GPN 2007800019
ISSN 1018-1393
9 771018 139006 03

MIS REVIEW Volume 17 Number 2 March 2012

MIS REVIEW
An International Journal

Vol.17 No.2 March 2012

MIS Review **Vol. 17, No. 2** Published in March 2012

Publisher: Samuel Chuen-Lung Chen

Published by: National Chengchi University, Department of Management Information Systems & Airiti Press Inc.

Editor-in-Chief: Eldon Y. Li

Executive Editor: Shari S.C. Shang

Assistant Editor: Chia-Jen Ting, Ting-Yu Chen

Printed by: Sincere Digital Printing, Co.

Cover Designer: Thomes Chen

Typesetting: Wen-Ling Liu

Publication Office:

National Chengchi University, Department of Management Information Systems
No. 64, Sec. 2, ZhiNan Rd., Wenshan District, Taipei City 116, Taiwan

Airiti Press Inc.
18F., No. 80, Sec. 1, Chenggong Rd., Yonghe District, New Taipei City 23452, Taiwan

Order Information:

Airiti Press Inc.
18F., No. 80, Sec. 1, Chenggong Rd., Yonghe District, New Taipei City 23452, Taiwan
Tel: +886-2-29266006
Fax: +886-2-22317711
E-mail: press@airiti.com

Price: NT$ 400

e-Journal: http://www.airitilibary.com

ISSN: 1018-1393

GPN: 2007800019

Printed in Taiwan

Editorial Board

Editor's Introduction

In this *MISR* issue, we are delighted to present four research papers concerning the following issues: the factors of ERP customization from consulting company's perspective, revisiting software requirements specifications, from CAATTs adoption to continuous auditing systems implementation, and the application of healthcare information system for comprehensive geriatric assessment. The summaries of the four papers are as follows.

Shaio Yan Huang, Hsueh-Ju Chen, An-An Chiu and Shih-Lung Hsieh in their paper "The Factors of ERP Customization from Consulting Company's Perspective" discuss ERP (Enterprise Resource Planning) customization issues and selects the largest ERP consulting company as the example for the case study, followed by a questionnaire survey exploring the decision factors of the consulting company for ERP customization. The data analysis shows consultants consider cost and time while assessing the possibility of ERP customization and they value more the necessity and usefulness in the long run than increase in benefit in the short run. The main contribution of their study is that in the future consultants may refer to the decision factors for ERP customization and design a customization-related check form in which these factors are operationalized to provide them with a quick reference for customization assessment.

Björn Johansson and Tanja Rolandsson in their paper "Revisiting Software Requirements Specifications -- What Could We Learn" introduce the importance of software requirements specifications (SRSs) and present an analysis of nine SRSs. The analysis shows that the overall structure of the SRSs either follows the IEEE (Institute of Electrical and Electronics Engineers) standard 830 (introduction -- overview -- list of requirements), or another structure (introduction -- references -- list of requirements), both with three main sections. They conclude that future SRSs should focus more on non-functional requirements since these are more difficult to describe and will probably play an even more important role when developing information systems in the future.

Chia-Ming Sun in his paper "From CAATTs Adoption to Continuous Auditing Systems Implementation: An Analysis Based on Organizational Routines Theories" studies the technological adaptation process of a company, which continuously implemented four Computer Assisted Auditing Technology and Techniques (CAATTs) projects in three years. The research results show that the documentation of CAATTs projects and group learning among different functions contribute to the routinization of automated auditing procedures; the continuous auditing system based on the automated auditing program also contributes to routinely audited tasks. However, the improvisational nature of auditing activities, implicit characteristics within general auditing software, and rigidity of automatic auditing programs cause the resistance of auditors on CAATTs use and impede the emergence and flexibility of computer-aided auditing procedures.

Nai-Wen Kuo and Yao-Yu Chung in their paper "The Application of Healthcare Information System for Comprehensive Geriatric Assessment" develop a healthcare information system for Comprehensive Geriatric Assessment (CGA), which integrates information technology and medical-related technologies. Their proposed system not only can process geriatric consultation services and ensure that all patient's information are stored in standardized format, but also provide medical personnel for statistical analysis and processing purposes. They use the Apriori algorithm of data mining to help doctors in finding out the relationship of geriatric syndrome. The system can enable organizations to meet their business objectives for increasing service capacity, cost control, revenue generation, while maintaining high quality of care for the patients in geriatric care.

We would like to thank all the authors and reviewers for their collaborative efforts to make this issue possible. Please note that the Chinese version of a paper, if any, is available online at the Airiti company website. We believe the bilingual format of paper submission would allow Chinese authors to better focus on their research process rather than being hindered by language barrier. It is our sincere wish that this journal become an attractive knowledge exchange platform for both Chinese and non-Chinese authors. Please render your continuous support and submit your papers to *MISR*. Finally, to our loyal readers around the world, we hope you find the contents of the papers useful to your work or research.

Dr. Eldon Y. Li
Editor-in-Chief and Professor

Dr. Shari S.C. Shang
Managing Editor and Associate Professor

Department of Management Information Systems
College of Commerce
National Chengchi University
Taipei, Taiwan
Spring 2012

MIS Review

March 2012 Vol.17 No.2

Contents

Research Articles

MIS Review Vol. 17, No. 2, March (2012), pp. 1-30

The Factors of ERP Customization from Consulting Company's Perspective

Shaio Yan Huang[1], Hsueh-Ju Chen[2], An-An Chiu[3], Shih-Lung Hsieh[3]

[1]*Advanced Institute of Manufacturing with High-tech Innovations, Department of Accounting and Information Technology, National Chung Cheng University*

[2]*Department and Graduate Institute of Accounting, National Chung Hsing University*

[3]*Department of Accounting and Information Technology, National Chung Cheng University*

ABSTRACT: *Enterprise Resource Planning (ERP) has become an essential information tool for modern enterprises. Enterprises should adjust existing processes to meet the ERP framework or keep them and customize the ERP when implementing it. The most practical way to implement it is partial ERP customization. This study selected the biggest ERP consulting company as the example for the case study, followed by a questionnaire survey to explore the decision factors of the consulting company for ERP customization.*

The analysis of results of the questionnaire shows consultants consider cost and time while they assess the possibility of ERP customization and they value more the necessity and useful results of it in a long run than rise in benefit in a short run. The situation analysis reveals consultants take a positive attitude to some particular industries like electronic, specific industries for ERP customization; relatively, a negative attitude to some non-representative traditional industries. It also suggests experienced consultants focus more on creating value in a long term considering ERP customization. Therefore, there is significant differentiation among attitudes of consultants from different seniorities for ERP customization.

The main contribution of this study is that in the future, consultants may refer to the decision factors for ERP customization in it and design a customization-related check form into which these factors they stress are translated to provide them with a quick reference index for customization assessment. This index may improve the quality of the assessment of consultants on consistency and preciseness.

KEYWORDS: *Enterprise Resource Planning System, ERP, Consultant, Customization, Code.*

1. Introduction

Enterprise Resource Planning (ERP) has become an essential information tool for modern enterprises. It was developed from data processing (DP), management information system (MIS), material requirements planning (MRPI), and manufacturing resource planning system (MRPII) into ERP in the 1990s. From the beginning, it was solely a system for processing data and then developed into a system for planning materials and production and eventually a multifunctional system with the integration of internal resources for sales, production and finance. ERP improves operational efficiency by integrating business processes and provides better access to integrated data across the entire enterprise (Davenport, 2000). While ERP integrates the processes by which businesses operate and saves time and expense, the failure rate of ERP implementation ranges from 40 percent to 60 percent (Liang et al., 2007).

The high failure rate of ERP implementation might be ascribe to the different interests between the businesses that aim to provide the optimal solutions for business problem and ERP consulting companies that prefer general solutions (Hong & Kim, 2002; Rajagopal, 2002). Packaged ERP systems are the dominant system used in many organizations (Mabert, Soni & Venkataramanan, 2000). However, packaged ERP system often does not meet the existing information processing requirements and demands. (Hong & Kim, 2002; Mabert, Soni & Venkataramanan, 2001). Thus, it is expected the implementation of ERP systems are associated with a problem of misfit such as the gaps between the functionality provided by the packaged ERP system and that needed in the business (Soh, Kien & Tay-Yap, 2000). For reasons of misalignment and strategic alignment, customizations of enterprise systems are necessary. One estimate is that 20% of the processes in an organization cannot be modeled in an ERP system without customization (Scott & Kaindl, 2000). Software modification and customizations are needed for the ERP system to meet the needs of the organization

In most ERP system implementations, some degree of system customization is needed. However, customizations that involve extensive additions to the ERP system or modifications of ERP system code may compromise the success because too much customization increase costs and limits maintainability (Rothenberger & Strite, 2009). Several prior studies discuss the issue of ERP customization and most of them are from the viewpoint of the business (Chang, 2002). Only few papers study ERP customization from the viewpoint of consultants. In typical ERP system practice, the percentage of the content meeting the need of enterprises is about 70% and the remaining 30% needs the guidance from experienced and professional consultants (Bingi, Sharma & Godla, 1999).

According to recent industry analysis, customizations still play an important role in ES implementations and can be a significant cost factor (Genovese, 2007; Ni, 2008; Zastrocky & Harris, 2008). The consulting company is responsible for considering whether the enterprise needs customization throughout the process of ERP implementation. There are certain steps to implement the ERP system: understanding and collecting, system planning, system assessment, implementation preparation, operation analysis and process design, education and training, operation simulation, production and verification and improvement (Liao, 1998). The step of operation analysis aims at analyzing the differences between practical processes and the system. The consulting company convinces enterprises to reduce costs by decreasing the customization. If customization is a must, what are the factors affecting the consulting company to agree on customization?

The issues this study aims to discuss are as follows: (1) Effects of consultants on the decision for ERP customization and what role they play; (2) Discuss the influence of customization on enterprises from the viewpoint of a consulting company; (3) The factors why a consulting company accepts the task of customization. Based on three issues above, this paper is going to study the decision factors for ERP customization from the viewpoint of a consulting company with exploratory study. An in-depth case study was conducted on a large-scale software company in Taiwan to find out the issues. This paper provides a reference for a consulting company on making decision during helping enterprises to customize their ERP system.

The purpose of this paper is to understand how the consulting company makes decisions and reaches an agreement with its clients when it is asked to customize the system during implementation with a case study of a large software company. We hope the consulting company will build up a customization mechanism which simplifies the project staff to earn the highest profits for the company and earn the trust of clients to create a win-win situation. The paper is divided into five sections: the first section: introduction; the second section: literature review; the third section: research designs and methodology; the fourth section: analysis of empirical study; the fifth section: conclusions.

2. Literature review

2.1 Enterprise Resource Planning (ERP)

Information technology grew rapidly in the 90s, which has contributed to the changes in enterprises to adapt the global competition. Enterprises started to implement the ERP system to catch up with the rapidly changing market and the growing price competition. American Production and Inventory Control Society (APICS) defines ERP: "ERP system is an accounting-oriented information system for identifying and planning the enterprise wide resources needed to take, make, ship and account for customer orders to expand its overall performance and reduce its costs (APICS The Association for Operations Management, 1998)." ERP is an enterprise-wide information system, which allows data that is not only from manufacturing or domestic operations to be shared throughout the enterprise. ERP also provides other modules of the global enterprise (Gould, 1997).

ERP is a cross-sectional enterprise system, which can integrate business processes (O'Brien, 2002). It helps enterprises to handle supply chain, which includes delivery of goods, stock management, customer orders management, production planning and management, shipping, accounting, human resources and other activities (Ragowsky & Somers, 2002). ERP imposes its modules based on the company's strategies, organization characteristics and culture to integrate all business activities to increase its profit (Davenport, 1998). Following MRPI and MRPII, the ERP is not a whole new domain, which integrates the operation information of each business unit to help decision makers get up to date information (Wang & Fei, 1999).

2.2 ERP customization

A common problem when adopting package ERP has been the issue of "misfits," that is, the gaps between the functionality offered by the package and that required by the adopting organization (Davis, 1988; Lucas, Walton & Ginzberg, 1988). Soh et al. (2000) observe there are three types of misfits: data, process, and output. Data misfits arise from incompatibilities between organizational requirements and ERP package in terms of data format, or the relationships among entities as represented in the underlying data model. Functional misfits arise from incompatibilities between organizational requirements and ERP packages in terms of the processing procedures required. Output misfits arise from incompatibilities between organizational requirements and the ERP package in terms of the presentation format and the information content of the output. When a misfit occurs, organization needs to choose either adapting to the new functionality or customizing the package.

Manhasset (2000) show that businesses change too much existing process to fit ERP system might cause implementation failure. In that case, proper customization might be a better solution. Levin, Mateyaschuk and Stein (1998) find that proper ERP customization can shorten the ERP implementation time. Moad (1995) points out businesses need to set flexible manufacturing system by customization. Proper customization might be the best solution. The proper application of customization is important for any packaged software system (Lucas et al., 1988); it is acute for ERP for two reasons (Haines, 2009). First, any misfit between organizational requirements and packaged system can be disruptive to an organization's operations. Second, ERP customization can be especially intricate and therefore difficult and expensive because of the complexity (Gattiker & Goodhue, 2002).

Customization can be distinguished into three types: configuration, extension, and modification (Haines, 2003). Configuration activities, which usually amount to changing entries in tables or configuration files, are supported by ERP consultant companies. Most ERP consultant companies also allow extension of their system by supporting common interfaces. Modification is an alteration that is usually not supported by ERP consultant companies. This includes code changes and other more invasive alterations.

Rothenberger and Srite (2009) mention that customization involving extensive additions to ERP system or modifications of ERP system code may compromise a project success because too much customization increases costs and limits maintainability. In particularly, upgrading customized project is labor-intensive. Stedman (1999) points out reasons for lessening customization are to fasten implementation time and to decrease the complexity of updating new versions. We summarize the effects of customization as follows:

1. Affecting the project progress: Before business implementing ERP system, gap analysis is needed because the functionality of packaged ERP system cannot satisfy all the business requirements. In the process of ERP implementation, the project progress will be delayed if there is higher need for customization.

2. Increasing project costs: Enterprises estimate the project costs before implementing the ERP system. During the implementation process, enterprises will need to increase the budget for customization to solve the differentiation issues. Apart from the software cost for building the ERP system, the invisible opportunity cost is even more astonishing (Stephen, 1999). A survey from Forrester Research on the U.S. top 1,000 enterprises found that "cost" is the major consideration for enterprises to decide whether to customize the system during implementation or not since customization brings extra project costs (Davis, 1998).

3. Difficulty in maintenance and inconvenience in system updates: Consultants focus on the overall planning in the beginning implementation period and help the end user to accustom to the new system. If customization is needed, consultants should use plug-ins to avoid changing ERP Script. Davis (1998) noted the reason avoiding ERP customization is the software for keeping and updating the customized program is more complicated and the software needs to ensure the customized program can be carried out normally under the updated system. Therefore, keeping the system will be more difficult after updating the system if the system structure is changed.

4. Increasing the workload for I.T. (Information technology) staff: Consulting company designs the urgent reports at the beginning of implementation and the remaining reports are developed by I.T. staff afterwards. The workload for I.T. staff will increase if there are too many customization requirements.

However, despite a growing awareness of the difficulties and cost implications of customizations (Wu, Shin & Wu, 2005) and concerted efforts to curb them, most organizations end performing at least some customizations. Nevertheless, it is also important to note that customization can play a positive role (Gattiker & Goodhue, 2005). All in all, enterprises should fully understand the necessity of customization suggested by end user during implementation and corporate with consultants to ensure a smooth production to reduce the mentioned effects.

2.3 ERP implementation strategy and process

To decide which ERP implementation strategy to use is the next step after enterprises decide to implement ERP system. The choice of implementation strategies depends on organizational characteristics such as size, structure, complexity and controls of organization. Welti (1999) mention that implementation strategy should be adopted based on availability of human resources, expertise, financial resources and time. There are three main implementation strategies: Step by step, Big bang and Roll out (Arinze & Anandarajan, 2003; Bancroft, 1996; Reijers, 2003; Welti, 1999). The difference between these implementation strategies is the risk an organization takes and the opportunity to reduce data clutter because of the incompatible. The advantages and disadvantages about these three strategies are presented in Table 1.

All ERP vendors have the clear and complete implementation. Therefore, the probability of implementation success will be raised as long as enterprises continue by the process required by ERP vendors. We introduce three example software companies: SAP, Oracle and DSC.

1. SAP: SAP is the world's biggest ERP software company. Its major product is R/3 (Realtime/3). ASAP (Accelerated SAP) Road map provides the methodology to optimize SAP System. ASAP is divided into five phases. The primary activities in each phase are shown as Table 2.
2. Oracle: Oracle proposed the AIM Method (Application Implementation Method) which has six phases. AIM implementation process is shown in Table 3.
3. Data Systems: The TIPTOP ERP developed by Data Systems employed the TIM method (TIPTOP Implementation Methodology) which includes six phases. The primary activities are shown in Table 4.

In conclusion, there is difference in the phases of the three mentioned companies though, the spirit is the same. The steps are similar including Project starting phrase, Current situation analysis phrase, User training phrase, Differential analysis phrase, Preparation before production phrase and System go-live phrase.

2.4 Effects of consultants on ERP customization decisions

When customization, consultants have a central place because they are the bridge linking enterprise and consulting company (Chang, 2002). They are the key factors to decide the success of implementation. During ERP implementation, Welti (1999)

Table 1 ERP Implementation Strategies

	Method	Advantages	Disadvantages
Step by step	Batch import of each module.	1. Llittle human resources are involved. 2. Module import enables employees to increase experience and knowledge. 3. Operation is not affected by the severe change. 4. Well-planned to avoid recurrence of failure.	1. Long duration may result in losing drive and patience of employees. 2. Difficult to identify the superiority of ERP system if modules are implemented in batches because ERP system stresses stresses on its holistic. 3. Holistic of system will be affected because employees try to customize the system because of convenience. 4. Employees are used to the old system if both old and new systems are in use.
Big bang	Phase out the old system and implement the new system.	1. Short period of implementation so employees are more aggressive. 2. Benefits are achieved faster. 3. Employees have higher sense of crisis if old system is phased out. 4. Draw the attention of senior management because all employees are asked to involve in implementation since time is urgent.	1. Enterprise can hardly afford the huge change. 2. Massive labor and time cost are involved. 3. Unsuccessful implementation will lead to enterprise crisis.
Roll out	Carry out the system in one of the internal company. Copy the experience to others after success.	1. Avoid massive change. 2. Implement the module of one company will show immediate benefits.	1. There is time lag between the integration of different companies. 2. There will be problems in consolidated statements easily.

Table 2 ASAP Implementation Process

	Phase	Primary activities	Key success factors
1	Project Preparation Phrase	1. Initiated project planning and technical requirements planning 2. Project program establishment and starting 3. Quality management	1. Executives Promise 2. Actual planning and expectation setup
2	Business Blueprint Phase	1. Project management 2. Project team training 3. System environment development 4. Organizational structure and process redefinition 5. Quality management	1. Availability of project team from customer side 2. Availability of experienced consultants 3. Clear definition of scope
3	Realization Phase	1. Project management 2. Project team training 3. Configuration setting 4. Interfaces setting 5. Integration and testing of the function of statement preparation 6. Documentation of end user manual and end user training 7. System management 8. Quality management	1. Quick decisions 2. No major change in project scope and focusing on the data transmission and interface
4	Final Preparation Phase	1. Project management 2. End user training 3. System management 4. Detailed project planning 5. Quality management	1. Interface testing 2. Acceptance of users (e.g., file export, interface)
5	Go Live and Support Phase	1. Prodction support and on-going support	1. Go live plan 2. End user training 3. Long-term strategy for assistance establishment

Source: Hiquet and Kelly (1998).

Table 3 AIM Implementation Process

Stage	Primary activities	Critical factors
1 Definition	1. Work on project plan and expectation. 2. Confirm project scope and strategy. 3. Build project team and work plan. 4. Evaluate risk of the project.	1. Executive promise and support.
2 Operation Analysis	1. Understand operation management and technical character. 2. Collect and analysis current operation. 3. Train the project team. 4. Build up blueprint.	1. Participation in training of related employees and successful installation of software and hardware. 2. Project team's understanding of Oracle Application.
3 Solution Design	1. Build up operation model. 2. Integrate technique and operation. 3. Train the project team to solve problems.	1. Involvement of professionals. 2. Acceptance of change.
4 Build	1. Build solid system structure. 2. Test the system at the same time.	1. Build up solutions. 2. Establish final solutions. 3. Test the solutions.
5 Transition	1. Test before production. 2. Train, prepare data and test in each unit. 3. Build up a formal operation environment.	1. Effective user training. 2. Error-free in each operation test. 3. Coordination of production process.
6 Production	1. Observe production status. 2. Plan for future direction.	1. Effective support from Oracle.

Source: Oracle AIM handbook (Buchan et al., 1999).

Table 4 TIM Implementation Process

	Stage	Primary activities
1	Project Committee and Executive Team	1. Establish division of labor of project team members and decision mechanism.
2	Kick-off Meeting	1. Advertise the goal and determination for ERP project. 2. Introduce team members and the division of labor. 3. Introduce the implementation schedule.
3	Survey and Diagnosis	1. Consultants from Data Systems to understand the operation model and information process.
4	Application Training	1. Consultants introduce the standard operation and functions of TIPTOP. 2. Suggestions on process.
5	Operation Training	1. Process planner to understand the operation. 2. Propose differences in detail.
6	S.O.P. Modeling	1. Set up S.O.P..
7	Differential Analysis & Customization	1. Review the between process and system and find out the solutions.
8	Data Migration & Entry	1. Convert data from the old system. 2. Key-in related information.
9	Business Flow Simulation	1. KM and KU to be familiar with the future operation model. 2. Found out potential differentiation.
10	Balance Check & Implementation	1. Calculate the balance of each module. 2. Key-in (or convert) the correct opening balance.
11	System Go-live	1. New system go live.
12	Project Review Meeting	1. Project costs and schedule control. 2. Review the goal in each phase. 3. Resolve the major outstanding issues.

Source: Data Systems (2000).

points that consultants are responsible for: (1) ensuring the project is in line with the expected schedule; (2) providing the project team with consultation, support and training; (3) showing, supervising and confirming the project planning; (4) solving the problems with the experts in software company; (5) system setup and customization; (6) keeping the quality of project; (7) recording the process documentation. As a result, the capabilities, behavior patterns and communication abilities of consultants are important. At the beginning of implementation, consultants need to frame a project schedule with the customers. However, the project costs are always more than expected because of irresistible factors in which customization is the major one. The importance of consultants is shown because they have to understand the differentiation between and the system the need of customers (Soh et al., 2000). The different stands between ERP suppliers and customers are the major failure factor (Swan, Newell & Robertson, 1999).

Most ERP consultants strongly support lessening implementation risk, reduce implementation cost, avoid negative impacts on system performance, ease adoption of future package upgrades, reduce maintenance costs, and foster adoption of process-oriented "best practices (Brehm, Heinzl & Markus, 2001)." Organizational users, however, often demand to have the ERP package customized to meet their operational needs, reduce disruption to established ways of doing things, and meet regulatory requirements and customer needs. Consultants need to assist customer to make best choice.

In typical ERP system practice, the percentage of the content meeting the need of enterprises is about 70% and the remaining 30% needs the guidance from experienced and professional consultants (Bingi et al., 1999). Consultants should aid the end user to understand the effects of customization on enterprise and help the end user in knowing which needs are necessary to lower the customization requirements. The major responsibility of consultants is to assist customers to implement ERP system successfully. While customization is only one of consultants' jobs, consultants focus on overall planning. Consultants should responsible for checking whether the system satisfies the organizational culture, industry characteristics and operation process after production and providing the best solutions without customization. During the implementation, consultants should tell the end user which customization requirements need to be done before production and others can be deferred and developed after production. Endless demands can increase the risk of implementation, delay in schedule and the costs of project and gradually affect the schedule and the result of ERP implementation.

2.5 Factors consulting company accepting ERP customization

The biggest problem most enterprises face is whether to choose customization or not at the beginning of implementing ERP system (Craig, 1999). A few users try to avoid customization to decrease the complexity of implementation and reduce the costs. Customization is the only way for the ERP system to meet all needs of customers. Referring to Haines (2009), we summarize there are six factors for consulting company to accept the customization appointment in three categories as follows.

2.5.1 Product factors

1. Enterprise's critical demand for customization: In the differential analysis conducted during ERP implementation, consultants discuss the functions that cannot satisfy the needs with the customers. They list out the unsolvable problems and ask for the coordination of customers. Consultants put the demands in order by the necessity of customization and habitability. Consultants develop the customization based on the importance, necessity and urgency of customers' needs to ensure the production is smooth.

2. Commonality demand: The development of ERP system is increasingly complete and mature. Apart from the sophisticated development of both software and hardware equipment and the techniques of development employees, absorbing practical experience through the coaching process of consultants play an important part. Customers are able to solve their problems with a more perfect system. Throughout the coaching process, consultants learn more about the market needs by discussing customization with customers.

