New Trends in Software Methodologies, Tools and Techniques
H. Fujita and G.A. Papadopoulos (Eds.)
IOS Press, 2016

doi:10.3233/978-1-61499-674-3-213

Cross-Tool Interoperability in Heterogeneous Application Lifecycle Management Systems

József KLESPITZ [a,1], Miklós BÍRÓ [b] and Levente KOVÁCS [a]
[a] *Obuda University, Budapest, Hungary*
[b] *Software Comptence Center Hagenberg, Hagenberg, Austria*
Johannes Kepler University, Linz, Austria

Abstract. Application lifecycle management (ALM) systems, used to support software development processes, can be composed from various components depending on the need of company. In this paper we focus on benefits of establishment of cross-tool interoperability with specific regard to heterogeneous platforms. We discuss how the different components can be connected for systemic tool collaboration with the help of Open Services for Lifecycle Collaboration (OSLC). This helps to improve traceability and consistency. Furthermore, workflow can be generated to correct the traceability gaps, consistency problems and even to implement modifications. Moreover, we present how the workload of developers can be decreased by integrating revision control system and development environment with the existing ALM system.

Keywords. application lifecycle management, development process improvement, open services for lifecycle collaboration, tool interoperability, tool integration

1. Introduction

Safety-critical software developments and development of software with high financial impact (e.g. bank- and telecommunication sector) requires mature development processes to create reliable and high quality software. Management of development processes is crucial which is solved by using application lifecycle management (ALM) systems. The aim of ALM system are (among others) to help supervision and interaction between different software versions and releases, tracking the status of tests, providing key process indicators, and to present quality for external organizations (e.g. quality assessors and customers) [1]. In other words, ALM system has to provide information for management and responsible people (governance), it has to present the aptitude of development processes for assessors and business partners (development) but it has to reduce the documentation burden and work effort (operation) as discussed in [2].

Establishment of improvement of an ALM system is a difficult task as universal solution is unavailable due to the custom needs. The needs and preferences of the company

[1]Corresponding Author: PhD student, Obuda University,H-1034 Bécsi str 96B Budapest, Hungary; E-mail: klespitz.jozsef@phd.uni-obuda.hu.

Figure 1. ALM providers and their capability [5]

has to be collected and the choice from a wide variety of vendors has to be proved. Considerable aspects can be read in [3,4]. Vendors on this market provide both comprehensive, general systems, and programs to solve particular problems at this field. On Fig. 1. some of the important vendors are compared regarding their completeness of vision and their ability to execute.

Depending on the choice(s) of the company one or more solutions are chosen where the artifact relationship management is usually solved partially with these single programs, but the systemic management has to be customized. In this paper we provide a method to implement cross-tool interoperability with Open Services for Lifecycle Collaboration (OSLC). OSLC supports the integration which results improved traceability and consistency among others. Furthermore, the interoperability makes it possible to generate workflow automatically to execute changes and to eliminate systemic deficiencies. Altogether, this will result a more transparent, more effective and more user friendly environment.

Due to the increased benefits we focus in this research on companies who established their ALM system from many different components with different vendors where the artifact relationship is not managed completely by these. However, companies who use only a few tools to establish their ALM system with built-in artifact relationship management can still benefit from the ideas presented in this paper, but the main targets are still heterogeneous ALM systems.

The paper is structure as the following: Firstly, it is discussed why the gained benefits are important (Chap. 2). Afterward, the importance of interoperability is presented (Chap. 3). Finally, we explain the advantage of using OSLC for system integration (Chap. 4) and we propose a system setup where our ideas can be evaluated (Chap. 5).

2. Importance of Improvement

Nothing shows importance of traceability better than standard IEC 61508 recommends [6] the use of forward and backward traceability (together bilateral or bidirectional traceability) for Safety Integrity Level (SIL) 1 and 2, and it is highly recommended for SIL 3 and 4 applications. Furthermore, traceability is required at various fields. In generic software development the Capability Maturity Model Integration (CMMI) [7] and standard ISO/IEC 15504 requires [8] traceability in development. In safety-critical software developments DO-178C has similar prescription for airborne systems [9], while ISO 26262 for road vehicles [10]. Moreover, Automotive SPICE [11] and MDevSPICE [12] has to be mentioned as further directives with traceability related ordinance for autos and medical devices.

Current paper focuses mostly on development of medical devices. In this domain the Medical Device Directive (MDD) of European Council [13] and guidance of the Food and Drug Administration (FDA) of the United States [14] are the most noticeable guidelines which involves traceability. Furthermore, standards IEC 62304 [15], ISO 14971 [16], IEC/TR 80002 [17], and ISO 13485 [18] has to be mentioned. More can be read about traceability in [19].

