A Case Study on Logistics Software Development

Sakgasit Ramingwong[*1], Narissara Eiamkanitchat[1], Lachana Ramingwong[1], Sakgasem Ramingwong[2]

[1]Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand

[2]Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand

[*1]sakgasit@eng.cmu.ac.th; [1]narisara@eng.cmu.ac.th; [1]lachana@gmail.com; [2]sakgasem@eng.cmu.ac.th

Abstract-**Due to many of their unique characteristics, software development projects have inferior rate of success. Developing, customizing and installing nine types of business software for twenty different entrepreneurs in a few months would be deemed as a mission impossible for many project managers. This paper describes a program which involves a development and customization of fourteen logistics software within less than three months. Twenty two entrepreneurs as well as six software houses were included in the program. Although a number of serious challenges surfaced during the development, the program ultimately succeeded. Key findings and success factors from this case study are discussed in this paper.**

Keyword- Information system development; Software; Logistics; Project management; Case study

I INTRODUCTION

Logistics has become a vital element for modern businesses. This essential discipline involves several critical activities such as transportation management, inventory management, warehouse management, production management and forecasting. Efficient management of logistics systems could yield a great competitive edge over competitors.

Realizing the importance of logistics, the Office of Strategy Management, Upper North Cluster 1 (OSM-1), Thailand, initiated a program called Development of Logistics Management System for Greater Mekong Subregion Cluster (LS) which aims to alert local entrepreneurs on this issue as well as increases their competitiveness. This program included a series of projects such as arranging workshops on logistics concepts, building a community of logistics entrepreneurs, and constructing logistics software. Development of logistics software and initiate projects for local software houses were also important objectives of this agenda.

OSM-1 is a governmental unit which patronizes Chiang Mai, Lampang, Lampoon and Mae Hong Son, four of the nine northern provinces of Thailand. Main incomes of these provinces are generated from agriculture, food, production, service, and tourism industries. Recently, the balance of the current local business is being agitated by the forming of ASEAN Economic Community (AEC), foreign investments and global economic crisis. This forces local enterprises to evolve their overall competitiveness in order to survive.

Interestingly, although the use of computer and enterprise software are not new in international businesses, this technology utilization is still very limited for the local entrepreneurs. During the requirement elicitation of this project, it was found that even several of the top wholesalers in the region were still operated manually while many others only use computer for a very limited functionalities such as simple word processing and spreadsheet. This left a large opportunity for improvement in order to equip and prepare them before the AEC age.

This paper describes the development of the LS program as well as its key findings. This project involves developing, customizing and installing nine types of business software for twenty different entrepreneurs within three months. This is an extremely challenging situation for the management team, especially in dealing the unique nature of software projects.

Software projects are well-known for their difficulties. Their abstract nature and dependency on human resource distinguish them from other forms of engineering projects [1]. Success rate of software projects were reported to be at approximately 30% [2]. A number of risks such as continuous change of requirements and lack of stakeholder commitment have potential to ruin an entire project [3]. Despite of these foreseeable risks, the LS program turned out to be interestingly successful.

The second section of this paper gives an overview of setting and development of the LS program. Then, problems, solutions and key findings are described in the third section. Finally, the fourth section concludes the paper.

II A LOGISTICS SOFTWARE DEVELOPMENT PROJECT

Aiming to increase sustainable competitiveness for local business, one major agenda of the LS is development of pilot logistics software for a pilot group of local entrepreneurs. In order to ensure flexibility and availability for future distribution, all software are required to be complied with open source licenses. A team of experts in logistics and information system from local universities such as Chiang Mai University (CMU) and North Chiang Mai University (NCU) were assigned as consultants for this program. These experts are specialized in various critical fields, such as logistic activities management, warehouse management, software development, project management, and database management. Due to some serious time constraints, all software must be completed within three months. A limited budget was also assigned for each software.

A. Software Description

Apart from the need of open source licensing, OSM-1 provided a considerable flexibility to the software requirements. The software must be related to competitive logistics activities. This included transportation management, inventory management, warehouse management, customer relationship management, human resource management, accounting, marketing, forecasting and many other critical business activities, depending on requirements from business sectors [4]. The development processes were to be closely supervised by the consultant team. The software houses also needed to provide appropriate documentations, trainings as well as six months of support to the pilot entrepreneurs. After the termination of this LS program, other local businesses can contact OSM-1 or the software houses if they wanted to adopt or further customize the software.

