We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,500
Open access books available

118,000
International authors and editors

130M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

20

Improving Quality Assurance in Automation Systems Development Projects

Dietmar Winkler and Stefan Biffl
Christian Doppler Laboratory
"Software Engineering Integration for Flexible Automation Systems" (CDL-Flex),
Institute of Software Technology and Interactive Systems,
Vienna University of Technology,
Austria

1. Introduction

Development processes of large-scale automation systems, e.g., power plants and manufacturing systems involve engineers from various disciplines, e.g., mechanical, electrical, and software engineers, who have to collaborate to enable the construction of high-quality systems (Biffl *et al.*, 2009a). Engineers in individual disciplines apply domain-specific tools, methods, and data models, which are typically not seamlessly linked to each other. For instance, electrical engineers use circuit diagrams and technical data sheets to model the electrical behaviour of the systems, process engineers focus on process workflows for the instrumentation of the system, and software engineers use software models to develop and test control applications of the system (Hametner *et al.*, 2011).

Because of the heterogeneity of individual disciplines and the missing links between them, project management (e.g., project observation and control) and quality assurance (QA) activities across disciplines become even more difficult. Nevertheless, a comprehensive view on the project, frequent synchronization of systems engineering artefacts between disciplines, and QA activities are success-critical factors for developing large-scale automation systems.

Observations at our industry partner, a hydro power plant systems integrator, showed that these overlapping project activities (i.e., project management and QA) are currently not supported sufficiently (Sunindyo *et al.*, 2010)(Winkler *et al.*, 2011). In typical industry projects in a distributed and heterogeneous environment synchronization between disciplines and QA activities across disciplines are conducted manually and require high effort by experts, who have to overcome media breaks between the outcomes of different tools and data models. In addition, we observed strong limitations of QA activities which leave important and critical defects unidentified. To support systems development activities in heterogeneous environments for project management (PM) and QA, we identified three main challenges:

- *Project and Process Management*. Heterogeneity of tools and data models requires time-consuming activities to assess the current project state across all involved disciplines. Because of a lack of tool support experts have to collect and analyze data manually. Therefore, the current project state based on real data and facts derived from manual project analysis is available very infrequently and on request only. Nevertheless, continuous analysis of engineering projects and the availability of project status reports are key requirements of project managers to (a) enable a comprehensive view on the overall project state(s) and (b) control the course of events based on the analysis results more effectively and efficiently (Moser *et al.*, 2011).

- *Change Management*. Decoupled disciplines and workflows make engineering processes and change management processes more difficult, in particular, if heterogeneous disciplines are involved. For instance, changing a hardware sensor (e.g., an oil pressure sensor) from a digital to an analogue device (executed by the electrical engineer) affects process engineers (required changes in hardware wiring), and software engineers (required change of software variables according to value ranges and data types). Therefore, a second key requirement is to improve collaboration and interaction between engineers (coming from various disciplines) with respect to propagating critical changes to affected disciplines in a controlled way within a short time interval (Winkler *et al.*, 2011).

- *Quality Assurance.* Typically engineers apply isolated QA approaches recommended by standards and industry best practices to assess and improve product quality with a focus on their individual application domain. For instance, electrical engineers apply simulation approaches of wiring and electrical signals (Sage *et al.*, 2009) and software engineers conduct reviews (Sommerville, 2007), inspections (Laitenberger *et al.*, 2000), and testing approaches (Meyers *et al.*, 2004) to identify defects in the artefacts efficiently and effectively. Isolated QA methods focus on an individual discipline and are well-established. Nevertheless, we observed strong limitations regarding QA activities across disciplines and tool borders. New mechanisms are required to support QA across disciplines. Therefore, the third key challenge focuses on enabling and supporting QA in heterogeneous engineering environments across disciplines and domain borders.

Common to all three challenges/requirements is the need to linking heterogeneous environments to support synchronization and QA across disciplines and tool borders. Figure 1 illustrates these challenges on the semantic level. Three basic roles (see Figure 1; positions 1a – 1c), i.e., electrical, process, and software engineers work within their disciplines using specific tools and methods including best-practice QA approaches. Nevertheless, there is a strong need to synchronize artefacts and disciplines (represented by the overlapping areas in Figure 1), which could address specific risks and quality issues. Observations at our industry partner confirmed that QA activities with focus on the overlapping areas of two or more (heterogeneous) disciplines are not sufficiently addressed yet (see Figure 1; position 2) (Biffl *et al.*, 2011).