2.5.2 Internal factors

1. Customer accepts higher customization cost: Consulting company will accept customization if the customization requirements do not affect the structure of the whole system; and customers accept the higher charges for customization.

2. Increase the output value of consultants to achieve performance: Consultants' performance is considered based on the coaching hours and the customization hours. Thus, consultants will also consider the value output other than the appropriateness of demands.

2.5.3 Customer relation

1. Positive interaction between two parties: The probability of system production increases when the interaction between consulting company and enterprise is fine. Consultants accept customization requests such as adjustment of reports or addition of fields.

2. Focus on future benefits: Not do the two parties cooperate in the present, but also focus on the future. Consulting company should accept customization request if it is sure the customer will bring greater benefits to the company in the future.

3. Research designs and methodology

3.1 Conceptual structure of research and the selection of case study

The analytical foundation of this paper is "the factors why the consulting company accepts the task of customization" to explore the factors in case company D during its assistance for ERP implementation. The analysis of case study is our basic model, followed by the questionnaire for employees in D Company. The data collected from the questionnaires is analyzed and authenticated to generalize the factors for the consulting company to accept the task of customization.

D Company was established in 1982. There are about 1,200 employees in present and its capital is about 1.2 billion. Its capital was raised to nearly 4.8 billion after it had been acquired by Whitesun in 2008 and it has become the biggest ERP application company in Taiwan. For getting market share in Chinese market, several operating bases were set up in Shanghai, Guangzhou, and Beijing in 2002; the Asia-Pacific market is also valued so the company set up operating base in Vietnam in 2008. In a survey conducted by Common Wealth Magazine in 2007, 53% of the top one thousand manufacturers in Taiwan are the customers of D Company; and around 31% of the top five hundred companies in the service industry are served by D Company as well.

The design and development of MRP was the business focus in the early establishment stage. After accumulating experience in software design and integration for years, D Company has designed suitable ERP products for large-sized, medium-sized and small-sized enterprises. Enterprises are able to choose appropriate products according to their business scales. D Company pays a lot of attention on Product research and development. Out of all employees, 22% of them are R&D staff, 21% of them are consultants and 43% of them are responsible for technical services. They put technique development and customer service into the first place so the consultants have rich consulting experiences. In 2002, D Company had 3,679 customers and in 2008, the number has increased to over 10,000. Its customers include manufacturing industry, trading industry, logistic industry and finance industry. This can explain why this paper selects D Company as the example to conduct exploratory study.

3.2 Questionnaire design and object

This paper carries out the analysis by conducting questionnaire of which the contents are designed based on the results in literature study while the object of this paper is consultants in D Company. The questionnaire is formed by descriptions of scenario simulation cases. Such scenarios are divided into three cases:

1. Company A is a significant electronics listed company. Consultants wish to establish the image of authority by the successful implementation although they have already had years of experience in this industry. Consultants are devoted in this case in hoping that other companies in this industry can gradually implement the ERP system provided by the consulting company. Company A has expanded so rapidly that the current system is no longer suitable to use. To meet Company A's demand, the consulting company is stressed to put the new system into practice by the very tight schedule.
2. Company B is a medium size company manufacturing power tools. Consultants wish to accumulate more experiences although he has enough experience in the industry and is able to help customers put their new system into practice quickly. During the coaching process, consultants find that the customer tends to customize the system to fulfill their demands; they do not care whether the operation process conforms to internal auditing and control. To meet Company B's demand, the consulting company is stressed to put the new system into practice by the very tight schedule.
3. Company C is a traditional textile company. The industry characteristics are complicated and diversified so consultants pay more attention to the industry characteristics and hope to learn more experience and knowledge in this case. To meet Company C's demand, the consulting company is stressed to put the new system into practice by the very tight schedule.

The three situations above are the basic of our questionnaire. Data structure aspect, system function aspect and report output aspect are used to explore the decision factors for customization. Cross-analysis and comparison are done on the differentiation of industries, industrial representative and the familiarity to the industry of the software company.

The contents of questionnaire are mainly divided into three parts. The first part: a brief description of the reasons and nature of the questionnaire and the situation simulation cases. The second part: design of the questionnaire contents. Contents are described with situation cases and three customization types are employed as aspects which are data structure aspect, system function aspect and report output aspect. There are six factors under each aspect, which are "critical demand for production customization," "commonality demand," "acceptance of higher customization charges," "increase consultant's output value to achieve performance," "good communication among two parties," "focus on future benefits." Likert Scale is employed for measurement and the five levels are "very unimportant," "unimportant," "average," "important" and "very important." In the score 1 to 5, 1 represents the lowest importance while 5 represents the highest importance. The contents of questionnaire are summarized in Table 5. The third part: the basic background of the questionnaire takers.

Table 5 Questionnaire Design

	Code	Variable	Definition
Data structure aspect	X1	critical demand for production customization	During the stage of differential analysis, consultant lists out the customization items in the order of essentiality to determine the order of customization. And whether the consultant accepts the order proposed by the customer?
	X2	commonality demand	During the consulting process, consultant accepts customization because it is necessary for most enterprises and reflects the actual needs of the market throughout the discussion with customer.
System function aspect	X3	acceptance of higher customization charges	If the demands are no necessary for production and not common, will customer still accepts the higher charges for customization?
	X4	increase consultant's output value to achieve performance	Performance is considered based on the coaching hours and also the customization hours. Thus, consultants will also consider the value output other than the appropriateness of demands.
Report output aspect	X5	good communication among two parties	Good coordination between enterprise and consultant during implementation and smooth process of project. Will consultant accept customization if the demands do not affect the system?
	X6	focus on future benefits	Ensure the customer will bring greater benefits to the company. Will consultant accept customization if future benefits are even greater although he has to pay the cost in the short run?

Source: summarized by this study.

4. Analysis of empirical results

This paper aims at exploring the decision factors for ERP customization from the viewpoint of consulting company. The research samples are consultants in D Company. Questionnaires were sent via e-mail. We would contact the test taker if we have not received the reply after 10 days of the delivery of questionnaire and asked whether he has received the questionnaire or he needed further explanations to finish it. We also contacted the test taker in the next week if he did not return the questionnaire. Ninety questionnaires were distributed and the responded questionnaires were sixty-one, the response rate is 68%. Exempting three inappropriate questionnaires, we had fifty-eight valid questionnaires. The response rate is 64%.

4.1 Descriptive statistics

Analysis results for the descriptive statistics are summarized in Table 6. (1) Gender: the proportion of male and female consultants in D Company is average; (2) Age: consultants in D Company are most likely between 31 ~ 40 year-old, in the next place are consultants between 26 ~ 30 (incl.) year-old. Hence, the professionalism of consultants is formed by experiences and continuous learning. (3) Academy degree: Most of the consultants' degrees are bachelor followed by master.

Table 6 Descriptive Statistics of Basic Information of Test Takers

Background of test takers		Number of people	Percentage
Gender	Male	32	52.5%
	Female	29	47.5%
Age	21 ~ 25 (incl.) year-old	2	3.3%
	26 ~ 30 (incl.) year-old	7	11.5%
	31 ~ 35 (incl.) year-old	23	37.7%
	36 ~ 40 (incl.) year-old	21	34.4%
	41 ~ 45 (incl.) year-old	8	13.1%
	46 ~ 50 (incl.) year-old	0	0%
	51 ~ 55 (incl.) year-old	0	0%
	56 or above	0	0%
Seniority of consultant	Above 10	11	18%
	5 ~ 10 years	23	37.7%
	Below 5	27	44.3%
Education	Doctor	0	0%
	Master	16	26.2%
	Undergraduate	38	62.3%
	Vocational school or below	7	11.5%
Residency	Northern Taiwan	11	18%
	Central Taiwan	36	59%
	Southern Taiwan	14	23%

Source: summarized by this study.

4.2 Measure of reliability and validity

4.2.1 Reliability analysis

The reliability test results of each aspect in this paper are shown in Table 7 and most of them reach 0.7 or above. The reliability for situation 1 is relatively low probably because the test takers were still unfamiliar with the questionnaire at the beginning. However, the average reliability of this questionnaire is over 0.9. The Cronbach's α coefficient is over 0.7, which implies the degree of internal consistency of the questionnaire is high (Nunnally, 1978).

Table 7 Reliability Analysis of Each Aspect

Situation / Aspect	1	2	3
		Cronbach's α	
Data structure aspect	0.591	0.721	0.702
System function aspect	0.801	0.848	0.871
Report output aspect	0.699	0.787	0.780
Reliability of situation	0.746	0.817	0.844
Total reliability		0.913	

Source: summarized by this study.

4.2.2 Validity analysis

Content validity represents a systematical test to examine the relevance of contents. This paper referred to related literature and collected measurements from literature. The questionnaire was amended after consulting the opinions of scholars who specialize in information management to ensure the items in questionnaire reflected the aspects in our research framework effectively. Face validity only shows the validity of test adaptors or test takers subjectively. To ensure all test takers understand the items in questionnaire, we asked two management and two seniors in IT department to take the pre-test. We further discussed the contents of the questionnaire and amended the questionnaire. Hence, both the content validity and the face validity of the questionnaire reach the significance level.

4.3 Analysis of questionnaire results

4.3.1 Descriptive analysis of overall aspects

From the summary in Table 8, we learnt that when consultant considers whether to customize the system, the total scores of "System function aspect" are the lowest, which are 199.84, 196.35 and 202.33 respectively. This implies that the consultant concerns more about “Data structure aspect” and “Report output aspect” rather than “System function aspect.” Moreover, the mean of “critical demand for production customization” and “demand commonality” is mostly above 4. We infer that consultant agrees with these two items more than the others. The result is consistent with the viewpoint that the communication between the consultant and the end user throughout the implementation process is important (Bingi et al., 1999). Furthermore, the mean of “focus on future benefits” is between 4 ~ 3.5 indicating that this is the second important issue for the consultant. A possible explanation is that the consulting company focuses on long-term profits because it cannot survive without making profits. The profit-making ability becomes a critical evaluation against consultant. Lastly, the mean of “increase consultant’s value to achieve performance” is the lowest which means that the consultant do not value “increase consultant’s value to achieve performance.” The result is consistent with the viewpoint that consulting company values long-term profits. An analysis of three situation simulations is illustrated as below.

Situation simulation 1 -- Electronics industry, industry representative, good familiarity with the industry, high pressure for production schedule:

Electronics industry is one of the most important industries in Taiwan. In Table 8 we found that when consultants are familiar with the industry, “Data structure aspect” is regarded as the most important consideration for customization though the pressure for production schedule is high. The total score for “Data structure aspect” is 231.66 and is way above “System function aspect” scoring 199.84 and “Report output aspect” scoring 224.00 because A company is representative in its industry. Data structure is the core of the package software. If the ERP system provided by the consulting company cannot fulfill A company’s demands, the reinvention will be needed. In addition, consultants recognize that “critical demand for production customization,” “demand commonality” and “focus on future benefits” are the most important ones. Their means are much higher than other items’ means. We can conclude that consultants wish to gain more experiences and learn the system build-up technology and after-service ability by serving indicative enterprises. Such experience will be converted to consultants’ own techniques and increase their competence in this industry. The successful experience can be copied to similar enterprises and shorten their implementation duration. In the meantime, consulting company can establish its reputation for seeking various benefits in the long run. In the contrast, “increase consultant’s value to achieve performance” is not the main factors for consultants in considering customization.

Situation simulation 2 -- Traditional industry in medium-small size, non-representative in the industry, fair familiarity with the industry, high pressure for production schedule:

Table 8 Summary of Descriptive Analysis

Situation / Items	1						2						3					
	Data structure aspect		System function aspect		Report output aspect		Data structure aspect		System function aspect		Report output aspect		Data structure aspect		System function aspect		Report output aspect	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Q1	4.16	0.778	3.62	1.051	4.15	0.853	3.90	0.831	3.36	1.212	4.03	0.856	4.30	0.863	3.48	1.233	4.02	0.904
Q2	4.18	0.695	3.52	1.043	4.05	0.872	3.75	0.856	3.28	1.113	3.87	0.939	4.33	0.851	3.51	1.090	4.00	0.856
Q3	3.49	0.924	3.03	0.875	3.57	1.040	3.51	0.829	3.10	0.961	3.82	0.866	3.69	1.025	3.38	0.986	3.70	0.919
Q4	3.16	1.098	2.85	1.030	3.20	1.123	3.10	1.150	2.87	1.040	3.28	1.067	3.03	1.169	2.77	0.990	3.31	0.992
Q5	3.69	1.009	3.16	0.952	3.52	0.993	3.47	0.833	3.20	0.963	3.61	0.971	3.72	0.878	3.23	1.071	3.60	0.887
Q6	4.10	0.831	3.46	0.993	3.88	0.904	3.59	0.920	3.51	0.977	3.95	0.956	4.07	0.793	3.54	1.010	3.85	0.853
Total scores	231.66		199.84		224.00		213.19		196.35		229.32		235.16		202.33		225.01	

Source: summarized by this paper.

Note: Q1: critical demand for production customization; Q2: demand commonality; Q3: accept higher customization charges; Q4: increase consultant's value to achieve performance; Q5: good communication with customers; Q6: focus on future benefits.

Referring to Table 8, we learnt that the scores of “Data structure aspect,” “System function aspect” or “Report output aspect” in Situation 2 are much lower than the other two situations. Due to the particularity of the industry, the duration, human resources and costs for system build-up cannot be copied in other enterprises. The costs cannot be shared. Meanwhile, the system for this industry requires more maintenance. The total costs increase but the derived benefits are limited. Consultants will be less willing to customize the system. For each aspect, the score of “Report output” is the highest. Consultants are more willing to provide “Report output” service because it involves less technical issue and less time. In each item, the scores of “critical demand for production customization,” “demand commonality” and “focus on future benefits” are the highest. This result is in accordance with the mindset and attitude of consultants during evaluation.

Situation simulation 3 -- Special traditional industry, non-representative in the industry, low familiarity with industry, high pressure for production schedule:

From Table 8, we learnt that the score for “Data structure aspect” is higher than the scores for “System function aspect” and “Report output aspect” in special traditional industry. It implies that consultants will face the same customization demands for ERP implementation in this industry because the industrial environment, characteristics and operation process of enterprises are the same. Such customization demands, important and urgent to the industry, can be solved by module design that will be one of the ERP system functions. This customization process is regarded as a learning opportunity for consultants. Consultants concern about this aspect because they can utilize their professional techniques and enhance their learning aptitude. In each item, the scores for “critical demand for production customization,” “demand commonality” and “focus on future benefits” are the highest. This result shows that consultants will design reports to meet the enterprises’ needs when it is requested for specific industry.

Summary of three situation simulations:

In both A and C company, consultant consider “Data structure aspect” as the most important factor of customization. In B company, consultant consider “Report output aspect” as the most important factor of customization. From the results, we realize that consultant emphasize on different factors of customization in different industries. In all three situation simulations, the scores for “critical demand for production customization,” “demand commonality” and “focus on future benefits” are the highest and “increase consultant’s value to achieve performance” is lowest.

4.3.2 Test on the effects of customer’s characteristics on customization

The results reflected in Table 9 indicate the level of differentiation when consultants consider customization based on “Data structure aspect,” “System function aspect” and

Table 9 Summary of the Effects of Customer's Characteristics on Customization

Situation			Mean	SD	Significance (two-tailed)
Data structure aspect	Mated pair 1	S11-S21	0.23611	0.52909	**0.001**
	Mated pair 2	S11-S31	-0.05738	0.63383	0.482
	Mated pair 3	S21-S31	-0.29722	0.66673	**0.001**
System function aspect	Mated pair 1	S12-S22	0.05738	0.64828	0.492
	Mated pair 2	S12-S32	-0.04098	0.73828	0.666
	Mated pair 3	S22-S32	-0.09836	0.68811	0.269
Report output aspect	Mated pair 1	S13-S23	-0.03333	0.58930	0.663
	Mated pair 2	S13-S33	-0.01695	0.53130	0.807
	Mated pair 3	S23-S33	0.01389	0.58488	0.855

Note: 1. S11, S12, S13: Customization for "Data structure aspect," "System function aspect" and "Report output aspect" in Situation 1.
2. S21, S22, S23: Customization for "Data structure aspect," "System function aspect" and "Report output aspect" in Situation 2.
3. S31, S32, S33: Customization for "Data structure aspect," "System function aspect" and "Report output aspect" in Situation 3.

"Report output aspect" towards different company situations. The T test for paired sample of "Data structure aspect" in Table 9 shows there is 1% in significance level in Situation 1 and Situation 3 comparing that of Situation 2. It implies consultants will do customization for "Data structure aspect" considering the characteristics of customers. Consultants are still willing to carry out customization although they have to perform under pressure due to the short schedule for production when the customers are electronics or specific industry, the customer has industry representative, or the consultants are not familiar with the industry. The result agrees with the previous result that consultants increase their understandings of industry characteristics, environment and business process through the customization of changing the core of ERP system. Consultants eventually extend their professionalism and increase their own values while assisting their customers.

Table 9 indicates there is no significance differentiation in three situations when consultants consider "Data structure aspect" and "Report output aspect." This implies consultants will not have different opinions on "Data structure aspect" and "Report output aspect" for customers of different natures. A clear budget plan and control is one of the key success factors for enterprises implement ERP system (Hsieh, 2000). Since customizing "System function aspect" goes against the standard operation process, consultants tend to suggest customers to reengineer business process under limited budget rather than customizing the functions of the system. In addition, consultants tend to customize report output if time and cost is sufficient. It is because special report format is required for different industries. Customizing report output will not change core code of the system that affects future maintenance and updates of system. Hence, consultants do not consider the previous characteristics of customers in evaluating customizing report output. In conclusion, not only do consultants consider the customer characteristics, but also the key elements that need to be customized.

4.3.3 Test on the effects of consultant's characteristics on customization

Table 10 explains the test result of whether gender of consultants affects ERP customization. There is no significant influence of consultant's gender on the customization decision for "Data structure aspect," "System function aspect" and "Report output aspect" regardless of customer's characteristics. Only when the customer belongs to specific industry and the consultant is unfamiliar with the industry, there is 10% differentiation level for customization in "Report output aspect" among male and female consultants. In another words, male consultants tend to customize report format than female consultants do.

Table 11 indicates the test result of whether the seniority of consultants affects ERP customization. In Situation 1, Table 4.4 shows the greatest differentiation happens in "Data structure aspect" and "System function aspect" in the item of each aspect, and their P values are 0.094 and 0.024 respectively. With further analysis, the greatest differentiation happens to consultants with seniority between 5 ~ 10 years and those below 5 years; whereas the differentiation for consultants with seniority of above 10 years and 5 ~ 10 years is not great. Such finding means consultants will have different attitudes on customization that needs to change the core program code depending on their seniority. Although consultants are under pressure of production schedule, the customers are in the electronics industry and have industry representative; consultants are familiar with the industry as well. The difference in handling customization of consultants may due to their coaching experiences. There is significant differentiation in working performance for different age groups (Huang, 2005). The golden age of consultants is around 5-10 years in which consultants are committed to create long-term value. Instead of considering the costs and benefits of solution to customization, consultants consider the benefits brought by cost benefit in a deeper aspect. Regarding consultants with less than 5 years experience, they are more conservative against the change of ERP core during customization because their professionalism and customer experience are still weak and insufficient. This can explain why different age groups show great differentiation in the attitude towards customization that changes the ERP core.

Table 10 T Test Statistics against Gender

	Gender	Mean	SD	*F* test	Significance
S11	Male	3.7760	0.57984	2.205	0.143
	Female	3.8218	0.44305		
S12	Male	3.2396	0.74648	0.045	0.832
	Female	3.3161	0.66270		
S13	Male	3.7344	0.54355	0.459	0.501
	Female	3.7321	0.69882		
S21	Male	3.5469	0.63250	2.518	0.118
	Female	3.5595	0.55037		
S22	Male	3.2240	0.83988	0.028	0.868
	Female	3.2126	0.74655		
S23	Male	3.8177	0.54929	0.307	0.582
	Female	3.6954	0.76515		
S31	Male	3.8490	0.69090	2.090	0.154
	Female	3.8621	0.47989		
S32	Male	3.3437	0.77244	0.274	0.602
	Female	3.2874	0.90402		
S33	Male	3.7527	0.45107	2.850	0.097
	Female	3.7474	0.77871		

Note: 1. S11, S12, S13: Customization for "data structure aspect," "system function aspect" and "report output aspect" in Situation 1.
2. S21, S22, S23: Customization for "data structure aspect," "system function aspect" and "report output aspect" in Situation 2.
3. S31, S32, S33: Customization for "data structure aspect," "system function aspect" and "report output aspect" in Situation 3.

In Table 11, it shows in Situation 2 and Situation 3 that the greatest differentiation in items of each aspect is "System function aspect" with P values of 0.033 and 0.049 respectively. This result represents when consultants are under schedule pressure in handling non-representative customers; regardless of their familiarity of the industry; the seniority of consultants do not affect their consideration for "Data structure aspect" and "Report output aspect" but the "System function aspect" is obviously affected. We found that consultants who have 5-10 years experience tend to construct standard functions of ERP system for that special industry to apply because there is no or the current ERP system cannot fulfill the needs of the specific industry. Consultants have considered the long-term benefits to achieve a win-win-win situation. Consultants with 5 or less years of experience are not professional enough so they are not confident in customization. Hence, there is great differentiation in the attitude and cognition of customization that changes ERP core for these two groups.

5. Conclusions and suggestions

The purpose of this paper is to explore the reasons and factors why companies want to customize their ERP system when implementing it and discover the decision factors for customization from the viewpoint of the consulting company by conducting the case study, employing the D company to investigate the issues we have pointed out by interviews and followed by a questionnaire survey to carry out a deeper analysis. The results of the questionnaire analysis indicate that consultants pay more attention on “data structure aspect” and “report output aspect” than “system function aspect.” That shows that consultants take into consideration cost and time when evaluating the ERP customization, especially for the one which will change the core function of the ERP system. Among the six factors, “critical demand for production customization” and “demand commonality” are most emphasize by consultants while “increase consultant’s value to achieve performance” is the least important. The major reason is that consultants are more concerned about the necessity and long-term benefits than short-term increase in performance.

According to the situation analysis, consultants’ willingness for customization is higher in mainstream industries such as the electronics industry because they could build up their reputation and at the same time copy the experience. Regarding specific industries, consultants are still willing to customize the ERP system due to their own drives for enhancing their ability and self-learning. For non-representative traditional industries, consultants’ willingness for customization are relatively lower because the manpower and time costs are higher while long-term benefits are not outstanding. Their service areas are limited to report output service which requires the minimum manpower and time costs.

Based on the analysis of the gender of the consultant, the result shows that gender has no significant influence on “data structure aspect,” “system function aspect” and “report output aspect” except that, when the customer is in specific industry and consultants have little understanding in that industry, male consultants tend to customize the format for report output compared to female ones. Moreover, based on the analysis of the seniority of the consultant, the result indicates great differentiation in the attitude of consultants with different seniority toward to customization which changes ERP core functions. The main reason is that the consulting experience varies with seniority. Senior consultants may

Table 11 Summary of One-Way Analysis of Variance

Situation simulation	Aspect	*F* test	P value	Multiple tests				
				Seniority of consultant		Mean difference	SD	P Value
S1	S11	2.469	**0.094**	Above 10 years	5 ~ 10 years	0.24769	0.18455	0.378
					Below 5 years	-0.06229	0.18007	0.936
				5 ~ 10 years	Above 10 years	-0.24769	0.18455	0.378
					Below 5 years	-0.30998	0.14285	0.085
	S12	3.987	**0.024**	Above 10 years	5 ~ 10 years	0.12978	0.27783	0.734
					Below 5 years	-0.39169	0.27109	0.378
				5 ~ 10 years	Above 10 years	-0.12978	0.27783	0.734
					Below 5 years	-0.57246	0.21505	0.027
	S13	0.076	0.927					
S2	S21	1.872	0.163					
	S22	3.622	**0.033**	Above 10 years	5 ~ 10 years	0.20883	0.27783	0.734
					Below 5 years	0.36364	0.27109	0.378
				5 ~ 10 years	Above 10 years	-0.20883	0.27783	0.734
					Below 5 years	-0.57246	0.21505	0.027
	S23	0.865	0.427					
S3	S31	1.356	0.266					
	S32	3.181	**0.049**	Above 10 years	5 ~ 10 years	0.23584	29411	0.703
					Below 5 years	-0.33502	28698	0.477
				5 ~ 10 years	Above 10 years	-0.23584	29411	0.703
					Below 5 years	-0.57085	22765	0.039
	S33	0.175	0.840					

Note: 1. S11, S12, S13: Customization for "data structure aspect," "system function aspect" and "report output aspect" in Situation 1.
2. S21, S22, S23: Customization for "data structure aspect," "system function aspect" and "report output aspect" in Situation 2.
3. S31, S32, S33: Customization for "data structure aspect," "system function aspect" and "report output aspect" in Situation 3.

concern more about creating long-term values in assessing customization than junior ones do.