Consistency of system was prescribed first by the Automotive SPICE published in 2015 [11]. They have defined consistency that the examined contents and semantics should not contradict each other. Although, we focus on medical development the benefits and the possibly appearance in medical standards and directives make worthy to use it at this domain as well.

The software under development has to fulfill traceability and consistency throughout the development as it is shown on Fig. 2.

Workflow generation is examined as another aspect. Companies have processes to eliminate traceability gaps and to implement requirement modifications. To control such process usually low or middle managers are responsible. Their job can be disengaged and redirected which means additional valuable labor. Furthermore, workflow generation for requirement change can be used to simulate possible design modification which provides useful information about efforts and costs for middle and top management.

3. Need of Cross-tool Interoperability

As it was already mentioned the research focuses on heterogeneous ALM systems which consists various incompatible components. The reason establishing ALM system from different tools may varies: Company may choose different vendors for best fitting or not to depend on a certain provider. If the components of the ALM system were bought at different times to solve arisen problem that time may cause inhomogeneity as well. The heterogeneity is fixed increasingly over time: tremendous amount of information is stored in ALM systems which means precious intellectual property. Migration, to use compatible components, may cause data corruption or even the entire database may become unavailable. Furthermore, people have to overcome reluctance against new interfaces and approaches.

In such a diverse environment it is inevitable that traceability, consistency and usability related problem will occur. In cross-tool relationships (e.g. connection between

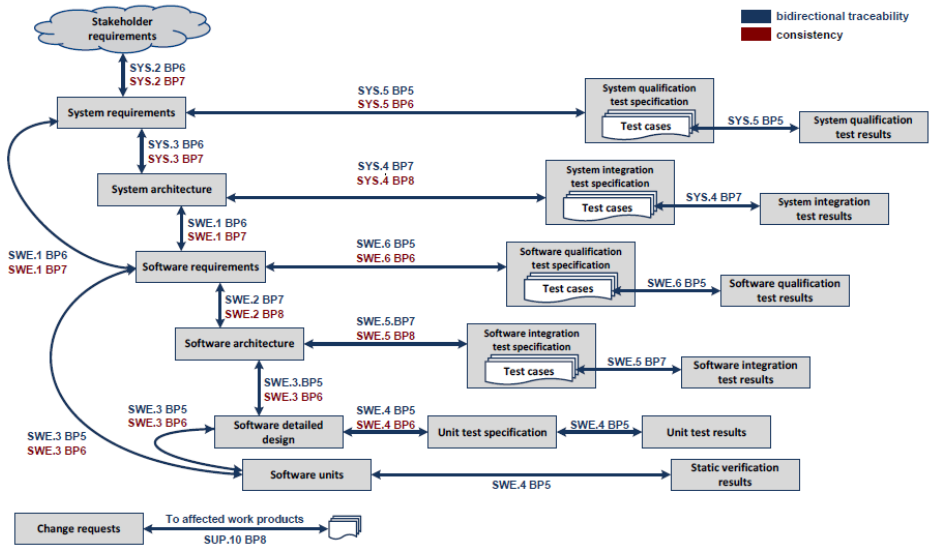

Figure 2. Traceability (blue) and consistency (red) when using V-model [11]

a requirement and its tests both at different tools) aging will erode the traceability, and could cause inconsistency. Furthermore, developers have to switch between the different tools to gather or register information, which decreases their overall effectiveness. Moreover, it is likely to have the same entry at multiple location and such redundancy further decreases the overall performance.

To eliminate these weaknesses cross-tool interoperability has to be established. By having a system where all of the components are capable to communicate with each other the above mentioned problems can be eliminated [20]. Direct links between different artifacts help to have clear traceability, where inspections of consistency and traceability can be easily run automatically. Furthermore, using the known relationships between the artifacts workflow can be created automatically for changes or fault corrections (where fault correction means both bug fixing and restoring consistency). What is more, in case of redundant entries or straightforward relationship certain artifacts can be generated. For example if a revision control system is connected to the tool chain the comment field can be used to describe the modification, explain and store the content in the issue tracking system and even the necessary automatic tests can be triggered.

4. Linked Data for Interoperability

The idea to use interoperability in not new [21,22]. However, we wish to point out the lacks of previous concepts and establish a more effective model. A certain information should exist in the system. Copying items unnecessarily consumes data storage and system resources, while increasing the chance of inconsistency if the copied items are not checked regularly. The transferred data should consist the minimal necessary information to avoid using complex metadata, which is nearly impossible to standardize. The used solution should be independent from vendors. The support is not always guaran-

teed and the intellectual property stored in ALM systems should not depend on a vendor. Moreover, an open solution provides integration any necessary tool without restricting other system providers.

