Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2014 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2014

AN EMPIRICAL STUDY OF LEAN AND AGILE INFLUENCES IN SOFTWARE CONFIGURATION MANAGEMENT

Usman Durrani
RMIT University, usman.durrani@rmit.edu.au

Zijad Pita
RMIT University, zijad.pita@rmit.edu.au

Joan Richardson
RMIT University, Joan.Richardson@rmit.edu.au

John Lenarcic
RMIT University, joan.lenarcic@rmit.edu.au

Recommended Citation

Durrani, Usman; Pita, Zijad; Richardson, Joan; and Lenarcic, John, "AN EMPIRICAL STUDY OF LEAN AND AGILE INFLUENCES IN SOFTWARE CONFIGURATION MANAGEMENT" (2014). *PACIS 2014 Proceedings*. 320.
http://aisel.aisnet.org/pacis2014/320

AN EMPIRICAL STUDY OF LEAN AND AGILE INFLUENCES IN SOFTWARE CONFIGURATION MANAGEMENT

Usman Durrani, School of BITL, RMIT University, Melbourne, Australia, usman.durrani@rmit.edu.au

Zijad Pita, School of BITL, RMIT University, Melbourne, Australia, zijad.pita@rmit.edu.au

Joan Richardson, School of BITL, RMIT University, Melbourne, Australia, joan.richardson@rmit.edu.au

John Lenarcic, School of BITL, RMIT University, Melbourne, Australia, joan.lenarcic@rmit.edu.au

Abstract

Based on a web based survey of 158 IT software development practitioners, mostly from different organizations, this paper examines the application of Software Configuration Management (SCM) process in adaptable software development environments. From the literature review six independent variables and five dependent variables were derived to answer the research question "how does organization size impact on the usage of SCM process in adaptable software development environments?" The six hypotheses confirmed that organization size does not impact on the usage of SCM process in adaptable software development environments. Also, all sizes of organizations consider SCM process valuable and have similar software traceability approaches with different process formalities and tools sophistications. This study helps SCM practitioners to build a better understanding of the relationships between adaptable software development environment, lean principles and practices, and the tools and technologies established to facilitate the coexistence of Agile and SCM practices.

Keywords: Adaptable Software Development Environment, Agile Software Development, Lean Thinking, Software Configuration Management, SCM, Traceability.

1 INTRODUCTION

Rapid advancement in software and hardware technology took information systems (IS) to a new era, where people were constrained to find ways to manage and control their software development processes. Consequently, a new discipline was evolved, the Software Configuration Management (SCM), the concept that apply to all items to be controlled and traceable, although there are some differences in implementation between hardware Configuration Management and software Configuration Management (Abran & Moore, 2001). The SCM process is considered as a discipline for controlling the evolution of complex systems by developing and using standards and procedures for managing an evolving software systems (Sommerville, 2007).

According to Leon (2005), SCM traceability approach should be implemented in all software projects – irrespective of size of the project and organization. The current approaches for traceability mainly rely on complete requirement specification artefacts. This is considered as a fundamental problem for the management of information in the agile environments. Hence, there is a need for new and specific approaches for traceability in agile software development methodologies, that may provide better integration among the various environments and tools to share traceability information (Aizenbud-Reshef, Nolan, Rubin, & Shaham-Gafni, 2006).

Studies have been performed to develop detailed traceability methodologies to address the question "how to improve" or more specifically "how to run a Software Process Improvement project" (McFeeley, 1996; Naveh & Erez, 2004; Shaikh, Ahmed, Memon, & Memon, 2009). Unfortunately, even these methodologies are still very generic and cannot directly answer questions, such as, "how to implement or improve the SCM process" or more specifically "how to run a Software Process Improvement project to implement or improve the SCM process".

This study focuses on empirical analysis of the SCM process in lean and agile software development environments in different size of organizations. This is achieved by analysing the relationship between six independent variables: organization size, lean thinking principles, agile methods, SCM practices, values of SCM process, and grades of software change traceability and five dependent variables: adaptable software development environment, software change traceability solutions, agile methods in adaptable software development environment, SCM practices in adaptable software development environment and values of SCM process in adaptable software development environment.

The study was only conducted in the software development organizations with the coexistence of agile methodologies and SCM process. For this study, the organization size was defined based on the number of employees directly related to the software development operations to study the complexity of software traceability strategies. Other economic contexts of organization size, such as market shares, organization financial worth etc. were left for the future study. This study will facilitate SCM practitioners by finding relationships between adaptable software development environment, lean principles and practices, and the tools and technologies (management systems) established to facilitate the coexistence of Agile and SCM process and practices.