The results show that, in ERP customization, consultants take account of long-term benefit for their clients instead of increasing short-term performance and senior consultants focus more on value creation in the long run. The result of this study can provide consulting companies an objective assessment mechanism on ERP customization. The main contribution of this paper is that in the future, consultants may refer to the decision factors for ERP customization in it and design a customization-related checklist form into which these factors they emphasize are translated to provide them with a quick reference index for customization assessment. This index may enhance the quality of the assessment of consultants on consistency and preciseness. This paper also suggests the consulting company to focus on “critical demand for production customization,” “demand commonality” and “focus on future benefits” when designing their own checklist focusing on so that consultants can get quick assessment to the customized situation and eventually increase the quality of communication with the appointment company. There are two suggestions we made. (1) We suggest that customized checklist can be designed separately based on various situations or enterprises because the assessment methods may vary with different industries. By doing this, consultants are able to pick the relevant checklist quickly to increase the quality and effectiveness of decision-making. (2) We suggest consulting company integrates and classifies consultants according to seniority, so apprenticeship practice is suggested. In this practice, senior consultants lead the new consultants to make the decision together. Such practice assists all consultants to have a uniform decision-making process and approach so as to grasp the customer’s needs and provide services with equal quality.

Limitations of this paper are as follows: (1) This paper chose only one consulting company in Taiwan as the research object although this company is the most representative one, but flaws may appear in this paper due to the characteristics and culture of this company. Future research could extend to other companies in the same industry in order to obtain more convincing results. (2) This paper used only T test and one-way ANOVA although the results showed clearly customization evaluation factors but other research methods such as regression analysis are required to further clarify issues for deeper implications. (3) Consultants were the only research object in this paper. Although the opinions and decisions of consultants are definitely effective, the management of consulting companies is the final decision makers. Future research could work on the differentiation between these two parties.

References

APICS The Association for Operations Management (1998) 'Defining enterprise resource planning', *APICS*, available at http://www.apics.org/OtherServices/articles/defining.htm (accessed 30 June 1998).

Arinze, B. and Anandarajan, M. (2003) 'A framework for using OO mapping methods to rapidly configure ERP systems', *Communications of the ACM*, Vol. 46, No. 2, pp. 61-65.

Bancroft, N.H. (1996) *Implementing SAP R/3: How to Introduce a Large System into a Large Organization*, Manning, London, UK.

Bingi, P., Sharma, M. and Godla, J. (1999) 'Critical issues affecting an ERP implementation', *Information Systems Management*, Vol. 16, No. 3, pp. 7-14.

Brehm, L., Heinzl, A. and Markus, M. (2001) 'Tailoring ERP systems: a spectrum of choices and their implications', *Proceedings of the Thirty-Fourth Hawaii International Conference on Systems Sciences (HICSS-34)*, Maui, HI, January 3-6, pp. 1-9.

Buchan, S., Born, G., Burns, G., Dehner, B., Goossens, L., Haase, T., et al. (1999) 'Application implementation method (AIM) method handbook -- release 3.0.0', Technical report, Oracle, Redwood City, CA.

Chang, S.-C. (2002) 'Decision behavior of business ERP customization', Unpublished Master's thesis, Chung Yuan Christian University, Taoyuan, Taiwan. (in Chinese)

Craig, S. (1999) 'College ERP success may take custom fit', *Computerworld*, Vol. 33, No. 48, p. 12.

Data Systems (2000) 'The issue of modern business management: ERP system', *Data Systems*, available at http://www.dsc.com.tw (accessed 1 July 2000). (in Chinese)

Davenport, T.H. (1998) 'Putting the enterprise into the enterprise system', *Harvard Business Review*, Vol. 76, No. 4, pp. 121-131.

Davenport, T.H. (2000) *Mission Critical -- Realizing the Promise of Enterprise Systems*, Harvard Business School Press, Boston, MA.

Davis, G.B. (1988) 'Commentary on information systems: to buy, build, or customize?', *Accounting Horizons*, Vol. 2, No. 1, pp. 101-103.

Davis, J. (1998) 'Scooping up vanilla ERP', *InfoWorld*, Vol. 20, No. 47, pp. 1-4.

Gattiker, T.F. and Goodhue, D.L. (2002) 'Software-driven changes to business processes: an empirical study of impacts of Enterprise Resource Planning (ERP) systems at the local level', *International Journal of Production Research*, Vol. 40, No. 18, pp. 4799-4814.

Gattiker, T.F. and Goodhue, D.L. (2005) 'What happens after ERP implementation: understanding the impact of interdependence and differentiation on plant-level outcomes', *MIS Quarterly*, Vol. 29, No. 3, pp. 559-585.

Genovese, Y. (2007) 'Planning for an SAP ERP 6.0 upgrade', *Gartner*, available at http://www.gartner.com/id=557925 (accessed 5 December 2007).

Gould, L. (1997) 'Planning and scheduling today's automotive enterprises', *Automotive Manufacturing and Production*, Vol. 109, No. 4, pp. 62-66.

Haines, M.N. (2003) 'Customization, configuration or modification? A taxonomy for information system specialization', paper presented at the Fourteenth IRMA International Conference, Philadelphia, PA, May 18-21.

Haines, M.N. (2009) 'Understanding enterprise system customization: an exploration of implementation realities and the key influence factors', *Information Systems Management*, Vol. 26, No. 2, pp. 182-198.

Hiquet, B.D. and Kelly, A.F. (1998) *SAP R/3 Implementation Guide: A Manager's Guide to Understanding SAP*, Macmillan Technical, Indianapolis, IN.

Hong, K.-K. and Kim, Y.-G. (2002) 'The critical success factors for ERP implementation: an organizational fit perspective', *Information and Management*, Vol. 40, No. 1, pp. 25-40.

Huang, C.-C. (2005) 'The research on job performance, job satisfaction and organizational commitment in relation with the age factor -- as Kaohsiung City United Hospital for example', Unpublished Master's thesis, I-Shou University, Kaohsiung, Taiwan. (in Chinese)

Levin, R., Mateyaschuk, J. and Stein, T. (1998) 'Faster ERP rollouts', *InformationWeek*, July 13, Vol. 691, p. 24.

Liang, H., Saraf, N., Hu, Q. and Xue, Y. (2007) 'Assimilation of enterprise systems: the effect of institutional pressures and the mediating role of top management', *MIS Quarterly*, Vol. 31, No. 1, pp. 59-87.

Liao, C.-T. (1998) 'ERP implementation -- practice world-class management', *Learning and Development*, Vol. 514, pp. 20-25. (in Chinese)

Lucas, H.C., Walton, E.J. and Ginzberg, M.J. (1988) 'Implementing packaged software', *MIS Quarterly*, Vol. 12, No. 4, pp. 537-549.

Mabert, V.A., Soni, A. and Venkataramanan, M.A. (2000) 'Enterprise resource planning survey of U.S. manufacturing firms', *Production and Inventory Management Journal*, Vol. 41, No. 2, pp. 52-58.

Mabert, V.A., Soni, A. and Venkataramanan, M.A. (2001) 'Enterprise resource planning: measuring value', *Production and Inventory Management Journal*, Vol. 42, Nos. 3/4, pp. 46-51.

Manhasset (2000) 'ERP apps must preserve flexibility', *Internet Week*, May 1, p. 39.

Moad, J. (1995) 'Let customers have it their way', *Datamation*, Vol. 41, No. 6, pp. 34-39.

Ni, R. (2008) 'How to avoid five costly mistakes when procuring software', *Gartner*, available at http://www.gartner.com/id=629210 (accessed 28 March 2008).

Nunnally, J.C. (1978) *Psychometric Theory*, McGraw-Hill, New York, NY.

O'Brien, J.A. (2002) *Management Information Systems: Managing Information Technology in the E-Business Enterprise*, McGraw-Hill, Boston, MA.

Ragowsky, A. and Somers, T.M. (2002) 'Enterprise resource planning', *Journal of Management Information Systems*, Vol. 19, No. 1, pp. 11-15.

Rajagopal, P. (2002) 'An innovation-diffusion view of implementation of enterprise resource planning (ERP) systems and development of a research model', *Information and Management*, Vol. 40, No. 2, pp. 87-114.

Reijers, H.A. (2003) *Design and Control of Workflow Processes: Business Process Management for the Service Industry*, Springer-Verlag, Berlin, Germany.

Rothenberger, M.A. and Srite, M. (2009) 'An investigation of customization in ERP system implementation', *IEEE Transactions on Engineering Management*, Vol. 56, No. 4, pp. 663-676.

Scott, J.E. and Kaindl, L. (2000) 'Enhancing functionality in an enterprise software package', *Information and Management*, Vol. 37, No. 3, pp. 111-122.

Soh, C., Kien, S.S. and Tay-Yap, J. (2000) 'Cultural fits and misfits: is ERP a universal solution?', *Communications of the ACM*, Vol. 43, No. 4, pp. 47-51.

Stephen, P.L. (1999) 'An ERP game plan', *Journal of Business Strategy*, Vol. 20, No. 1, pp. 32-37.

Swan, J., Newell, S. and Robertson, M. (1999) 'The illusion of "best practice" in information systems for operations management', *European Journal of Information Systems*, Vol. 8, No. 4, pp. 284-293.

Wang, Y.-H. and Fei, H.-T. (1999) 'The trend of financial information system in twenty first century', *Accounting Research Monthly*, Vol. 158, pp. 116-122. (in Chinese)

Welti, N. (1999) *Successful SAP R/3 Implementation: Practical Management of ERP Projects*, Addison-Wesley, Boston, MA.

Wu, J.-H., Shin, S.-S. and Wu, C.-C. (2005) 'COTS-Based systems: a methodology for evaluating data and output misfits', *Proceedings of the Thirty-Eighth Hawaii International Conference on System Sciences (HICSS'05)*, Big Island, HI, January 3-6, pp. 1-10.

Zastrocky, M. and Harris, M. (2008) 'Steady-State ERP costs for higher education', *Gartner*, available at http://www.gartner.com/id=696607 (accessed 16 June 2008).

About the authors

Shaio Yan Huang is an Associate Professor, Department of Accounting and Information Technology, National Chung Cheng University. His research areas are auditing, accounting information systems and financial accounting. His papers appeared in *Journal of Computer Information Systems*, *Computer in Human Behavior*, *Total Quality Management and Business Excellence* and elsewhere.

Hsueh-Ju Chen is an Associate Professor, Department of Accounting, National Chung Hsing University. Her research areas are auditing, managerial accounting, enterprise resource planning. Her papers appeared in *Industrial Management and Data Systems*, *International Journal of Accounting Studies* and elsewhere.

An-An Chiu is a Ph.D. Student, Department of Accounting, National Chung Cheng University. Her research areas are auditing, accounting information systems and financial accounting.

Shih-Lung Hsieh is a Master of Accounting Information Technology, National Chung Cheng University. His research areas are accounting information systems and enterprise resource planning.

MIS Review Vol. 17, No. 2, March (2012), pp. 31-57

Revisiting Software Requirements Specifications -- What Could We Learn

Björn Johansson[1], Tanja Rolandsson[2]
[1]*Department of Informatics, School of Economics and Management, Lund University*
[2]*The Swedish Armed Forces*

ABSTRACT: *Software requirements specifications (SRSs) are important documentations that reports results of system requirements determination (SRD) when developing software. It forms a base for subsequent activities in a system development process. In order to increase the knowledge of SRS and how such documentation could be structured we present an analysis of nine SRSs. From the analysis of similarities and differences in composition and requirements organization in the SRSs we aim at giving some advice on how a SRS could be improved and thereby supporting development of information systems better. The analysis shows that the overall structure of the SRSs either follows the IEEE (Institute of Electrical and Electronics Engineers) standard 830 with three main sections (introduction – overview -- list of requirements), or another structure (introduction -- references -- list of requirements). However, how specific requirements then are structured and presented differ from SRS to SRS. The most frequent type of requirements is functional requirements, which is not a big surprise. However, more unpredictable is that non-functional requirements are getting less attention. One conclusion is that even though using standards might not be the only way to formulate SRSs, they are being used and serve their purposes, at least to some extent. However, it can also be concluded that the high focus on functional requirements in standards could be seen as an influential factor explaining why SRSs have such a high focus on functional requirements. The main conclusion is that future SRSs should spend more focus on non-functional requirements since these are both more difficult to describe and will probably play an even more important role when developing information systems in the future.*

KEYWORDS: *Software Requirements Specification, Requirements Engineering, Functional Requirements, Non-functional Requirements, IEEE 830.*

1. Introduction

Information systems development consists of a number of steps. The first one is the system requirements determination (SRD) step. SRD is described by Duggan and Thachenkary (2003) as the overall process of finding, analyzing, and documenting requirements, and includes several activities. It can clearly be stated that the SRD is important in the entire systems development process, since it forms a basis for subsequent activities, affects the design of the system architecture and contributes to the quality of the system (Hull, Jackson & Dick, 2005; Wiktorin, 2003). Hull et al. (2005) describe several reasons for project failure, and amongst those they state that having incomplete requirements is one major reason for failed projects. However, even if projects succeed, insufficient requirements can have several unwanted consequences, such as lower systems quality, that development take longer time, become more expensive than expected as well as unsatisfied users of the developed system (Eriksson, 2007; Jackson, 1995). In addition to the benefit of overcoming unwanted consequences, another benefit of well-formulated requirements is that they can be one of the most important reasons for project success (Hull et al., 2005; Wiktorin, 2003). Apparently, the SRD is of great value to the systems development process, and adequate requirements can be claimed being a necessity for project success.

However, once requirements have been collected, they need to be documented in such a way that they could be used successfully. This documentation usually results in a requirements specification. As part of the SRD, the software requirements specification (SRS) is an important document in the development project (Véras et al., 2010). The SRS can be a channel of communication, conveying the characteristics of the system between developers and users. It can also be part of a contract and as such be used to evaluate performance of systems. Moreover, it can be used to estimate time and cost for the development project (Daniels & Bahill, 2004; Smith, Lai & Khedri, 2007). Even if there are templates available it is not obvious how SRSs should be formulated. For example, the Institute for Electrical and Electronic Engineers (IEEE) has constructed a standard called 830 which lines up a number of attributes SRSs should have in order to be a well-formulated, understandable document (The Institute of Electrical and Electronics Engineers [IEEE], 1998a). This standard is widely mentioned in the requirements engineering literature (Eriksson, 2007; Smith et al., 2007; Wiegers, 1999; Wiktorin, 2003). However, Power (2002) claims that practitioners use a variety of methods, both stylistically and structure-wise, when documenting requirements. In addition, Smith et al. (2007) state that no universally accepted way of documenting requirements exist.

Research on requirements determination (Avison & Fitzgerald, 2006; Cysneiros & do Prado Leite, 2004) has shown that functional requirements, as one certain type of requirements, are specified more clearly and to a larger extent than nonfunctional requirements in requirements specifications. The explanation stated is that nonfunctional requirements are more difficult to identify than functional requirements. From the discussion so far the following questions, that are the focus of this article, can be formulated: (1) How are software requirements specifications structured, and how are requirements in them organized, and (2) What is the most common category of requirements in actual software requirements specifications, and (3) What could we learn from an investigation of real SRSs?

These questions lead to the purpose of the research, which is to gain knowledge on how to compose and structure software requirements specifications (SRSs) in order to be able to develop "better" software in the future. This is done by identifying similarities and differences in SRS composition and requirements organization, illustrating what types of requirements are most frequent, and showing how requirements are formulated.

The next section presents a selection of literature on SRSs and requirements in order to create a theoretical background for the analysis. Method is discussed in Section 3. In Section 4 results from each analyzed SRS are presented, followed by an analysis and discussion of the results. Finally, there are some concluding remarks on what we have learnt from revisiting SRSs.

2. About specifications and requirements

Requirements specifications have been researched before. Besides the research by Power (2002), Franko and Hansson (2006) did an examination of impact from organizational rules when formulating a requirements specification and Dahlstrand, Fredborg and Leandersson (2009) suggested a framework for developing a requirements specification when using a particular systems development technique called Co-design. These last two, however, focus on how to create well-formulated SRSs. This research deals with the actual structure and composition of SRSs. In order to do so, the next section presents information necessary to build a theoretical background for the analysis.

2.1 Fundamentals of software requirements specifications

Software requirements specifications (SRSs) are important documents for systems development, used by different groups of people for different purposes; by customers, to know what to expect, by the software developers, to know what to build and how, by test groups, to test and evaluate the system and so forth (Hull et al., 2005; McIlroy & Stanton, 2011; Wiegers, 1999). The SRS can act as a channel of communication between developers and customers and help to ensure that the system satisfies customer needs (Adisa et al., 2010). Moreover, it creates the baseline upon which following systems development activities are based (Nicolás & Toval, 2009). It is obvious that every system has requirements, and the SRS exists to make these requirements possible to build (Daniels & Bahill, 2004; Hull et al., 2005).

It is quite clear that an SRS is one part of the overall systems requirements determination process which in its turn is part of the entire systems development process. However, this does not explain what an SRS is. Eriksson (2007) describes an SRS as a document produced when a system is built from scratch, or if there are major changes being made to an existing system. That might be useful information, but it is a very brief description. Wiktorin (2003) writes that "a requirements specification consists of several parts." That sounds rather unclear and not of help when trying to understand the concept of requirements specifications. However, it does reveal that there needs to be more than one activity when creating the document. Another description, also rather short, is given by Duggan and Thachenkary (2003): "Requirements specification: representing the results [of the previous steps in the SRD process] in a document." This explains where an SRS comes from, but not what information lies within an SRS.

One explanation of the SRS and its contents is given by Wieringa (1996), who states the following: "A requirements specification consists of a specification of product objectives and a specification of required product behavior." In other words, an SRS shows the purpose of the system, and how it is supposed to behave -- its functionality, which is described by De Carvalho, Johansson and Parthasarathy (2010) in the following way: an SRS should describe the "what" of a system, not the "how." Wiegers (1999) states that since the SRS is important for the following activities in systems development , it needs to have a detailed description of system behavior. Smith et al. (2007) use a similar definition as Wieringa; they state that the SRS should describe essential system requirements of the software and its external interfaces, such as functions, performance, constraints and quality attributes. Another similar description of the SRS is given by IEEE in standard 12207: "the systems requirements specification shall describe: functions and capabilities of the system; business, organizational and user requirements; safety, security, human-factors engineering (ergonomics), interface, operations, and maintenance requirements; design constraints and qualification requirements (The Institute of Electrical and Electronics Engineers [IEEE], 1998b)." To sum up, an SRS is a document created when a system is built or rebuilt, containing purpose and behavior of the system as well as descriptions of the system and its desired functions.

2.2 Recommended contents of requirements specifications

The IEEE (Institute of Electrical and Electronics Engineers) standard 830 called Recommended Practice for Software Requirements Specification is a standard where an outline for a requirements specification structure is given (IEEE, 1998a; Wiktorin, 2003). IEEE 830 is also mentioned recurrently in other literary works (Eriksson, 2007; Smith et al., 2007; Wiegers, 1999; Wiktorin, 2003). According to this standard, requirements specification should contain three sections; introduction, overview and list of requirements.

Wiegers (1999) presents a modified version of the IEEE standard 830, extending the list of requirements into (a) external interface requirements; (b) system features; (c) other non-functional requirements and finally (d) other requirements. Paragraphs (1), introduction and (2), overview are, with a couple of minor modifications, the same as presented by Wiktorin (2003). There are most likely a large number of other templates available as well, but since IEEE 830 has been frequently mentioned, it acts as the base for the analysis in this research.

Wiktorin (2003) as well as Wiegers (1999) suggests that requirements should be organized in different groups. Wiktorin (2003) also states that since functional requirements usually are numerous, it is necessary to have several sub-categories of functional requirements, making interpretation and understanding of requirements easier. Wiegers (1999) points out that it is necessary to give each requirement a unique identifier. The simplest way is to use a sequential number showing both type of requirement and number. Another way is hierarchical numbering, which is claimed by Wiegers (1999) to be the most common way to label requirements.

From a language point of view, requirements can be documented in different ways. Both formal methods, where mathematically formal syntax and semantics is used to describe the requirements, and informal methods, where the requirements are described in natural language exists (Smith et al., 2007). One example of a formal language is the Vienna Development Method Specification Language (VDM-SL). Figure 1 gives one example when such a language has been used to formulate requirements.

```
functions

42.0   wf-first-S-no-sectimetimedev : RS* → 𝔹

  .1   wf-first-S-no-sectimetimedev (s) △
  .2       s (1).sectime = nil ∧
  .3       s (1).timedev = nil
  .4   pre  len s ≥ 1 ;
```

Figure 1 Example of VDM-SL in Use (Droschl, 2000)

The natural language approach, is arguably the most common one in contemporary SRSs (Nicolás & Toval, 2009). Wiegers (1999) seems to assume that requirements will be written in natural language, and gives a number of guidelines for how to formulate requirements, such as “state requirements in a consistent fashion” and “avoid comparative words.” Level of formality in SRSs does not only depend on what the analyst wants to write. It also depends on the complexity of the system. According to Daniels and Bahill (2004) highly complex systems require a higher level of formality.

As stated above, by categorizing requirements readability of SRSs can be improved. Requirements can have very varying characteristics, so categorizing them by dividing them into different types would be a logic way. But first, it is necessary to define what a requirement is.

Machado, Ramos and Fernandes (2005) present the following definitions of the term requirement: "(1) a condition or capability needed by a user to solve a problem or achieve an objective; (2) a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents; (3) a documented representation of a condition or capability as in (1) or (2) (Machado et al., 2005). "

The first and second of these definitions are applicable to requirements in SRSs, and are hereafter used in this research. The third definition appears to be what is considered as an SRS, and does therefore not apply to the term requirement.

Requirements are not all the same kind, which results in that there are many ways to classify them. Eriksson (2007) divides requirements into three broad categories, where the second category has a number of sub-categories. These different types of requirement categories are functional, non-functional (with the sub-categories usability, reliability, performance and supportability) and design restrictions. Wiktorin (2003) divides requirements into functional and non-functional ones. Avison and Fitzgerald (2006) describe a quite similar division: on one hand there are functional requirements, on the other hand non-functional requirements with a number of sub-categories. Another categorization is made by Grady (1992), who describes the FURPS+ model. FURPS is an acronym for Functionality, Usability, Reliability, Performance, Supportability (Grady, 1992), which is the same types as used by Eriksson (2007). The "+" sign was added to the model to "extend the acronym to emphasize various specific attributes (Grady, 1992)." Yet another division is presented in the IEEE standard 830 (IEEE, 1998a). This standard suggests the following groups of requirements: external interfaces, functions, performance requirements, logical database requirements, design constraints and software system attributes. The last category has five sub-categories; reliability, availability, security, maintainability and portability (IEEE, 1998a).

This research will henceforth use the categorization from Eriksson (2007) to describe different types of requirements. One modification has been done though; the division Eriksson (2007) makes on supportability into maintainability and testability is not used as shown in Figure 2.

To summarize this section, we started out by explaining what an SRS is, stating that it is a documentation of requirements when developing an information system or making major changes to a system. The SRS describes the purpose and functionality of the system. After this, we stated that an SRS needs to be structured in some way to simplify

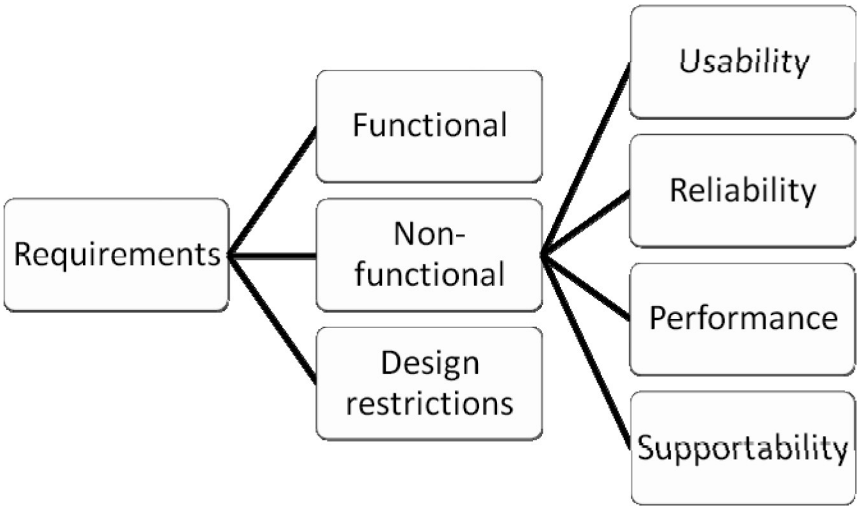

Figure 2 Hierarchy of Requirements Adopted from Eriksson (2007), Sub-categories of Supportability Deleted

the readability, and the IEEE standard 830 was presented as an example, along with an extended version, as presented in Figure 2.

