Lean traceability solution through SLAM model

A case study of a hybrid delivery team in a hybrid cloud computing environment

Usman Durrani, Joan Richardson, John Lenarcic, Zijad Pita
School of Business IT and Logistics
RMIT University
Melbourne, Australia
usman.durrani@rmit.edu.au

***Abstract*—"Cloud Computing" is the latest IT buzzword representing the next generation of internet computing paradigm. Along with this paradigm shift came the new waves of opportunities and challenges and a need to explore more innovative ways to manage software development lifecycle. At present there is a lack of studies in the area of traceability in cloud computing environments. This research presents a longitudinal case study of a large Australian project, where Agile and Plan driven project teams, coexisted, and worked together to deliver a consolidated product with end-to-end traceability capabilities in a hybrid cloud computing environment. It will contribute by presenting a Software configuration through Lean and Agile Management "SLAM" model used in the case study. From practical aspects, this study enhance the understanding of IT practitioners, and cloud service providers to establish management systems aligned with the standard software configuration management practices through the use of lean principles and tools.**

Keywords—Cloud Computing; Software Configuration Management; Lean Thinking; Agile; Traceability; Strategies; Case Study; Model; Management System; Process

I. INTRODUCTION

Cloud computing is defined as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimum management effort or service provider interaction" [1]. It represents a shift towards next generation of internet computing paradigm. Along with this paradigm shift, comes new wave of opportunities and challenges and a need to explore more innovative ways to manage traceability in software development lifecycle.

While there are number of benefits of using cloud computing, such as, cost effectiveness due to multi-tenancy, real time provisioning and pay-as-you go services, it has its share of complexities and concerns in terms of security, privacy, performance, availability, loss of control, and integration [2, 3]. Through this paper we highlight yet another concern regarding the traceability of software development lifecycle in a cloud computing environment. Cloud computing is designed for reuse and flexibility, and can present challenges for Software Configuration Management (SCM), especially in a highly regulated and audited environments [4].

SCM is a software engineering process which originally was developed for the traceability needs of classical software development methodologies in standalone software development environments. In general, it is defined as a discipline of "applying technical and administrative direction and surveillance, to identify and document the functional and physical characteristics of a configuration item, to control changes to those characteristics, to record and report change processing and implementation status, and to verify compliance with specified requirements"[5]. Unlike cloud computing environment which is quick to develop and deploy, SCM must be planned for deployment [4].

Sparked by our interest to investigate hybrid cloud computing environment [6, 7] and the existence of traceability capabilities through SCM process, we embarked on a longitudinal case study of a large Australian IT project. The case study participants comprised of agile and plan driven project teams, coexisted, and worked together under a common program to deliver a consolidated billing system.

This paper is structured as follows: In next section, we review the literature and present the theoretical framework and the associated Software configuration through Lean and Agile Management (SLAM) model. We then present the research methodology for this study. Following section will describe the case analysis, including the brief background of case study, which then follows with the explanation of each of the construct of the SLAM model in the context of the case study. We will then conclude the case study in the last section.

II. REVIEW OF LITERATURE

This section comprises of three subsections. Each of these subsections will define and discuss the concepts relevant to the study and will later facilitate in the development of theoretical framework.

A. Traceability requirements, issues and concerns

Traceability is "the ability to describe and follow the life of a requirement, in both a forward and backward direction"[8]. As stated by Gotel, et al. [9] the traceability effort for a project must be "fit for purpose". It means that since the traceability decisions in agile and plan driven projects are different and there is no one-size-fits-all traceability solution currently available for the purpose.

In order to have “fit for purpose” traceability solution, alignment should be made as per traceability granularity required by the customer [10]. This alignment is performed by adjusting the complexity and sophistication of traceability process and practices along with the associated tools and technology implementation, focusing first and foremost on what the business customers expect, now how it is delivered [4].

According to [11], in order to ensure a systematic development process, so that the software system remains well defined at all times with accurate specifications and verifiable quality attributes requires (holistic traceability) knowledge from two perspectives. First is the product knowledge perspective, which represents the information about the artifacts of the software system and their relations with each other. Second perspective is linked with process knowledge, which represents knowledge about the process followed in the development of these artifacts [11]. At present, there is a lack of traceability models that encompasses both process and product perspectives and a holistic traceability model was already proposed by the authors of this paper to address it [12].

