

Adaptable Software Configuration Management: An Investigation on Australian Agile Software Development Organizations

Usman K. Durrani, Joan Richardson, and John Lenarcic

Abstract—There is a general misconception on the coexistence of Agile and Software Configuration Management (SCM) practices, and a very little reporting exists about how practitioners can benefit from their coexistence or how such coexistence can improve the organizational capabilities.

This paper represent phase one of the research study, with the aim to identify adaptable software configuration management process in Australian agile software development organizations. This as a result will facilitate later phases by identifying suitable candidate organization(s) for case study and providing the basis for validation and generalization of Adaptable Configuration Management Traceability Model (AcmTm) through additional quantitative study. This research as a whole will support SCM practitioners in particular and IT practitioners in general by providing better understanding of SCM Process in agile environments. It will also contribute knowledge towards theory of lean thinking.

Index Terms—Adaptable process, agile software development, lean thinking and principles, software configuration management.

I. INTRODUCTION

Close monitoring of current technological trends reveals continuous improvement in software development methodologies [1]-[4], but it also shows that they have not yet reached the desired satisfaction level of the practitioners [5]-[7]. Hence, the software engineering process will continue evolving. Therefore, the objective of reaching a consensus on any one software development methodology is not viable at present.

Many researchers have already attempted to address whether Agile software development methods are just an ad hoc approach or are truly based on some sound principles [8]. They suggest that, although Agile methods have evolved out of best practice, they are in fact consistent principles as proposed by organizational theories [8].

Careful observation of Software Configuration Management (SCM) practices [7], [9] and Agile software development practices [4], [10]-[12] reveal that these are a combination of various interrelated practices and managerial policies. According to [13], regardless of the software development method in use, it is important that it remains

under control, and SCM is the method of bringing this control to the software development process.

To allow for the coexistence of such practices, an organization needs to develop an adaptable process. This adaptability comes with the application of Lean Thinking [14] that promotes the culture of implementing only those practices that can add value for the process, project, and organization. But such strategies often come with associated physical and policy constraints [13], and organizations need to overcome these constraints to attain of their goals.

The development of an adaptable software configuration management process that promotes the coexistence of Agile and SCM practices is the key area of this research. This study will help to describe and understand the phenomenon of alignment and implementation [15] of Software Configuration Management in the Agile software development environment. A sample population of sixty organizations having both Agile and SCM practices in place and considers both practice domains as important for their software development activities were surveyed.

II. BACKGROUND AND EXISTING WORK

A. Agile Practices vs. Software Configuration Management Practices

While “software” is very important for all aspects of the modern world, software development itself is not yet perfect. Despite continuous efforts to introduce new development methodologies, software development is still very costly due to poor quality (defects, uncontrolled changes, traceability etc.), poor project management, poor processes, and poor requirement analysis [16] that often lead to delayed and/or failed outcomes. Both the proponents and critics of Agile methods emphasize the source code as the main focus for practices. Agile proponents often argue that the source code is the only deliverable that matters, and they marginalize the role of analysis and design evolution. However, critics point out that the emphasis on source code can lead to corporate memory loss. This is because there is little emphasis on producing good documentation and models to support software creation and evolution of large, complex systems [7].

Software Configuration Management is a method of bringing control to the software development process, and is recognized as an inseparable part of quality-oriented product development regardless of the development method. However, SCM is often considered a bureaucratic method; an approach that causes additional work and more

Manuscript received September 14, 2012; revised November 9, 2012. This work was supported in part by the School of Information Technology and Logistics, RMIT University, Melbourne, Australia.

The authors are with the School of BITL, RMIT University, Melbourne, Australia. (e-mail: usman.durrani@rmit.edu.au, joan.richardson@rmit.edu.au, john.lenarcic@rmit.edu.au).

documentation [17]. Nonetheless, the value of SCM should not be underestimated in the case of Agile software development methods [13]. The potential for having the engagement of Agile software development methods together with the SCM process exists more in organizations having an adaptable software development environment than in those still using traditional ways of development.

For the purpose of this study, the researcher has investigated agile values and principles [18] along with agile and SCM practices [19]. Table I and II lists the most widely used Agile and SCM practices that were included as part of the research survey tool.