Common practices for synchronizing different disciplines focus on these overlapping areas, where experts have to discuss and exchange data to bridge these technical and semantic gaps manually (Biffl *et al.*, 2009b). Therefore, we see the need to support this synchronization process by providing inspection and testing approaches with focus on these

overlapping areas to improve QA activities by means of increasing product quality and decreasing effort and error-proneness (caused by the manual synchronization process). Note that the goal of synchronizing individual disciplines is to focus on the overlapping areas, leaving discipline-specific data within their assigned tools. In this chapter we present an approach for identifying these common concepts as foundation for addressing these overlapping areas and show benefits for QA activities, i.e., defect detection across disciplines and project observation and control.
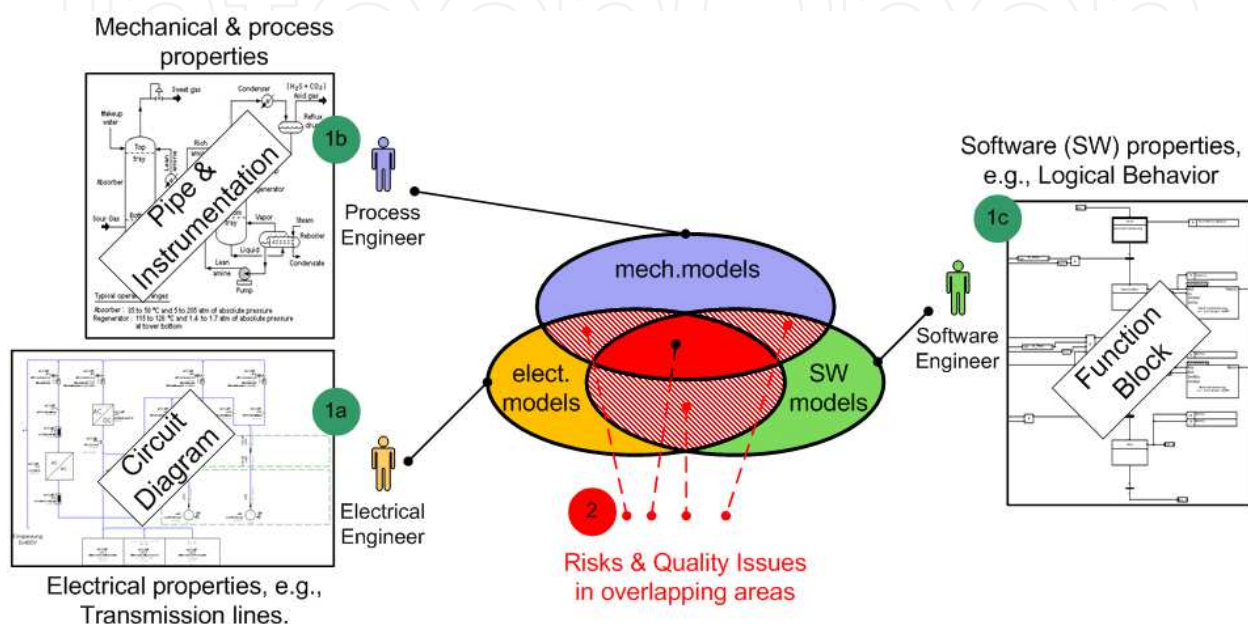

Fig. 1. Risks and Quality Issues in overlapping areas in heterogeneous engineering environments (Biffl *et al.*, 2011).

The reminder of this chapter is structured as follows: Section 2 provides an overview on the related work and Section 3 highlights the research issues. Section 4 describes the basic concepts of the Automation Service Bus (ASB) and Section 5 presents a pilot application for improving QA aspects based on the ASB. Finally, Section 6 summarizes, concludes and identifies future work.

2. Related work

This section summarizes related work of automation systems development processes and software QA as lessons learned from business IT software development for application in large-scale automation systems engineering projects.