The scope of this paper is to present a case of “fit for purpose” traceability solution [10] using management systems (as proposed in this paper) to address product and process knowledge perspective of traceability [11, 12].

B. *Lean thinking principles and tools*

Lean thinking is a business philosophy that demands the total and systematic elimination of waste and to improve the flow of value in a given context. According to [13], lean thinking is based on five principles that must be thoroughly understood and agreed upon before lean implementation can begin. They are, 1) to specify what the customer defines as value, 2) to identify the value stream for the product, 3) all value to flow through the value stream without delays or barriers, 4) allow the customer to pull value, rather than the manufacturer producing to forecast, and 5) pursue and practice continuous improvements

As concluded by [14], at present there is no agreed definition of lean and the set of tools or characteristics for the formulation of lean adaptable environments depends on the relevant goals. In the scope of this case study, we describe the use of a subset of lean tools selected based on the goals of the case study and listed in Table 1. We will later present the implementation of these lean tools in the context of the case study in section V.

C. *Cloud computing environment and associated challenges*

Cloud computing refers to both applications delivered as a service over the internet and the hardware and systems software in the data centers [15]. It use of different virtualization technologies by defining images of operating systems, middleware, and applications to represent physical machines and are usually pro-allocated to an available server [16].

Research studies in the past have identified various advantages and benefits of using cloud computing, such as, cost effectiveness due to multi-tenancy, real time provisioning and pay-as-you go services etc. [17, 18]. On the contrary, it has also presented different challenges, issues, vulnerabilities, and complexities, such as, security, privacy, performance, availability and integration [19, 20].

As stated by [21], the existing software development process models and framework activities are not adequate from the cloud computing aspects, and the suggestion was made to involve cloud service providers during the software development lifecycle for the establishment of different processes (such as, configuration management, change management, and quality assurance) and the provisioning of different project related environments.

TABLE I. LEAN TOOLS FOR THE SLAM MODEL IMPLEMENTATION

Lean Tools	Meaning
6S tool (Sort, Stabilise, Shine, Standardise, Safety, Sustain)	Lean foundational tool of lean used during identification of activities in the value stream
Sustaining continuous flow (Heijunka, Takt Time)	Creating flow and eliminating waste
Jidoka	Continuous integration and intelligent automation
Kanban	Visual management and pull system
Kaizen	Continuous improvement (“change for the better”)
Yokoten	Best practice sharing across everywhere
Gemba	The real place (Go and see the work)

III. THEORETICAL FRAMEWORK

Theoretical framework for this study, as shown in Fig. 1, is based on generic traceability process model [9] and the lean implementation model for business process change [22]. In addition, we also proposed a SLAM model (as shown in Fig. 1) for the purpose of implementing end-to-end lean traceability solution facilitated through lean tools (as listed in Table 1) and lean principles [13].

The remaining section will describe various phases of traceability process model and also introduce associated constructs of the theoretical framework. Detail of each constructs and how the lean tools (as listed in Table 1) were applied to the case study is described in section V.

1) Initiation Phase: To establish an adaptive traceability process requires a strategic initiative, where the top managers act as leaders in defining and communicating the vision of the process and the resulting changes [9, 22]. The planning activities of such an iterative strategy includes, determining the needs, resourcing, planning, implementing, and assessing and reassessing the traceability needs with the value-based mind-set.

2) Creation Phase: In order to establish lean traceability solution, an emphasis should be made to established adaptable environments with its basis in learning capacity, cultural readiness, relationship balancing, and IT leveragability and knowledge capacity [22]. The use of lean principles in combination with other context specific processes also facilitate in the development of adaptable environments [23, 24].

