ERP systems between organizations that may call for further study, particularly in regard to outsourcing of business processes such as information technology and software choice. While ERP systems provide a good example in which to examine this issue, there exists room for research in many other areas of software choice.

While open source software (OSS) was the focus of the quality model described in [2], this model may still be applied to custom software for acceptance testing, in the case where access to the source code is a condition of the outsourcing arrangement. However, it is not as readily applied to COTS software. This model examined the characteristics of software by examining characteristics, specifically four major characteristics and ten sub-characteristics of the potential software choice. It determined that functionality, usability, portability and reusability were the major quality characteristics of open-source software, with ten subcomponents (suitability, security, understandability, operability, learnability, adaptability, installability, functional commonality, co-existence, and lawfulness) also being examined. In the case of custom software, lawfulness is probably not going to be a marker of quality; but the nine remaining sub-components identified for OSS are likely to be relevant. Nonetheless, a software quality model that is directly applicable to custom software contracted from an outsourcer would still be more appropriate and preferred in this case, as it would allow for consideration of contractor-specific issues such as code complexity and maintainability.

Requirements determination is one of the most important parts of software choice. There are a number of techniques that may be used to determine the requirements for software. One such process is BPM, in which the business processes that the software will be integrated into are modeled and examined in order to extract requirements. Business process simulation (BPS), as described by [3] may also be used to determine software requirements. Authors in [3] went a step further than BPM, using a simulation technique to determine the requirements for a given manufacturing system. This

978-1-4244-2108-4/08/$25.00

simulation examined the entire process of manufacturing a given element, allowing the authors to build business process models and determine requirements before the manufacturing system was actually in place. This could prove invaluable for examination of a complex system or a system that has not yet been implemented. Although the process described by [3] was simulation of a physical manufacturing process, it could easily be applied to modeling of any business process, including entirely software-based models. This technique should be explored more thoroughly in terms of software choice.

Another means of testing the quality of software, or of determining some or all of the initial requirements, is to model the standards compliance of the software. This may be particularly useful when assessing initial requirements or when the software requirements were incomplete or there is no evidence of how they would have been met in the development process. Reference [4] discussed the use of standards as a basis for modeling quality in software, an approach that can be used for either COTS or custom software provided from any source. The authors in [4] specifically examined the issue in the context of teaching business process modeling to students in a university environment, using the SEI CMMI-SW standard to demonstrate the BPM construct and process. While this was examined specifically in the context of teaching BPM, it could be extended outward to the case in which the full requirements of a given system are no longer available. It may also be useful for the organization conducting the requirements gathering process to use quality standards as an input to the initial BPM process, particularly if it is the case that the organization requires adherence to these standards.

III. Outsourcing

One choice for software acquisition is contracting the development of custom software through an outsourcing arrangement. Outsourcing has been shown to be a popular choice for companies in high human resource cost regions, including the United States, Europe and Great Britain, who often experience great cost savings by engaging in outsourcing their software development rather than in-house development [5]. The issue of software outsourcing has been examined from a supply chain perspective [5]; however, an examination of this issue from a business process modeling perspective, in which the organization's business processes and those of the outsourcing firm's are compared, contrasted and integrated, may provide some useful information.

Outsourcing may be a good choice for custom development, particularly if an organization lacks the resources to perform its own software development and testing; however, it is not without its challenges and may not be appropriate in all cases. Reference [6] discussed some of the challenges involved in the requirements determination phase of a direct client-vendor outsourcing project, as undertaken by many firms outsourcing software development. Process-specific difficulties discussed included: conflicting client-vendor goals, including scheduling and priority setting; low client involvement in the development process; conflicting requirements engineering processes; and misalignment of client commitment with project goals. Other, more general difficulties with outsourcing were discussed, such as communications difficulties, disavowing of responsibility for design decisions or other issues; disagreements in tool selection; and expectations differentials [6]. Outsourcing may also lead to more complex software as compared to in-house development. An examination of outsourced and in-house development of two branches of the same software package was conducted by [7]. The authors determined that the branch of code provided by the outsourcing firm contained a larger number of lines of code and greater complexity skewness; whilst the branch of code produced in-house contained less number of lines of code and was less complex. However, this may not be an issue if the firm in question will not be integrating or maintaining the outsourced code.