The Open Services for Lifecycle Collaboration (OSLC) provides such a solution [23], Fig. 3. It uses web architecture, and creates linking between artifacts via linked data. Single entries can be kept as they can be reached via URIs without copying. The used standard is accepted by many important vendors. OSLC becomes more popular in the last years and a growing number of adaptor can be found on the market developed by the vendors themselves (e.g. IBM developed adaptors for Atlassian JIRA or Git) or by third party (e.g. Tasktop).

Applicability of OSLC is researched and proofed to be effective [24,25,26]. However, the existing solutions do not exploit the full potential. Therefore, we wish to expand interoperability for further benefits.

We wish to carry out a case study to prove our theory. In our setting four tools will be integrated: a requirement management system, a test management system, an issue tracking system, and a revision control system both from different vendors. By connecting the requirement- and test management systems we expect to have in-creased traceability and run automatic inspection. Looking for missing links with ancestors or requirement-test counterpart can be easily solved in such a transparent environment. The traceability supervision can be extended with semantic checks which are useful for consistency analysis as well. All of the previously mentioned checks can be run automatically.

If any problem is found it has to be fixed. The workflow to restore traceability and/or consistency can be created automatically by analyzing the descendants and related tests of a requirement. Furthermore, such workflow can be generated if a requirement changes: its ancestors has to be reviewed if they are not obsolete or contradicting and similarly all of its descendants have to be checked. The related tests have to be re-executed. Again, all of these tasks can be explored programmatically and they can be interlinked to a workflow in the issue tracking system.

By integrating the revision control system it is possible to partially automatize the issue tracking. If the commit comment contains the identifier of the related issue or requirement then the other part of the commit message can be used to describe the issue and push the workflow to the next state. Moreover, with semantic checks even the requirement can be analyzed if it is in accordance with the change or not.

Though, it is slightly related, the development environment itself can be integrated into the tool chain as well. With the help of delegated user interfaces of OSLC usability can be increased. The open issues of a developer can be shown just inside of the development environment together with the regarding requirements and/or tests. Such solution could increase productivity as the developer does not have to navigate among different clients, but everything would be at hand.

The requirement management system, the issue tracking system and the test management system have already OSLC adapters. However, OSLC adapter has to be created for the revision control and the delegated user interfaces has to be programmed as well. These tools are completed with a fifth one, which is Tasktop Sync. This program was chosen to provide the necessary adaptors and create the linking between the artefacts. As its name suggests it was designed to synchronize data be-tween different tools. It is not only capable to copy certain objects there-and-back and keep the updated, but it provides the possibility to connect different artifacts via web links. Furthermore, all of the neces-

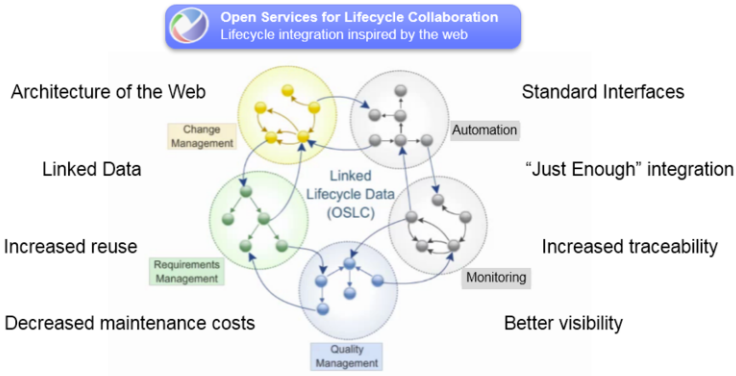

Figure 3. Benefits of OSLC

sary information can be accessed with its help externally [27]. Application of this tool helps us to focus on the relevant tasks (trace-ability and consistency check, workflow generation) as it provides an effective inter-face.

We believe that the case study will prove that not only the quality of the software increases by realizing the above mentioned cross-tool interoperability, but it will also noticeably decrease the human workload and enhance the developers experience.

5. Conclusion and Further Works

The need of ALM systems in modern software development is unquestionable. They support both the management in governance, the development by organizing information, and operation by providing the necessary information.

Nowadays many vendor is competing for the companies. Although some of them provide more or less solutions it is still common that a company use different tools from various producers. The reason for this can be various: incrementing introduction of ALM system, best tailored ALM system by choosing the most suitable components, completing an existing system, afraid to change not to lose intellectual proper-ty, etc.