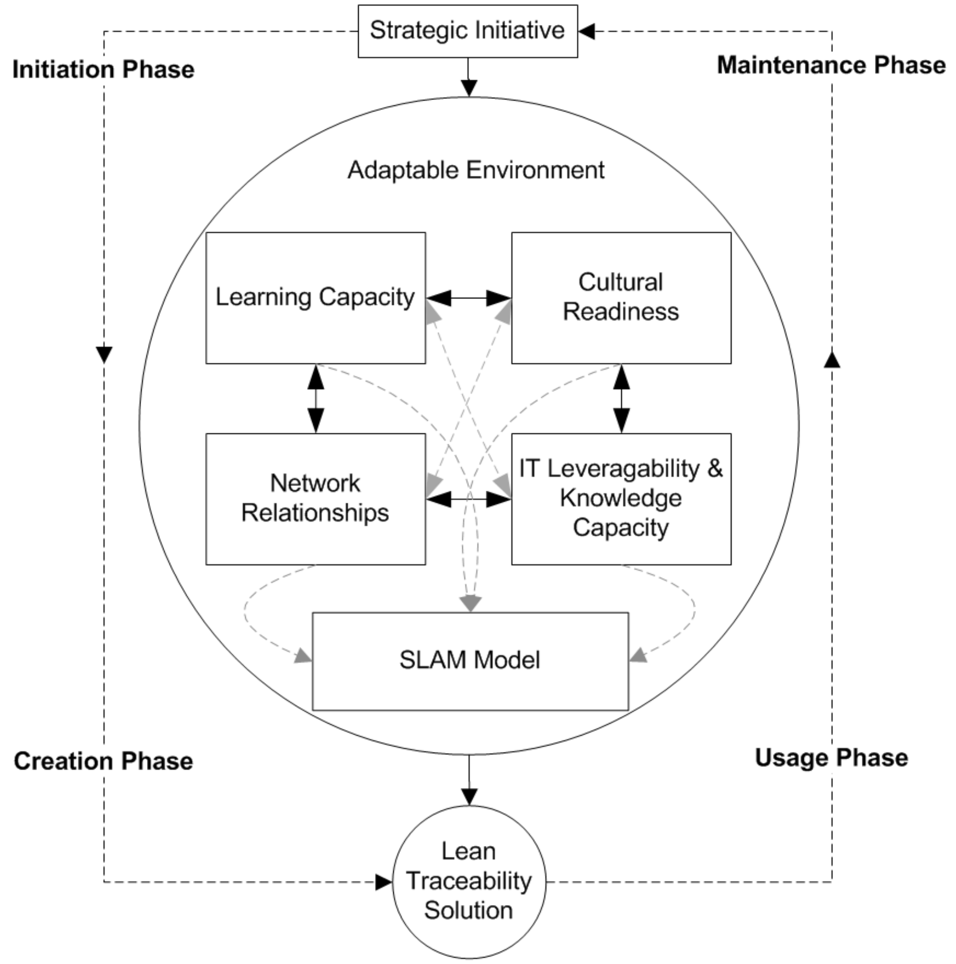

Fig. 1. Theoretical framework

Based on the identifications of needs and requirements of the customers, this phase creates a holistic lean traceability process covering both product and process knowledge perspectives [11]. For this study, we proposed to use management systems to implement these knowledge perspectives and defined 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".

Also as a result of our emphasis on the management systems for lean traceability implementation, we define traceability as "the ability of tracing from one management system containing specific type of objects and the sources to another based on defined syntax, semantic relations and implementation context to gain transparency from product and process perspectives". Where syntax refers to the structure of the traceability element and semantic represents the purpose [9].

3) Usage Phase: Once the adaptable environment is established along with the lean traceability solution, the availability and usefulness of traces between management systems has to be ensured to allow for on-going use throughout the system lifecycle [9]. The management systems established in a given context should be capable to retrieve traces either in isolation or as a constituent part of event and to render in some meaningful way. According to [9], an important part of this phase is the quality assessment of the established traceability of the management system in terms of the fitness for purpose. It should be recorded and reported as part of the continuous improvement process.

4) Maintenance Phase: This phase reflects the maintenance of overall adaptable environment. It covers the changes required to all constructs in the theoretical framework (as shown in Fig. 1), such as, learning capacity, cultural readiness, relationship balancing, IT leveragability and knowledge capacity, established management systems and the associated traceability artefacts, traces, and objects [9, 25]. Based on the nature of changes required, strategic realignment is performed to adapt to the changes in order to make it valuable for the stakeholders.

IV. METHODOLOGY

For this study, we performed a single in-depth case study via a large Australian IT project. Through it, we analyzed lean strategies for the application of traceability in hybrid cloud computing. Following [26], a qualitative and single case study approach was chosen primarily to provide rich data to other researchers and an opportunity to analyze management decision on how, what, and when to apply lean traceability process.