Participation in this program is free of charges. OSM-1 was responsible to all expenditure, including development and consulting costs.

B. *Selection of Stakeholders*

There are thousands of local businesses under supervision of OSM-1, ranging from very small entities to large-scale production. In order to select the most appropriate ones for this pilot program, a series of workshops was organized. Entrepreneurs who were interested in the program were required to show their commitment by having executives or high-level managers participated in the workshops. During the workshops, the participants were trained on fundamentals of logistics and its importance as well as possible application of technologies on logistics. They were also informed that they can further participate in the LS program without any costs. Then, at the end of the workshops, the ventures that were willing to join the LS program were asked to construct mind maps which described their own business and potential areas of improvement. This was done under close supervision of the experts.

Approximately two hundreds entrepreneurs joined the program. Sixty of them expressed interests in the LS program and participated in the mind mapping workshops. Ultimately, twenty two highest ranked participants were selected to be pilot businesses. Five more organizations which scored slightly lower than them were also kept as reserves as potential replacements. The selection was based on the applicants' ranks of determination, potential of infrastructure, business needs and potential success deemed by the experts. Most of these selected businesses indicated that they need software for inventory management, warehouse management and transportation management. These subjects were considered as initial business requirements for further selection of software developers. Figure 1 illustrates these initial business requirements.

There are more than two hundred software houses in OSM-1 cluster. Another two-day seminar was arranged in order to select potential software developers for the program. During this seminar, the objectives, initial requirements, timeframe and budget of the program were disclosed to these developers. Twenty software houses participated and presented their proposals in this seminar. Six of them were chosen based on their determination, experiences, technology and suitability of proposal regarding initial business requirements. At this stage, several software houses withdrew their proposals due to the policy on open source license, time constraints and limited budgeting.

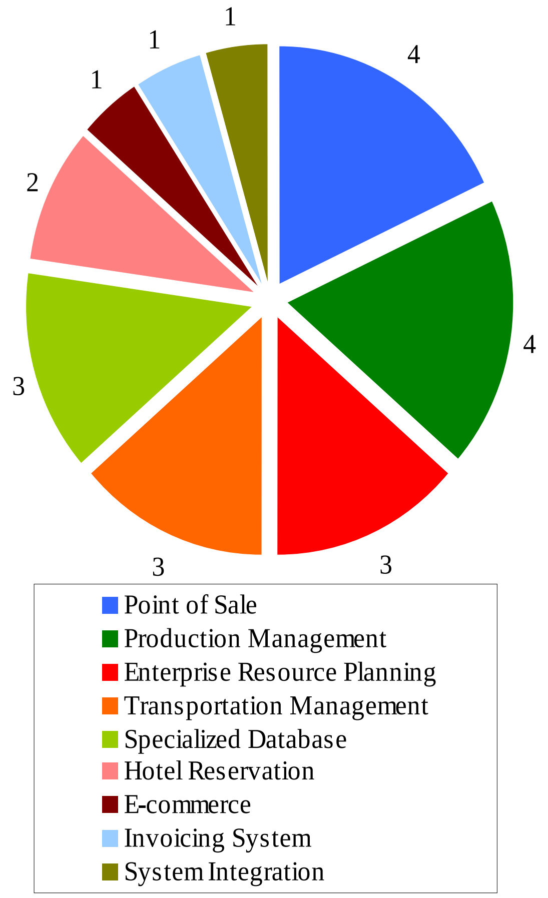

Fig.1. Initial requirements after mind mapping workshop

C. *Business Matching*

After the selection of potential stakeholders, a final workshop was organized. The main objective of this workshop was to match entrepreneurs to appropriate software houses. Executive representatives from each business were given approximately one hour to discuss their in-depth requirements with a software house which was pre-arranged by the experts. At least one consultant from the expert team was assigned to moderate the conversation. Their primary role was to ensure feasibility and fairness of each project. Interestingly, most of the entrepreneurs began to either change or increase their requirements. Yet, most of the matching succeeded smoothly after a degree of negotiations. For instance, two hotels which proposed a hotel reservation system enhanced their scope to include several other tourism aspects. Entrepreneurs who demanded enterprise resource planning (ERP) system were advised to reduce their scope to only inventory management. All of there alterations were based on appropriateness of time, budget, expertise of software vendors and overall feasibility.