The rest of paper proceeds as follows. The next section is the literature overview, followed by analysis of data based on a web based survey of 158 software development companies. Then the results of research hypotheses are presented. Finally, a summary of the survey results, limitation and the possibilities for further research is presented.

2 LITERATURE REVIEW

2.1 Software Configuration Management (SCM) Process and Framework

SCM is a software engineering process that makes software development traceable. SCM is central and essential services to all the major processes of systems and software engineering, such as, project management, requirement management, design, implementation, integration, verification, release/transition, and operation & maintenance. SCM comprises of seven primary lower-level processes and two special instances of applying those lower-level processes. Seven primary lower-level processes include: planning, management, configuration identification, configuration change control, configuration status accounting, configuration auditing, and configuration release management. Two special instances are interface control and supplier configuration item control (IEEE.Std.828, 2012).

Even though SCM activities can be initiated at any stage during a product's life cycle, it is better to have the SCM system in place from the beginning (Leon, 2005). As compared with early SCM systems, modern SCM systems provide a wide range of high-level functionality, such as, construction, structure, auditing, components, team, process, accounting, and controlling (Dart, 1990).

SCM framework is a layered structure which comprises of policy, process and practices to manage the life cycle of any software system. The framework, defines how to relate the requested system requirements to system components, and the methods used to identify different versions of the system. In other words, for a system to work effectively and efficiently, a SCM framework must be defined and implemented regardless of the software development methodology used.

Most SCM standards have an embedded assumption that a waterfall model will be used for system development (Bersoff & Davis, 1991). Regardless of its nature, the SCM process should be implemented by all projects without considering their size, complexity, or the stage (conceptual, design, development, testing or maintenance) of the project (Leon, 2005). Also, the standards have to be adapted to modern software development approaches, based on incremental specification and development.

2.2 Lean Thinking Principles, Tools and Practices

Leanness stands for the fundamental approach capable of minimizing the risks and side-effect in reengineering business process to adapt change in any direction (Luo, Zhang, & Ren, 1996). Lean is a way of thinking that enables organizations to "specify value, line up value creating actions in the best sequence, conduct these activities with interruption whenever someone requests them, and perform them more and more effectively"(Womack & Jones, 1996). Principles of lean are universal and are broadly accepted across many disciplines (Poppendieck, 2011). The main purpose of implementing lean manufacturing is to increase productivity, reduce lean time and cost, and improve quality (Wong, Wong, & Ali, 2009).

The five principles that drive this lean thinking process are: 1) define each product such that it precisely meets customer requirements (value), 2) identify the value stream for each product, 3) all value to flow through the value stream without delays or barriers, 4) all the customer to pull value rather than the manufacturer producing to forecasts (pull), 5) and pursue perfection and practice continuous improvement (perfection) (Womack & Jones, 1996).

2.3 Software Traceability

Traceability is "the ability to describe and follow the life of a requirement, in both a forward and backward direction"(Gotel & Finkelstein, 1994). At the most basic level, traceability is simply the opportunity to relate information that is recorded between objects, artefacts or management systems of some kind, along with its ability to examine the relationships within (Ramesh & Jarke, 2001).

Studies have investigated the phenomenon of traceability using different terminologies and looking at the relations between software documents (Nguyen & Munson, 2003), software artefacts (Spanoudakis), objects (Ramesh & Jarke, 2001), products of the development process (IEEE.Std.610.12-1990, 1990), product fragments, project item (Gills & Borzovs, 2005), sources and stakeholders (Ramesh & Jarke, 2001) etc. This research will look at traceability through SCM process from management systems perspective and their relations within the context of organization and project levels.

The basic idea behind using this perspective is that management systems are the entities that contain different types of sources which are then associated with various types of objects depending on syntax and semantics (Ramesh & Jarke, 2001). Also, the resulting management systems are managed, used, and administered by different kinds of stakeholders. This, as a result, covers traceability in a given context both, from process and product knowledge perspectives (Mohan, Xu, Cao, & Ramesh, 2008). This study defines the traceability as: "the ability of tracing from one management system containing particular types of objects and the sources to another based on defined syntax, semantic relations and implementation context".