TABLE I: AGILE SOFTWARE DEVELOPMENT PRACTICES SELECTED FOR RESEARCH

Agile Practice Variables
Prioritization of Requirements with Customers or Representatives
Negotiation of Change Requirements with Customer or Representative
Incremental Development Containing User Stories
Simple Solutions
Unit Testing
Pair Programming
Dedicated Meeting Space
Refactoring Source Code
Self-Directed Project Team
Self-Organized Multidisciplinary Team
Collective Code Ownership
Continuous Integration
Coding Standards Followed
40 Hours Work
Direct Communication with Customer or Representative
Central Change Database
Daily Team Meetings
Involvement of all Project Stakeholders
Post and Pre Increment Reflect Workshops
Concurrent Development Activities
Milestones to Track Progress

TABLE II: SOFTWARE CONFIGURATION MANAGEMENT PRACTICES SELECTED FOR RESEARCH

SCM process variables
Configuration Identification
Version and Baseline Management
Software Change Categorization
Central Change Authority
Configuration Accounting
Configuration Status Auditing
Build Management
Release Management

B. Theory of Lean Thinking and Adaptable Process

According to [14], three key principles of lean production or thinking are: the identification of value, the elimination of waste, and the generation of flow (of value to the customer). These principles, along with others, when applied appropriately can provide an adaptable environment in which Agile capabilities can be developed [20].

On the basis of our understanding of the above Lean Thinking concept, we can also align the SCM process in

terms of general manufacturing waste categories based on [21]-[22] as shown in Table III:

TABLE III: SCM WASTE CATEGORIES MAPPED WITH GENERAL MANUFACTURING WASTE CATEGORIES

Manufacturing waste	SCM waste
Overproduction	Non valued implementation of SCM process only to meet improvement standards such as ITIL, Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI) etc.
Waiting (time on hand)	Queued changes waiting for Change Control Board (CCB) approval before implementing
Unnecessary transport or conveyance	Involvement of multiple stakeholders, such as SCM and system support departments for the build and deployment of software changes in managed environments
Over processing	Heavy weight SCM framework including policies, processes, and practices for each and every configuration management activity without the any flexibility of customization
Unnecessary movement	Process making it mandatory for developers to log change requests first in any management system and waiting for CCB and management for their decisions and communication with the customer
Defects	Developers have to wait for testing team to verify the changes. Testing team have to wait for SCM team to compile the required software changes and build through manual or semi-automated build process

From the context of this theory, the researcher will study the culture of sample organizations to understand the adaptable software development environment where they implement SCM process in an agile environment without compromising the values of either one.

III. CONCEPTUAL FRAMEWORK

To allow for the coexistence of Agile and SCM practices, an organization needs to develop an adaptable process. This adaptability comes with the application of Lean Thinking [14] that promotes the culture of implementing only those practices that can add value for the process, project, and organization. But such implementation often comes with associated physical and policy constraints [13], and organizations need to overcome these constraints for the attainment of their goals (as shown in Fig. 1). The development of an adaptable software development environment is the motivation of this study.

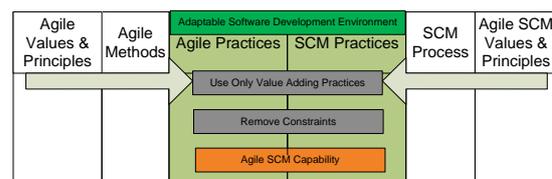


Fig. 1. Conceptual framework.

Hypotheses for this research are described as follow:

- 1) H1: There is wide variation in the extent of implementation of Agile software development practices within different size of Australian software development

organizations. (Which mean, it is hypothesized that medium and small organizations use agile methods more frequently than the large organizations)

- 2) H2: There is wide variation in the extent of implementation of Software Configuration Management practices within different size of Australian software development organizations. (Which mean, it is hypothesized that large organizations use SCM Process more frequently than the medium and small organizations)
- 3) H3: Some SCM practices are considered more important than others in agile software development organizations

IV. METHODOLOGY APPROACH

This section presents the research method used for collecting data from the respondents and the associated technique used to descriptively analyze the current state of Australian Software Development Organizations.

In our research, identification of the respondents was a challenge. This is because Agile software development is still an emerging approach that has gained popularity only in the last seven to eight years. In organizations, often only a few teams are practicing Agile software development along with the Software Configuration Management process. Therefore, it was a challenge to identify the specific respondents who would qualify to respond to our questionnaire. After a considerable amount of research on this issue, the “linkedin” online community was identified for the collection of data.

Since this was an exploratory study, the researcher gathered data from software professionals who have worked with both Agile methodology and the Software Configuration Management process. This type of sampling method is called judgment sampling, a type of purposive sampling. According to [23], “judgment sampling occurs when a researcher selects sample members to conform to some criterion”.