3. Methodological considerations

The data for the investigation of SRSs consists of a collection of real case requirements specifications available online. Selected requirements specifications were found by using a regular Google search; however, the selection followed a rigorous plan as described below. Selected requirements specifications are from software products of different types. They were stated as being the final requirements specification for its specific product. The search was made with the keywords "software requirements specification" and "system requirements specification." However, these terms are not very specific, resulting therefore in many hits. Instead of limiting the search to get a manageable number of hits, the first 25 pages with hits were skimmed and sites which seemed relevant were opened. This resulted in a list of 16 possible candidates. From this list 10 SRSs were selected. The six not selected turned out to be either school projects, too old, or not requirements specifications for software, following the idea of a purposeful approach (Maxwell, 2005) finding a representative sample for the research, and ensuring that the theoretical background is applicable (Eriksson, 2007; Maxwell, 2005). Choosing data available on Internet can be risky and to get reliable data it is necessary to be critical to the data when collecting it. One of the major principles in source criticism is to understand *who* the author of something is and the origin of the data (Svenning, 2003).

To avoid getting unreliable data, only requirements specifications which measured up to a certain standard were collected. The criteria we used were the following three: the SRS needed an origin, it had to be authentic (i.e., no mock-ups), and it had to be written in the last decade (year 2000 or later).

Each requirements specification needed to have a traceable origin, or at least a possibility to find out who the owner was or who had created it. To further establish the reliability and to ensure that it was okay to use the specification, an e-mail was sent to the owner of the document.

The second criterion was related to authenticity. The requirements specifications had to be from software development projects, no school projects were collected. It was easy to identify school projects; they usually had course name, student name and teacher name stated on the first page. The final criterion, that they needed to be written no earlier than year 2000, was easy to check since almost all SRSs found had a date and version number on them. Undated documents were rejected. The result of this process was that the final data set consisted of nine SRSs.

Analysis is generally based on finding patterns and a keyword in qualitative analysis is sorting. In order to be able to analyze it, collected data needs to be sorted. This is of great importance when analyzing texts (Halvorsen, 1992; Svenning, 2003). Another critical point Svenning (2003) makes regarding analyzing texts, is the need for a theoretical basis.

In this research two research questions were raised. The first was to examine how specific SRSs were structured and in what way specific requirements are organized. The second was to see if some type of requirements is more common than other types, and is most common. To answer the first question properly, it is necessary to go back to the fundamentals of SRSs. Then the questions are if the SRSs follow the IEEE 830, or not? If not, how does the structure differ from IEEE 830? Is the language formal or informal? The third paragraph in IEEE 830, list of requirements, can be organized in different ways. This is the most interesting part of question number one. Are requirements categorized by type, importance, or some other way? Are they given unique identifiers?

This leads to question number two. According to Avison and Fitzgerald (2006) as well as Cysneiros and do Prado Leite (2004), functional requirements are more common than non-functional requirements. It could be qustioned if this is consistent with reality. To say something about that, the list of requirements in each SRS was analyzed, different types of requirements identified and then analyzed from the statement that certain types of requirements are more common than others. In order to answer these questions, the outline for analysis presented in Table 1 was constructed.

Table 1 Outline for Analysis and Presentation of Results

Label of SRS <Abbreviated Name>	**Categorization of Requirements: <Description of Categorization> Section 5.3**		
	Structure of SRS: <IEEE 830/Other> Section 5.1	Language: <Natural/ Constructed>/ Identification of Requirements: <Type of Identifier> Section 5.2	Number of Different Categories of Requirements: <Functional: Usability: Reliability: Performance: Supportability: Design Restrictions: > Section 5.4

4. Presentation of the results

This section presents the results from respectively SRS one by one, in alphabetical order by the name of the system the SRS is related to. Each system is briefly presented and some examples of different types of requirements are presented. The results of the questions in the outline for analysis (Table 1) of the SRS are shown in Table 2, which is a summary of each SRS. The results organized by each question in the table are then analyzed and discussed in the section that follows.

4.1 APAF

APAF stands for ASPERA-3 Processing and Archiving Facility. ASPERA-3 is an instrument package for a space mission to Mars and the APAF is a system which is designed to process the telemetry collected by ASPERA-3. The software receives data from ASPERA-3, processes it, distributes it, presents it on a web-display and finally submits it for storage. The SRS, which is version 1.0 is from 2007 and to some extent follows the IEEE 830 standard. What is in Section two -- overview in IEEE 830 -- is here included in the first section. The second section is called “requirements specification description” with different requirements for requirements. Section three is the list of requirements, as in IEEE 830. Following is one section called “notes.” Some examples of functional requirements are: “*APAF-FR-02 The APAF system shall process all ASPERA-3 science data into IDFS data sets*,” and “*APAF-FR-07 Web-based displays of the most current ASPERA-3 data shall be provided for public view*,” and an example of supportability requirement is: “*APAF-LR-02SwRI shall provide software support for the APAF system*.”

4.2 CTBTO_WMO_SEA

CTBTO_WMO_SEA (Comprehensive Nuclear Test-Ban Treaty Organization World Meteorological Organization Special Event Analysis) is a software package designed

Table 2 Presentation of the Analysis of the Nine SRSs

SRS Name	Categorization of Requirements		
	Structure of SRS	Language / Identification of Requirements	Number of Different Categories of Requirements
APAF	(1) Capability/Functional Requirements; (2) External Interface Requirements; (3) Internal Interface Requirements; (4) Internal Data Requirements; (5) Security & Privacy Requirements; (6) Computer Resource Requirements; (7) Logistics-Related Requirements; (8) Delivery Requirements; (9) Other Requirements Considered.		
	(1) Scope; (2) Requirements Specification Descriptions; (3) Requirements; (4) Notes.	Natural/Letter-Number Combinations	Functional: 28 Usability: 0 Reliability: 0 Performance: 0 Supportability: 2 Design Restrictions: 4
CTBTO_WMO_SEA	(1) Functional Requirements; (2) Acceptance Testing Requirements; (3) Documentation Requirements; (4) Security Requirements; (5) Portability Requirements; (6) Performance Requirements.		
	(1) Scope; (2) References; (3) Functional Requirements; (4) Acceptance Testing Requirements; (5) Documentation Requirements; (6) Security Requirements; (7) Portability Requirements; (8) Performance Requirements; (9) Terminology (Glossary, Abbreviations and Appendices).	Natural/Hierarchical Numbers	Functional: 92 Usability: 6 Reliability: 0 Performance: 5 Supportability: 3 Design Restrictions: 2

Table 2 Presentation of the Analysis of the Nine SRSs (continued)

SRS Name	Categorization of Requirements		
EVLA CB	(1) External Interface Requirements; (2) Performance Requirements; (3) Reliability/Availability; (4) Serviceability; (5) Maintainability; (6) Scalability, Security; (7) Installation & Upgrades; (8) Documentation.		
	IEEE 830: (1) Introduction; (2) Overall Description; (3) Specific Requirements.	Natural/Hierarchical Numbers	Functional: 66 Usability: 2 Reliability: 8 Performance: 12 Supportability: 13 Design Restrictions: 0
I-15 RLCS	(1) External Interface Requirements; (2) Functional Requirements; (3) Performance; (4) Logical Database Requirements; (5) Design Constraints; (6) RLCS Application Software Attributes.		
	IEEE 830: (1) Introduction; (2) Overall Description; (3) Specific Requirements.	Natural/Hierarchical Numbers	Functional: 85 Usability: 0 Reliability: 10 Performance: 20 Supportability: 3 Design Restrictions: 6
MDOT VII DUAP	(1) Input Services; (2) Administrative Services; (3) Dynamic Data Services; (4) Computational Services; (5) Persistent Data Services; (6) Output Services; (7) Presentation Services; (8) Design Constraints; (9) Quality Characteristics; (10) External Services.		
	IEEE 830: (1) Introduction; (2) Overall Description; (3) Specific Requirements.	Natural/Letter-Number Combinations & Level of Priority (Low, Medium, High)	Functional: 92 Usability: 0 Reliability: 0 Performance: 0 Supportability: 4 Design Restrictions: 6

Table 2 Presentation of the Analysis of the Nine SRSs (continued)

SRS Name	Categorization of Requirements		
NPOESS DE	(1) Required States and Modes; (2) Capability Requirements; (3) External Interface Requirements; (4) Internal Interface Requirements; (5) Internal Data Requirements; (6) Adaption Requirements; (7) Security and Privacy Requirements; (8) Computer Resource Requirements; (9) Operator-Related Requirements; (10) Other Requirements.		
	(1) Scope; (2) Referenced Documents; (3) Requirements; (4) Requirements Traceability (Appendices).	Natural/Hierarchical Numbers	Functional: 88 Usability: 0 Reliability: 5 Performance: 13 Supportability: 3 Design Restrictions: 4
OSSAFFCM	(1) Required States and Modes; (2) CSCI Capability Requirements; (3) CSCI External Interface Requirements; (4) CSCI Internal Interface Requirements; (5) CSCI Internal Data Requirements; (6) Adaption Requirements; (7) Safety Requirements; (8) Security and Privacy Requirements; (9) CSCI Environment Requirements; (10) Computer Resource Requirements; (11) Software Quality Factors; (12) Design and Implementation Constraints; (13) Personnel Requirements; (14) Training-Related Requirements; (15) Logistics-Related Requirements; (16) Other Requirements; (17) Packaging Requirements; (18) Precedence and Criticality Requirements.		
	(1) Scope; (2) Referenced Documents; (3) Requirements; (4) Qualification Provisions; (5) Requirements Traceability; (6) Notes.	Natural/Hierarchical Numbers	Functional: 19 Usability: 7 Reliability: 1 Performance: 0 Supportability: 5 Design Restrictions: 10

Table 2 Presentation of the Analysis of the Nine SRSs (continued)

SRS Name	Categorization of Requirements		
SRS2XE SAMPLE FLAG	(1) Functional Requirements; (2) Acceptance Testing Requirements; (3) Documentation Requirements; (4) Security Requirements; (5) Portability Requirements; (6) Performance Requirements.		
	(1) Scope; (2) References; (3) Functional Requirements; (4) Acceptance Testing Requirements; (5) Documentation Requirements; (6) Security Requirements; (7) Portability Requirements; (8) Performance Requirements; (9) Terminology (Glossary, Abbreviations and Appendices).	Natural/Hierarchical Numbers	Functional: 49 Usability: 6 Reliability: 0 Performance: 3 Supportability: 4 Design Restrictions: 3
STEWARDS	(1) System Features; (2) External Interface Requirements; (3) Other Non-functional Requirements.		
	IEEE 830: (1) Introduction; (2) Overall Description; (3) Specific Requirements.	Natural/Letter-Number Combinations	Functional: 85 Usability: 1 Reliability: 1 Performance: 2 Supportability: 2 Design Restrictions: 9

to (1) initialize and forward calculations from a larger system and (2) provide a web-based interface for collecting, comparing, source locating and reporting data received from meteorological centers. The SRS with version no. 1.0 is from 2009, and follow an organization-specific template. Section one, introduction, is quite similar to IEEE 830. Section two is not an overview of the system, but a list of references to related documents. Requirements are not collected under one section, instead different types of requirements have their own headings. Examples of functional requirements are: "*1.3.2 The software shall allow for post processing of the virtual RN station measurements*," "*2.2.2 The software shall monitor the automated collection of the data received in response to the request*," and "*2.2.4 The web tool shall be capable to generate WMO Centre comparison statistics: Consolidation of auto-reporting on the web page of the exercise*, *including plot generation*, *and integration of the standard display of MMFORs inter-comparison statistics*." One example of a performance requirement is: "*18.4 A full daily update of the reporting web page shall not take longer than 12 hours per 10 x 10 SRS data request examined*." And an example of design restriction is: "*3.1 The CTBTO_WMO_SEA software shall run under UNIX/LINUX while making use of the native C and FORTRAN compiler package gcc running at the PTS*."

4.3 EVLA CB

This SRS describes requirements for a system which is an in-between system and the main component of a data processing pipeline in a system called EVLA (the VLA Expansion Project). The CB stands for Correlator Backend. The system is supposed to receive, assemble, format, process and finally deliver data in a suitable way. Data is sent to EVLA CB from a monitor and control system, and after processing it the EVLA CB delivers it to an end-to-end system. In the SRS it is stated that it follows IEEE 830. The SRS with the version number 2.0 is from 2002. The following examples of requirements is typical examples of functional requirements: "*3.2.1.1 Monitor and Control System -- The BE shall acknowledge receipt of all data received from M&C*," "*3.2.2.7 Data Invalid -- The BE shall replace all invalid data with zero values*," "*3.2.2.31 Reboot network -- The BE shall be able to initiate a reboot of any internal network*," and a typical example of performance requirement is: "*3.3.2.1 Input -- The BE System shall be capable of accepting an aggregate data input stream from the Correlator of a minimum om 1.6 Gbytes/sec*. *This must be done simultaneously with the output stream*, *but not necessarily over the same interconnects*. *This is an initial deployment specification and will be increased over time*," an example of supportability requirement is: "*3.6.1 Software tools -- software tools and pre-built applications that do not have source code available shall come with a complete diagnostic package and customer support*."

4.4 I-15 RLCS

The Interstate 15 Reversible Lane Control System (I-15 RLCS) is designed to control opening and closing of reversible lanes on the Interstate 15. It was developed since the previous system was getting old and was impossible to extend. I-15 RLCS also provides a graphical user interface, process control and monitoring, sequencing, data processing and security as well as reporting. The SRS is from 2004; however, it does not contain any information about versions. Examples of functional requirements are: "*3.2.2.1 The RCLS shall monitor all field device sensors*, *and shall process operator requests for changing field device status*," and "*3.2.2.6.1 The RCLS software shall initialize each control unit and device sensor as it is identified*." Examples of performance requirements are: "*3.3.1.1 The external server data store containing RLCS status for use by external systems shall be updated once per minute*," and "*3.3.4.1 At a minimum of every 60 seconds*, *the system shall check the current date and time against a list of scheduled events for the current mode to determine if any event should be executed*." Examples of design restrictions are: "*3.5.1.1 The data processing and security*, *and reporting functions of the RLCS application software shall be implemented with commercial off-the-shelf software*," and "*3.5.2.3 The MD5 algorithm shall be used to secure application data and software in the controllers and the application servers*."

4.5 MDOT VII DUAP

MDOT VII DUAP is an abbreviation of Michigan Department of Transportation's Vehicle Infrastructure Integration Data Use Analysis and Processing system. The system is a research program, designed to examine the impact on traffic operations, asset management and transportations planning by new VII data. DUAP is supposed to collect, convert and communicate data to different people to support MDOT. The SRS investigated is version 1.02 from 2007. Examples of functional requirements are: "*IS-010 The System shall collect probe vehicle data*. *H*," "*IS-070-003 The DUAP data elements shall include roadway event information data fields corresponding to the SAE J2354 structure as enumerated un APPENDIX F -- SAE J2354 Event Information Elements*. *H*," and "*AS-140 The DUAP System shall be able to organize the sequence of execution of computational modes*." One example of supportability requirement is: "*The DUAP System shall be capable of adding new data sources*. *L*," and examples of design constraints are: "*DC- 010-010 The DUAP System shall use a Java software foundation*. *M*," and "*DC-040 The DUAP System shall use Michigan Geographic Framework geo-references*. *H*." Most of the requirements in this SRS also have a source (another document) and some of the requirements were commented.

4.6 NPOESS DE

The NPOESS (National Polar-Orbiting Operational Environmental Satellite System) Data Exploitation (DE) is a system that aims at distributing data from NPOESS observations to civilian customers and operational and climate communities. It receives data, process and packages and finally delivers the data to the right people. It is also supposed to be part of customer service. The SRS is from 2007 and marked with version 1.0. Sections one and three are similar to IEEE 830. Section two, which in IEEE 830 is called overview, is sort of included in the first section and instead Section two is called referenced documents. After Section three which lists requirements, there is a fourth section called requirements traceability. Examples of functional requirements are: "*3.2.1 The System shall be capable of defining Data Products for Ingest*," and "*3.3.1 The System shall be capable of receiving data and products from IDPS*." One example of performance requirement is: "*3.4.3 The system shall be capable of executing 99 % of its scheduled tasks in any consecutive 30 day period*" and a example of supportability requirement is: "*3.6.2 The System shall be capable of adding additional capacity without redesign of its infrastructure*." Example of design restriction: "*3.8.2.2 The System shall be constructed using COTS and Open Source software where it is possible, practical and approved by the Government*."

4.7 OSSAFFCM

OSSAFFCM is an abbreviation of Open Source Sustainability Assessment Framework Format Converter Module. This software is, as the name reveals, a format converter module in an open source framework. It is designed to calculate sustainability for different products and thus see environmental impact of the product. The purpose of the format converter is to convert data formats from one of four types into another without any loss of data. The SRS has version no. 1.1 and is from 2007. What is included in IEEE 830 Section two, "overview," is included in the first section. The second section contains a list of related documents. Section three lists requirements, similar to IEEE 830. After that follows three sections, called "qualification provisions," "requirements traceability" and "notes." Examples of functional requirements are: "*3.2.1.6 The inventory calculation will be able to cope with loops in the product system*," and "*3.2.4 The converter will allow conversion between important LCI (Life Cycle Inventory) data formats*." Examples of usability requirements are: "*3.12 The software will be documented in English*," and "*3.13.3 The software will be designed to be used by people interested in Life Cycle Assessment and sustainability assessment*." Example of design restriction is: "*3.9 The Framework and the converter will be designed to run in Windows environments (Win 2000, Win XP, Win Vista), Macintosh, and Linux OS. A Java Virtual Machine (JVM) of version 5 or higher needs to be installed. A MySQL database needs to be installed as well. Both JVM and MySQL will be made available on the website*."

4.8 SRS2XESAMPLEFLAG

The SRS (Source-Receptor Sensitivity) 2XESAMPLEFLAG is an extension to a web-based metrological analysis tool. When a known emitter releases xenon which might affect a xenon sample, the software is designed to flag this sample. SRS2XESAMPLEFLAG is supposed to run on any Unix-based system. The SRS from 2009 has version no. 1.0. The structure follows an organization-specific template (Same as CTBTO_WMO_SEA). The introduction, Section one, is fairly similar to IEEE 830. Section two is not an overview of the system, but a list of references. The requirements are not collected under one section, instead different types of requirements have their own headings. Examples of functional requirements are: "*1.6 The software shall be capable to choose between those available ATM (Atmospheric Transport Modelling) models that were utilized to generate the SRS (Source Receptor Sensitivity) data pertaining to the sample. Further requirements are specified in requirements 5.x*," and "*3.1 The software shall be capable to parse the information of the emissions from known xenon sources via a xenon sources inventory input file, parsed by the command line option -e<xenon-emissions-file>*." Examples of usability requirements are: "*14. All documentation will be written in English in MS Word format*," and "*20. There shall be a help function implemented that is called with the '–help' option and upon any users' choice of options that is non-compliant with the required syntax*." Example of a performance requirement is: "*25. Processing of a 21 days SRS field and preparation of the data shall not last longer than 5 minutes per sample if the full 21 days release (emission) period is examined*."

4.9 STEWARDS

STEWARDS (Sustaining the Earth's Watersheds -- Agricultural Research Data System) is a system designed to manage data related to water, soil, management and economics. It was developed to be part of the Conservation Effects Assessment Project (CEAP) which assesses the environmental effects of a conservation program implementation. The SRS with version no. 1.1 is from 2006. Examples of functional requirements are: "*FR-2.3: Browse, query, and download individual sampling station data and metadata. Provide access to the data via browsing of sites, stations, and instruments; allow for simple queries to individual datasets; provide a metadata search tool to query dataset parameters; and allow for downloading of datasets (full or partial)*," and "*FR-2.5: Generate tabular reports of selected data. Provide access to CEAP-related reports, tables and project documents*." Example of a performance requirement is: "*PR-2: Loading speed: The data system shall load as quickly as comparable productivity tools on whatever environment it is running on*," and a example of supportability requirement is: "*SQ-1: Portability: This database will be built for a particular system and may not be portable but results to queries will be portable between many environments*." Example of design restriction: "*SR-5: Relational Database Management System -- As the primary data storage mechanism for the corporate standard relational database management system, Microsoft SQL Server will be required to support system functionality*."

5. Analysis and discussion

In this section we discuss the above presented results. It follows the order of the presented questions in the outline for analysis as presented in Section two.

5.1 The structure of the different SRSs

Four of nine SRSs followed IEEE 830 -- EVLA CB, I15 RCLS, MDOT VII DUAP and STEWARDS, and except from MDOT VII DUAP, this was also stated in the SRS. Of the remaining five, four of the SRSs -- CTBTO_WMO_SEA, NPOESS DE, OSSAFFCM and SRS2XESAMPLEFLAG -- were structured in a fairly similar way. These four all started with an introductory section called scope, continued with references or referenced documents in Section two, and the requirements began in Section three. In NPOESS DE it was stated that it was based on the IEEE standard 12207. However, in that standard no given structure is described, so it must have been used in some other way than following a structure. In the SRS, it was also stated that the requirements management tool DOORS was used to create the SRS. NPOESS DE was the only SRS where it was stated that some kind of tool had been used. The last SRS -- APAF -- was structured almost as the other four in the non-IEEE 830-group, with the exception that Section two was called "requirements specification descriptions." For the SRSs who did not follow IEEE 830, including APAF, all had system overview and document overview in the first section. This is fairly similar to the sub-headings "system type" and "overview of the subsequent parts of the specification" in IEEE 830. Beyond these two subheadings, all SRSs, both the ones following IEEE 830 and the rest, had other subheadings as well in the first section. These were things like project identification, rationale or definitions.

Another similarity between all SRSs is that they had the specific requirements starting in Section three. The CTBTO_WMO_SEA and SRS2XESAMPLEFLAG turned out to have the same author, which explains why they are structured similarly. These SRSs follow an organization-specific template, called DSTD, which is stated in the document overview-section of the SRSs. In no other SRSs such information was given.

Biggest differences in the outlines of the SRSs were within the second section and how requirements were divided into groups. The second section of the SRSs was, as stated above, either "overview," as in IEEE 830, or a section with references to other documents. APAF, which stood out a bit from the others, had a second section that looked nothing like any of the other ones. It was called "requirements specification description," an explanation of necessary attributes for requirements. These attributes were for example that each requirement should have a unique identifier, be necessary for the system or formulated in a clear and concise way. It can be claimed that such statements feel a bit superfluous and belong in the requirements engineering literature and not in an SRS. Others seem to agree on that, especially since no other SRSs had such explanations of the nature of requirements.

So, except from APAF, the SRSs appeared to follow either IEEE 830, starting with an introduction, followed by a system overview and then the actual requirements, or another, unnamed structure, starting with scope, followed by a list of referenced documents, and then the actual requirements.

5.2 Language and identifier

All SRSs and requirements in them are written in an informal language, which is consistent with the claim made in chapter two, that contemporary SRSs use natural, informal language. Maybe the use of informal language also is because the systems are not extremely large or complex. Some of them were just small extensions to existing systems, for example the SRS2XESAMPLEFLAG or the OSSAFFCM. These "smaller" systems also had a fairly low number of requirements.

All requirements had unique identifiers, that in most cases were hierarchically numbered, but in APAF, MDOT VII DUAP and STEWARDS there were letter-number combinations. This is consistent with the statement both Wiegers (1999) and Eriksson (2007) makes; that hierarchical numbering is the conventional way to label requirements. No other types of labeling were present in the SRSs analyzed.

5.3 Categorization of requirements

In all cases, requirements were divided into groups depending on character of the requirements. These groups were not the same as the ones in chapter two; instead the requirements were grouped differently in each SRS, with the exception of CTBTO_WMO_SEA and SRS2XESAMPLEFLAG, who, as mentioned above, have the same outline for the entire SRS. Except for MDOT VII DUAP, one group of requirements was named almost the same in all SRSs: capability/functional requirements. Another group called external interface requirements was also very common. This group is mentioned by Wiegers (1999) as the first group of requirements in the extended IEEE 830, and appears in six of the nine SRSs. A third group of requirements was also present in six of the nine SRSs, namely security and privacy requirements. Table 2 presents how requirements were structured in the SRSs. In most SRSs, requirements were divided into 6-10 groups. It would seem logical if systems with lots of requirements had a larger number of groups;

however, this is not the case. Interesting to note is that, for example, OSSAFFCM with a fairly low number of requirements had the largest number of groups of requirements (18), while I-15 RLCS, with a great number of requirements, had a much smaller number of groups of requirements (6).

The MDOT VII DUAP varies from the others since the naming of the groups of requirements was done much differently. Most categories were called "services" of some kind, such as input services, administrative services and computational service. Also, the major part of requirements in this SRS had a reference document and some of them had comments to help the interpretation of them.

When analyzing the SRSs, it was quite difficult to identify type of requirement. Especially the non-functional ones were tricky to categorize into Eriksson's (2007) categories. None of the SRSs used the same categorization as Eriksson (2007), which made it necessary to categorize requirements from the category given in the SRS to the corresponding category in the outline for analysis. However, in most cases the name of the group of requirements corresponded to one of the FURPS+ components or one of its keywords. For example, EVLA CB had groups of requirements called "serviceability" and "maintainability," which both are FURPS+ keywords for supportability. Functional requirements were sometimes also necessary to identify. The category "security and privacy requirements" which appeared in six of the nine SRSs were usually stated under its own heading. Since "security" is one of the FURPS+ keywords for functionality and requirements usually followed a noun-verb construction such as "the System shall be capable of generating backups for all NDE data, procedures, and software (requirement 3.7.2 in NPOESS DE)," they were counted as functional requirements. However, in STEWARDS, the category "security requirements" is listed as a subcategory of "other non-functional requirements." When looking at specific requirements, there was for example one that looked like this: "*SCR-4: Availability The fourth consideration for security requirements is availability. The system must be available to the intended audience 24 hours per day, 7 days a week with, 99% availability and a tolerance of -5% (not less than 50% of working hours in any week). For this system, availability will be concerned with the reliability of the software and network components. Intentional "denial of service attacks" is not foreseen as a significant concern* (STEWARDS, p. 13)." This could be seen as a non-functional requirement, which in the categorization belongs to the group "reliability." This example shows that the categorization of requirements can differ a lot.