This study also defines management system as "the implementation of a certain degree of process, practices and work instructions, facilitated by tools and technology to deliver specific and valuable information about defined sources and objects to the target stakeholders for the operational and strategic alignment of the business".

3 RESEARCH METHODOLOGY

The survey was emailed via Qualtrics tool and administered by RMIT University to the population Linkedin groups, as well as to the IT professional in the researcher's direct Linkedin network contacts. Total population for the main survey was approximately 1400 respondents. Majority of the participants were selected using purposive/judgement sampling through researcher's direct contacts. According to Cooper and Schindler (2003), judgement sampling occurs when a researcher selects sample members to comply to some criterion. The total number of surveys returned was 158, representing 11.28% response rate.

3.1 Research Questions

This study investigates the research question: how does organization size impact on the usage of SCM process in adaptable software development environments? Sub research questions (SRQs) that help to later address the main research question are:

- SRQ1: Does the organization size affect the implementation of adaptable software development environment?
- SRQ2: Does the organization size affect the implementation of SCM practices in the adaptable software development environment?
- SRQ3: Is the SCM process considered valuable by all size of organizations?
- SRQ4: Do different sizes of organizations have different software change traceability capabilities?
- SRQ5: Does the importance of various management systems associated with SCM process vary in different sizes of organizations?
- SRQ6: Does the empowerment of project teams to customize SCM process, associated management system linkage, and roles to perform SCM process, vary in different sizes of organizations?

3.2 Survey Structure

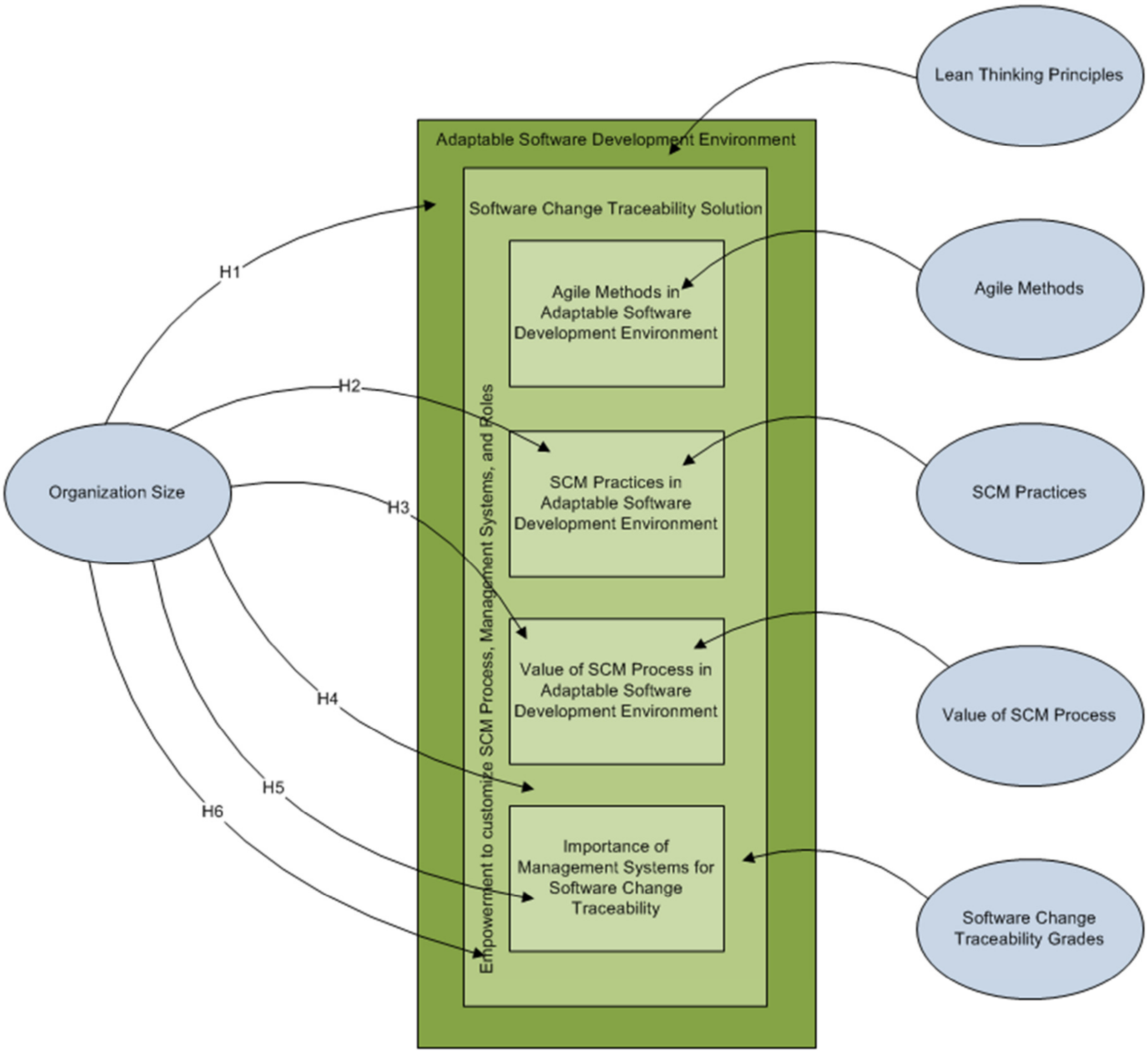
The survey structure include four sections consisting of ten demographic items, four Likert-type scale items that addressed Lean Thinking principles, SCM practices, value of SCM process, and grades of software change traceability solution, thirty five nominal items, and fourteen open-ended written response items.