In order to increase validity of the survey, the researcher also conducted a pilot test survey with a small user group. Based on the feedback of the pilot group, the content of the survey was then updated and full-scale survey was conducted.

Since the research question (*Does the Adaptable SCM process and practices exist in Australian Agile Software Development Organizations?*) requires exploration of the phenomenon, descriptive statistics was used to assess the current state of ASCM process in Australian software development organizations. Descriptive statistics are commonly used in survey studies [24].

SPSS (Statistical Package for the Social Sciences) software was used to input, code, and summaries the descriptive data. Frequencies, percentages, measures of central tendency, and measure of variations were calculated. The unit of analysis for this research study was the Agile organization having the coexistence of Agile and SCM practices in their software development environment.

After the validation and verification of the online survey, over the period of two months, questionnaires were posted to 183 practitioners, mostly from different organizations, selected through personal references. From this mail out, 76

practitioners responded but 17 responses were excluded from further analysis because they did not represented organizations that used both agile and SCM practices; 107 practitioners did not respond.

V. RESULT

Various organizational variables (organization size, software development team size, geographical location, and nature of industry) were selected for the research study in order to understand their association with the adoption of Agile and SCM practices, and the importance given to both these set of practices. This also helped in the understanding of how such software development organizations provide an adaptable software development environment that promotes the development of dynamic capabilities for rapidly changing software development requirements.

Regardless of the size of the organization and/or the size of the development team, in general, all the participant organizations did considered tools and technology as either important or very important for their software development activities.

TABLE IV: VALUE ADDED BY SCM PROCESS IN PROJECT/PROCESS/ORGANIZATION

Value Categories	Respondent's Frequency
Value for the Project/Products	17
Value for the Process	16
Value for the Organization	20
No Value	7

From Table IV, we can infer that SCM process/practices add value to organization, project/products and other software development processes. Only 7 respondents mentioned SCM process as non-value adding for any operational level.

The ANOVA test was used to compare the means of different size organizations (small-, medium-, and large-scale) in terms of their implementation of Agile practices. Rather than comparing the means for each of the 22 Agile practices presented in the survey (Table I), the researcher used the technique of combining the weights as given by each respondent (in various organization size categories) against all 21 Agile practices. These combined Agile practice weights were then used to perform the ANOVA test, and the respondents were grouped according to their organization sizes.

Since we have three groups of organizations by size (small, medium and large), the possible combinations for the mean comparison are “Small and Medium”, “Small and Large”, and “Medium and Large”. Levene’s test (Table V) for equality of variance was performed prior to ANOVA testing.

TABLE V: LEVENE’S TEST FOR EQUALITY OF VARIANCE

Levene Statistic	df1	df2	Sig.
.602	2	55	.551

There were no significant results from the ANOVA test between the three groups of organizations, as shown in Table

VI.

Therefore, based on the test performed above, the hypothesis H1 is not supported, that “there is wide variation in the extent of implementation of Agile software development practices within different size of Australian software development organizations”. In other words, software development organizations, regardless of their size, consider Agile software development practices important and use these frequently for their software development operations.

After the significance of Levene’s test (Sig. 0.065), the ANOVA test was performed between the three organizational groups proved significant, as shown in Table VII.

TABLE VI: ANOVA TEST RESULT FOR SIGNIFICANCE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	120.899	2	60.450	.576	.565
Within Groups	5770.222	55	104.913		
Total	5891.121	57			

TABLE VIII: ANOVA TEST RESULT FOR SIGNIFICANCE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	128.011	2	64.006	3.271	.045
Within Groups	1076.26	55	19.568		
Total	1204.27	57			

Based on the tests performed above, the H2 hypothesis is accepted, that “there is wide variation in the extent of implementation of SCM practices within small, medium, and large size organizations”. In other words, large software development organizations consider Software Configuration Management practices (Table II) comparatively more important than medium and small software development organizations. However, these medium and small software development organizations are gradually realizing the benefits of the SCM process and are adopting selective SCM practices for their operations using value-based mindset.

For the purpose of H3, researcher has performed one sided t test along with pair-wise t test. We observed that the respondents opined that the establishment of version management and baseline for the project and releases is the most important practice of agile software development organizations. This was followed by “coordination of software releases through well established procedures”, and then the “identification of project artifacts at the start of project”, and “execution of well-defined software build procedure for a particular version of software”. Hence, based on the tests performed above, the H3 hypothesis is accepted, that “some SCM practices are considered more important than others in agile software development organizations”.