We conducted our study using triangulation of data gathered through as many sources as possible improve accuracy and reliability. Data were collected by, 1) observing daily "stand-up" meetings, 2) semi-structured and casual interviews with program managers, platform delivery leans, platform configuration managers, developers, project managers, business analysts, and 3) documentation analysis reflecting the actual environment.

V. CASE ANALYSIS

A. Case Study: background and motivation

For the purpose of this case study, we coded the data, including the names of the participating program, platforms and teams to protect their identities. Here, Program-B refers to the participating research program that was established to build and operate new infrastructure, to enable advanced super-fast digital services nation-wide. Program's rollout commenced in 2010 and comprised of design, development, implementation, and on-going support and enhancement services of the business and operational support systems required for the processes of activating, assuring and billing services.

The program delivery was sub-divided into product releases such as, R1, R2, and R3A and the work streams were delivered through five platforms using a combination of different out-sourced organizations and subcontractor resources both onsite and offsite. These delivery platforms were: Platform-S, Platform-C, Platform-A, Platform-R, Platform-F.

The Program-B technical environment was based on infrastructure using virtual servers in a hybrid cloud computing environment. These virtual servers were created by Program-B Data Centre (DC) Engineers (Cloud Service Providers) on the demand of the program manager who specified the requirements in an Infrastructure Service Request. Other services were also requested for databases, network services and user provisioning. Different management systems as part of SLAM model (as shown in Fig. 2) were also deployed through the same process in a hybrid cloud environment and at defined trusted zones.

Based on initial meetings with the program manager and the associated platform delivery managers, the requirements for the lean traceability solution were identified and documented as part of Program-B Technical Management Plan". At later stages a more detailed SCM specific policies were release by Program-B Configuration Manager and documented in the "Program-B SCM Plan".

One of the key concerns raised by the program manager was the implementation of the SCM process with in a diverse nature of the Program-B delivery team, having five platforms containing multiple vender teams each using either Agile or Plan driven methodologies. In words of the program manager:

"…the value for our customer is to have the end to end traceability of changes, but not at the expense of delays in the delivery of the project milestones. We need to find a flexible way of providing the configuration management solution which is suitable for all the platform delivery teams regardless of the software development methodology they use. Things get more complicated and challenging when it needs to be delivered in a cloud computing environment".

B. Framework constructs: definition and analysis

This section discusses each construct of the theoretical framework as illustrated in Fig. 1 and Fig. 2. Aligned with the research objectives, questions were asked from the relevant participants, and/or observed in the environments, and/or confirmed through program documentation analysis. The research findings of how the case study incorporated these constructs and the usage of lean tools (Table 1) and principles are discussed below.

1) Construct 1. Strategic initiatives: Strategic initiatives are associated with the vision and/or expectations of the senior management team [27]. This could be a result of a need to provide adequate customer support or a proactive push to leverage potential opportunity [28]. The focus of resulting traceability strategy should be to consolidate the product and process related traceability knowledge to cover holistic software development lifecycle [11].

The program management for the case study had a very strong commitment to lean thinking and was aware of the long term commitment required for making lean implementation successful across the program. Decisions were made by the program and platform level management to utilize the lean approaches by hiring outside expertise, continual trainings of the permanent resources, and the culture of continuous improvement at different operational levels.

During the strategic discussions with stakeholders, all value-creating management systems were identified and placed in a tight sequence using 6S tool (Table 1). In an initial discussion, a strategic decision was taken by the program and platform management "…business has already acquired IBM Rational Team Concert (RTC) and Rational Asset Manager (RAM) tools (after discussion with the customer) with the intension of using it in this program, from day one. Unfortunately, we cannot operationalize these tools until all teams in the program get a good understanding of using it as per our processes. Let's initiate our software development activities using the existing or simple tools for initial phases. In parallel, plan continual training sessions to educate program resources (on these newly acquired tools). We need to deliver our final solution using RTC and RAM". Consequently, a phased approach was taken to incrementally introduce these value-creating management systems and the associated practices.