Initial section addressed the participants' organization demographics and the identification of lean principles, ASD methods and its usage. The five point Likert-type scale was used for lean principle item. This followed by a section which addressed the participants' usages of SCM process, and the associated practices, along with reference to different improvement standards, usage frequency for the projects in the organization, and the value for the organization. The four and five point Likert-type scale was used for SCM practices and SCM process respectively.

The subsequent section then addressed the organizations' or projects' traceability capabilities and structure. It also identified the types of management systems used along with their collaboration or independence functioning. The five point Likert-type scale is used for grades of existing software traceability solution item. The last section of the survey included items 30 through 59 that identified specific tools used in various management systems categories and it perceived importance for the organization.

4 CONCEPTUAL FRAMEWORK

This study introduces six independent variables, five dependent variables and their associations presented as a conceptual framework (Figure 1) based on a literature review. The independent variables represented as the ovals and the rectangles represent dependent variables.



Lean Thinking Principles
Adaptable Software Development Environment
Software Change Traceability Solution
Agile Methods in Adaptable Software Development Environment
Agile Methods
H1
H2
SCM Practices in Adaptable Software Development Environment
SCM Practices
Organization Size
H3
Empowerment to customize SCM Process, Management Systems, and Roles
Value of SCM Process in Adaptable Software Development Environment
Value of SCM Process
H4
H5
H6
Importance of Management Systems for Software Change Traceability
Software Change Traceability Grades

Figure 1. Conceptual Framework

Once all the values of the dependent variables are identified through their associated independent variables, six hypotheses associated with independent variable "organization size" will statistically test the mean difference between three different organizational size groups, small organizations (1 to 20 employees), medium organizations (21 to 200 employees), and large organizations (more than 200 employees).

5 DATA COLLECTON AND ANALYSIS

The survey participants were located globally; therefore, Qualtrics was proved to be the best tool choice to collect data for the geographically diverse participants. This approach reduced the time and cost of the survey and allowed the participants to take the survey at their own convenience. The total length of the pilot and the main survey was 3 months. The target population was taken from the Linked-in groups as well as the IT professionals in the researchers' direct Linked-in network.

5.1 Assessment of Validity and Reliability of Survey Data

Since this research is of exploratory nature, a reliability analysis to test the internal consistency of the data is necessary for the independent variables so that each and every factor is ensured of a high level of reliability. The Cronbach's alpha as the most common method is used for calculating internal consistency reliability (Rubin & Babbie, 1997). For exploratory studies, it is agreed that a lowest Cronbach's alpha level of 0.6 could be deemed acceptable (Nunally, 1967). Reliability analysis was performed only on the independent variables with Likert-scale. All variables had the acceptable level of reliability, ranging from .776 to .938.

5.2 Descriptive Statistics and Demographic Analysis

The demographic data collected from the survey were categorized into five perspectives: organization profile; organization's software development environment and the associated culture; demographics of the SCM process in the agile organizations; its usage of process improvement standards, frequency of SCM process usage, and the customization to the process for the valuable implementation; traceability solutions which exist in the organizations; demographics of the management systems in place in the actual environment along with its value and perceived importance for the organizations.

5.2.1 Lean Practices, Agile Software Development Environment and Culture

Respondents were asked about various lean principles and practices in their software development environment and the associated organizational culture (Liker, 2004; Womack & Jones, 1996). Majority of the respondents (from all size organizations) either agreed or strongly agreed with the existence and the influence of these principles and properties.

