Enhanced traceability and consistency with Augmented Lifecycle Space

József Klespitz[1], Miklós Bíró[2,3], Levente Kovács[1]
[1] Óbuda Univesity, Budapest, Hungary
[2] Software Competence Center Hagenberg, Hagenberg, Austria
[3] Johannes Kepler Universität, Linz, Austria
klespitz.jozsef@phd.uni-obuda.hu, miklos.biro@scch.at, kovacs.levente@nik.uni.obuda.hu

Abstract — **Modern software development is unimaginable without the help of Application lifecycle management systems (ALMs) due to the heavy documentation and communication needs, especially in safety-critical developments. In this paper the idea of Augmented Lifecycle Space is presented together with its use and benefits. The main target of application is to enhance traceability and consistency throughout the development with minimal or reduced human effort. The idea can be realized easily in homogeneous ALM systems. In case of heterogeneous systems, which is the main scope of this research, some kind of interoperability is required among the used tools. Open Services for Lifecycle Collaboration (OSLC) is a possible solution and it is used to illustrate a possible setup for further studies.**

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

I. Introduction

In modern software development the aim of the developers is to deliver high quality software reliably as fast as possible. In safety-critical developments this is completed with the minimization of risk until a level as low as reasonable practicable (ALARP). Therefore, mature development processes are inevitable where the operation of final product has some potential hazard (such in aerospace, automotive or medical industry) or the software could have high financial impact (e.g. bank- and telecommunication sector). It is crucial to manage development and to ensure customers about the appropriate operation. Therefore, companies use *application lifecycle management* (ALM) systems. The tasks of ALM systems can be formulated with the following three activities: governance, development and operation [1]. This means that ALM system supports the management by providing crucial information such as key process indicators, helps in distribution and management of tasks of developers and it provides the necessary documentation to prove the satisfactory operation according to the customer needs or to the standards and directives [2].

ALM systems consist more software components where the service provider is not necessarily the same. It is a hard choice to select the most suitable or best customizable solution from the selection of various vendors. Some guidance can be found for evaluation in [3, 4], while Fig. 1. presents some competitors by comparing their vision and their ability to execution. The choice can vary in a wide spectrum: Complete solutions can be bought from a single vendor or even all system components can be obtained from independent service providers.

However, it is sure that relationship management has to be solved; In case of comprehensive service, the vendor (usually) has an individual solution to establish connection between artifacts. In case of multiple providers, such connection has to be created tailored to the used software.

In such heterogeneous systems are the benefits of Augmented Lifecycle Space the most clear. Hereby, the existing artifact relationships are used to find the deficiencies of the system. The missing connections and artifacts then can be either generated automatically or can be created manually. Nevertheless, the Augmented Lifecycle Space shows the ideal state of the system and the steps for correction can be planned.

In this paper we present how the Augmented Lifecycle Space can be created together with its benefits. In our research we focus on developments, where heterogeneous ALM system is established and the development environment is highly regulated. In our research we recommend to use *Open Services for Lifecycle Collaboration* (OSLC) to establish artifact relationship management where it is missing.

The rest of the paper is structured as the following: At first, the regulation environment and possible developments methods are presented (Chapter 2). Afterward, the importance of traceability and consistency is highlighted together with the special challenges for assessors (Chapter 3). Thereafter, the idea of Augmented Lifecycle Space, its realization and its benefits are discussed (Chapter 4). Finally, the paper ends with conclusion and possible enhancements.

II. Importance of Traceability and Consistency

For safety-critical software the forward and backward traceability (together bilateral or bidirectional traceability) is required by standard IEC 61508 [5]. For Safety Integrity Level (SIL) 1 and2 it is recommended while for SIL 3 and 4 it is highly recommended. The importance of traceability is clear as it is prescribed not only in the aforementioned standard, but in many others. The Capability Maturity Model Integration (CMMI) [6] and standard ISO/IEC 15504 [7] have provision for general software development. Similar provision can be found for safety-critical developments: standard DO-178C [8] has requirements for airborne system, while standard ISO 26262 [9] is responsible for road vehicles.

978-1-5090-1216-9/16/$31.00 ©2016 IEEE

Figure 1. Important vendors and their performance [29]

Moreover, Automotive SPICE [10] and MDevSPICE [11] has to be mentioned as further directives with traceability related ordinance for autos and medical devices.

Most important regulation for medical device developments are also highlighted, as these are the main target of this research. In this domain the Medical Device Directive (MDD) of European Council [12] and guidance of the Food and Drug Administration (FDA) of the United States [13] are the most noticeable guidelines which involves traceability. Furthermore, standards IEC 62304 [14], ISO 14971 [15], IEC/TR 80002 [16], and ISO 13485 [17] has to be mentioned. More can be read about traceability in [18].