To summarize this, it can be concluded that none of the SRS had the same structure of requirements as described in the theoretical background in Section two. Instead, all SRSs had individual organizations of requirements, except from CTBTO_WMO_SEA and SRS2XESAMPLEFLAG who had the same organization of requirements, probably because they have the same author. The great variety of ways to organize the requirements into groups made that a re-categorization was done to fit them in the outline for analysis and be able to compare results, especially in the counting of the requirements.

5.4 Number of different categories of requirements

Functional requirements were more frequent than non-functional ones and design requirements, which confirms statements made earlier. It also supports the statement that non-functional requirements are difficult to evaluate and measure, since they involve so many aspects. Another claim is that functional requirements are specified clearer. In some cases, the functional requirements were extremely detailed, while there were only a few non-functional requirements in the same document. This was particularly noticeable in CTBTO_WMO_SEA, where functional requirements were very detailed with a great number of sub-requirements: "*1.2.1.4.1.1.1 For the nuclide it shall be possible to enter the nuclide symbol name explicitly (e.g. Xe-133). The name will be translated into the nuclide specific half-life time (via a lookup table offered by the commission). If 'Tracer' is given, a zero half-life time shall be applied (indicated by a '999.9' in the configuration file 'SPECIES')*" while a non-functional requirement or design restriction could be as short as this: "*12 Source code will be documented in the code according to suitable standards*." However, these two requirements are, especially the latter one, rather extreme and should not be viewed as facts, only as examples or end-points of a scale on how detailed requirements can be. Most of the time, functional and non-functional requirements seemed to have been given the same level of detail.

Eriksson (2007) divided non-functional requirements into four groups; usability, reliability, performance and supportability. Of these four groups, performance requirements were most frequent. In four of the SRSs, performance requirements were labeled as an own group of requirements. This was the only group of non-functional requirements that was consistent with Eriksson's classification. As discussed above, the division of requirements into different groups in each of the SRSs seemed to follow individual templates.

Design restrictions requirements were not very common. One reason for that could be that they are formulated to a high extent in other documents. For example, in STEWARDS there is a reference to a site called "USDA Web Style Guide," MDOT VII DUAP has a reference to a document called "Systems Engineering Methodology version 1.0" and in the SRS for EVLA CB there is a reference to a document called "EVLA Correlator Monitor and Control System, Test Software and Backend Software Requirements and Design Concepts."

Regarding non-functional requirements, the most frequent were performance requirements. Maybe this is because performance requirements often are expressed with time and thereby the easiest one to measure. One example on this is the requirement 3.10.2.3 from NPOESS DE: "*The System shall deliver A-DCS telemetry data from IDPS to the US Global Processing Center within 1minute of their receipt*." Comparing this to a requirement from OSSAFFCM: "*3.11.2 The software will be easily learned and used. This usability will be supported by documentation accompanying the software*." It is clear that that is hard to measure. As Avison and Fitzgerald (2006) state, nonfunctional requirements can be hard to evaluate, especially ones that not are about time.

Wrapping this up, functional requirements were more common than non-functional ones and design restrictions. This confirms statements about functional requirements and that they are the most frequent ones in SRSs. Sometimes functional requirements were more detailed than non-functional ones, but this was only apparent in special cases and not something consistent. According to the results, it is not true that functional requirements are described clearer or in more detail than non-functional requirements or design restrictions.

6. Conclusions

The purpose of this research was to identify similarities and differences in SRS composition. From the analyses of nine SRSs it can be concluded that there seems to be two major types of structures for SRSs. They either follow IEEE 830 with its introduction -- overview -- list of requirements structure, or have an outline which is introduction -- references -- list of requirements. There was no similar pattern regarding the organization of the actual requirements. With the exception of CTBTO_WMO_SEA and SRS2XESAMPLEFLAG who have the same origin and were structured the exact same way, no other SRSs had a similar way to organize requirements. The only similarity in organization of requirements was that in all cases, requirements were divided into groups, and they had unique identifiers. Three groups of requirements were frequently used: functional/capability requirements, external interface requirements and security/privacy requirements. The differences in SRS composition are, however, not very extreme. This finding is not consistent with the results reported by Power (2002), who claims that SRS structures vary a lot.

Looking at specific requirements, functional requirements outnumbered the other categories by large. Even combined, there were less non-functional requirements than functional ones. This does confirm the claim that it might have been better to follow Wiktorin (2003) suggestions dividing functional requirements into groups. However, there were few requirements in the SRSs that clearly could be regarded as functional requirements. Since the functional requirements category thus became much broader than the different categories of non-functional requirements, it was not very surprising that there were many more functional requirements.

It can also be concluded that number of requirements is not a reason for the choice of structure of SRSs. For example, the OSSAFFCM which had the largest number of groups of requirements had a very small number of actual requirements. It can be stated that it seems unnecessary to have so many groups of requirements when in some cases there was only one or two requirements in each category, but the reason for this is not clear. A possible suggestion for further research would then be to pick out a couple of SRSs and contact the originators to get the answer on such questions.

It can be stated that following standards, like the IEEE 830, simplifies documentation of requirements. If every requirements engineer documented requirements in his/her own way, the development team would have to put a lot of time in interpreting the SRS for every new project. This is of course why standards are developed in the first place, and it can be concluded that standards are useful, as long as they are applicable to the organizational context and allow modifications. Another benefit of using standards is that they do not need to be organization-specific. If the development team works independently and is not bound to one specific company, they can use the same template for different projects in different organizations.

Regarding the specific groups of requirements, it might be possible to use templates as well. The literature showed a number of different ways to categorize requirements (Avison & Fitzgerald, 2006; Eriksson, 2007; IEEE, 1998a; Wiktorin, 2003), and the results in this research showed a great variety in the organization of specific requirements. The main conclusion from the research on SRSs is that a more general classification might be useful, particularly for independent development teams, or when development teams from different companies cooperate in projects. However, it can also be concluded that a classification should focus more on non-functional requirements and thereby decrease the risk of having a too high focus on functional requirements in future systems development processes.

Acknowledgements

We would like to express our thanks to the anonymous reviewers and Professor She-I Chang, whom acted as guest editor, for their comments and suggestions during the process of finalizing this article. This paper is a revised and expanded version of a paper entitled “Revisiting Software Requirements Specifications -- What Could We Learn” presented at

2010 International Conference on Accounting and Information Technology (ICAIT) July 5-7, 2010, Chiayi, Taiwan.

References

Adisa, F., Schubert, P., Sudzina, F. and Johansson, B. (2010) 'Living requirements space: an open access tool for enterprise resource planning systems requirements gathering', *Online Information Review*, Vol. 34, No. 4, pp. 540-564.

Avison, D. and Fitzgerald, G. (2006) *Information Systems Development: Methodologies, Techniques & Tools*, McGraw-Hill, Berkshire, UK.

Cysneiros, L.M. and do Prado Leite, J.C.S. (2004) 'Nonfunctional requirements: from elicitation to conceptual models', *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 328-350.

Dahlstrand, M., Fredborg, H. and Leandersson, S. (2009) 'Co-design av co-design -- förslag på riktlinjer för arbetssättet och utformningen av kravspecifikationer', Unpublished Bachelor's thesis, University of Borås, Västra Götaland, Sweden.

Daniels, J. and Bahill, T. (2004) 'The hybrid process that combines traditional requirements and use cases', *Systems Engineering*, Vol. 7, No. 4, pp. 303-319.

De Carvalho, R.A., Johansson, B. and Parthasarathy, S. (2010) 'Software tools for requirements management in an ERP system context', *Journal of Software Engineering and Technology*, Vol. 2, No. 2, pp. 101-106.

Droschl, G. (2000) 'Formal specification of a security system module in VDM-SL', Technical report, Institute for Software Technology, Technical University of Graz, Graz, Austria.

Duggan, E.W. and Thachenkary, C.S. (2003) 'Higher quality requirements: supporting joint application development with the nominal group technique', *Information Technology and Management*, Vol. 4, No. 4, pp. 391-408.

Eriksson, U. (2007) *Kravhantering för IT-System*, Studentlitteratur, Malmö, Sweden.

Franko, S. and Hansson, M. (2006) 'Verksamhetsregler vid utformning av kravspecifikation', Unpublished Bachelor's thesis, Lund University, Scania, Sweden.

Grady, R.B. (1992) *Practical Software Metrics for Project Management and Process Improvement*, Prentice Hall, Upper Saddle River, NJ.

Halvorsen, K. (1992) *Samhällsvetenskaplig Metod*, Studentlitteratur, Lund, Sweden.

Hull, E., Jackson, K. and Dick, J. (2005) *Requirements Engineering*, Springer, London, UK.

Jackson, M. (1995) *Software Requirements & Specifications: A Lexicon of Practice*, *Principles and Prejudices*, ACM Press, London, UK.

Machado, R.J., Ramos, I. and Fernandes, J.M. (2005) 'Specification of requirements models', in Aurum, A. and Wohlin, C. (Eds.), *Engineering and Managing Software Requirements*, Springer, Berlin, Germany, pp. 47-68.

Maxwell, J.A. (2005) *Qualitative Research Design: An Interactive Approach*, Sage Publications, Inc., Thousand Oaks, CA.

McIlroy, R.C. and Stanton, N.A. (2011) 'Specifying the requirements for requirements specification: the case for work domain and worker competencies analyses', *Theoretical Issues in Ergonomics Science*, doi: 10.1080/1463922X.2010.539287

Nicolás, J. and Toval, A. (2009) 'On the generation of requirements specifications from software engineering models: a systematic literature review', *Information and Software Technology*, Vol. 51, No. 9, pp. 1291-1307.

Power, N.M. (2002) 'A grounded theory of requirements documentation in the practice of software development', Unpublished Ph.D. dissertation, Dublin City University, Dublin, Ireland.

Smith, S., Lai, L. and Khedri, R. (2007) 'Requirements analysis for engineering computation: a systematic approach for improving reliability', *Reliable Computing*, Vol. 13, No. 1, pp. 83-107.

Svenning, C. (2003) *Metodboken* (5th ed.), Lorentz Förlag, Eslöv, Sweden.

The Institute of Electrical and Electronics Engineers (1998a) 'IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements Specifications', Technical report, IEEE, New York, NY.

The Institute of Electrical and Electronics Engineers (1998b) 'IEEE/EIA Standard 12207.0-1996: (ISO/IEC 12207) Standard for Information Technology -- Software Life Cycle Processes', Technical report, IEEE, New York, NY.

Véras, P.C., Villani, E., Ambrósio, A.M., Pontes, R.P., Vieira, M. and Madeira, H. (2010) 'Benchmarking software requirements documentation for space application', in Schoitsch, E. (Ed.), *Computer Safety*, *Reliability*, *and Security*, Springer, Berlin, Germany, pp. 112-125.

Wiegers, K.E. (1999) *Software Requirements*, Microsoft Press, Redmond, WA.

Wieringa, R.J. (1996) *Requirements Engineering: Frameworks for Understanding*, John Wiley & Sons, Chichester, UK.

Wiktorin, L. (2003) *Systemutveckling på 2000-Talet*, Studentlitteratur, Lund, Sweden.

About the authors

Björn Johansson holds a Ph.D. in Information Systems Development from the Department of Management and Engineering at Linköping University. Currently he works as Associate Senior Lecturer at Department of Informatics, Lund University. Previously he worked as a Post Doc. at the Center for Applied ICT at Copenhagen Business School. He is a member of the IFIP Working Groups IFIP 8.6 and IFIP 8.9.

Tanja Rolandsson received her B.Sc. in Informatics and B.A. in English Linguistics from Lund University in 2010. She is currently working for the Swedish Armed Forces as a systems engineer at the Armed Forces Command and Control Regiment with development, implementation and administration of several different weather information systems.

Appendix 1: SRS Sources

All websites visited January 3, 2010.

1. ASPERA-3 Processing and Archiving Facility
   http://www.aspera-3.org/idfs/APAF_SRS_V1.0.pdf

2. CTBTO_WMO_SEA
   http://www.ctbto.org/fileadmin/user_upload/procurement/2008/RFP2009-0339-CTBTO_WMO_SEA_Software_Requirements_Specification-DISCHENDORF.pdf

3. EVLA Correlator Backend
   http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/Corr_bkend_Req_Soft.pdf

4. I 15 Reversible Lane Control System
   http://www.dot.ca.gov/dist11/operations/I15RLCS/RFP_DOT2040_SecVID_Apr21_2004.pdf

5. MDOT Vehicle Infrastructure Integration Data Use Analysis and Processing (VII DUAP)
   http://www.michigan.gov/documents/mdot/MDOT_DUAP_SysReq_Final_220099_7.pdf

6. NPOESS Data Exploitation
   http://projects.osd.noaa.gov/NDE/pub-docs/SystemRequirementsDoc.pdf

7. OSSAFFCM Open Source Sustainability Assessment Framework, Format Converter Module
   http://www.openlca.org/uploads/media/Software_Requirements_v1.1_3Feb07.pdf

8. SRS2xESampleFlag
   http://www.ctbto.org/fileadmin/user_upload/procurement/2008/RFP2009-0337-SRS2XeSampleFlag_Software_Requirements_Specification-DISCHENDORF.pdf

9. STEWARDS
   ftp://ftp-fc.sc.egov.usda.gov/NHQ/nri/ceap/stewardssystemdesign030206.pdf

MIS Review Vol. 17, No. 2, March (2012), pp. 59-85

From CAATTs Adoption to Continuous Auditing Systems Implementation: An Analysis Based on Organizational Routines Theories

Chia-Ming Sun
Department of Accounting, Yunlin University of Science and Technology

ABSTRACT: *Previous studies on the use of computer assisted auditing technology and techniques (CAATTs) mostly focus on computer skills and perceived usefulness of auditors, and the influential factors from organizational environments; however, they seldom emphasize the group level. In this research, we study the technological adaptation process of a case company, which continuously implemented four CAATTs projects in three years. We summarized and analyzed the routinization process of how the case company adapted their computer-aided audit procedures from an experimental action to daily usage. An approach based on organizational routines theories was adopted to study group learning and interactions among project members, and to understand how they integrated automated auditing techniques and mechanisms into the existing manual auditing procedures. The process also reveals the incremental progress of an emerging routine from CAATTs adoption to continuous auditing systems.*

The research results show that the documentation of CAATTs projects and group learning among different functions contribute to the routinization of automated auditing procedures; the continuous auditing system based on the automated auditing program also contributes to routinely audited tasks. However, the improvisational nature of auditing activities, implicit characteristics within general auditing software, and rigidity of automatic auditing programs cause the resistance of auditors on CAATTs use, and also impede the emergence and flexibility of computer-aided auditing procedures.

KEYWORDS: *CAATTs, Continuous Auditing, Information System Control, Technological Adaptation, Organizational Routines.*

1. Motivation

The popularity of information technology in organizations, driven by the force of the global competitive information technology market, has created the era of automated auditing techniques (Bhimani, 1996; Elliott, 2002; Zhao, Yen & Chang, 2004). It has been commonly accepted by scholars and practitioners that the use of computer assisted auditing techniques (CAATTs) can reduce auditing costs and improve efficiency, and also help auditors to focus on high-risk business activities (Braun & Davis, 2003; Debreceny et al., 2005). However, Alles, Kogan and Vasarhelyi (2009) propose that the inadequate understanding of the nature of audit automation techniques is the greatest obstacle for organizations in adopting CAATTs.

Automated auditing techniques can be roughly classified as computer-assisted auditing techniques (CAATTs). They are widely adopted by accounting firms and internal auditing departments and are emerging auditing techniques in continuous auditing. Computer-assisted auditing techniques are the tools and techniques which are used to assist auditing. Mostly, they include tools and techniques to audit computer application, and the software to extract and analyze data (Braun & Davis, 2003). As many organizations use sophisticated information technology to assist business activities and to improve information processing efficiency (Ramamoorti & Weidenmier, 2004), how auditors effectively use CAATTs to review and monitor the effectiveness of internal control systems has become an important research issue (Braun & Davis, 2003; Gehrke & Wolf, 2010; Lindow & Race, 2002).

Scholars and practicers have high expectations on the emerging assurance service based on continuous auditing techniques. Continuous auditing is defined as a type of auditing that produces auditing results simultaneously with, or a short period of time after, the occurrence of relevant events (Alles et al., 2006; Kogan, Sudit & Vasarhelyi, 1999; Rezaee, Elam & Sharbatoghlie, 2001; Vasarhelyi, 2002). Because continuous auditing is implemented as a fully automated process on a computer application, it will save auditors substantial costs when verifying a large volume of transaction data. Due to continuous auditing techniques adopting a more immediate approach to execute control testing and risk estimation, they are believed to enhance the quality of auditing evidences. If supported by an artificial intelligence analysis tool in the verification procedure, they will also enhance the efficiency of auditing staff, and improve the quality of audit reports (Elliott & Jacobson, 1997).

However, continuous auditing is not, as Alles, Kogan and Vasarhelyi (2002) expected it to be, highly respected by external auditors -- internal auditors have become the main promoters of continuous auditing techniques. For the United States, accounting firms focused on the activities derived from Sarbanes-Oxley Act Section 404 and lost interest in the development of continuous auditing; also, section 201 requests strong independence for external auditors, so it is argued: will the new continuous assurance service influence the independence of external auditors? In contrast, internal auditors are eager to find resources to cope with the new responsibilities conferred on them by Section 404 (Chang, Wu & Chang, 2008); thus, in promoting the project on continuous auditing, they will not be restricted by independence requirements (Alles, Kogan & Vasarhelyi, 2008).

However, in the IIA's "2009 IT Audit Benchmarking Study" report, only about a quarter of organizations who have adopted CAATTs use continuous auditing techniques; and although many organizations have tried to use CAATTs, most organizations still consider the use of CAATTs to be associated with high cost, implementation difficulties, and low short-term benefits. As a result, less than fifty percent of organizations use computer assisted auditing software for fraud detection and prevention (The Institute of Internal Auditors, 2009).

Previous studies mostly focused on the characteristics of auditors as well as the design of CAATTs software to analyze the difficulties within CAATTs adoption (Huang, Hung & Tsao, 2008; Janvrin, Lowe & Bierstaker, 2008; Mahzan & Lymer, 2010). However, in addition to the computer skills of auditors and CAATTs software training, some scholars pointed out that perhaps the more fundamental problem is that people underestimate the impacts of automated auditing on auditing procedures and standards? Also, promotions of continuous auditing usually emphasize the "redesign" of existing auditing procedures, causing continuous auditing to be an unrealistic application having few impacts on current auditing procedures and methods. **Therefore, the incremental evolution process, which extends and integrates traditional auditing procedures with the experimental usage of CAATTs, might contribute to the adoption and development of continuous auditing.**

As a result, Alles et al. (2009) state that dramatically reengineering existing audit procedures is not feasible. They suggest that the continuous usage of CAATTs might help explore how to adopt continuous auditing techniques. Therefore, continuous auditing techniques are not new technologies to replace CAATTs, but rather as extensions of CAATTs. To explore how to adopt continuous auditing techniques, scholars and practitioners might start from the study of CAATTs usage and consider how to integrate them with the daily activities of auditors.

The issues mentioned above are the motivations for why we focus not only on the skills of auditors but also auditing activities which include computer auditing procedures. We consider the continuous use of CAATTs as evolutionary organizational routines, which start from the initial adoption of CAATTs and result in the transformation of traditional auditing. Therefore, the scope of this study covers the activities of CAATTs project groups, which contain the auditors, the chief of auditing executives, Information Systems (IS) staff, and IS managers. In addition, the theoretical framework mainly focuses on the evolutionary process of auditing practices, not only on the organizational characteristics of leadership style, supports from managers, and others discussed in previous studies.

In other words, we consider auditing activities as a kind of organizational routine, which will help us predict and analyze the patterns of organizational activities. Since the formation and evolution of organizational routines involves the learning process of organizations, the analysis of auditing practices will help us to understand how the usage of CAATTs becomes a part of daily activities and becomes the opportunity to change existing auditing procedures.

In summary, this study examines the interactions and group learning among CAATTs project members which are concerned with automating the traditionally manual procedures of auditing through integration with CAATTs. We examine how they start from the trial usage of CAATTs, and then proceed with experiments on continuous auditing and monitoring and finally work with CAATTS as emerging routines in auditing practices.

2. Literature

2.1 Current status and challenges of CAATTs

In the modern business environment, auditors use computer-assisted audit techniques (CAATTs) to assist their tasks of reducing audit risks and cost (Braun & Davis, 2003). Among the computer-assisted auditing techniques, General Audit Software (GAS) is easy to operate and can fit for different users and environments, therefore it has become the most widely used of CAATTs tools (Braun & Davis, 2003). GAS has many advantages, including ease of use, a huge data processing capability, a read-only aspect (it does not change raw data), the ability to import data presented in various way, and provisions of required auditing functions such as statistics, sampling, audit trails and so on. Despite this, GAS is still not widely applied in Taiwan, by private auditors and companies. For example, in the investigation of internal auditors in Taiwan, Huang et al. (2008) found that only 23.7% of organizations adopt GAS.

Lanza (2005) summarized the difficulties in using CAATTs and explained why CAATTs is still unpopular among auditors. The time spend for data acquisition and conversion almost equals to the time spend by the auditor in field audits. Secondly, when information systems change, the automated auditing programs have to be changed as well. Also, it is very difficult to obtain electronic data when there is a lack of support from IS staff. Finally, it is a time-consuming task for auditors to write automated auditing programs.

Many studies have explored the personal factors influencing auditor's adoption of CAATTs. Huang et al. (2008), for example, adopted the Technology Acceptance Behavior Model (TAM) (Davis, Bagozzi & Warshaw, 1989) to investigate the personal factors impacting on an internal auditor's use of CAATTs. The factors mainly consist of “perceived usefulness” and “perceived ease of use.” The IT skills of auditors have also been recognized as critical factors affecting adoption. Many scholars and practitioners advocate that auditors become familiar with CAATTs (Burnett, 2003; Gallegos & Looho, 2000; Lord, 2004). Debreceny et al. (2005) and Ramamoorti and Weidenmier (2004) state that auditors with sufficient IT skills will significantly improve effectiveness of auditing while reducing the dependence on IT staff.

From the literature, we find that, while some scholars emphasize the ease use of GAS and the benefits of using CAATTs, others emphasize the importance of IS staff involvement and the IT skills of auditors. Therefore, the implementation of CAATTs requires the involvement of auditors and professionals in different fields, such as systems analysts, database managers, application software vendors, and others (Rezaee et al., 2001).

In summary, perhaps the difficulties of promoting CAATTs relate to how professionals from different areas can be combined to learn from each other and work together, which involves cross-section collaboration and integration of different fields of organizational practices. Therefore, the implementation of CAATTs could be considered as an adaptive organizational that involves interaction among project members which modify existing organizational routines (original auditing procedures) and becomes the new organizational routines (combining previous auditing procedures with automated auditing procedures).

2.2 Technological adaptation, organizational routines and improvisation activities

Many scholars have explored the use of adaptation process technology, such as Rice and Rogers (1980), who indicated that with the continuous use of technology users will discover and reinvent the application and usage of the same technology. They suggest that these adaptation activities should be considered as part of the innovation process, and that these activities are important for users in improving their satisfaction (Johnson & Rice, 1987). Tyre and Orlikowski (1994) referred to these related adaptation activities as "technology adaptation," and defined them as "the modification and changes of technology usage after adoption," which involves the characteristics of technology, operation activities among users and the change of assumptions, knowledge and experiences. These changes may mainly come from user involvement or the interactions between users and developers.

The technological adaptation approach emphasizes the importance of follow-up modification after the adoption of technology, which will assist the effective use of technology and improve operational efficiency (Dutton & Thomas, 1984). The adaptation process is not only about affecting the usage of technology but also to changing the organizational structure and work practices (Van de Ven, 1986). Tyre and Orlikowski

(1994) also indicate that the longitudinal view of the technological adaptation process may differ with a cross-sectional view. Because the experiences through established practice will simplify the decision-making process, by focusing on new information while ignoring other parts, individuals will ignore misfit problems in their usages of the technology and over time these problems be integrated into the organizational process and evolve as organizational routines.