In regards to use of agile software development methodology, out of 143 respondents, 44% selected "mixed agile methods". Scrum and FDD were identified as the next most widely used agile methods with a response rate of 28.7% and 10.5% respectively.

Based on the response received, we can argue that all sizes of software development organizations consider lean practices as a key ingredient in the development of agile organizational culture. It is evident from the fact that majority of the respondent organizations (all sizes) are using mixed agile software development methodology for their operations. Their practices include continuous assessment through value stream to identify value-adding or waste practices in their software development methodologies in alignment with the customer expectation.

5.2.2 SCM Process and Practices

From the total respondents, 74.1% confirmed that they have SCM process in their organizational environments.

Further analysis was performed to identify distribution of SCM implementation in terms of organization size based on employees directly related to software development. Only 11.6% from the sample group of small size organizations confirmed the existence of SCM process in their organization, while in the case of medium and large size organizations the existence of SCM process was 89% and 100% respectively.

Based on the responses received, we can argue that although the large software development organizations are still the biggest consumer of the SCM process, the evidence has shown that the small and especially the medium sized organizations are catching up in terms of SCM process maturity to their larger competitors. The key motivator for this maturity was the involvement of these small and medium size organizations in the software development projects where traceability was either mandated by the project sponsor or it was seen as an internal component of a quality system engineering process.

5.2.3 Software Traceability Solutions and the associated Management Systems

Majority of the respondents (86.8%) used SCM process as a key mechanism for providing software change traceability in their organizations. Only 8% responses were received from the small organizations. Remaining responses either belonged to group medium organizations 76% or large organizations 87%.

In regards to the motivation behind having the software change traceability, 8.5% respondents indicated that such traceability requirement was "Mandated from project sponsor", 47.2% respondents choose "Component of a quality system engineering process", and 35.8% respondents mentioned "Both" as a motivation behind software traceability solution. Only 6.6% respondents indicated "Neither" of the above option and two respondents have selected "Other". Based on the analysis of the responses received, it was revealed that software traceability solutions using SCM process exists in all size of the organizations but the process formality and the tool or technology sophistication varied based on the requirements mandated by the project sponsor or the improvement standard followed as a component of internal quality system engineering process.

5.2.4 Management Systems and its Value and Perceive Importance

Respondents were presented with a list of most commonly used categories of management systems (as initially identified through the literature review) and were asked to select the systems that exist in their organizations for the perspective of software change traceability. Respondents were further asked to identify if these management systems works "Independently" (means, that it neither expect any input from the external management system nor provide any output to the external management system), or "Collaboratively" (means, that it works in a cooperative mode of exchanging the information with external management systems). The response rate of 36.3% for independent use was reported and the response rate for collaborative use was 63.7%. Hence, we can argue that in most case the management systems exist in harmony and operate through collaboration with each other.

6 HYHPOTHESES RESULT

Parametric procedures are used and considered appropriate because the assumption of independence of data, normality of the distribution and homogeneity of variance were satisfied for all of the independent variables.

An analysis of variance (ANOVA) test was conducted to examine the lean thinking principles and properties, SCM practices, values of SCM process, and the grades (features) of software change traceability solutions that exist in agile software development organizations. Chi-Square test was also performed to verify the independence of organization size from the importance of different management systems studied.

6.1 Hypothesis 1

The first hypothesis was used to investigate “SRQ1: Does the organization size affect the implementation of adaptable software development environment?” The null hypothesis is: H01: Organization size has no affect in the establishing adaptable software development environment.

The results were not significant F(2, 155)=1.46, p >.05 (see Table 1) and indicated that lean thinking and properties exists in all sizes of software development organization. Respondents from small organizations (n=25) had a (Mean) M=50.82 and (Standard Deviation) SD=5.37. Respondents from medium organizations (n=47) had M=49.01 and SD=5.64, and the large organizations (n=86) had M=48.34 and SD=7.12; however, differences were not significant.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	119.694	2	59.847	1.434	.241
Within Groups	6467.643	155	41.727		

Table 1. ANOVA test for adaptable software development environment

On the basis of above ANOVA test we accept the null hypothesis (H01), “organization size has no effect in the establishing adaptable software development environment”. In other words, all sizes of organizations considers lean principles, tools, and practices (as stated in section 2.2) important for the development of adaptable software development environment. In addition, these organizations are frequently using these practices to identify value-adding software development processes, lining up these processes in the best sequence to deliver business goals.