As a result of above analysis, therefore, in order to facilitate next phases and to better understand adaptable software configuration management process, it is more feasible to undertake case study in the large software

development environments as compared to medium and small. However, to validate and generalization the results, it is recommended to also consider small and medium organizations for the case studies as well.

VI. DISCUSSION AND CONCLUSION

This quantitative exploratory study investigated the existence of Adaptable Software Configuration Management process in Australian software development organizations. The comparative analysis was performed between various organizations based on their sizes (small, medium, and large) and the practices they performed, along with the frequency and importance. It also explored the different tools and technologies used and their importance for these organizations in different software development areas, such as, programming language, software development IDEs, SCM system, software build management system, software defect/issues system, and project management system etc. In addition to that, various SCM dynamic capabilities were also studied along with the importance of each.

This study has found that Agile methods and the associated practices are considered important and used frequently by all-sized organizations (small, medium, and large). The outcome of this research is in sync with the result of the workshop conducted by [25] with large organizations, in which it was unanimously agreed that Agile methods helped them to become more flexible and adaptive to change [26]. In the researcher’s opinion, more discipline and tool support is required for Agile practices, and they are generally far more prescriptive than traditional development practices [27]. We found that all participants in this research are using various tools and technologies extensively for software build management, software defect and issue management, and release management activities.

Another critical finding of this research is that, although large organizations are still the highest consumers of traditional processes such as SCM process, an increasing number of medium and small organizations are adopting a customized version of SCM process for their specific software development needs. SCM process is one of the key supporting software development processes that provides discipline to Agile methods and helps in the development of capabilities to integrate, build, and reconfigure the competencies for rapidly changing software requirements.

VII. RESEARCH CONTRIBUTION

While the Agile software development methods are gaining more popularity compared with classical software development methods, the evidence of their success has been mostly subjective, and it is very difficult to obtain hard numbers [1]-[2]. Similarly, very scant empirical studies exist of Software Configuration Management in Agile methods [13], [28]. By using the quantitative method approach, this study aims to generate empirical evidence to contribute to the body of knowledge in the relevant areas.

On the practical side, this research can also provide support to IT businesses in general, and Software

Development Organizations in particular, with the streamlining of the internal operational environment for the facilitation of an adaptable software development environment and the resulting coexistence of value-added Agile and SCM practices.

VIII. FUTURE RESEARCH

This study was conducted mainly from the perspective of organizational size (small, medium, and large) and other perspectives such as project size and project complexity were not considered during the research scope. Hence, further research is required from these perspectives.

There is a further need to investigate how different strategies are used by these organizations in order to develop adaptable software development environments through the application of Lean Thinking. It is also necessary to understand different constraints faced (and resolutions performed) by such organizations during the implementation of Software Configuration Management practices in accordance with Agile values and principles [18], [29].

ACKNOWLEDGMENT

This work has benefited greatly from discussions with my senior supervisor Joan Richardson and second supervisor John Lenarcic from RMIT University.

REFERENCES

- [1] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New directions on agile methods: a comparative analysis," in *Proceedings of 25th International Conference on*, 2003, pp. 244.
- [2] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, *Agile software development methods: Review and Analysis*. Espoo, Finland: Technical Research Centre of Finland, VTT Publications pp. 478.
- [3] A. Cockburn, *Agile Software Development*, Boston, Addison-Wesley, 2002.
- [4] J. Highsmith, *Agile Software Development Ecosystems*, Addison-Wesley, 2002.
- [5] R. N. Charette, "Why software fails," *IEEE Spectr*, vol. 42, no. 9, pp. 42-49, 2005.
- [6] D. Davis and V. Venkatesh, "Toward preprototype user acceptance testing of new information systems: implications for software project management," *IEEE Transactions on Engineering Management*, vol. 51, no. 1, pp. 31-46, 2004.
- [7] D. Turk, R. France, and B. Rumpe, "Limitations of agile software processes," in: *Proc. of 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering—XP2002*, Italy, 2002.
- [8] L. Cao and B. Ramesh, "Agile Software Development: Ad Hoc Practices or Sound Principles?" *IT Professional*, vol. 9, no. 2, pp. 41-47, 2007.
- [9] S. Fraser, B. Boehm, J. Jarkvik, E. Lundh, and K. Vilkki, *How Do Agile/XP Development Methods Affect Companies?* Springer-Verlag Berlin Heidelberg, 2006.
- [10] K. Beck, *Extreme Programming Explained: Embrace Change*. Boston, MA, Addison-Wesley, 1999.
- [11] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. NJ, Prentice-Hall, 2002.
- [12] J. Stapleton, *Dynamic Systems Development Method -- The Method in Practice*, Addison-Wesley, 1997.
- [13] J. Koskela, *Software configuration management in agile methods*, VTT Publication, Finland, 2003.
- [14] J. Womack and D. Jones, *Lean Thinking - Banish Waste and Create Wealth in Your Corporation*, New York, Simon & Schuster, 1996.
- [15] M. N. Ravishankar, L. P. Shan, and E. L. Dorothy, "Examining the Strategic Alignment and Implementation Success of a KMS: A Subculture based Multi-Level Analysis," *Information Systems Research*, pp. 1-23, 2010.