Kwon and Zmud (1987), based on the Change Model of Lewin (1952) proposed the Innovation Diffusion theory, which includes six stages: initiation, adoption, adaptation, acceptance, routinization and infusion. The initiation stage, corresponding to the unfreezing phase of the Change Model, indicates that, when organizations face problems or opportunities, they began to search for useful solutions. The adoption and adaptation stages, corresponding to the change phase of the Change Model, indicate that organizational members are willing to implement the technology and adapt to its impacts. The acceptance, routinization and infusion stages correspond to the frozen phase of the Change Model and refer to the period when users accept the use of the technology; their acceptance can be recognized through an evaluation of performance and satisfaction. Through these stages, the wide use of the technology becomes part of the organizations routines (Cooper & Zmud, 1990).

Despite the technological adaptation model of Kwon and Zmud (1987) constituting six stages, most studies focus on the prior stages of initiation, adoption, adaptation and acceptance. Fewer studies focus on the routinization and infusion stages, when users combine technology use with organizational routines and gain most benefits from the integration. Organizational routines, however, have been studied broadly by scholars of organizational behavior. Thus, related organizational routines theory will be helpful in an exploration of the routinization stage of the technological adaptation process.

According to a dictionary definition, routines are "detailed regulations and compliance activities of the regular course, such as standard operating procedures." Organizational routines have the following four characteristics: (1) repetition; (2) recognizable pattern of action; (3) multiple participants; (4) interdependent actions (Cohen, 1991).

Organizational routines can save costs and improve efficiency for our daily activities. However, organizational routines are also the main cause of rigidity when environments change. They might make mistakes or lost opportunities for innovation. Organizations could be inefficiency without integration by organizational routines. In the other hand, people usually get suboptimal solutions and reduce performance when excessively rely on organizational routines. Therefore, organizational routines could be considered as a double-edged sword (Cohen & Bacdayan, 1994), rapidly developing new routines or adopting appropriate routines are the sources of competitive advantage and also the characteristics of successful organizations (Cohen, 1991). Nelson and Winter (1982) proposed that organizational routines are similar with skills, therefore organizations can be regarded as social networks embedded with organizational routines.

In addition, organizational routines are presented in the forms of documentation of standard operating procedures, shared practices or norms within community members. They might be coded with explicit texts, or shared in implicit ways. Organizational routines usually only cover general principles and rules of activities, and they do not cover the special cases with different situations. As a result, in different situations, the actual behaviors of participants might be different according to their interpretation. Organizational routines inevitably retain the characteristics of improvisation. Thus, organizational routines are repetitively executed in organizations, but their contents are not necessarily constant (Feldman & Pentland, 2003).

Organizational routines are the results of organizational learning (Argote, 1999). The Changes of organizational routines, regardless of emerging routines or amendments of prior routines, are the accumulation of organizational learning process (Levitt & March, 1988; Nelson & Winter, 1982). In other words, current organizational routines are built by past learning, error correction, and continuous improvement. Managers have to appropriately execute the daily activities with repetitive procedures (routines) and creative tasks (innovations) for the tradeoffs between efficiency and flexibility (Feldman & Pentland, 2003).

The tasks of auditors consist of a series of repetitive procedures, among which include: audit planning, risk assessment, auditing procedure design, data collection and analysis, working papers and auditing report writing, and follow-up tracking. These auditing activities are organizational routines with specific patterns and interactions among people from different departments. To enhance the efficiency and quality of auditing, there are many principles and standards guiding the procedures of auditing activities, but the characteristics of improvisation, risk assessment and professional judgments are very important for the execution of auditing, which makes the tasks of auditors more complex than other organizational routines.

From the view of organizational routines theory, although CAATTs help organizations get the benefits of automation, accuracy, and quick analysis, some auditors might question that automated auditing is based on explicit audit procedures, whereas many unexpected situations with variant rules happen in the processes of auditing, thus making it unfeasible to define the complete auditing procedures which cover various scenarios in advance. So, one of the critical factors is to provide the flexibility to auditors (improvisational behavior) together with the explicit and precise procedures of automatic computer programs (predefined organizational routines).

From the literature discussed in this section, we can find that the adaptation process from CAATTs adoption to the implementation of a continuous auditing system is affected by the auditing environment, skills of auditors, decisions by chief auditing executives, and the interaction between auditors and IS staff. The scope of the research target is not limited to CAATTs itself but also includes business processes, control activities, auditing procedures and enterprise information systems. The phenomenon of CAATTs adoption and use could be considered as technological adaptation processes, which involve adoption improvisation activities to change prior organizational routines into new organizational routines.

3. Study design

In order to understand the adaptation process of enterprises using CAATTs, our research design adopts an exploratory study via a single case. The selected case company is referred to as 'Alpha Company' in this study. The period of analysis from 2006 to 2009 covers the time when Alpha Company decided to adopt CAATTs and their continuous use of CAATTs. Alpha Company developed the automatic auditing procedures for the four transaction cycles, and continues to fulfill the enhancement. Therefore, the technological adaptation process of Alpha Company could help us understand how companies adapt through the adoption and adaptation stages and the acceptance and routinization stages.

This study starts from July 2008 and ends in May 2009. The sources of data come from interviews (see Table 1), internal documents and public information. The researchers interviewed the Chief Audit Executive (CAE) and Chief Information Officer (CIO) and the members of CAATTs projects. To increase the accuracy of the interviewee’s responses, and go back to the initial situation of use, the researchers first asked the interviewees the course of events for adoption and implementation of CAATTs, and followed up with the

Table 1 The Interview Activities and Interviewee List

Interviewee	Job Title	Frequency	Interview Time
CAE	Auditor-General	2	December 2008/March 2009
Internal Audit Staff	Senior Auditor	2	October 2008/March 2009
Internal Audit Staff	Financial Auditor	1	March 2009
CIO	Associate Manager	2	December 2008/March 2009
IT Staff	Director	1	March 2009
IT Staff	Database Administrator	2	December 2008/March 2009
Project Consultants	Project Manager	2	July 2008/May 2009

adaption and routinization processes. The researchers used open questions and focused on emerging concepts and themes in the technological adaptation process.

In addition to the transcripts of interviews, the researchers also analyzed internal documents and public information, as the following shows, to understand the usage of CAATTs in Alpha Company:

1. Company information: company profiles, organizational charts, annual financial reports, etc.
2. Documentation of CAATTs projects: flowcharts, project task item lists, audit item lists, audit programs, data dictionary, minutes of meetings, etc.
3. Documentation of auditing: documentation of the internal control system, audit plans, working papers and audit reports, etc.

4. Case description and analysis

4.1 The initial and continued use of CAATTs

We divide the technological adaptation process of Alpha Company into two periods for easy comparisons: one is the initial period of use, which includes the initiation, adoption and adaptation stages, and the other is the continued period of use, which includes the acceptance, routinization, and fusion stages. In the initial use phase, Alpha Company first tried to adopt CAATTs on their sales and receivables cycle, and in the continued use phase, Alpha Company continuously implemented three projects, which includes production, procurement and payroll cycles. Each phase is described as the sequences of pre-implementation, implementation and ex-implementation phases. The objectives, tasks and factors of each stage are summarized in Figure 1.

4.1.1 The initial use period: pre-implementation phase

Alpha Company has a highly integrated information system; the management values the monitor and control of information processing activities. The IT staffs maintain the enterprise information system by themselves. The CAE of Alpha Company thought manual auditing is constrained by manpower and budget. For the purpose of increasing the effectiveness of the internal control system and decreasing the expenditure of auditing, the CAE started to implement the CAATTs project in 2006.

The auditors of Alpha Company have been trained in the use of CAATTs software; however, they still did not how to use CAATTs in practice. So the CAE hired an external consulting team to assist the project and decrease the workload of the auditors during the implementation phase. The consultant thought it was not easy to discuss the computer

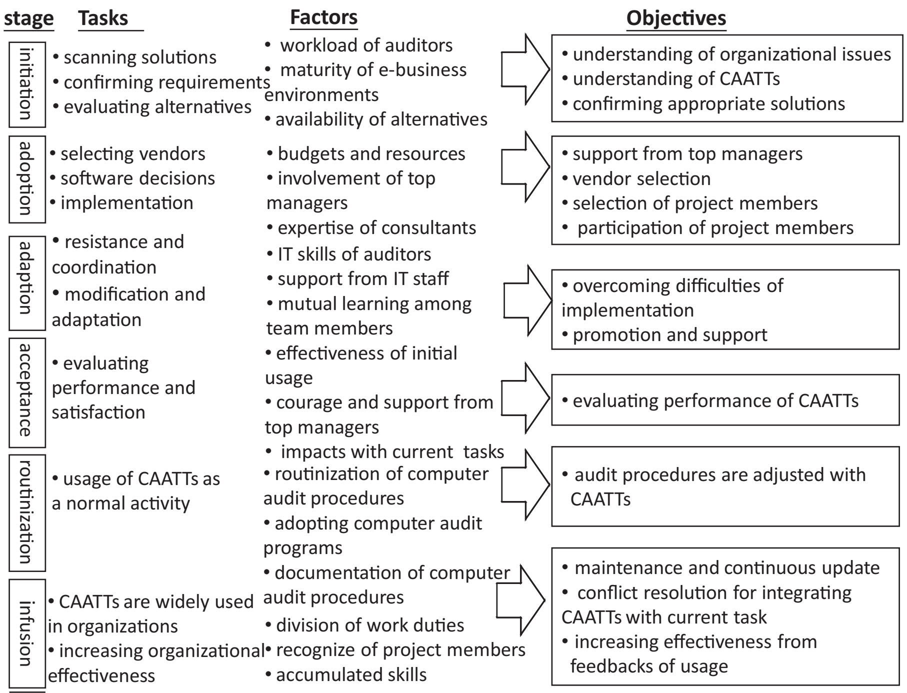

Figure 1 Objectives, Influencing Factors, and Tasks in Each Stages of the Technology Adaptation Process

auditing procedures only with CAATTs software, so he designed a series of documents as the analysis and design tools.

At the beginning of the initial period of use, the consultant arranged a kick-off meeting with the CAO and the CIO to obtain the consensus of both. Although the CIO did not recognize the need for computer auditing, and also questioned the performance of the CAATTs software, after the explanations of the consultant and with the strong support of CAE, the CIO was willing to assist the project. The meeting was concluded with the following agreements: the audit items would be prepared by the auditors, whereas the IT staff would help auditors to understand the current system processes and provide the required data files, and the consulting team would be responsible for the design, testing and documentation of computer auditing programs.

4.1.2 The initial period of use: implementation phase

Initially, because of the lack of experience of the project members, the progress of project implementation was slow. The project meetings were chaired by auditors, IT staff, and the consulting team. In these meetings, project members discussed together setting project plans, auditing scope, auditing items, risk assessment for each auditing item, and the feasibility of analysis using computer auditing procedures.

During these meetings, consultants found that the auditors were used to the pervious field detection and were not familiar with the process flow of the enterprise information systems and the embedded application controls. Therefore, some auditing items were not applicable for computer auditing; some controls had been embedded in the information systems, but the auditors were not informed. So they required the assistance of IT staff to confirm the feasibility of the computer auditing procedures.

On the other hand, the consulting team is based on these feasible auditing items and recognized required data files from IT staff, and also confirmed the definition of the fields in these data files. However, it was difficult to communicate between IT staff and auditors, because IT staffs were not familiar with the auditing rules, and auditors were not familiar with the relationships of fields among multiple tables. The consultant played the important role of collaboration to explain the auditing objectives and technical illustrations to both. In addition, impacted by the heavy workload of IT staff, data confidentiality and authorization issues, and time-consumption associated with downloading and converting data from the mainframe, the schedule of the project resulted in serious delays.

After obtaining electronic data files, the consulting team used CAATTs software to execute the control test, data was filtered by auditing rules, and the abnormal transactions were extracted. The transaction data was sent to the auditors to confirm the reasonableness and reasons. If necessary, the auditors would modify the auditing rules and re-check several times for the extracted results. When the steps of analysis were confirmed as reasonable, the consulting team would write down the automatic procedures as computer programs. Meanwhile, the auditors would track the reasons of unusual records based on the analyzed results, and prepare the working paper for the auditing report. The consulting team wrote the analysis documentation according to the auditing procedures and the auditing objectives.

4.1.3 The initial period of use: post-implementation

The first CAATTs project needed 9 months to be completed because of the unfamiliarity with CAATTs by project members and the huge discrepancy in knowledge background among them; however, CAE and the auditors were satisfied with the achievements. The CAE confirmed that the automatic auditing programs can decrease the time and resource required for repeated auditing procedures and increase the efficiency of auditors with more coverage of data analysis. He also praised the consultants for the systematic analysis and documentation methods, and promoted this methodology as a guideline for following projects. He expected the auditors could apply the methodology and develop other automatic auditing procedures by themselves.

4.1.4 The period of continued use: pre-implementation phase

After the successful completion of the CAATTs project, the CAE wanted the scope of automatic auditing to be extended to multiple transactions cycles. He hoped the auditors could familiarize themselves with the development of computer auditing programs and be proficient with CAATTs software. Therefore, in addition to engaging the consulting teams to continue developing the other cycles, such as procurement, production and payroll cycles, the CAE requested the project members to take CAATTs training, and take active roles in the subsequent projects.

4.1.5 The period of continued use period: implementation phase

Since the auditors experienced the effects of CAATTs in their project, they knew what kinds of auditing items were applicable for computer auditing. So it was more efficient for the auditors to make a list of computer auditing items and to proceed with risk assessments. After an internal department meeting, they engaged the consultants with the IT staff for feasibility analysis in the first project meeting.

As the auditors were unable to grasp the updated status of business application controls, they still needed the assistance of IT staff to identify the appropriateness and feasibility of each computer auditing item. This step helped to remove these issues that were under control. So, the awareness of IT staff of internal control systems and acceptance with the values of CAATTs projects is very important for the implementation of projects. For example: IT staff with a supportive attitude not only explained the information flows and processing rules, but also enthusiastically pointed out the weak areas of current application controls. However, another IT staff serving as database administrator might focus on the efficiency of database servers and the security issues of information, but would not be concerned with whether the data elements are applicable for the auditing items.

Alpha company then built a new database server, whose data is synchronized with the mainframe; consequently making it is easier to retrieve data directly from the new database server. The auditors could use the CAATTs software to download the authorized data by themselves. This increased the efficiency and independence of auditors. However, the auditors still required the assistance of IT staff to confirm the meaning of data fields and location of data files, such as the status codes for each transaction forms, and classification codes used for analysis. This information was essential for computer auditing procedures.

The consulting team was still responsible for the tasks of data importing, validation, and analysis. Although different transaction cycles were assigned to different staff, most of the staffs were part-time students, who just learned how to operate CAATTs software without auditing experiences. However, by referring to the documentation on previous projects and examples of similar computer auditing programs, they could get started quickly, and completed the required programs on schedule.

The involvement of the auditors and familiarity with CAATTs software still played an important role in the CAATTs projects. Based on the rotation system in Alpha Company, most auditors come from various business units. Some of them are determined not to be auditors in the long term, and some of them might only assist field auditing by collecting information; hence, they would not be active in CAATTs training. That is to say, some auditors keep a wait and see attitude for these projects.

4.1.6 The period of continued use: post-implementation phase

After the initial use of the CAATTs project on sales cycle and a following up with three projects on different transaction cycles, the auditing items developed in these projects covered the major areas of regular auditing activities in Alpha Company. The auditors began to use these automated auditing programs, which were developed by the consulting team, as pre-audit activities to help find out the abnormal transaction, and then conducted follow-up auditing.

The CAE was satisfied with the outcome of these CAATTs projects, but still recognized the necessity of self-reliance for auditors in implementing CAATTs projects. From the interviews, we find that the key for CAATTs implementation is not the insufficiency of computer skills for auditors, but the fact that they worry that the increase of automated auditing programs will become a heavy workload in terms of maintenance. The auditors hope the IT staff to undertake the maintenance and development of automated programs. On the other side, the CIO wanted the auditors to have the ability to use CAATTs tools and maintain autonomy, while the IT staffs focus on the enterprise information systems. The IT staff is only responsible for data file access authorization and database server maintenance. The consultant also believed that auditors should not only rely on reports generated by automated programs, although they still require the ability to use CAATTs for analysis conducted by themselves. Thus, either error messages would be caused by changed data environments or new auditing items could be upgraded and maintained by the auditors, fundamentally enhancing their independence and effectiveness as auditors.

4.2 Case analysis

In Figure 2, we present the process flow of Alpha company, which integrates the computer auditing process with the field auditing activities, computer control test, and tracking items. Alpha company implemented the CAATTs by projects at first, and gradually changed as continuous auditing became embedded in the routine auditing

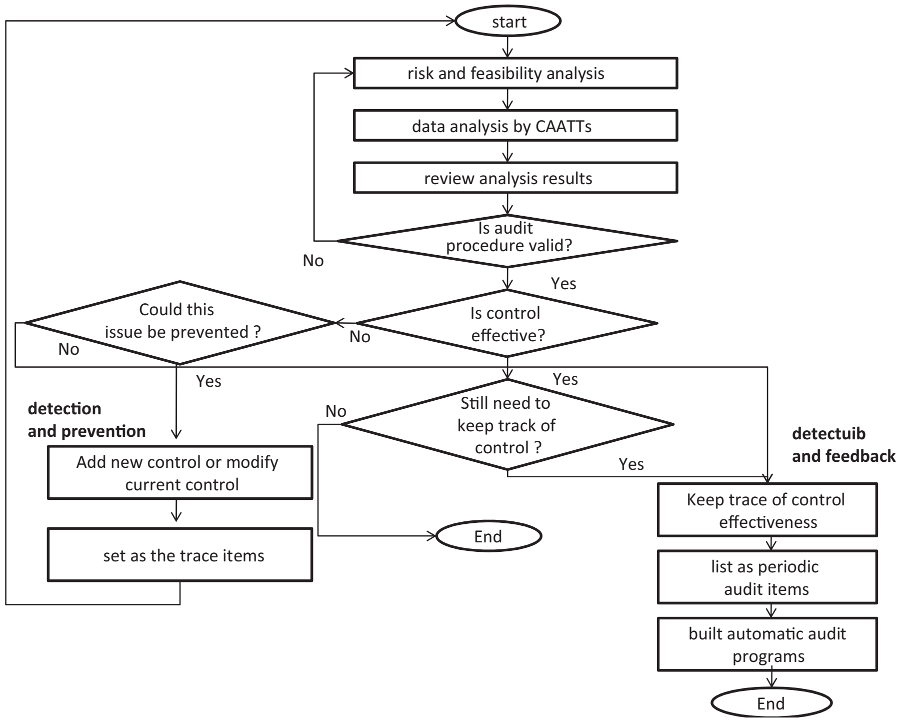

Figure 2 The Integration of Manual Auditing Procedures and Computer Auditing Procedures

activities of different transaction cycles. The implementation of each project helped the auditors to view the current status of application controls, whether manual or computer based controls. The initial analysis of each project helped to modify the detective rules of abnormal data and improve the effectiveness of current controls. After the cause-effect analysis of abnormal data, the auditors discussed with IT staff or staff of related departments to add or modify current control procedures and list them as tracking items. If the risk of some operations could not be decreased by preventative control procedures, and the risk was over the tolerance, then the related computer auditing procedures were added in the continuous auditing system, which would be executed by timely automated programs to monitor the current status of critical operations.

Compared to traditionally manual form auditing, we found that Alpha Company changed its auditing activities from information gathering and preliminary risk assessment to control tests (these could be considered as the evolution of routines). The auditors were more familiar with IT application controls through cooperative learning with IT staff. Thus, whereas the auditors used to ask their clients modify their errors in the past (old routine), now the auditors designed new application controls to prevent the mistakes (emergence of new routine). In addition, the cost of time and effort of routine activities were reduced by the use of the continuous auditing system, thus the auditors could focus on other auditing tasks with high risks (evolution of routines).

While the auditors were not familiar with CAATTs at first, now they recognized the efficiency and effectiveness of CAATTs. The IT staff was initially resistant to convert and download data for the auditors, but they also found a more efficient way to provide data. In other words, Alpha Company developed an effective methodology to reduce the time and costs of CAATTs implementation. They also used the features of the continuous auditing system, such as real-time control and monitoring, to increase the benefits of CAATTs implementation. The new auditing methodology from CAATTs adoption to the use of a continuous auditing system is the result of the technological adaptation process through organizational learning.

4.3 Research findings and discussions

As mentioned in the section of literature review, whether or not computer auditing techniques can be utilized in organizations, the key issue is how to support and enhance current routines, and then convert them into new organizational routines. Figure 3 presents the development of computer auditing routines and related factors among the adaption process. The research findings and discussions are based on the drivers and resistance of routine adaptation, following with the emerging routines and related constraints.

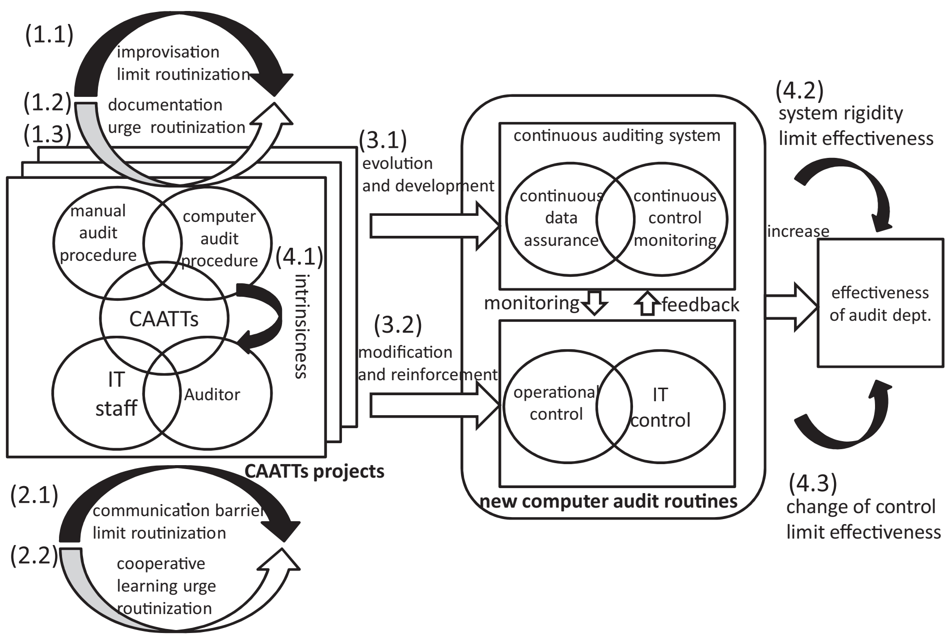

Figure 3 The Research Framework and Findings

4.3.1 The drivers and resistance of computer auditing routines

[Finding 1.1] The characteristics of improvisation in auditing activities will reduce the wiliness to use CAATTs. The balance and complementary between human risk assessment and automated auditing are the key for the development of computer auditing routines.

In the case company, most of the auditors maintain the attitude of wait and see for CAATTs adoption. The CAE thought maybe the auditors were not familiar with the software. However, even with training and compulsory use, some auditors were still negative about CAATTs. After interviews, we found that the senior auditors value the knowledge and experiences garnered from field auditing. These help auditors identify problems and confirm the auditing scope. In other words, senior auditors emphasize the improvisational characteristics of risk assessment in auditing activities. Risk assessment is the response for the information from target areas with personal judgments or suggestions from other members. Decisions making activities for risk assessment are executed in timely pressure and limited information.

In auditing activities, risk assessment and determining the causes of abnormal data need impromptu decision-making with situational characteristics. The characteristics of structural analysis with predefined procedures for automated programs make it not easy to solve undefined situations (Searcy, Woodroof & Behn, 2003). Thus, the improvisational characteristics of auditing activities will reduce the wiliness to use CAATTs. Alles et al. (2009) also considered how to utilize the efficiency and accuracy of automated auditing programs, while maintaining the flexibility of result interpretation for auditors, retrieving data according to different situations, analyzing with real-time judgments are the bottlenecks for the use of a continuous auditing system.

On the other side, to lower the improvisational characteristics of auditing activities was also the motivation of the CAE to adopt CAATTs. From the opinions of the CAE, the manual auditing methods are more dependent on the experiences of auditors, are not easy to learn for new comers, and sometimes the tasks are done in a muddled manner because of the of half-heartedness. However, from the thorough examination of computer auditing programs, there is the realization that the auditing scope is more comprehensive and reliable. Computer auditing programs help reduce the uncertainty of human decision-making and help the collaboration among auditors with efficiency and objectivity. In summary, the balance and complementary between human risk assessment and automated auditing are the key for the development of computer auditing routines.

[Finding 1.2] The characteristics of explicitness from computer auditing documentation will enhance the efficiency of the CAATTs project, and the emergence of computer auditing routines.

[Finding 1.3] The knowledge engineering features of computer auditing activities will help transform personal implicit knowledge into explicit knowledge shared by groups.

In the four CAATTs projects of the case company, we find that: The documentation of CAATTs project built in the first project is very helpful for the implementation of subsequent projects. Even though members involved in the follow-up projects are new entrants who lack experiences, after they refer to the previous documentation as guidelines, they can quickly and effectively apply the same method on other different projects.