6.2 Hypothesis 2

The second hypothesis is used to investigate “SRQ2: Does the organization size affect the implementation of SCM practices in the adaptable software development environment?” The null hypothesis is: H02: Organization size has no influence in the adoption of SCM practices.

The results were not significant F(2, 103)=0.89, p >.05 (see Table 2) and indicated that SCM practices exists in all sizes of software development organization. Respondents from small organizations (n=13) had an M=38.32 and SD=6.14. Respondents from medium organizations (n=28) had M=37.52, and SD=3.75, and the large organizations (n=65) had M=38.98, and SD=5.00; however, differences were not significant.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	42.307	2	21.153	.894	.412
Within Groups	2435.785	103	23.648		

Table 2. ANOVA test for Software Configuration Management

On the basis of above ANOVA test we accept the null hypothesis (H02), “organization size has no influence in the adoption of SCM practices”. In other words, the result confirms that it is a misconception that the software configuration process is only suitable for the large organizations and/or complex projects. Also, from this point of view, the embedded assumption that most SCM standards will use waterfall model for the system development can be challenged (Bersoff & Davis, 1991). It also confirms the statement of Leon (2004) that the software configuration management process should be implemented in all software projects irrespective of the size of the project and organization because the change is inevitable in all projects, and an unmanaged change is likely to cause trouble.

6.3 Hypothesis 3

The third hypothesis is used to investigate "SRQ3: Is the SCM process considered valuable by all size of organizations?" The null hypothesis is: H03: Organization size has no influence in the valuable usage of SCM process.

The results were not significant F(2, 103)=0.48, p >.05 (see Table 3) and indicated that values of SCM process is realized in all sizes of software development organization. Respondents from small organizations (n=13) had M=26.24, and SD=4.94. Respondents from medium organizations (n=28) had M=26.27 and SD=3.08, and the large organizations (n=65) had M=25.34 and SD=5.20; however, differences were not significant.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	21.096	2	10.548	.476	.623
Within Groups	2283.083	103	22.166		

Table 3. ANOVA test for Value of Software Configuration Management Practices

On the basis of above ANOVA test we accept the null hypothesis (H03), "organization size has no influence in the valuable usage of SCM process". In other words, researcher has not only found the evidence of the existence of software configuration management process in all sizes of organizations, but as a result of hypothesis 3, we can state that all sizes of organizations also considers software configuration management as valuable for their software development activities.

6.4 Hypothesis 4

The fourth hypothesis is used to investigate "SRQ4: Do different sizes of organizations have different software change traceability capabilities?" The null hypothesis is: H04: All sizes of organizations have similar level of software change traceability capabilities.

The results were not significant F(2, 103)=0.24, p >.05 (see Table 4) and indicated the existence of sophisticated software change traceability solution in all sizes of agile software development organization. Respondents from small organizations (n=13) had M=39.32 and SD=8.93. Respondents from medium organizations (n=28) had M =39.39 and SD=6.68, and the large organizations (n=65) had M=40.50 and SD=8.59; however, differences were not significant.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	32.021	2	16.010	.239	.788
Within Groups	6889.085	103	66.884		

Table 4. ANOVA Test for the software change traceability capabilities

On the basis of above ANOVA test we accept the null hypothesis (H04), "all sizes of organizations have similar level of software change traceability capabilities". In other words, all sizes of organizations have either agreed or strongly agreed that their software traceability solution have the characteristics of covering the entire software development lifecycle, simple, affordable, maintainable, accessible to all relevant stakeholders, scalable, customizable, auditable, and helping to reduce the cost and time to deliver the product.

On the other hand, all sizes of organizations have also given the similar responses in terms of the granularity of their existing traceability solution. This includes trace of product and process knowledge from:

- stakeholder and the requirements raised
- requirements raised and the iterations performed to complete the work
- requirement raised and all other related requirements
- requirements raised and code developed
- requirements raised and baseline under version control

- code developed and the associated baselines
- requirements raised and test performed
- test performed and the associated baseline
- baseline to build/release package
- build/release package and the target managed environment
- target managed environment to stakeholder evaluation

Regardless of the similarities that were identified in terms of the capabilities and granularities in different sizes of organizations, one aspect was also identified to be different in most organizations. It was their strategy to implement different traceability tools and the level of sophistication provided by the each tool. As a result, the overall management systems also varied in terms of how much information it records manually or automated.