On the other hand, several entrepreneurs made major unexpected changes to their requirements, resulting in a few consequential switches of business matching. For instance, an organization which indicated needs on system integration changed their requirement to a customer relationship management system. Additionally, two entrepreneurs greatly increased their scope from basic point-of-sale to complete inventory management systems. These basic requirements are exhibited in Figure 2.

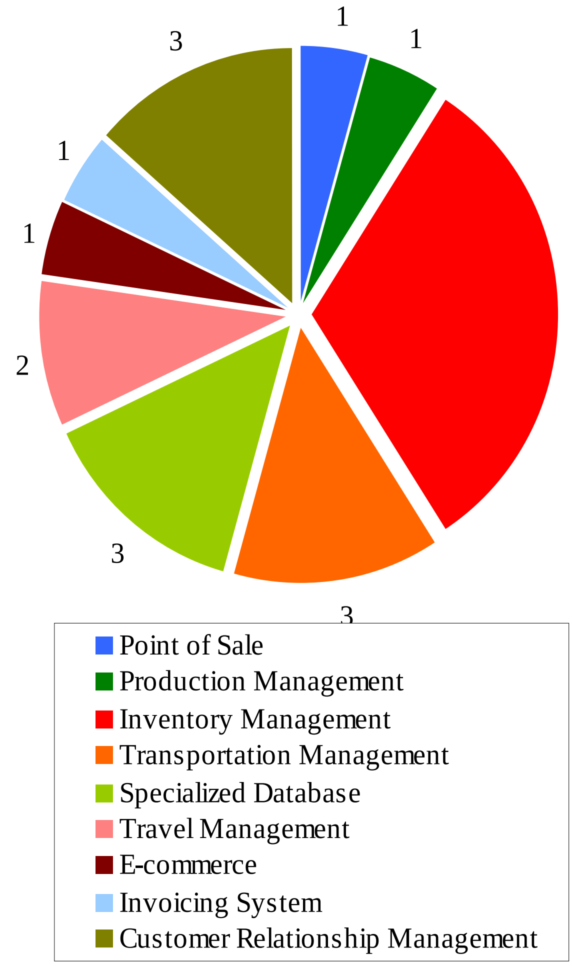

Fig. 2 Basic requirements after business matching session

Regardless of minor or major changes, these basic requirements were finally agreed. The stakeholders then made appointments with the software developers for an on-site in-depth requirement analysis.

D. Contract Signing

Immediately after the matching session, a contract was delivered to each entrepreneur. This contract was signed separately from requirement documents. Although there were absolutely no costs for participating in the program, a clause which aimed to prevent potential lack of coordination was included in the contract. According to this clause, if the entrepreneurs cancelled the project before completion, they were required to pay actual costs of software development and consultancy. Furthermore, the contract also highlighted that the finished software would hold an open source license.

Unexpectedly, the project cancellation clause was found to be extremely irritating for a few entrepreneurs. Several businesses negotiated for lower potential compensation. Eventually, after being convinced by the experts, most of them concurred to the original clause. Yet, one of them ultimately cancelled their project.

Another participant surprisingly withdrew from the program due to the open source policy. They stated that their information structures and working processes were highly confidential. Thus, if the finished software would be distributed freely, their competitors might be able to imitate their invaluable knowledge. A few other organizations also expressed uneasiness on this option. Yet, after being convinced that only the main engine of the software would be distributed while their business processes and specific customization would not, most stakeholders finally agreed to sign the contracts.

Two new participants were pushed from a list of reserved organizations in order to replace the withdrawals. The final sets of basic requirement are displayed in Figure 3. Figure 4 illustrate a comparison between requirements in each stage.