As most auditing activities have clear procedures, analysis procedures and reporting principles, using CAATTs documentation to support auditing activities is similar to using manual auditing programs. The documentation helps integrate computer audit procedures with manual processes and also helps auditors understand the procedures of automated programs.

In addition to guiding the implementation of projects, the documentation, such as meeting minutes and analysis documents, also help project members clarify the implicit rules of fuzzy decisions into accurate procedures, which could be executed by computer programs. These analysis processes could be considered as the formalization process of knowledge engineering activities (Foguem et al., 2008).

Dowling and Leech (2007) also proposed the similar opinion: The key elements for auditing automation are how to clarify the vague knowledge hidden in manual auditing procedures and specify the knowledge as explicit rules. In other words, the analysis and design of automated auditing procedures, will transform the personal implicit knowledge, such as the operations and controls of each departments, the processing flow of enterprise information systems, etc., into explicit knowledge shared by groups after the project meetings. The explicit knowledge is easy to learn or imitate by new entrants (the emergence of new routines).

[Finding 2.1] The communication barriers caused by the differences in expertise limit the emergence of computer auditing routines.

[Finding 2.2] Cooperative learning among project members helps the emergence of computer auditing routines.

From the case company we find that: Because computer auditing activities are different from the traditional manual form of auditing, activities such as retrieval of electronic data, information system control tests, and examination of abnormal data require the assistance of IT staff. They cannot be done alone by auditors. However; the communication barriers caused by different expertise prohibit the implementation of CAATTs projects. For example, IT staffs tend to use the same platform and programming language with enterprise information systems, so auditors find it difficult to accept the software. Also, auditors are unfamiliar with tools with which automated programs and continuous auditing systems are developed and how maintenance should be conducted. Most IT staffs also do not understand the risk assessment and control testing procedures for auditing activities; thus, sometimes they will project a negative attitude toward CAATTs projects. In addition, the analysis needs the help of IT staff to confirm the causes of abnormal data. However, when the auditors recognize the areas of control weakness, they usually will be considered as the lack of control in information systems. The auditing results will add the workload of IT staff in terms of modifying the information system. Therefore, the IT staffs, who are the facilitators for CAATTs projects, embarrassedly become as the negligence holders. This reduces the willingness for IT staffs to participate in the CAATTs projects.

Though the knowledge gap among different experts causes barriers in communication, it also becomes the starting point for cooperative learning among project members. Through discussions and problem solving, individual skills and professional knowledge emerge and are infused into the organizational routines (Julian, 2008). For example, through the assistance of IT personnel, the auditors were able to quickly understand the logic of application controls, which is the key for effective risk assessment; the IT staff, for their part, used the opportunities to re-check the correctness and effectiveness of application controls by participation in the project. These helped to amend or reinforce the control mechanisms of operations.

In Alpha Company, we also found that the common training courses of CAATTs, which teaches auditors how to operate the software, are insufficient for auditors. However, after the implementation of several projects and continuous use, the project members accumulated the computer auditing techniques for various scenarios. In addition, with the assistance of consulting teams, project members with different professional backgrounds learned how to develop computer auditing procedures fitting the business contexts of Alpha Company.

This phenomenon supports the argument of Becker et al. (2005), which clarifies the difference between organizational routines and personal skills by stating that collective nature is the distinctive feature for organizational routines. The projects, involving many departments, will help these functional communities, which work in different vertical divisions, gradually form a new community of computer auditing. Thus, the cooperative learning among project members help the emergence of computer auditing routines.

4.3.2 The emergence of computer auditing routines

[Finding 3.1] The computer auditing programs developed in the CAATTs project are helpful for auditors transforming CAATTs applications into continuous auditing systems, and evolving them into emerging routines.

[Finding 3.2] The results of computer auditing with cause-effect analysis of abnormal data, contribute to updating and reinforcing control activities and operations.

From the case analysis, we found that the scripts of CAATTs software could be converted to the automated auditing system, which can be executed regularly. So the auditors could use the automated auditing system to assist regular auditing tasks, and these programs could be executed more frequently than manual auditing. Thus the auditors could timely and with less cost find the abnormal data. This improved the efficiency of auditors and made the auditors more willing to use CAATTs.

In addition, as the scope covered by the automated programs continued to extend, the auditors gradually changed their auditing procedures. They used the reports of the automated auditing system as the basics of risk assessment and activities arrangements. The use of the automated auditing system shortened the auditing periods, increasing auditing frequency. In Alpha Company they use the workflow system to provide real-time exception reports by definition. So the CAE and department managers can use the system to monitor the status of operations. The automated auditing system has become the continuous auditing system of Alpha Company.

Although Hammer (1990) proposed that the implementation of information systems should redesign or reengineer current processes to avoid a focus only on the automation of current operations, which would limit great potential benefits, the rules of auditing activities are implicit and improvisational, so to redesign the overall auditing activities in advance is not easy. The hybrid approach of integrating manual and computer auditing is easier for promoting automated auditing in daily auditing activities (Alles et al., 2009).

From the above, the computer auditing programs developed in the CAATTs project are helpful for auditors transforming CAATTs applications into continuous auditing systems and evolving them as emerging routines. The results of computer auditing, with cause-effect analysis of abnormal data, contribute to updating and reinforcing control activities and operations.

4.3.3 The restrictions of computer auditing routines

[Finding 4.1] The implicit features of the automated auditing system restrict auditors in reviewing and understanding the meaning of auditing programs. These increase the rigidity of computer auditing programs.

[Finding 4.2] The rigidity of automated auditing systems limits the maintenance and expandability of the systems.

Despite what has been reported, we still cannot be too optimistic on the use of automated auditing systems. The limitation of CAATTs software and continuous auditing systems also emerged in the case company. For example: in the case company, CAATTs software could support the analysis of auditing activities, but manipulation by data fields did not easily indicate the logical relationships among data fields. In other words, for the auditors, who are not familiar with computer programs and data structures of databases, it is difficult to quickly grasp the semantic meaning of computer programs (Li, Huang & Lin, 2007). The implicit features of CAATTs manipulation and computer programs also hide the risk factors and the rule of auditing procedures. This will increase the difficulty of maintenance and expandability for computer auditing procedures and become the barriers for the integration with manual auditing procedures. Therefore, the rigidity of computer auditing programs restricts the improvisation of auditing activities.

In addition, as the auditing scenarios changed, the automated programs continued to be modified with more complexity and were more difficult to maintain. So the auditors argued with the IT staff as to who should be responsible for maintenance of the automated auditing system? Although automated auditing systems can be executed repetitively and rapidly, and improve the efficiency of regular auditing, the design and maintenance of automated auditing systems is more complicated than manual auditing procedures. This causes the resistance of auditors, who intend to avoid burden of system maintenance. As the auditor of the case company stated, "Our expertise is to focus on risk assessment and control test. We are not professional IT staff, so the burden of maintenance will decrease our performance."

As the statements indicated in Finding 1.1, how to keep the flexibility of auditing activities with the automated auditing system is the key in terms of conducting computer auditing. In particular, the risk of companies will alter with changes in environments and organizations. The lack of semantic meanings in computer auditing programs hide the factors of risk assessment and the rules of the auditing procedures, therefore, it also has significant impacts on the benefits of continuous use for automated auditing systems.

[Finding 4.3] The change of operations and information systems will affect the long-term benefits of automated auditing systems.

In the case company, the rules of application control were not recorded in detail in the documentation of internal control. In addition with the implicit features of computer programs, the operation risks will alter when operations or information systems changes. This will produce the weakness of control or even result in the lost of functions of automated audit systems.

Current automated auditing systems operate on the basis of data field manipulation and computer programs embedded within the structure of the database. Process redesign, organizational restructuring and the replacement or updating of information systems will cause the errors or dysfunction of systems and increase the burden of maintenance. For example, when the sales departments of the case company become separate business units, the location and data files of sales transactions change after the adjustments. This would lead to full revision of the overall computer auditing programs.

Alles et al. (2006, 2009) also proposed the similar opinions. Systems using general auditing software are based on the data oriented approach (named as CDA: Continuous Data Assurance), which makes it not easy to monitor the real-time status of controls. How to develop continuous monitoring systems with continuing changes in information systems and control activities remains a major bottleneck for continuous auditing systems.

5. Conclusion

5.1 Contributions of this study

Previous studies on CAATTs usage mostly focused on personal skills, attitudes and the support of top managers. Fewer studies explore the drivers and resistance of CAATTs implementation on a group level. In this study, we analyzed the technological adaptation process for CAATTs in a case company, which implemented four CAATTs within three years. We took the theoretical viewpoints of organizational routines to explore how the auditors integrated computer auditing procedures with manual auditing activities, and transform the CAATTs manipulation into the continuous auditing system. The technological adaptation process extends previous routines and leads to the emergence of new organizational routines.

From the case study and a discussion thereof, we conclude: Cooperative learning among project members and the documentation of projects contributes to the emergence of computer auditing routines; the continuous auditing system based on computer auditing programs also contribute to the emergence of routines. However, the improvisational characteristics of auditing activities, the implicit features of computer auditing programs, and the rigidity of continuous auditing systems causes the auditors to resist using CAATTs tools, and restricts the emergence of computer auditing routines and the flexibility of auditing activities.

The main contributions of this study are as follows:

For the companies planning to adopt CAATTs or who have tried to use CAATTs, this study will help organizations comprehend the difficulties which they may face and how to overcome them. In addition, this study provides a solution on how to implement CAATTs projects, and transform them into continuous auditing systems. The study also revealed the interactions among continuous auditing systems, control mechanisms, information systems, auditing procedures and work adjustments of related personnel. These research results are in accordance with the suggestions provided by Brown, Wong and Baldwin (2007): Case studies are necessary for the areas of continuous auditing. They help understand the key issues for building continuous auditing systems, and also the critical factors underpinning CAATTs implementation.

Concerning the arguments of previous studies relating to the ease of use of CAATTs and the importance of IT skills for auditors, we find that this contradiction is caused by the specialization of both tasks and knowledge. Although the general auditing software is easy to use for auditors, without the assistance of IT staffs, the auditors cannot successfully complete the analysis of electronic data extracted from enterprise information systems. In addition, the lack of semantic meaning in the software manipulation and computer auditing programs mainly restrict the accurate review and maintenance of automated auditing systems.

5.2 Suggestions for practitioners

The practical suggestions for companies planning or who have tried to use CAATTs are as follows:

(1) The documentation of computer auditing procedures will help CAATTs to be integrated with manual auditing procedures, and ease the burden of maintenance of computer programs.

(2) It is insufficient to build the computer auditing routines only by documentation, because it is not easy to change the auditing methodology of auditors. It still depends on the cooperative learning environment of project members.

(3) The implementation of CAATTs projects requires different areas of expertise. The duties and tasks of each project member should be clear and consistent with their expertise. This will enhance the willingness of participants and encourage collaboration and mutual learning with each other.

(4) The implicit features of computer auditing program increase the rigidity of computer auditing procedures and the continuous auditing system. It is suggested that auditors use the documents with clearly defined format and legend. This will help staffs to effectively review the correctness and reasonableness of computer auditing procedures.

(5) After the improved understanding of application control by auditors, and enhanced comprehension of auditing procedures for IT staff, the CAATTs project will be implemented more smoothly and will also be helpful in terms of the emergence of computer auditing routines. The conflict among project members will decrease and there will be less misunderstanding among them.

Acknowledgements

Thanks for the helpful suggestions of reviewers and the assistance of Yi-Shan Lin, and thank the National Science Council grant funding for this research, grant project No. NSC-98-2410-H-224-005. In addtion, this paper is a revised and expanded version of a paper entitled 'From CAATTs Adoption to Continuous Audit Systems Implementation: The Analysis Based on Organizational Routines Theories,' presented at International Conference on Accounting and Information Technology (ICAIT 2010) which was hold at Chung Cheng University, Chiayi, Taiwan, in 7 July, 2010.

References

Alles, M.G., Kogan, A. and Vasarhelyi, M.A. (2002) 'Feasibility and economics of continuous assurance', *Auditing: A Journal of Practice and Theory*, Vol. 21, No. 1, pp. 125-138.

Alles, M.G., Kogan, A. and Vasarhelyi, M.A. (2008) 'Audit automation for implementing continuous auditing: principles and problems', paper presented at the Ninth International Research Symposium on Accounting Information Systems, Paris, France, December 13.

Alles, M.G., Kogan, A., and Vasarhelyi, M.A. (2009) 'Principles and problems of audit automation as a precursor to continuous auditing', Working paper, Department of Accounting, Business Ethics & Information Systems, Rutgers Business School, Newark, NJ, September.

Alles, M.G., Tostes, F., Vasarhelyi, M.A. and Riccio, E.L. (2006) 'Continuous auditing: the USA experience and considerations for its implementation in Brazil', *Journal of Information Systems and Technology Management*, Vol. 3, No. 2, pp. 211-224.

Argote, L. (1999) *Organizational Learning: Creating, Retaining and Transferring Knowledge*, Springer, New York, NY.

Becker, M.C., Lazaric, N., Nelson, R.R. and Winter, S.G. (2005) 'Applying organizational routines in understanding organizational change', *Industrial and Corporate Change*, Vol. 14, No. 5, pp. 775-791.

Bhimani, A. (1996) 'Securing the commercial internet', *Communications of the ACM*, Vol. 39, No. 6, pp. 29-35.

Braun, R.L. and Davis, H.E. (2003) 'Computer-assisted audit tools and techniques: analysis and perspectives', *Managerial Auditing Journal*, Vol. 18, No. 9, pp. 725-731.

Brown, C.E., Wong, J.A. and Baldwin, A.A. (2007) 'A review and analysis of the existing research streams in continuous auditing', *Journal of Emerging Technologies in Accounting*, Vol. 4, No. 1, pp. 1-28.

Burnett, S. (2003) 'The future of accounting education: a regional perspective', *Journal of Education for Business*, Vol. 78, No. 3, pp. 129-134.

Chang, S.-I., Wu, C.-C. and Chang, I.-C. (2008) 'The development of a computer auditing system sufficient for Sarbanes-Oxley section 404 -- a study on the purchasing and expenditure cycle of the ERP system', *Information Systems Management*, Vol. 25, No. 3, pp. 211-229.

Cohen, M.D. (1991) 'Individual learning and organizational routine: emerging connections', *Organization Science*, Vol. 2, No. 1, pp. 135-139.

Cohen, M.D. and Bacdayan, P. (1994) 'Organizational routines are stored as procedural memory: evidence from a laboratory study', *Organization Science*, Vol. 5, No. 4, pp. 554-568.

Cooper, R.B. and Zmud, R.W. (1990) 'Information technology implementation research: a technological diffusion approach', *Management Science*, Vol. 36, No. 2, pp. 123-139.

Davis, F.D., Bagozzi, R.P. and Warshaw, P.R. (1989) 'User acceptance of computer technology: a comparison of two theoretical models', *Management Science*, Vol. 35, No. 8, pp. 982-1003.

Debreceny, R., Lee, S.-L., Neo, W. and Toh, J.S. (2005) 'Employing generalized audit software in the financial services sector: challenges and opportunities', *Managerial Auditing Journal*, Vol. 20, No. 6, pp. 605-618.

Dowling, C. and Leech, S. (2007) 'Audit support systems and decision aids: current practice and opportunities for future research', *International Journal of Accounting Information Systems*, Vol. 8, No. 2, pp. 92-116.

Dutton, J.M. and Thomas, A. (1984) 'Treating progress functions as a managerial opportunity', *Academy of Management Review*, Vol. 9, No. 2, pp. 235-247.

Elliott, R.K. (2002) 'Twenty-First century assurance', *Auditing: A Journal of Practice and Theory*, Vol. 21, No. 1, pp. 139-146.

Elliott, R.K. and Jacobson, P.D. (1997) 'Adding value to audits', *CA Magazine*, Vol. 130, No. 9, pp. 35-37.

Feldman, M.S. and Pentland, B.T. (2003) 'Reconceptualizing organizational routines as a source of flexibility and change', *Administrative Science Quarterly*, Vol. 48, No. 1, pp. 94-118.

Foguem, B.K., Coudert, T., Béler, C., and Geneste, L. (2008) 'Knowledge formalization in experience feedback processes: an ontology-based approach', *Computers in Industry*, Vol. 59, No. 7, pp. 694-710.

Gallegos, F. and Looho, A. (2000) 'IS audit training needs for the 21st century: a selected assessment', *Journal of Computer Information Systems*, Vol. 41, No. 2, pp. 9-15.

Gehrke, N. and Wolf, P. (2010) 'Towards audit 2.0 -- a web 2.0 community platform for auditors', paper presented at the Forty-Third Hawaii International Conference on System Sciences, Koloa, HI, January 5-8.

Hammer, M. (1990) 'Reengineering work: don't automate, obliterate', *Harvard Business Review*, July-August, pp. 104-112.

Huang, S.-M., Hung, Y.-C. and Tsao, H.-H. (2008) 'Examining the determinants of computer-assisted audit techniques acceptance from internal auditors' viewpoints', *International Journal of Services and Standards*, Vol. 4, No. 4, pp. 377-392.

Janvrin, D., Lowe, D.J. and Bierstaker, J. (2008) 'Auditor acceptance of computer-assisted audit techniques', Working paper, College of Business, Iowa State University, Ames, IA, December.

Johnson, B.M. and Rice, R.E. (1987) *Managing Organizational Innovation: The Evolution from Word Processing to Office Information Systems*, Columbia University Press, New York, NY.

Julian, J. (2008) 'How project management office leaders facilitate cross-project learning and continuous improvement', *Project Management Journal*, Vol. 39, No. 3, pp. 43-58.

Kogan, A., Sudit, E.F. and Vasarhelyi, M.A. (1999) 'Continuous online auditing: a program of research', *Journal of Information Systems*, Vol. 13, No. 2, pp. 87-103.

Kwon, T.H. and Zmud, R.W. (1987) 'Unifying the fragmented models of information systems implementation', in Boland, R.J. and Hirschheim, R.A. (Eds.), *Critical Issues in Information Systems Research*, John Wiley and Sons, New York, NY, pp. 227-251.

Lanza, R.B. (2005) 'What are the common reasons to not use audit software?', *AuditSoftware. Net*, available at http://www.auditsoftware.net/community/why/articles/commonNOT.htm (accessed 3 March 2010).

Levitt, B. and March, J.G. (1988) 'Organizational learning', *Annual Review of Sociology*, Vol. 14, No. 1, pp. 319-338.

Lewin, K. (1952) 'Group decision and social change', in Swanson, G.E. (Ed.), *Readings in Social Psychology*, Holt, New York, NY, pp. 459-473.

Li, S.-H., Huang, S.-M. and Lin, Y.C. (2007) 'Developing a continuous auditing assistance system based on information process models', *Journal of Computer Information Systems*, Vol. 48, No. 1, pp. 2-13.

Lindow, P.E. and Race, J.D. (2002) 'Beyond traditional audit techniques', *Journal of Accountancy*, Vol. 194, No. 1, pp. 28-33.

Lord, A.T. (2004) 'ISACA model curricula 2004', *International Journal of Accounting Information Systems*, Vol. 5, No. 2, pp. 251-265.

Mahzan, N. and Lymer, A. (2010) 'Adoption of computer assisted audit tools and techniques (CAATTs) by internal auditors: current issues in the UK', *JPS Accounting Forums*, available at http://www.jps-dir.com/Forum/uploads/1364/Mahzan.pdf (accessed 3 March 2010).

Nelson, R.R. and Winter, S.G. (1982) *An Evolutionary Theory of Economic Change*, Belknap Press of Harvard University Press, Cambridge, MA.

Ramamoorti, S. and Weidenmier, M.L. (2004) 'The pervasive impact of information technology on internal auditing', in Bailey, A.D., Jr., Gramling, A.A. and Ramamoorti, S. (Eds.), *Research Opportunities in Internal Auditing*, The Institute of Internal Auditors Research Foundation, Altamonte Springs, FL, pp. 301-377.

Rezaee, Z., Elam, R. and Sharbatoghlie, A. (2001) 'Continuous auditing: the audit of the future', *Managerial Auditing Journal*, Vol. 16, No. 3, pp. 150-158.

Rice, R.E. and Rogers, E.M. (1980) 'Reinvention in the innovation process', *Science Communication*, Vol. 1, No. 4, pp. 499-514.

Searcy, D., Woodroof, J. and Behn, B. (2003) 'Continuous audit: the motivations, benefits, problems, and challenges identified by partners of a big 4 accounting firm', *Proceedings of the Thirty-Sixth Hawaii International Conference on Systems Sciences (HICSS' 03)*, Big Island, HI, January 6-9, pp. 210-219.

The Institute of Internal Auditors (2009) '2009 IT audit benchmarking study: executive summary and report', *GAIN Annual Benchmarking Study*, available at http://www.theiia.org/guidance/benchmarking/gain/knowledge-services/2009-it-audit-benchmarking-study/ (accessed 3 March 2010).

Tyre, M.J. and Orlikowski, W.J. (1994) 'Windows of opportunity: temporal patterns of technological adaptation in organizations', *Organization Science*, Vol. 5, No. 1, pp. 98-118.

Van de Ven, A.H. (1986) 'Central problems in the management of innovation', *Management Science*, Vol. 32, No. 5, pp. 590-607.

Vasarhelyi, M.A. (2002) 'Concepts in continuous assurance', in Arnold, V. and Sutton, S.G. (Eds.), *Researching Accounting as an Information Systems Discipline*, American Accounting Association, Sarasota, FL, pp. 257-271.

Zhao, N., Yen, D.C. and Chang, I.C. (2004) 'Auditing in the e-commerce era', *Information Management and Computer Security*, Vol. 12, No. 5, pp. 389-400.

About the author

Chia-Ming Sun is an Assistant Professor in the Department of Accounting at the National Yunlin University of Science and Technology in Taiwan. He received his Ph.D. in information management from National Chiao Tung University. His research interests include information technology auditing, information technology governance, and enterprise information systems. He has industrial experiences and worked as consultant, information system manager and information system auditor in manufacturing companies.

MIS Review Vol. 17, No. 2, March (2012), pp. 87-98

The Application of Healthcare Information System for Comprehensive Geriatric Assessment

Nai-Wen Kuo, Yao-Yu Chung
Department of Information Management, Chinese Culture University

ABSTRACT: *This paper is to integrate information technology and medical-related technologies to develop a healthcare information system for Comprehensive geriatric assessment (CGA). This system not only can process geriatric consultation services and ensure that all patient's information are stored in standardized format, but also provide medical personnel for statistical analysis and processing purposes. Moreover, this paper uses the Apriori algorithm of data mining for helping doctors to find out the relationship of geriatric syndrome. The systems of this study can enable organizations to meet their business objectives for increasing service capacity, cost control, revenue generation, while maintaining high quality of care for the patients in geriatric care. Furthermore, making the theories and applications of medical informatics will be more extensive and convenient for researches and healthcare-related industry.*

KEYWORDS: *Comprehensive Geriatric Assessment, Healthcare Information System, Data Mining, Association Rules, Elderly, Hospitals.*

1.Introduction

The number of older persons in Taiwan requiring healthcare will progressively increase over the next few decades. However, the population group that currently constitutes the majority of the health-care workforce will remain virtually stable (Kuo, 2010). Information technology (IT) plays an increasingly central role in the Taiwan healthcare industry. Using IT in a strategic and innovative manner to support health-related decision making represents a serious challenge for health care organization management, as well as for system developers (Chen, 2009). The hospitals also consider how to handle mutual relationship with patients by IT planning and evaluation. Thus, it is key issues to establish good connection to meet the needs by means of decision support information system (Dowling, 1980). The applications concentrate on giving an organization an IT-based strategy for meeting competitive challenges (such as by using emerging Web technologies to integrate healthcare organizations internally and externally) (Yang & Hwang, 2006).

Comprehensive geriatric assessment (CGA) is a core procedure in specialist geriatric care. There is evidence that this process improves functional recovery, reduces morbidity and attenuates demand for long-term institutional care (Stuck et al., 1993). It is central to geriatric consultation services which are the vehicle for delivering CGA to hospital patients who are not located in specialist geriatric units. Evidence of the impact of geriatric consultation on patient outcomes is mixed (Gray, 2007). However, geriatric consultation incorporates important triaging decisions to inpatient geriatric assessment, rehabilitation, long-term institutional care and complex community support programs. The system of this paper can upgrade administration information, increase service capacity, reduce personnel costs, and improve the quality of patient care in geriatric medicine. The system not only can help medical institutions to collect patient's relevant information and allow the contents of standardized assessment's form in order to increase diagnostic accuracy, but also enhance the quality of geriatric care environment in medical centre.

2. Background

2.1 Comprehensive geriatric assessment

CGA is a kind of integration of a variety of professional areas of assessment methods. It's a multidimensional assessment consisting of functional, emotional and cognitive components, and focused on a few of the many possible domains that are often reflected in the geriatric and gerontology literature (Overcash et al., 2005).