6.5 Hypothesis 5

The fifth hypothesis is used to investigate "SRQ5: Does the importance of various management systems associated with SCM process vary in different sizes of organizations?" The null hypothesis is: H05: Importance of various management systems associated with SCM process does not vary in different sizes of organizations.

A Chi-Square test was performed to investigate whether the importance of six management systems, such as, Release Management System (RMS), Defect/Issue Management Systems (DIMS), Version Control Management System (VCMS), Build Management System (BMS), Release/Deployment Management System (RDMS) and Content Management System (CMS) are independent from the organization size (OS). All six chi-square tests resulted in the acceptance of the null hypothesis that the organization size and six management systems variables being tested are independent of each other, with x2=11.644 and p-value=0.070 for OS versus RMS, x2=3.813 and p-value=0.432 for OS versus DIMS, x2=2.796 and p-value=0.592 for OS versus VCMS, x2=7.647 and p-value=0.265 for OS versus BMS, x2=10.036 and p-value=0.123 for OS versus RDMS, and x2=2.940 and p-value=0.816 for OS versus CMS.

Based on the feedback and as a result of chi-square test, it was identified that organizations regardless of their size considers all six categories of management systems as either important or very important in their software development environment. The only difference between different sizes of organizations was the capability of the management systems based on the combination of process capability and the tool sophistication.

6.6 Hypothesis 6

The sixth hypothesis is used to investigate "SRQ6: Does the empowerment of project teams to customize SCM process, associated management system linkage, and roles to perform SCM process vary in different sizes of organizations?" The null hypothesis is: H06: Empowerment of project teams to customize SCM process, associated management system linkage, and roles does not vary in different sizes of organizations.

A Chi-Square test was performed to investigate whether the empowerment of project teams to customize SCM process, associated management systems, and roles are independent from the organization size - OS. All three chi-square tests resulted in the acceptance of the null hypothesis that the organization size and three customization variables being tested are independent of each other, with x2 = 0.42 and p-value = 0.979 for OS versus empowerment to SCM Process customization, x2 = 0.894 and p-value = 0.640 for OS versus empowerment to customize Management Systems linkage customization, x2 = 1.733 and p-value = 0.420 for OS versus empowerment to customize the SCM Role.

In other words, questions were asked regarding the empowerment given to the project team to customize the software configuration management process, or how different management systems connects with each other to share the information, or how the software configuration management role will perform in the context of the project environment. The responses were received from all sizes of organizations and based on the chi-square test we can state that all sizes of organizations delegate the power to the project teams to customize software configuration practices according to their project context. It includes using the existing tools and process in alignment with customer or management goals. It was also indicated by the respondents from all sizes of organizations that their organizations have given them the empowerment to define the workflow between different establish management systems in their organization according to the context of the project and using the value based mind set to only include what add-values to the goals of the project. In addition, respondents have also indicated that they were given the empowerment to define who is going to perform configuration management role in their project rather than taking a fixed resource to perform the job.

7 CONCLUSION AND FUTURE WORK

Through the use of six research hypotheses, this study found that organization size does not impact on the usage of software configuration management process in adaptable software development environments. It is also identified that all sizes of organizations consider SCM process valuable in their environments and have similar software traceability approaches with different process formalities and tools sophistications. Finally, all organizations regardless of their sizes believe in the importance of management systems for the traceability and give empowerment to the project teams for the customization of SCM process and the workflow of the associated management systems according to the customers' expectations.

It was identified that most of participants emphasized on the implementation of software change traceability from the management systems perspective instead of directly looking at the software development processes and tools required in the environment. It was observed, that after the identification of the required management systems in the software development environment, these participant organizations have adjusted their existing software traceability capabilities through the lean and agile principles aligning it with the customer expectation.

Study also identified the existence of collaborative management systems for the establishment of the software configuration management process and its valuable contribution towards software traceability in their environments. The key strategy identified was the "empowerment of project teams" to customize their software configuration management process, associated roles and the relevant management systems in order to establish the software traceability. These project teams had taken a value-based approach to identify and assess their value-streams and then scaled- up or down software configuration management process formality and the tools sophistication keeping the management systems fully compliant with their prevailing software improvement standards.