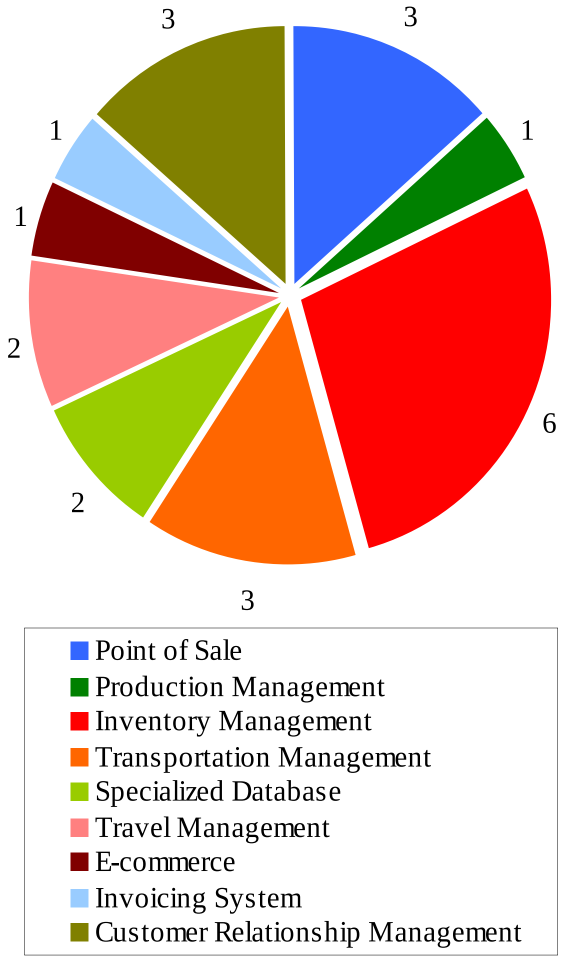

Fig. 3 Basic requirements after contract signing

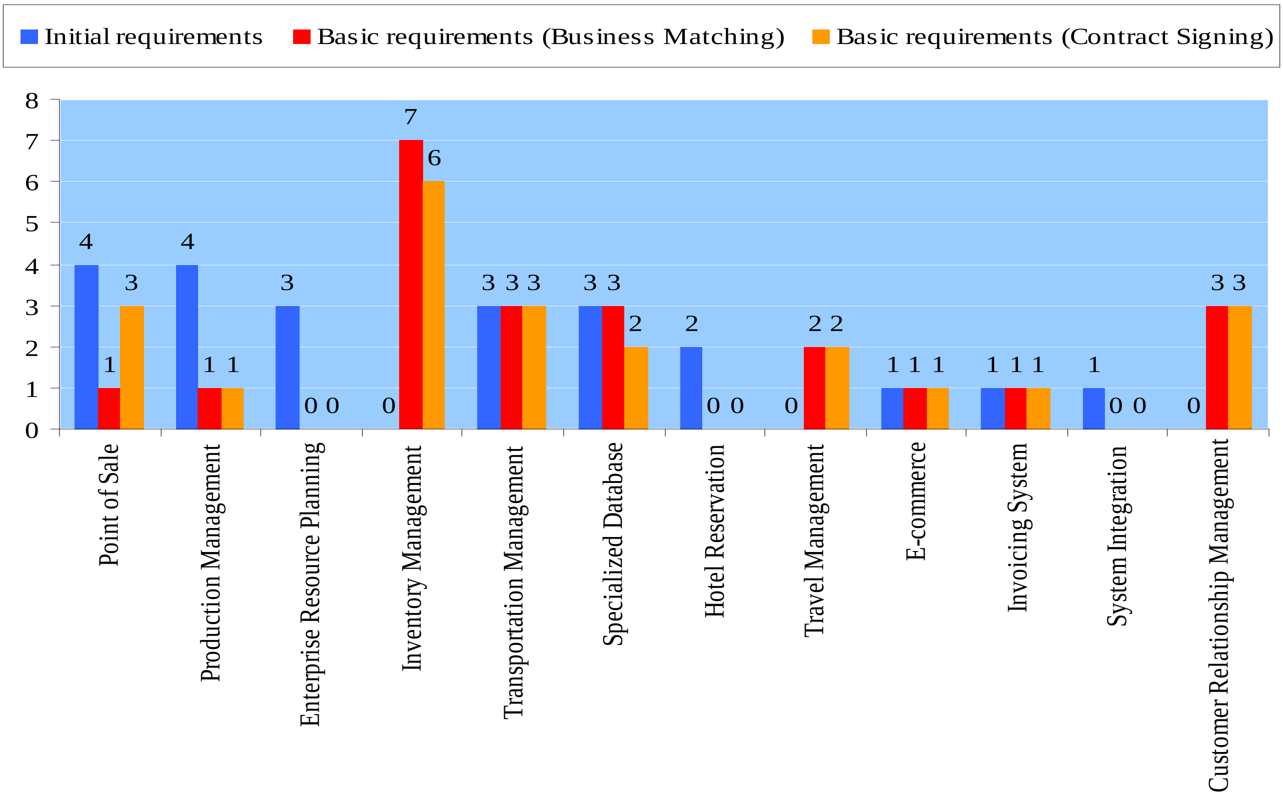

Fig. 4. Comparison of initial requirements (mind mapping), basic requirements (business matching) and basic requirements (contract signing).

E. Software Development

As aforementioned, due to time constraints, only three months were given for the software development process. Depending on their development strategies, the software houses and the entrepreneurs were encouraged to make appointments as appropriate.

During this development period, the team of experts was also assigned to supervise all projects. At minimum, each project was to be visited by one software development consultant and another logistics system expert for a total of six man-days. Generally, this implied three times of half man-day visits with the appearance of both mentors. The first visits usually happened during the on-site requirement engineering on the first weeks of the project. Then, the second appointments took place roughly during development stage on the middle of the second month. Subsequently, the final visits involved the final software inspection or the acceptance test on the last weeks of the project. Additional expert visits were to be organized if needed.

Ten projects involves moderate to heavy customization of existing open source software such as OFBiz and Openbravo. In contrast, the rest of the projects involved freshly development of software which were specifically tailored to entrepreneur's needs.

Many challenges surfaced during the software development. Unsurprisingly, scope creeps and change of requirements occurred in almost all of the projects. A few projects suffered from critical problems such as incompatibility of the new software and the legacy system, lack of user experiences and lack of involvement from customer executives.

The first two weeks of the program were spent on hierarchical process of contract signing. During this period, several software houses risked wasting their efforts by immediately starting their development process while the others waited for the completion of the contract. After two months of development, most of the project progressed smoothly. However, a serious delay occurred to one project, signaling that finishing it in time would be highly unlikely. A major conflict between the software house and the entrepreneur was reported. After the program executives were alerted by this problem, the responsible software house was replaced. Additionally, a new aggressive policy was issued in order to ensure that other software would finish within the deadline. According to this new policy, the acceptance testing of all projects was hastened up by two weeks. This stirred noticeable unrests amongst the software houses.