In order to ease the difficulty involved with requirements setting in the outsourcing environment, the organization could use a model such as the one proposed in [8] to pre-select their requirements. The model uses a risk-based requirements model to determine the earliest requirements of the project, and then allows for change during further evaluation and implementation. The approach used a defect detection and prevention (DDP) approach, which is derived from a hardware manufacturing discipline, but may also be applied to software development or other production tasks [8]. In particular, this approach heads off expectation/reality mismatches like those discussed above, including mismatches in capabilities and requirements, mismatches in stakeholder perceptions regarding priorities and comparative value of a given task or requirement, and a consideration of other potential roadblocks to success.

IV. Areas for Further Research

There is ample room for further research within the existing field of knowledge regarding software acquisition, particularly in utilizing BPM and cost modeling as tools. One such area is the cost modeling of software outsourcing as compared to in-house development, as an extension of the work conducted in [5] on supply chain analysis. A second area of further research might include the modeling of an outsourcing firm's processes as compared to in-house development, in order to determine the reasons behind the increase in complexity within the code observed by [7]. While much attention has been paid to the choice of ERP software, other seemingly more mundane software choices have been neglected. For example, determining the differences in business processes which lead to the choice of COTS software as compared to those which lead to the contracting of customized software could lead to some interesting information regarding the requirements definition process and the relative value of software processes. Another area of improved research could be the integrated BPM of client organizations and outsourcing firms. This examination could lead to some insights into the process of outsourcing, including information on why difficulties between outsourcers and clients as described above occur and how they may be resolved from a business process perspective.

V. Conclusion

The issue of software evaluation and choice has not been extensively studied from a modern BPM perspective, and many aspects of the process have changed since the first frameworks for software choice were constructed in the 1990s. With the

rapidly increased use of outsourcing and "semi-custom" software, it is vital that further research be conducted into determining the process of software choice and establishing new frameworks for understanding the current business process. The tools of business process modeling and cost modeling offer the perfect fit for determining the reasons behind software choice and providing a framework or blueprint for firms engaging in these choices in future.

REFERENCES

[1] V. Vathanophas, "Business process approach towards an inter-organizational enterprise system," Business Process Management Journal, vol. 13, no. 3, 2007, pp. 433-450.

[2] W. J. Sung, J. H. Kim, and S. Y. Rhew, "A quality model for open source software selection," IEEE, Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information Technology, 2007, pp. 515- 519.

[3] A. Azadeh, M. Haghnevis, and Y. Khodadadegan, "Design of the integrated information system, business and production process by simulation," Journal of the Amercian Society for Information Science and Technology, vol. 59, no. 2, 2008, pp. 216-234.

[4] T. Rozman, R. V. Horvat and I. Rozman, "Modeling the standard compliant software processes in the university environment," Business Process Management Journal, vol. 14, no. 1, 2008, pp. 53-64.

[5] W. Gan and X. Gan, "Empirical analysis on supply chain of offshore software outsourcing from China perspective," IEEE, Proceedings of the 2007 IEEE International Conference on Service Operations, Logistics and Informatics (SOLI2007), 2007, pp. 1-4.

[6] J. M. Bhat, M. Gupta, and S. N. Murthy, "Overcoming requirements engineering challenges: lessons from offshore outsourcing," IEEE Software, September/October 2006, pp. 38-44.

[7] A. Capiluppi, J. Millen, and C. Boldyreff, "How outsourcing affects the quality of mission critical software," IEEE, Proceedings of the 13th Working Conference on Reverse Engineering (WCRE), 2006, pp. 1-3.

[8] M. S. Feather, S. L. Cornford, K. A. Hicks, J. D. Kiper, and T. Menzies, "A broad, quantitative model for making early requirements decisions," IEEE Software, 2008, pp. 49-56.