Program-B infrastructure team played a very important role to establish the strategy regarding privacy and security of the

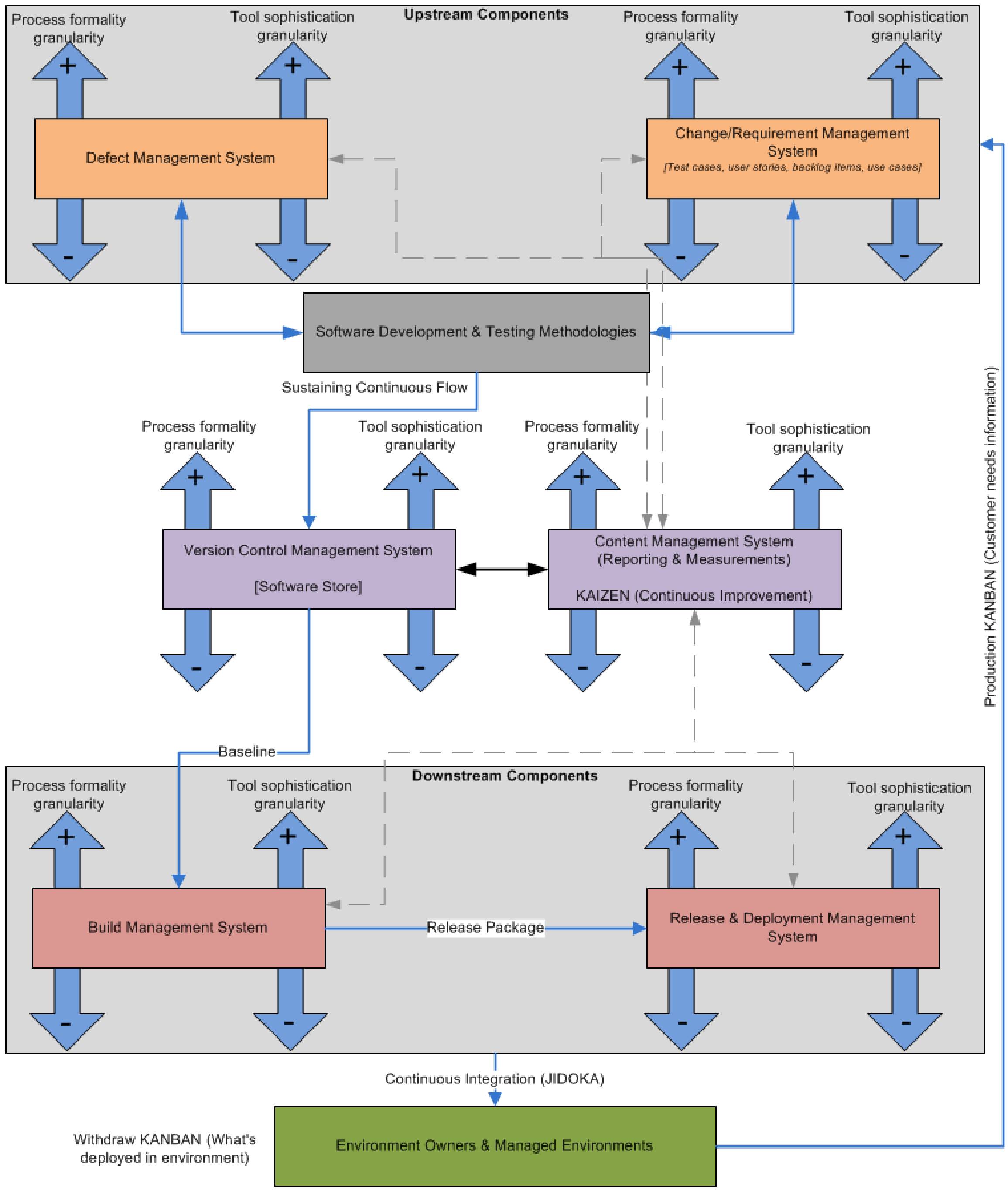

Fig. 2. SLAM model for Lean traceability solution

cloud computing environment and the associated deployment policies.

2) Construct 2. Learning capacity: As stated by [29], the major goal to provide learning is to establish the positive outcomes through effective adaptation to environmental changes and improved efficiency in the process of learning. Such adaptation can includes making technological enhancements through management systems to implement end-to-end traceability and also learning from the best relevant practices in the industry.