- [16] D. B. Cao, "An Emperical Investigation of Critical Success Factors in Agile Software Development Projects," Doctor of Philosophy dissertation, School of Business & Technology, Capella University, 156, 2006.
- [17] B. Appleton and S. Berczuk, "The Need for Agility in SCM," *CM Journal*, vol. 8, 2003.
- [18] K. Beck, M. Beedle, *et al.*, "Manifesto for Agile Software Development," [Online]. Available: <http://www.agilemanifesto.org/>, 2001.
- [19] *IEEE Standard for Software Configuration Management Plans*, IEEE pp. 828, 2005.
- [20] M. Poppendieck, "Principles of lean thinking," [Online]. Available: <http://www.poppendieck.com>, 2009.
- [21] R. Morien, "Agile Management and the Toyota Way for Software Project Management," in *Proc. of 3rd IEEE International conference on Industrial Informatics (INDIN)*, 2005.
- [22] B. Sproull, *The Ultimate Improvement Cycle: Maximizing profits through the integration of lean, six sigma, and theory of constraints*, New York, CRC Press: Taylor & Francis Group, 2009.
- [23] R. Cooper and S. Schindler, *Business research methods*, New York, NY: McGraw-Hill/Irwin, 2002.
- [24] M. Weimer, *Enhancing scholarly work on teaching and learning: Professional literature that makes a difference*, San Francisco: Jossey-Bass, 2006.
- [25] B. Boehm, "Get Ready for Agile Methods, with Care," *Computer*, vol. 35, no. 1, pp. 64-69, 2002.
- [26] S. Fraser, B. Boehm, J. Jarkvik, E. Lundh, and K. Vilkki, *How Do Agile/XP Development Methods Affect Companies?* Springer-Verlag Berlin Heidelberg, 2006.
- [27] S. Fraser, *Software Best Practices: Agile Deconstructed*. Springer-Verlag Berlin Heidelberg, 2009.
- [28] A. Back and O. Bodin, "Djupstudie EDA270 - Agile Software Configuration Management," *CM Journal*, 2006.
- [29] A. Cockburn and J. Highsmith, "Agile Software Development: The People Factor," *Computer*, pp. 31-133, 2001.



Usman K. Durrani is currently doing his PhD from School of Business Information Technology and Logistics RMIT University in Melbourne, Australia. He is a seasoned management consultant, working in the area of software configuration management for more than twelve years. He has been involved in many high-tech government and private projects in US, China, Australia, and Pakistan. His research interests include agile software development methodologies software configuration management, lean thinking concepts and principles, and management and control aspects of cloud computing. Mr. Usman has got two Masters Degrees (CS & IT) and a Bachelor Degree in 2003, 2000 and 1997 respectively from RMIT University Australia, UCP Pakistan and Punjab University Pakistan. He also holds international certifications, such as, PMP, and ITIL v3.



Joan Richardson has been an academic for the last 20 years. She completed a PhD on 'An episode of change in a contemporary higher education setting' and has published widely on IT education issues. She has completed a number of learning and teaching projects funded by Learning and Teaching Infrastructure Funds and was awarded with an Australian Learning and Teaching Council (ALTC) citation in 2010. She is the main author of the book *Business Computing 1*, which is to date an innovation in learning and teaching with material presented on CD Rom, websites and a set of online resources. Dr. Joan has worked with Pearson Education Australia to produce the text book, and has been working on learning and teaching funded projects to innovate learning resources. She also supervises PhD students and is the deputy Head of School responsible for all Learning and Teaching matters.



John Lenarcic is a physicist and applied mathematician by training, an IT academic by fortunate accident and an armchair philosopher by conscious choice. He currently teaches usability analysis and computer ethics in the School of Business IT and Logistics at RMIT University in Melbourne, Australia. Dr. Lenarcic's research interest includes social media, linguistic issues in systems development, design thinking in software engineering and the social and ethical implications of information systems.