Fortunately, with all the hard works and cooperation between the entrepreneurs, the software houses and the experts, the acceptance process of twenty-one projects completed successfully within the new deadline. Furthermore, although it took much longer than the others, the troubled project was also finished.

Benefits from this program included improved efficiency of logistics activities, new logistics functions, reduced working time, reduced expenditures, decreased losses, expanded market reaches, and systematized databases and reports. Several software were expected to yield more than three to five times of their budget.

III PROBLEMS AND LESSON LEARNT FROM THE CASE STUDY

From the program initiation to completion, a number of problems were encountered. Most of these challenges involve management issues. Only a few threats were related to technical aspects. This section highlights crucial problems, their mitigation strategies as well as learnt lessons from this program.

A. Threats of open source policy

The agenda of open source became both benefits and risks in this project. Although open source software could be widely and freely distributed, several businesses exhibited their reluctance on their contract signing. They indicated that this could cause leaks of knowledge can lead to further problems on their competitiveness. On the other hand, several developers terminated their participation in the program due to this policy. They suggested that the allocated budget was insufficient for developing an open source software. They also indicated that they would consider rejoining the program if this policy changed and the software were to be licensed as proprietary.

In order to reduce this risk on the business side, the consultant team approached each troubled entrepreneur. The consultants convinced these entrepreneurs that all of their confidential information, especially business processes and specific customizations, would not be included in the open source code. Only general modules such as general ledgers and inventory database would be released. Fortunately, with this resolution, most of the anxious participants were satisfied and decided to retain their position in the program. Nevertheless, despite of this convincing attempt, one entrepreneur eventually withdrew from the program. This was due to the uniqueness of their proposed research-oriented inventory management system which might be difficult to be released without revealing critical knowledge.