Program-B environment showed tendency towards creating a learning environment by appropriately responding to enhancements in the traceability process and the usage of phased implementation of management systems. The adequacy of lean implementation was evident through the self-motivated trainings to all the employees at platform and program levels and through the institution of open door suggestion culture. All platforms have adopted the solution by learning and sharing

(Yokoten) their experiences with each other. Configuration management teams at program, platform and vender levels coordinated closely with data center for the establishment of their relevant management systems and its deployment in the defined trusted zones.

3) Construct 3. Cultural readiness: According to [30], organizational culture either promotes or prevents the integration of individual learning with enterprise learning by influencing the organization's ability to learn, share information and make decisions. Openness of communication and sharing information can promote a common culture regardless of the difference in the nature of teams organizations, and encourage the innovative behavior in the environment[29].

For the case study, the initiative for lean traceability approach came from both program management and platform management. Teams at program, platform and vender levels were assembled to ensure successful implementation of traceability with minimum impact to the actual product delivery operations. The key job of these teams was to map their relevant platforms' artifacts with the established management systems using incremental approach for the flow of information. For example, program team was responsible for providing the implementation of enterprise level management systems before the final stages of the program, interim implementation of management systems was the responsibility of the platform teams, while venders provided their product specific build scripts to interact with both enterprise and platform level management systems (during the transition) for building and deploying specific version of the product in target managed environments.

4) Construct 4. Network Relationships: As stated by [31], in most circumstances cooperative, interpersonal and group behaviour results in superior performance. Also the organizations that can manage the aspects of competition and cooperation continuously can benefit from employees incentives and controls, as well as instil change more effectively [29]. For the case study, the program and platform management teams worked very closely with the cloud service providers and venders to establish managed environments and for the delivery consolidated product in the managed environments, respectively.

In order to resolve problems or to discuss the implementation of phased enhancements, management representatives from program, platform, and venders met on regularly basis. These meetings included daily stand-up meetings, weekly status meetings, and casual on desk meetings. Venders' teams were also given the trainings on lean traceability practices and the knowledge on using established management systems for their purpose. Several key venders played active roles to implement management systems and for the automation of build and deployment aspects using their product specific build scripts. Cloud service providers (DC) played a dominant role as part of Program-B infrastructure team to establish the security policies for the deployment of various products and management systems.

5) Construct 5. IT leveragability and knowledge capacity: Evidence as suggested by [32] identified that IT projects often fails to capture the business and human dimensions of processes, and are likely to fail. To address this failure, socio-technical design approach was proposed and recommended for the synergy between the business, human, and IT dimensions of an organization [33, 34].

For the purpose of case study, visual management (Kanban) which is one of the lean tools (as shown in Table 1) was applied for better monitoring and control of the traceability process. It was treated as a communication mechanism that allowed the program to view processes starting from the recording of requirement by the end customer down to its deployment in the target production environments. Some non IT stakeholders initially resisted the implementation of such visual management, as they thought it may hampers productivity. Once such obstacles were conquered through different levels of training on reporting and auditing capabilities of the established management systems, these stakeholders saw the positive value of traceability solution and their associated visual controls.

6) Construct 6. SLAM model for lean traceability solution: According to [33], successful process management requires process measurements (improvement feedback loop, process audit etc.), tools and techniques (quality control tools, CASE tools etc.), and documentation (process flow chart analysis, root cause analysis etc.).

For the case study SLAM model was proposed based on requirements raised in the initiation phase. It comprised of six collaborative management systems (as shown in Fig. 2) established through a principle defined as, "the level of tool and technology sophistication and the formality of the process and practices associated with each management systems should be directly proportional to change traceability granularity required by the customer, and should be aligned with the business goals and customer expectations".

The process and tool control arrows (as shown in Fig. 2), which we called "Volume Adjustment Monitor (VAM)" was linked with each of the management systems and facilitated in the application of principle (as mentioned above) for the pragmatic implementation of sound SCM principles to serve the need of the business [35]. Consideration were made to add nothing but value (eliminate waste), center on the people who add value, flow value from demand (delay commitment), and optimization of the target management system for Yokoten (across the program) [24].

As illustrated in Fig. 2, "upstream components" or "manufacturing components" of SLAM model comprised of Change/Requirement management system, and defect management system. These management systems were linked with the "Production Kanban" to represent a "pull" system to avoid overproduction [23]. Once a requirement was recorded through the upstream component as a result of pull from the environment owners, project development team members were assigned the workload through the established workflow channel (such as, Kanban stand-up meetings, and/or email notification of assignment through management systems etc.). All software changes were released through a Sustainable continuous flow and stored under the "version control management system" [23].