The various programs raise a challenge for everyone. It is not only hard to keep updated databases which are not directly connected, but the developers has to use many different environment as well. In order to enhance traceability, consistency, and usability it is practical to integrate the used platforms.

Nowadays, the most advanced technology is to use web architecture to create connection between the artefacts. The use of linked data prevents many problems of the previous solutions such as redundant data storage or aging of databases. Open Services for Lifecycle collaboration (OSLC) provides a widely accepted standard for this.

In this paper we have prepared the theoretical base of a case study where the applicability and benefits of OSLC will be examined.

We are expecting to:

- have better traceability and consistency,
- explore traceability and consistency defects automatically,
- generate workflow automatically to repair these defects,

- generate workflow automatically if a requirements is changed or created,
- integrate revision control system to eliminate redundant documentation,
- show every necessary information for the developers in the development environment.

The case study will carry out on an ALM system with tools from four different vendors. The integration will be created with the help of Tasktop Sync, which will provide the framework for this research and provides the possibility to access every necessary information out of databases.

6. Acknowledgment

The authors are grateful for the support of Research and Innovation Center of buda University. The work is supported by the European Research Council Starting Grant ERC-StG 679681.

The research reported in this paper has been supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

References

[1] Sandeep Chanda and Damien Foggon. Application lifecycle management. In *Beginning ASP. NET 4.5 Databases*, pages 235–249. Springer, 2013.
[2] David Chapell. White paper: What is application lifecycle management. 2008.
[3] József Klepsitz, Miklós Biró, and Levente Kovács. Aspects of improvement of software development lifecycle management. 2015.
[4] Jim Brown. White paper: A plm systems comparison of the industry's top tools. Techtarget, 2016.
[5] Shane Hastie. White paper: Gartner and software advice examine agile lifecycle management tools. InfoQ, 2015.
[6] Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010.
[7] Cmmi for development, 2010.
[8] Information technology - process assessment, 2011.
[9] Software considerations in airborne systems and equipment certification, 2012.
[10] Road vehicles functional safety, 2011.
[11] VDA QMC Working Group 13. Automotive spice process assessment / reference model, 2015.
[12] Celtic MedTech Ltd. Mdevspice, 2015.
[13] Directive 93/42/eec of the european parliament and of the council concerning medical devicesl, 2007.
[14] Chapter i - food and drug administration, department of health and human services sub-chapter h - medical devices, part 820 - quality system regulation, 2015.
[15] Medical device software - software life-cycle processes, 2006.
[16] Medical devices - application of risk management to medical devices, 2009.
[17] Medical device software - part 1: Guidance on the application of iso 14971 to medical device software, 2009.
[18] Medical devices - quality management systems - requirements for regulatory purposes, 2003.
[19] Jane Cleland-Huang, Orlena Gotel, and Andrea Zisman. *Software and systems traceability*, volume 2. Springer, 2012.
[20] Weiqing Zhang, Vincent Leilde, Birger Moller-Pedersen, Joel Champeau, and Christophe Guychard. Towards tool integration through artifacts and roles. In *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*, volume 1, pages 603–613. IEEE, 2012.
[21] Amit Fisher, Ben Williams, and Schuyler Matthews. Linked lifecycle data–connecting distributed engineering data and harvesting its inherent value. In *INCOSE International Symposium*, volume 23, pages 331–345. Wiley Online Library, 2013.

[22] Bernhard K Aichernig, Klaus Hormaier, Florian Lorber, Dejan Nickovic, Rupert Schlick, Didier Simoneau, and Stefan Tiran. Integration of requirements engineering and test-case generation via oslc. In *Quality Software (QSIC), 2014 14th International Conference on*, pages 117–126. IEEE, 2014.

[23] OSLC Core Specification Workgroup. Oslc core specification version 2.0. *Open Services for Lifecycle Collaboration, Tech. Rep*, 2010.

[24] Mehrdad Saadatmand and Alessio Bucaioni. Oslc tool integration and systems engineering–the relationship between the two worlds. In *Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on*, pages 93–101. IEEE, 2014.

[25] Mehrdad Saadatmand and Mikael Sjödin. Master thesis: Tool integration using oslc. 2013.

[26] Gilbert Regan, Miklos Biro, Fergal Mc Caffery, Kevin Mc Daid, and Derek Flood. A traceability process assessment model for the medical device domain. In *Systems, Software and Services Process Improvement*, pages 206–216. Springer, 2014.

[27] Task linking configuration user guide. Tasktop Technologies, 2016.