The need for consistency is straightforward, yet only the latest version (published in July, 2015) of Automotive SPICE [10] prescribes it. According to their definition consistency means the lack of contradiction among the analyzed contents (requirements, test cases, etc.). Although, this regulation is not related to medical developments, yet it is highly possible that it will appear in the near future. Therefore, it is practical to consider it already in our research. The need (and check points) for analyzing traceability and consistency can be seen on Fig. 2. in case of the classical V model.

In [19] Gotel at al. visions seven requirements for the ultimate traceability. By analyzing the properties of the ideal traceability it is clear that every participant benefits from it: High managers are capable to trace the development process, low managers can establish and track workflows, developers and testers have up-to-date and inspected requirements or test cases, and the users are assured that every step was executed as necessary. However, instead (or next to) end users it is a common practice to validate the development process with the help of assessors.

Hereby, the assessors face with great challenge: they have to get familiar with the development quickly, they have to navigate among many (often unknown) tools and they can only see snapshots at certain points.

The aim of this research targets every participants. Our goal is ensuring everybody that every step in the development process is planned, reviews, implemented, tested and validated. This is done with automatized analyzes where human factor is eliminated.

III. Augmented Lifecycle Space

Companies may choose to setup their ALM system from various components even if the heterogeneous system will most probably have incompatibilities. The first and most important factor is suitability: the best fitting or best customization may require software from different vendors. Elder companies have rarely established ALM systems completely from nothing. System components bought in different times are rarely abandoned as people have to overcome reluctance against new interfaces and approaches. Furthermore, the amount of stored information makes the migration hard. Moreover, the date stored in ALM system is intellectual property which is hard to replace. The fear of losing data often prevents improvement.

It is inevitable to avoid consistency and traceability related problems in such heterogeneous environment if the system components are unconnected. Cross-tool relationship is especially important to avoid erosion of traceability and inconsistency. (Such event occurs if the requirements and test reports are handled in different tools and the artifacts changes do not affect each other.) Another aspect which can hinder development is distributed environment. Developer using different tools have to navigate among various windows which reduces efficiency. Finally, storing the same entry at multiple location without proper and automatic maintenance require unnecessary human workload and it is a possible starting point for erosion.

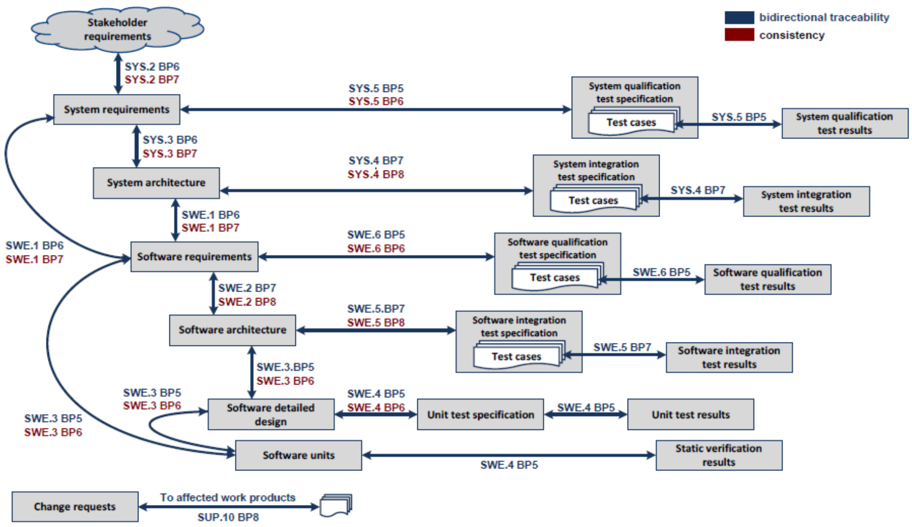

Figure 2. Traceability (blue) and consistency (red) when using V-model [10]

The idea of Augmented Lifecycle Space was invented to get rid of the aforementioned problems. However, its application presumes artifact relationship managements. The idea to use interoperability in not new [20, 21] and it inevitable according to the foregoing. Hereby, we suggest to use Open Services for Lifecycle Collaboration (OSLC) if possible [22]. OSLC is ideal for our research for many reasons: it has high industrial support and many relevant vendor is compatible with this standard. It uses web architecture and linked data, this way it requires minimal data traffic and copying of artifacts can be avoided which results reduced data storage need. Sent information is reduced to the minimal necessary amount, this way it is again undemanding in resources. The benefits of using OSLC can be seen on Fig. 3.

Applicability of OSLC is researched and proofed to be effective [23-25]. It has become 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). Augmented Lifecycle Space was invented to utilize the full potential of OSLC.