An important observation is that although these established management systems enhanced the visual management capabilities, but physical Kanban boards still played a key role to reflect teams' status and the progress. These physical boards were used during the daily Kanban stand-up meetings to share information with the co-located stakeholders, and then the relevant management systems were updated to disseminate updated status to the wider audience (including teams located in other local offices and the offshore teams).

Once the software changes were stored inside the version control system, it remained there for a short period till an instruction for "Withdraw Kanban"; a pull event representing a "wish" of a target environment owner to deploy selective changes into their relevant managed environments [23].

"Downstream components" or "distribution components" of the SLAM model comprised of Build management system and the Release & Deployment management system (as shown in Fig. 2). The role of the build management system was to initiate the pull event by extracting all the relevant software changes through a particular "baseline/snapshot/tag" and packaging/publishing it as per build instructions. As a second stage of the downstream components, release & deployment management system was triggered either manually or at a defined regular interval of time to deploy the newly published release package through custom product build scripts in the target managed environment, here represented as Jidoka (continuous integration).

Software changes once deployed in the managed environment went through a "stakeholder acceptance process" and the "production Kanban" signal were triggered to reflect the updated state of the requirement and often resulted in another pull signal for the upstream component.

Although each of the management system was capable of producing their reporting of the contained artifacts, but a consolidated centralized reporting mechanism was also established through the content management system. These individual and centralize reporting systems contributed towards Kaizen (continuous improvement) for the pursuit of perfection and conveyed in the constant "change for the better" [23, 36].

Gemba also played an important role in the program's software development environment. Since the management and client represented were co-located with the platform and venders delivery teams, they were directly aware of the various business and technical issues and constraints. Due to awareness, decisions were taken very quickly with minimum time wasted on lengthy meetings.

VI. CONCLUSION

At the end of last phase (R3A), all requirements of Program-B were delivered two months before the actual delivery time along with end-to-end traceability through program's enterprise-level management systems. The program delivery operations were handed over to the maintenance team with members' located in onshore and offshore offices.

Soon after R3A, the process to establish new cloud environments or deployment of new tools or technology got more simplified. It adopted a single channel of change request process through the Program-B maintenance team instead of multiple requests from program, platform, and venders representatives.

Program is still maintaining an approach of sustainable continuous flow through the use of Heijunka and Takt Time and has defined three deployment windows on daily basis for the approval and promotion of change requirements in the target managed environments. Through the use of VAM, the level of tool and technology sophistication and the formality of the process and practices associated with each management systems were reassessed during maintenance phases and the complete migration was performed from tools like SVN, ANT and TRAC to RTC and RAM as per client requirement.

VII. FUTURE WORK

In our future work, we are planning to conduct more case studies considering various process improvement standards, computing environments, software development methodologies, and organization sizes. It will help us to generalize our findings as identified in this paper. In addition, in the next stage of our research we are applying quantitative measurements in the research context to validate and add more rigors to our qualitative findings.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication,* vol. 800, p. 145, 2011.

[2] S. B. Chebrolu, "How Do Cloud Capabilities Impact Various Aspects of IT Effectiveness?," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012, pp. 932-940.

[3] A. Behl, "Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation," in *Information and Communication Technologies (WICT), 2011 World Congress on*, 2011, pp. 217-222.

[4] R. E. Raygan, "Configuration management in a system-of-systems environment delivering IT services," in *Engineering Management Conference, 2007 IEEE International*, 2007, pp. 330-335.

[5] IEEE.Std.828, "IEEE Standard for Configuration Management in Systems and Software Engineering," *IEEE Std 828-2012 (Revision of IEEE Std 828-2005),* pp. 1-71, 2012.

[6] S. Rajan and A. Jairath, "Cloud Computing: The Fifth Generation of Computing," in *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, 2011, pp. 665-667.

[7] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications,* vol. 34, pp. 1-11, 2011.

[8] O. C. Gotel and C. Finkelstein, "An analysis of the requirements traceability problem," in *Requirements Engineering, 1994., Proceedings of the First International Conference on*, 1994, pp. 94-101.