Although the large organizations were still identified as the biggest consumers of the software configuration management process, small and medium organizations have also shown a strong tendency towards adopting this process for the software traceability purpose. Results have shown that organizations of all size are taking a value-based approach to implement the software configuration management process based on the customer expectation and also consider that as a component of a quality system engineering process.

Regardless of the approaches taken, in all cases, management systems were established primarily because of value-adding contribution in the given scope rather than for the purpose of introducing a particular software improvement standard. In most cases, once the value-adding management systems were in place, project teams were empowered to customize their solutions to incorporate software quality or improvement standards for the purpose of compliance audits.

The limitation of this study is inability to identify the large sample of the small size organizations (1 to 20 employees) having the software configuration management process in their agile software development environment. As a result, the study could not compare the effectiveness of the software configuration management process in small organizations with their counterparts in medium and large organizations. Hence, overcoming this limitation will be a part of our future research work.

References

Abran, A. and Moore, J. (2001). SWEBOK: Guide to the Software Engineering Body of Knowledge. 1st Edition. California: IEEE Computer Society Press.

Aizenbud-Reshef, N., Nolan, B. T., Rubin, J. and Shaham-Gafni, Y. (2006). Model traceability. IBM Systems Journal, 45(3), 515-526.

Bersoff, E. H. and Davis, A. M. (1991). Impacts of Life Cycle Models on Software Configuration Management. Communications of the ACM, 34(8), 104-118.

Cooper, D. R. and Schindler, P. S. (2003). Business Research Methods. 7th Edition (McGraw Hill International Edition ed.). New York: McGraw Hill International/Irwin.

Dart, S. (1990). Spectrum of functionality in configuration management system. CMU/SEI-90-TR-11.

Gills, M. and Borzovs, J. (2005). Software Testing and Traceability. University of Latvia.

Gotel, O. C. and Finkelstein, C. (1994). An analysis of the requirements traceability problem. In Proceedings of the First International Conference on Requirements Engineering, 1994.

IEEE.Std.610.12-1990. (1990). IEEE Standard Glossary of Software Engineering Terminology.

IEEE.Std.828. (2012). IEEE Standard for Configuration Management in Systems and Software Engineering. IEEE Std 828-2012 (Revision of IEEE Std 828-2005), 1-71. doi: 10.1109/ieeestd.2012.6170935

Leon, A. (2004). Software configuration management handbook. Boston/London: Artech House, Inc.

Leon, A. (2005). Software Configuration Management Handbook. Boston/London: Artech House.

Liker, J. (2004). The Toyota Way—14 Management Principles from the World's Greatest Manufacturer. New York, NY: McGraw-Hill.

Luo, S., Zhang, X. and Ren, S. (1996). A paradigm for engineering business process and its information system. IEEE International Conference on Systems, Man, and Cybernetics.

McFeeley, B. (1996). IDEAL: A User's Guide for Software Process Improvement: DTIC Document.

Mohan, K., Xu, P., Cao, L. and Ramesh, B. (2008). Improving change management in software development: Integrating traceability and software configuration management. Decision Support Systems, 45(4), 922-936.

Naveh, E. and Erez, M. (2004). Innovation and attention to detail in the quality improvement paradigm. Management Science, 50(11), 1576-1586.

Nguyen, T. N. and Munson, E. V. (2003). A model for conformance analysis of software documents. In Proceedings of the Sixth International Workshop on Principles of Software Evolution, 2003.

Nunally, J. (1967). Psychometric Theory. New York: McGraw Hill.

Poppendieck, M. (2011). Principles of lean thinking. IT Management Select, 18.

Ramesh, B. and Jarke, M. (2001). Toward reference models for requirements traceability. Software Engineering, IEEE Transactions on, 27(1), 58-93.

Rubin, A. and Babbie, E. (1997). Research Methods for Social Work. Pacific Grove, California: Brooks/Cole Publishing Company.

Shaikh, A., Ahmed, A., Memon, N. and Memon, M. (2009). Strengths and weaknesses of maturity driven process improvement effort. International Conference on Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09.

Sommerville, I. (2007). Software Engineering.8th Edition. Addition-Wesley.

Spanoudakis, G. Supporting the Construction of Decision Models for Software Selection: Towards a Probabilistic Traceability Approach.

Womack, J. P. and Jones, D. T. (1996). Lean thinking: Banish waste and create wealth in your organisation. Simon and Shuster, New York, NY, 397.

Wong, Y. C., Wong, K. Y. and Ali, A. (2009). Key practice areas of lean manufacturing. International Association of Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09.