Assuming an ALM system with proper cross-tool interoperability provides a lifecycle space. This consist every artifacts and their mutual connections in the system. This real lifecycle space consist faults and deficiencies which need to be eliminated. To get rid of these problems the real lifecycle space is copied virtually.

This virtual space will be completed (augmented). In order to do this problems have to be found; Firtsly, the existing artifacts has to be categorized according to a chosen model. (This model can be a classical V-model as in [10] or may use a completely different approach as in [26].) According to the applied model the existing relationships and the artifacts has to be analyzed to find the missing components. Missing artifacts have to be automatically created, this way augmenting the system. Missing links have to be created in the same manner if no any artifact is required.

This Augmented Lifecycle Space can be used for more purposes. The most trivial application is to use the augmented components to correct the flaws of the existing system. In order to do this the necessary workflows can be created automatically as every necessary information is at hand to do this. On the other hand, the Augmented Lifecycle Space can be used for project planning as well. The labor needs can be simulated if it contains a new feature or a modified requirement. In such case the Augmented Lifecycle Space contains every modification which is necessary for the implementation without modifying the original database. Moreover, the workflows can be generated similarly as in the previous case, this way the management can have a complete vision of execution.

Altogether, application of Augmented Lifecycle Space has the following benefits:

- explore missing artifacts and relationships,
- ensure a complete and flawless system,
- generates workflows for fault correction,
- simulates the effect of a new feature/feature modification without disturbing the original database,
- predict labor needs for future modifications.

All of the aforementioned problems are solved automatically.

System improvement done this way has additional benefits: Code level traceability can be easily achieved through cross-tool interoperability. Low managers are disencumbered as workflows are generated automatically. If redundancy is inevitable it can be copied and maintained automatically. Finally, the different development tools can be integrated into a single one, where every user could see every necessary information, which highly increases usability and efficiency.

A case study is planned to validate our concepts. In the planned research four tools will be involved: requirement management system, test management system, issue tracking software and a revision control system all of them from different vendors. A fifth tool *Tasktop Sync* [27] will be used to provide the required OSLC adaptors.

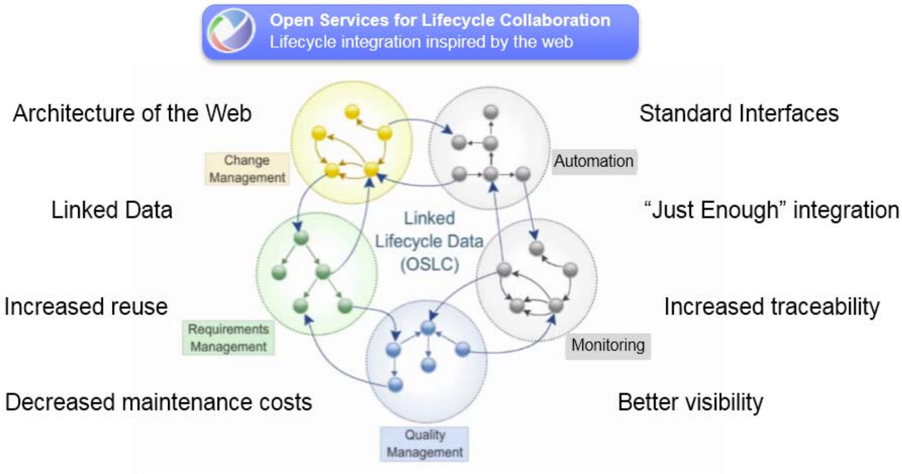

Figure 3. Benefits of OSLC

IV. Conclusion and Further Work

Modern software development is unimaginable without ALM systems. Traceability and consistency are cardinal feature which are prescribed by related standards and directives. In this research we are looking for answer how to eliminate traceability gaps and inconsistencies while helping everyday work and improving efficiency.

In heterogeneous systems (and in homogeneous systems as well) cross-tool interoperability is upmost important to have an effective artefact relationship management. We suggest to use OSLC to establish interconnection between system components as it provides an ideal platform for further improvements.

In such environment the idea of Augmented Lifecycle Space can be realized. Hereby, the missing artefacts and relationships are explored according the used development model. The missing items can be created in the Augmented Lifecycle Space and workflows can be generated automatically to achieve it in reality. The idea can be used to solve existing problems and to plan feature improvement as well.

In the future we are going to test our idea in practice with four system components all from different vendors. Results will be shared in the form of a case study.

Acknowledgment

The authors are grateful for the support of Research and Innovation Center of Óbuda University and Doctoral School of Applied Informatics and Applied Mathematics. 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] D. Chappell. *What is application lifecycle management*?, white paper, 2008, available at http://www.davidchappell.com/WhatIsALM--Chappell.pdf

[2] S. Chanda, F. Damien. *Application Lifecycle Management*. In Beginning ASP. NET 4.5 Databases. Apress, 2013. p. 235-249.