[9] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, and P. Mäder,

"Traceability Fundamentals," *Software and Systems Traceability,* pp. 3-22, 2012b.

[10] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, and J. Maletic, "The Grand Challenge of Traceability (v1. 0)," *Software and Systems Traceability,* pp. 343-409, 2012c.

[11] K. Mohan, P. Xu, L. Cao, and B. Ramesh, "Improving change management in software development: Integrating traceability and software configuration management," *Decision Support Systems,* vol. 45, pp. 922-936, 2008.

[12] U. K. Durrani, J. Richardson, J. Lenarcic, and Z. Pita, "Towards holistic product and process knowledge traceability model," *unpublished*.

[13] J. P. Womack and D. T. Jones, "Lean thinking: Banish waste and create wealth in your organisation," *Simon and Shuster, New York, NY,* vol. 397, 1996.

[14] J. Pettersen, "Defining lean production: some conceptual and practical issues," *The TQM Journal,* vol. 21, pp. 127-142, 2009.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM,* vol. 53, pp. 50-58, 2010.

[16] P. Kalagiakos and P. Karampelas, "Cloud computing learning," in *Application of Information and Communication Technologies (AICT), 2011 5th International Conference on*, 2011, pp. 1-4.

[17] Z. Yaming and L. Haiou, "Applications and business value analysis of cloud computing," in *E-Business and E-Government (ICEE), 2011 International Conference on*, 2011, pp. 1-4.

[18] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—The business perspective," *Decision Support Systems,* vol. 51, pp. 176-189, 2011.

[19] S. Patidar, D. Rane, and P. Jain, "Challenges of software development on cloud platform," in *Information and Communication Technologies (WICT), 2011 World Congress on*, 2011, pp. 1009-1013.

[20] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications,* vol. 1, pp. 7-18, 2010.

[21] R. Guha and D. Al-Dabass, "Impact of web 2.0 and cloud computing platform on software engineering," in *Electronic System Design (ISED), 2010 International Symposium on*, 2010, pp. 213-218.

[22] W. J. Kettinger and V. Grover, "Special section: toward a theory of business process change management," *Journal of Management Information Systems,* vol. 12, pp. 9-30, 1995.

[23] J. K. Liker, "The Toyota way: 14 management principles from the world's greatest manufacturer. (pp. 35-41)," ed: New York: McGraw-Hill., 2004.

[24] M. Poppendieck, "Principles of lean thinking," *IT Management Select,* p. 18, 2011.

[25] B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," *Software Engineering, IEEE Transactions on,* vol. 27, pp. 58-93, 2001.

[26] K. M. Eisenhardt, "Building Theories from Case Study Research. ," *The Academy of Management Review,* vol. 14, pp. 532-550, 1989.

[27] I. P. Kotter, "Leading Change: why transformation efforts fail," *Harvard business review,* 1995.

[28] M. J. Earl, "The new and the old of business process redesign," *The Journal of Strategic Information Systems,* vol. 3, pp. 5-22, 1994.

[29] S. Guha, V. Grover, W. J. Kettinger, and J. T. Teng, "Business process change and organizational performance: exploring an antecedent model," *Journal of Management Information Systems,* pp. 119-154, 1997.

[30] R. H. Kilmann, M. J. Saxton, and R. Serpa, "Issues in understanding and changing culture," *California Management Review,* vol. 28, pp. 87-94, 1986.

[31] D. W. Johnson and R. T. Johnson, *Cooperative learning*: Wiley Online Library, 1990.

[32] M. L. Markus and M. Keil, "If we build it, they will come: Designing information systems that people want to use," *Sloan Management Review,* vol. 35, pp. 11-11, 1994.

[33] J. Motwani, "A business process change framework for examining lean manufacturing: a case study," *Industrial Management & Data Systems,* vol. 103, pp. 339-346, 2003.

[34] E. Mumford, "New treatment or old remedies: is business process reengineering really sociotechnical design," *Journal of Strategic Information Systems,* vol. 3, pp. 313-326, 1994.

[35] B. Appleton and R. Cowham, "Defining Agile SCM: Past, Present & Future," *CM Journal.,* 2008.

[36] U. Dombrowski and T. Zahn, "Design of a lean development framework," in *Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on*, 2011, pp. 1917-1921.