A CGA can be effective only if there is a process for identifying elderly patients who may benefit from it. In most cases, they are elderly individuals who are frail and disabled or have multiple interacting conditions, as opposed to relatively healthy older people (including those whose health conditions are addressable by usual medical approaches) and those with serious focal chronic conditions for whom disease management by primary care with input by other subspecialists is appropriate (Leung, 2004). Examples include an older patient who appears to be on a rapid downward trajectory toward nursing home placement and a previously functional senior who is requiring increasing assistance to accomplish daily tasks, triggered by the project through the assessment concluded that the results with a variety of professional diagnostic and disposal of frail elderly people to discover the physical, psychological, social and functional problems, and pinpoint the problems and to present a complete package of programs and appropriate treatment, thus improving the elder's disease symptoms (Maas et al., 2007).

CGA offers a healthcare model that integrates medical care with social support, and could be carried out in wide variety of settings including: acute hospital units, chronic hospitals, inpatient geriatric consultation, outpatient department, nursing home, and home visits, helping medical personal to assess older patients include the measure of ability in different areas, monitor changes, evaluate treatment effects, quality assurance, care planning and funding arrangement. Table 1 provides a list of basic components usually included in a CGA. While the detailed elements vary, virtually all CGAs -- whether relatively simple multidimensional assessments for screening purposes or fully elaborated team assessments -- include medical, psychological, social, and environmental components, as well as functional components (at the level of activities of daily living [ADLs] and instrumental activities of daily living [IADLs]) (Solomon, 1988).

Table 1 Components of Comprehensive Geriatric Assessment

Component	Elements
Medical assessment	Problem list Comorbid conditions and disease severity Medication review Nutritional status
Assessment of functioning	Basic activities of daily living Instrumental activities of daily living Activity/exercise status Gait and balance
Psychological assessment	Mental status (cognitive) testing Mood/depression testing
Social assessment	Informal support needs and assets Care resource eligibility/financial assessment
Environmental assessment	Home safety Transportation and telehealth

2.2 Data mining

The successful application of data mining in highly visible fields like e-business, marketing and retail have led to the popularity of its use in knowledge discovery in databases (KDD) in other industries and sectors. Among these sectors that are just discovering data mining are the fields of medicine and public health. This research paper provides a survey of current techniques of KDD, using data mining tools for healthcare and public health. It also discusses critical issues and challenges associated with data mining and healthcare in general. The research found a growing number of data mining applications, including analysis of health care centers for better health policy-making, detection of disease outbreaks and preventable hospital deaths, and detection of fraudulent insurance claims (Bellazzi & Zupan, 2008; Huang, Chen & Lee, 2007).

In computer science and data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions, or having no timestamps (DNA sequencing). As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a “bottom up” approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found (Agrawal & Srikant, 1994; Delen, 2009).

2.3 The aims of the system

The system for delivery of CGA was developed in response to three primary challenges (Kuo, 2011; Kuo & Chung, 2010):

1. The assessment information of patient is not easy to save and hard to further data analysis (e.g., data mining).
2. Geriatric consultation is delivered by geriatricians and gerontic nurses, sometimes with support from other allied health specialists. This process is time consuming and involving much human resource.
3. The dependence on rare and expensive geriatrician time is reduced without compromising quality of care or accuracy of decisions.

To resolve these problems, this paper combined information technology and medical-related technology to develop a healthcare information system. The system is a computer-based medical information system. It also is an interactive computer information system that can help medical personal to complete their job.

3. The proposed system

3.1 Case description

This case study, selecting a medical center of Taiwan Teaching Hospital in northern. This medical center was established in 2005, which includes 10 bed geriatric units, 4 clinic rooms, with four attending physicians, a number of specialist physicians, case specialist and assistant to provide elder patients a full range of services.

The geriatric consultation service assesses patients who may require health care or permanent residential care. Case specialist provides the initial assessment and comprehensive geriatric assessment, and assistant entered online by the information system. After finished the assessment, the report will show on the system, the patient's data can further statistics analysis automatically and generating further reports immediately. These comprehensive reports can replacement for hand-written progress notes. The attending physicians or specialist physicians will assess the patient's symptoms and propose data, with the physician's own professional knowledge and long-term accumulated experience to do a correct medical diagnosis.

3.2 The basic schema

The proposed system basic schema for CGA is as follows (Figure 1).

1. The patient is initially assessed by a training doctor of the clinic.
2. Assessments are administered by nurses with expertise in aged care. They draw on information derived from patient interview and observation, interview of direct care staff and family members and from the medical record.
3. Enabling assessment record sharing among the consultation team and other stakeholders.

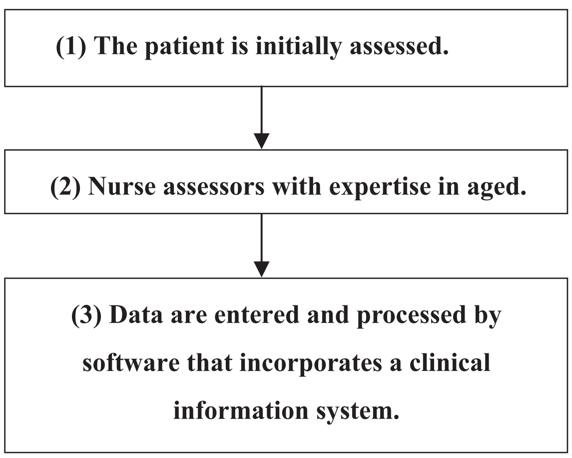

Figure 1 The Proposed System Basic Schema

3.3 System function framework

The proposed system is divided into three subsystems: data management, follow-up, query and analysis.

3.3.1 Data management subsystem

It contains the patient data management, patient functional assessment, drug evaluation, the overall project evaluation and treatment satisfaction survey.

3.3.2 Follow-Up subsystem

It contains the patient recorded the data of assessment, follow-up management and satisfaction survey the views of inspection. The subsystem mainly uses the object as an assistants and case specialists. Follow-up patients' change is also an important part of the assessment, it can be an efficient filter to help users keep track of patient and record whether the patient's body function decline of the situation, and necessary in order to provide patient-related assistance and treatment.

3.3.3 Query and analysis subsystem

This subsystem contains the results of patient assessment information, the assessment scale analysis diagram and satisfaction survey analysis diagram. The subsystem of the main user is attending physicians and specialist physicians. Query and analysis subsystem can assist doctor quick overview an assessment of each patient's complete information in order to give a proper diagnosis and follow-up treatment of movement; the subsystem can also analysis the contents of the database, integrated the information into a clear image, so that physicians can be targeted medical center for all patients, making a comprehensive assessment in order to understand the majority of patients prone to common diseases. The completely function interface of healthcare information system see Figure 2.

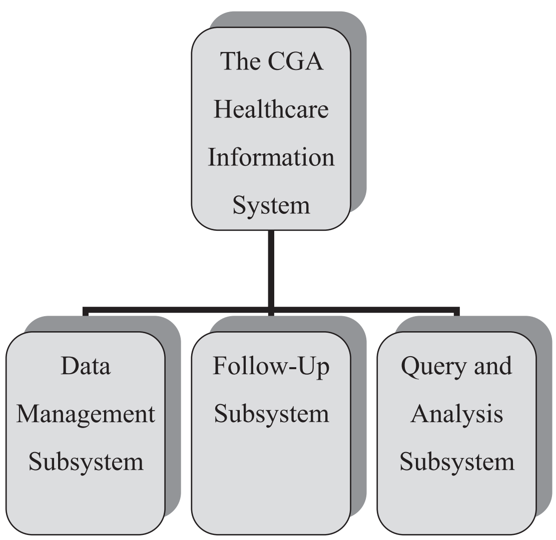

Figure 2 The System Function Framework

4. Result discussion

The 980 samples of assessment scale in this paper will be collected and categorized, which contained all elderly patients over 65 years old (630 males and 350 females). Using the Weka software tool for further analysis (data mining), which with built-in association rules of the Apriori algorithm (Figure 3). Weka is an open source data mining framework, integrates multiple algorithms for classification, clustering, association rule, etc., and supports abundant data Input/Output (I/O) and visualization functionalities (Sigurdardottir, Jonsdottir & Benediktsson, 2007).

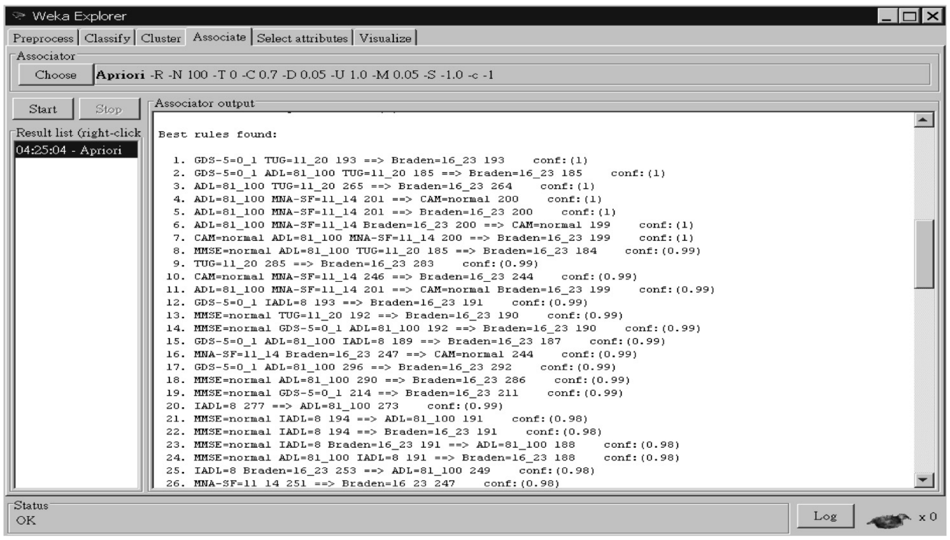

Figure 3 The Association Results of Apriori Algorithm

Making multiple association rules analysis and summarized a holistic integration result, we found out five common aging diseases (symptoms) for helping doctors to find out the relationship of geriatric syndrome to identify potential patients in the assessment and the relevance of the information (Table 2).

The five symptoms were dementia, disability living, malnutrition, falls and polypharmacy as follows:

4.1 Dementia

Clock drawing test (CDT) contains the evaluation function of action behavior, planning and implementation, graphics construction, visual-spatial understanding, number sense and other cognitive mental status, so the evaluation can directly judge the relationship with dementia. Mini nutritional assessment (MNA) and falls risk assessment

Table 2 The Five Aging Symptoms Analysis

	Dementia	Disability living	Malnutrition	Falls	Polypharmacy
MMSE (mini mental state examination)	--	●		●	
CDT (clock drawing test)	●				
GDS (geriatric depression scale)		●	●	●	●
ADL (activities of daily living)		--	●		
MNA (mini nutritional assessment)	●		--	●	●
Braden					
Falls (falls risk assessment)	●	●		--	
TUG (timed up and go test)		●		●	

(Falls) leading dementia of the elderly caused by malnutrition may often forget to eat food or a pica. On the other hand, the elderly fear of falling to reduce activity or prolonged bed rest, will cause some problems of variation in the metabolic response, decreased exposure to outdoor activities and the opportunity to interact with others, which will lead to the cognitive problems or dementia at the psychological level.

4.2 Disability living

The main reason of mini mental state examination (MMSE) function leading disability living is abnormal for the ability of patient's memory, construction, operation and execution. This will affect the function of activities of daily living normal execution, for example, patients may have some problems to perform some daily activities, such as brushing teeth, bathing, walking.

Geriatric depression scale (GDS) may be complicated because of severe depression symptoms or dementia caused by depression cases. Patients with depression may have delusions, auditory hallucinations and other psychotic symptoms, which leading to inappropriate behavior, such as difficulty walking or unable to interact with others, these acts can lead to disability living. Patients with the elderly most common diseases of limbs problems, musculoskeletal problems, a degenerative joint disease or osteoporosis are usually easy to fall, and may be loss of the ability of daily activities.

Falls can easily connect with disability living. The elderly patients with timed up and go test (TUG) problem often limit walking distance caused to the poor activities of daily living, it will also lead to functional disability of daily living. Sherman (2001) pointed out that if the subjects can fulfill assessment within ten seconds, we can predict the patients' activities of daily living (ADL) function will be able to maintain stability.

4.3 Malnutrition

GDS related to the elderly suffering from depression, because of emotional and psychological stress impacting, it leading unwilling to eat, appetite and other phenomena, or even serious anorexia, which making elderly patients occur malnutrition caused by nutrition ingestion insufficiency.

The main reason of ADL affecting malnutrition is reducing daily activity. Whenever patients have abnormal phenomena of eating, shopping, food preparation or action function, it will let the patient being nutrition risk caused by nutrition ingestion insufficiency. Usually, these cases occurred in elderly people of living alone or lacking proper care.

4.4 Falls

MMSE and GDS are associated with falls risk. It is very important information that whenever the elderly patient with abnormal mental state, they may increase the risk of falls, we should improve the psychological impact of the falls risk disease. In addition, MMSE is not entirely abnormal psychological problems, it also occurs in neural disorders, so sometimes need to do further assessment of physical condition.

MNA associated with falls risk patients may be physical weakness caused by malnutrition. TUG and falls risk shows more direct relationship. For example, the elderly patients with poor standing in walking test performance, which often have moving problems and need rehabilitation therapist for further careful evaluation. Moreover, basic information of the elderly patients reveals the visual impairment and sleep problems also related to the falls risk.

4.5 Polypharmacy

Polypharmacy is common problem of elderly patients. The association rules shown GDS, MNA and sleep problems are associated with polypharmacy. GDS is a mental illness, there would be more easily with complicated on medication.

The elderly may take some medications (such as antibiotics, Aspirin and other drugs) causing weight loss, loss of appetite, nausea, and malabsorption problems. Sleep problems can be directly associated with sleep aids drugs. The three drugs need to be careful consideration to avoid elderly people taking too many drugs causing other physical or psychological problems.

5. Conclusion

Comprehensive geriatric assessment is a core procedure in specialist geriatric care. There is evidence that this process improves functional recovery, reduces morbidity and attenuates demand for long-term institutional care. This paper is intended to integrate information technology and medical-related technologies to develop a comprehensive geriatric assessment healthcare information system for geriatric care. This proposed system is divided into three subsystems: data management, follow-up, query and analysis subsystem, which not only can process geriatric consultation services, and ensure that all patient's information are stored in standardized format, but also provide medical personnel for statistical analysis and processing purposes. Making multiple association rules analysis and summarized a holistic integration result, we will find out some common aging diseases for helping doctors to find out the relationship of geriatric syndrome to identify potential patients in the assessment and the relevance of the information. Moreover, coupled with professional medical team to assess and provide comprehensive healthcare service plans for improving the situation of the elderly patients. The systems of this study can enable organizations to meet their business objectives for increasing service capacity, cost control, revenue generation, while maintaining high quality of care for the patients in geriatric care. Preliminary evaluation suggests the system to be reliable, safe, efficient and appealing to clinicians.

We have the following conclusion:

5.1 Electronic medical information record

This system not only can store patient's complete information in the database, but also ensure that all patients' information stored in standard format and easy to follow-up. It not only can achieve paperless assessment, but also reduce the personnel and resource costs.

5.2 Standard geriatric assessment

The assessment system therefore acts as a clinical decision support system. For less-experienced practitioners it serves the purpose of interpreting basic observations to enhance the diagnostic and evaluation performance of the assessor. It also ensures that all patient information is stored unified format to provide medical personnel for further statistical analysis purposes.

5.3 Enhance the quality of geriatric care environment

Use of information system's assessment results to assist the medical personnel based on individual patient's needs and symptoms, formulate a comprehensive patient care plan, coupled with our professional medical team to assess and provide comprehensive health care service plans to help improve the situation of the elderly patients. The system of this paper can upgrade administration information, increase service capacity and diagnostic accuracy, it also enhance the quality of geriatric care environment in medical centre.

5.4 Identify the potential symptoms of patients

Making multiple association rules analysis and summarized a holistic integration result, we found out five symptoms (dementia, disability living, malnutrition, falls and polypharmacy) for helping doctors to find out the relationship of geriatric syndrome to identify potential patients in the assessment and the relevance of the information.

References

Agrawal, R. and Srikant, R. (1994) 'Fast algorithms for mining association rules in large databases', *Proceedings of the Twentieth International Conference on Very Large Data Bases*, Santiago, Chile, September 12-15, pp. 487-499.

Bellazzi, R. and Zupan, B. (2008) 'Predictive data mining in clinical medicine: current issues and guidelines', *International Journal of Medical Informatics*, Vol. 77, No. 2, pp. 81-97.

Chen, L.-K. (2009) 'The health challenges of aging society', *Taipei Veterans General Hospital Magazine*, Vol. 144, pp. 2-8.

Delen, D. (2009) 'Analysis of cancer data: a data mining approach', *Expert Systems*, Vol. 26, No. 1, pp. 100-112.

Dowling, A.F., Jr. (1980) 'Do hospital staff interfere with computer system implementation?', *Health Care Management Review*, Vol. 5, No. 4, pp. 23-32.

Gray, L. (2007) 'Geriatric consultation: is there a future?', *Age and Ageing*, Vol. 36, No. 1, pp. 1-2.

Huang, M.-J., Chen, M.-Y. and Lee, S.-C. (2007) 'Integrating data mining with case-based reasoning for chronic diseases prognosis and diagnosis', *Expert Systems with Applications*, Vol. 32, No. 3, pp. 856-867.

Kuo, N.-W. (2010) 'A decision support system in geriatric care', *Proceedings of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2010)*, Darmstadt, Germany, October 15-17, pp. 688-691.

Kuo, N.-W. (2011) 'Geriatric care in healthcare information system', *Advanced Materials Research*, Vols. 181-182, pp. 151-155.

Kuo, N.-W. and Chung, Y.-Y. (2010) 'Construction of comprehensive geriatric assessment information system -- a case study of medical center in Taiwan', *Proceedings of the 2010 INFORMS Service Science Conference*, Taipei, Taiwan, July 7-10, pp. 255-261.

Leung, E.M.F. (2004) 'Team approach and comprehensive geriatric assessment', *Geriatrics and Gerontology International*, Vol. 4, Suppl. 1, pp. S41-S44.

Maas, H.A.A.M., Janssen-Heijnen, M.L.G., Rikkert, M.G.M.O. and Wymenga, A.N.M. (2007) 'Comprehensive geriatric assessment and its clinical impact in oncology', *European Journal of Cancer*, Vol. 43, pp. 2161-2169.

Overcash, J.A., Beckstead, J., Extermann, M. and Cob, S. (2005) 'The abbreviated comprehensive geriatric assessment (aCGA): a retrospective analysis', *Critical Reviews in Oncology/Hematology*, Vol. 54, No. 2, pp. 129-136.

Sherman, F.T. (2001) 'Functional assessment: easy-to-use screening tools speed initial office work-up', *Geriatrics*, Vol. 56, No. 8, pp. 36-40.

Sigurdardottir, A.K., Jonsdottir, H. and Benediktsson, R. (2007) 'Outcomes of educational interventions in type 2 diabetes: WEKA data-mining analysis', *Patient Education and Counseling*, Vol. 67, Nos. 1-2, pp. 21-31.

Solomon, D. (1988) 'National institutes of health consensus development conference statement: geriatric assessment methods for clinical decision-making', *Journal of the American Geriatrics Society*, Vol. 36, pp. 342-347.

Stuck, A.E., Siu, A.L., Wieland, G.D., Rubenstein, L.Z. and Adams, J. (1993) 'Comprehensive geriatric assessment: a meta-analysis of controlled trials', *The Lancet*, Vol. 342, No. 8878, pp. 1032-1036.

Yang, W.-S. and Hwang, S.-Y. (2006) 'A process-mining framework for the detection of healthcare fraud and abuse', *Expert Systems with Applications*, Vol. 31, No. 1, pp. 56-68.

About the authors

Nai-Wen Kuo is an Associate Professor in Department of Information Management at Chinese Culture University in Taiwan. His research interests include: E-commerce, healthcare, system dynamics.

Yao-Yu Chung got M.S. Degree from Chinese Culture University in Taiwan.

CALL FOR PAPER

MIS Review: An International Journal

Published 2 Issues Annually by Airiti Press Inc.

MIS Review is a double-blind refereed academic journal published jointly by Airiti Press Inc. and Department of Management Information Systems, College of Commerce, National Chengchi University in Taiwan. The journal is published both in print and online. We welcome submissions of research papers/case studies in the areas including (but not limited to):

1. MIS Roles, Trends, and Research Methods

Roles, positioning and research methods of management information systems, and the impacts & development trends of information technology on organizations.

2. Information Management

Information infrastructure planning and implementation, information technology and organizational design, strategic applications of information systems, information system project management, knowledge management, electronic commerce, end-user computing, and service technology management.

3. Information Technologies

Database design and management, decision support systems, artificial intelligence applications (including expert systems and neural networks), software engineering, distribution systems, communication networks, multimedia systems, man-machine interface, knowledge acquisition & management, data mining, data warehouse, cooperative technology, and service science & engineering.

4. Information Applications and Innovations

The applications and innovations of business functional information systems (e.g., production, marketing, financial, human resources, and accounting information systems), enterprise resource planning, customer relationship management, supply chain management, intellectual capital, geographic information systems, and integrated information systems.

5. Information Technology Education and Society

Information education, e-learning, and information impacts on society.

6. Others

Other MIS-related topics.

INSTRUCTIONS FOR SUBMISSION

1. Papers can be prepared in either Chinese or English. If your paper is written in Chinese, it will be translated into English once it is accepted for publication.

2. There is no submission deadline for MIS Review. All papers will be double-blind reviewed by at least two reviewers, who will be recommended by the Editorial Board. The processing time for the first-round formal reviews is about six weeks. Subsequently rounds of reviews tend to be faster.

3. To simplify file conversion effort, PDF or Microsoft Word 2000/2003 (for Windows) format is advised. Then, please submit your paper via the MIS Review website (URL: http://www.icebnet.org/misr/).

4. MIS Review is an academic journal. According to international practice, once an article is accepted and published, MIS Review will not give or take any payment for the publishing. An electronic copy of the paper will be sent to the article author(s) for non-profit usage.

5. The submitted and accepted paper should follow the author guidelines for paper submission format provided on the MIS Review website.

The submitted paper should include the title page, abstract, key words, the paper body, references, and/or appendices. You must submit three files. The information of author(s) should not appear anywhere in the paper body file, including page header and footer.

1. On a separate (cover letter) file, please follow the author guidelines provided on the MIS Review website to prepare the letter.

2. On a separate (title page) file, please note the title of the paper, names of authors, affiliations, addresses, phone numbers, fax numbers, and E-mail addresses.

3. On a separate (paper body) file, please include the paper title, an abstract, a list of keywords, the paper body, the references, and/or appendices. The abstract must contain the research questions, purposes, research methods, and research findings. The abstract should not exceed 500 words and the number of keywords must be 5-10 words.

4. The submitted and accepted paper should follow the author guidelines for paper submission format provided on the MIS Review website.

CONTACT

Editorial Assistant
Department of Management Information Systems
College of Commerce, National Chengchi University
No. 64, Sec. 2, ZhiNan Road, Wenshan District,
Taipei 11605, Taiwan R.O.C.
Phone: +886-2-29393091 ext.89055
E-mail: misr@mis.nccu.edu.tw

airiti press Subscription Form

MIS Review

You may subscribe to the journals by completing this form and sending it by fax or e-mail to

Address: 18F., No. 80, Sec. 1, Chenggong Rd., Yonghe District, New Taipei City 23452, Taiwan (R.O.C.)
Tel: +886-2-29266006 ext. 8301 Fax: +886-2-22317711 E-mail: press@airiti.com Website: http://www.airitipress.com

PERSONAL				LIBRARIES / INSTITUTIONS			
	Europe	US/CA	Asian/Pacific		Europe	US/CA	Asian/Pacific
1 Issue	€ 34	US$ 41	US$ 38	**1 Issue**	€ 53	US$ 65	US$ 62

Vol.		No.		~	Vol.		No.		Copies		US$		Total US$	

*All Price include postage

PLEASE NOTE

- Issues will be sent in two business days after receiving your payment.
- Please note that all orders must be confirmed by fax or email.
- Prices and proposed publication dates are subject to change without notice.
- Institutions include libraries, government offices, businesses, and for individuals where the company pays for the subscription.
- Personal rates are available only to single-user personal subscribers for personal and non-commercial purposes.
- Airiti Press reserves its right to take appropriate action to recover any losses arising fromany intended or unintended misrepresentation of the term "Personal Subscriber".

BILLING INFORMATION

Name	
Company	
Tel	Fax
E-mail	
Shipping Address	

INTERNATIONAL PAYMENTS

Pay by Credit Card	
Card Type ☐JCB ☐MasterCard ☐Visa	
Card Name	
Card Number	
Expiry Date ___________ / ___________	CVV number
Signature	

Direct Bank Transfer	
Beneficiary	AIRITI INC.
Address	18F., No. 80, Sec. 1, Chenggong Rd., Yonghe District, New Taipei City 23452, Taiwan (R.O.C.)
Bank Name	E.Sun Commercial Bank, Ltd.Yong He Branch
Account No	0107441863017
Swift Code	ESUNTWTP
Bank Address	No.145, Zhongzheng Rd., Yonghe District, New Taipei City 23454, Taiwan (R.O.C.)