For the developers, the consultant team approached them by inducing that although the source would be available to freely adopt, they still had opportunities to extend their profits. Obviously, the developers could gain income from customization and technical supports for adopters. Furthermore, participating in this program increased their chance of advertising and strengthening their governmental portfolio. Unfortunately, these reasons were not convincingly enough for several providers. On the other hand, other software houses, including some highly experienced as well as promising ones, still expressed their enthusiasm to the program and eventually signed the contract.

B. Limitation of resources

Budget and time limitations were major obstacles on this program. The allocated budget of each software was limited and not negotiable. For most of the software, this amount of budget was just sufficient. However, for systems which involve sophisticate workflows such as inventory management and production management, the budget was significantly low. As a result, instead of developing an entirely new system, the software houses chose to adopt and customize available open source systems.

Time management was even more problematic to the program more than budget management. Developing a software within three months without completed requirements was arguably one of the worst case scenario imaginable. Furthermore, due to the delay in contracting process during the first weeks and the urgent contingency policy on the last month, the actual development time of each software was shortened to a mere two months. Fortunately, many of the software houses were implementing agile philosophy. Agile's change-oriented nature positively facilitated fast-pace development [5]. This effectively helped the software house in managing changes and risk mitigation. Frequent communication with the customer also significantly maintained the flow of the project as well as improved the developer's negotiation ability.

C. Scope creeps

Scope creeps is arguably the most anticipated risk in software development [6]. Due to the abstractness of computer software, it is unlikely for the customers, especially the ones who did not possess sufficient knowledge in information technology, to foresee and appreciate the product prior to the release. As a result, a great deal of requests for requirement change happened during the program. Some of these requests included minor fixes such as formatting of report or user interface. Others involved major amendments, e.g., changes in database structures or business processes.

While software houses could handle minor requests, major changes could cause serious consequence on the program. This could lead to delay, additional costs and even conflicts. The expert team played a vital role on this critical issue. They acted as an unbiased bridge for moderating and negotiating changes between software houses and entrepreneurs. Certain requirements which differed significantly from the basic requirements could cause serious problems to the project were mostly advised to be postponed from the program. However, in this case, instead of risking problems by force implementing the extra requirements, the software were designed to facilitate such requirements. As a result, the stakeholders could negotiate and develop them in the future.

According to interviews of the software houses, including the expert team in the negotiation greatly simplified the development processes. The experts also significantly eased customer requirements. Interestingly, a few software houses suggested that opinions from the experts seemed to be more convincing to the entrepreneurs than their suggestions, despite the fact that they are the same opinions. Some software houses claimed that the presence of the consultants sometimes halved the negotiation time and usually resulted in feasible solutions within time and budget limitation.

D. Heavy customization

Although some software were developed from established open source, they suffered from heavy customization. A number of businesses had extremely unique approach on certain activities. For example, company X proposed a development of a point-of-sale system. Yet, despite of the simplicity of general point-of-sale, they required that their system must be able to issue different pricing for different classes of customers. This also included specific giveaway for different regulars. Moreover, some businesses involved separated transaction of barcode packages, such as selling only one cigarette separately from its pack. These sophisticate configurations made the customization increasingly difficult.

Not surprisingly, the available open source software were unable to cope with all of these aforementioned cases. As a result, a few software houses put extra resources in order to satisfy these requirements. These created unexpected costs and efforts. Indeed, this problem might be reduced if all of these special requirements were cleared from the beginning.

E. Inadequate communication

A few cases of inadequate communication were experienced during the program. A major case surfaced when one entrepreneur reported that the software house rarely contacted them. At the end of the second month, excluding the initial business matching, they met each other only twice. The software vendor also failed to demonstrate the system prototype at the end of the last meeting as planned. This upset the customer and led to several escalating problems. Amongst these, the worst issue was deteriorating trust between both stakeholders. The entrepreneur later reported to the expert and asked for a substitute vendor. In the end, a vendor replacement was made and the project regrettably missed the deadline.