[3] J. Klespitz, M. Biró, L. Kovács. *Aspects of improvement of software development lifecycle management*. In 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI), 2015: p. 323-327.

[4] J. Brown. *A PLM systems comparison of the industry's top tools*, Techtarget, available at http://searchmanufacturingerp.techtarget.com/feature/A-PLM-systems-comparison-of-the-industrys-top-tools

[5] IEC, IEC-61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. 2010, Geneva, Switzerland.

[6] SEI, CMMI® for Development, Version 1.3. 2010, Software Engineering Institute

[7] ISO/IEC, ISO/IEC 15504: *Information Technology - Process Assessment*. 2011: Geneva, Switzerland.

[8] RTCA/DO, RTCA/DO-178C: Software Considerations in Airborne Systems and Equipment Certification, 2012.

[9] ISO, ISO 26262: *Road vehicles – Functional safety*. 2011, Geneva, Switzerland.

[10] Automotive, S. I. G., VDA QMC Working Group 13 Automotive SPICE Process Assessment / Reference Model, v3.0, 2015:470.

[11] Celtic MedTech Ltd., *MDevSPICE*, 2015, Dundalk, Ireland.

[12] .European Commission, *Directive 93/42/EEC of the European Parliament and of the Council concerning medical devices*, in OJ o L 247 of 2007-09-21. 1993: European Commission, Brussels, Belgium.

[13] FDA. *Chapter I - Food and drug administration, department of health and human services subchapter H - Medical devices*, Part 820 - Quality system regulation; Available from: http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820.

[14] IEC, IEC 62304: Medical Device Software - Software Life-Cycle Processes. 2006, IEC: Geneva, Switzerland

[15] ISO, ISO 14971 - Medical Devices - Application of Risk Management to Medical Devices 2009, ISO: Geneva, Switzerland.

[16] IEC, IEC TR 80002-1 - Medical Device Software - Part 1: Guidance on the Application of ISO 14971 to Medical Device Software. 2009, IEC: Geneva, Switzerland.

[17] ISO, ISO 13485: Medical Devices - Quality Management Systems - Requirements for Regulatory Purposes. 2003, ISO: Geneva, Switzerland.

[18] J. Cleland-Huang, O. Gotel, A. Zisman. *Software and systems traceability*. Vol. 2(3). Springer, 2012.

[19] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, J. Maletic. *The grand challenge of traceability (v1. 0)*

Software and Systems Traceability, Springer London 2012. p. 343-409.

[20] W Zhang, V. Leilde, B. Moller-Pedersen, J. Champeau, C. Guychard. *Towards tool integration through artifacts and roles* In 19th Asia-Pacific Software Engineering Conference *(APSEC),* 2012, vol. 1, IEEE, 2012, p. 603-613.

[21] A. Fisher, B. Williams, S. Matthews. *Linked Lifecycle Data– Connecting Distributed Engineering Data and harvesting its inherent value*. In INCOSE International Symposium, vol. 23(1), 2013, p. 331-345.

[22] OSLC Core Specification Workgroup. *OSLC core specification version 2.0*. Open Services for Lifecycle Collaboration, 2013.

[23] B. K. Aichernig, K. Hormaier, F. Lorber, D. Nickovic, R. Schlick, D. Simoneau, S. Tiran. *Integration of requirements engineering and test-case generation via OSLC*. In 14th International Conference on Quality Software (QSIC), IEEE, 2014, p. 117-126.

[24] M. Saadatmand, A. Bucaioni. *OSLC Tool Integration and Systems Engineering--The Relationship between the Two Worlds*. In 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2014, p. 93-101.

[25] M. Saadatmand, M. Sjödin. *Tool integration using OSLC*. master thesis, 2013.

[26] J. J. Jiang, G. Klein, H. G. Hwang, J. Huang, S. Y. Hung *An exploration of the relationship between software development process maturity and project performance*. Information & Management, Vol. 41(3), 2004, p. 279-288.

[27] Tasktop Technologies, *Task Linking Configuration – User Guide*. 2016, available at http://help.tasktop.com/help/index.jsp?topic=%2Fcom.tasktop.sync.help%2Fuserguide%2FTask-Linking-Configuration.html&cp=1_0_7

[28] G. Regan, M. Biro, F. Mc Caffery, K. McDaid, D. Flood. *A traceability process assessment model for the medical device domain*. In Systems, Software and Services Process Improvement, Springer Berlin Heidelberg, 2014, p. 206-216.

[29] S Hastie, *Gartner and Software Advice examine Agile Lifecycle Management Tools* In InfoQ, 2015, available at http://www.infoq.com/news/2015/02/agile-management-tools