Another form of inadequate communication also caused minor troubles in a few other projects. Unlike the aforementioned case, this communication issue occurred on the entrepreneurs' side. It was reported that lack of internal communication caused significant misconception between two representatives thus caused minor conflicts on the acceptance tests. Specifically, this entrepreneur appointed a middle manager to act as the main source of requirements since the head executive was often occupied. During the development of this project, the middle manager coordinated and made all related decisions. However, at the acceptance testing, the head executive took charge and questioned several extra issues which were not included in the requirements. This caused a minor disruption to the project but later amended by the consultancy from the experts.

Similar to several other risks, the expert team was a key solution on inefficient communication. The experts needed to monitor and moderate not only information flow between stakeholders but also internal communication of the entrepreneurs.

IV CONCLUSION

Development of Logistics Management System for Greater Mekong Subregion Cluster (LS) was a program funded by the Office of Strategy Management, Upper North Cluster 1 (OSM-1), Thailand. The main objectives of this program were to increase competitiveness of local entrepreneurs, generating income of local software houses, as well as creating a network of logistics stakeholders. LS program mainly focused on developing or customizing logistics systems for local businesses. Twenty-two entrepreneurs and six software houses participated in the program. The software included several logistics activities such as inventory management, transportation management, point-of-sales and customer relationship management.

A number of expected risks surfaced during the development. Change of requirements, inadequate communication, resources limitation and heavy customization were amongst the most encountered problems. Interestingly, a policy which intended to increase future applicability such as open source development turned out to be a major threat for several stakeholders. As described in the previous section, these challenges were mitigated by several means such as increasing communication, implementing software development methodology which facilitated changes, and involvement of experts from local universities. In the end, twenty-one software were developed successfully on time, within the budget and satisfied the scopes. These logistics software are now in used and ready for further distribution.

ACKNOWLEDGEMENT

The authors would like to sincerely thank the Office of Strategy Management, Upper North Cluster 1 (OSM-1), Thailand, who initiated the Development of Logistics Management System for Greater Mekong Subregion Cluster (LS) program. This great program was a critical step to effectively enhance competitive capacity of local businesses. Additionally, we would like to pay our gratitude to all program stakeholders, especially the software houses, local entrepreneurs and experts for these invaluable lessons learnt.

REFERENCES

[1] K. Ewusi-Mensah, "Critical Issues in Abandoned Information System Development Project," Communications of the ACM, vol. 40, pp. 74-80, September, 1997.

[2] J. L. Eveleens and C. Verhoef, "The Rise and Fall of the Chaos Report Figures," IEEE Software, vol. 27, pp. 30-36, January-February, 2010.

[3] B. Boehm, "Project Termination Doesn't Equal Project Failure," Computer, vol. 33, pp. 94-96, September, 2000.

[4] D. B. Grant, D. Lambert, J. Stock, and L. Ellram, Fundamentals of Logistics Management, European Edition ed.: McGraw Hill Higher Education, 2006.

[5] K. Schwaber, Agile Project Management with Scrum: Microsoft Press, 2004.

[6] D. Davison, "Top 10 Risks of Offshore Outsourcing." SearchCIO.com, 2004 [Online]. Available: http://searchcio.techtarget.com/originalContent/0,289142,sid19_gci950602,00.html.

Sakgasit Ramingwong received his Ph.D. from the University of New England, Australia, in 2009. He is currently a lecturer at Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand. His main research focuses on software project management, risk management, software process improvement and general software engineering aspects.

Narissara Eiamkanitchat is working as a lecturer at Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand. She received her Ph.D. degree from Chiang Mai University in 2010. Her main research areas involve neuro-fuzzy, computational intelligence for data analysis and data mining for business.

Lachana Ramingwong earned her Ph.D. from the University of New England, Australia, in 2008. She is a lecturer at Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand. Software process analysis, software testing and human-computer interaction are her main research interests.

Sakgasem Ramingwong is an assistant professor of Department of Industrial Engineering, Chiang Mai University, Chiang Mai, Thailand. He received his Ph.D. from the Royal Melbourne Institute of Technology (RMIT), Australia in 2003. His main research focuses on logistics and supply chain management, logistics and supply chain systems development and supply chain and business enhancement.