2.1 Automation Systems development processes

Automations Systems (AS), such as power plants and industrial automation systems for manufacturing purposes, include distributed software components to control systems behavior (Biffl *et al.*, 2009b). Increasing complexity of software products require well-defined processes and methods for software and systems construction and verification and validation. Various software and systems engineering processes support engineers by

providing sequences of steps for project planning and control, e.g., GAMP (Gamp, 2008), W-Modell (Baker *et al.*, 2008), eXtreme Programming (Beck *et al.*, 2004), Scrum (Schwaber, 2004), and V-Modell XT[1]. Nevertheless, process standards focus on the organizational structure of software and systems engineering projects with limitations on method support, tooling and synchronizing various and heterogeneous disciplines.

Observations at our industry partner showed a basic sequential engineering process in Automation Systems Engineering (ASE) development projects (see Figure 2 for details). The observed system development process includes a set of sequential steps including isolated (discipline-specific) QA activities conducted by experts or groups of experts in the individual domain. Because of the sequential process structure, changes from late phases of development (e.g., during test and/or commissioning) can have a major impact on previous phases of the project and can lead to project delays in case of critical changes. Note that these effects are common to sequential and waterfall-like development processes in homogeneous engineering environments (Sommerville, 2007).
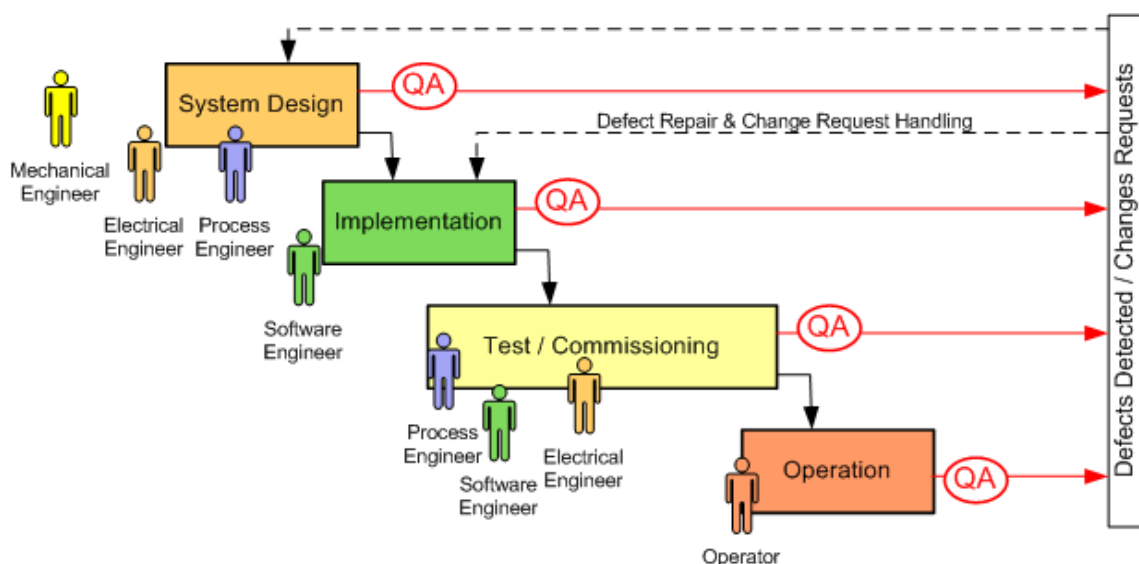

Fig. 2. Sequential Engineering Process with isolated Quality Assurance (QA) Activities.

Considering AS development projects, where experts work distributed in heterogeneous environments, effects of late changes are more critical, more risky, and error prone. Engineers from individual disciplines work concurrently during the development project. Therefore, frequent synchronization between these disciplines is a success-critical issue during development and change request handling (see Figure 3a). For instance, a wrong alarm indicator of an oil pressure sensor in the control center (identified during test and/or commissioning) might affect software engineers (because data handling could have been implemented incorrectly), electrical engineers (wrong alarm sensor type used or incorrectly wired) or the process engineer (incorrect sensor planned). This kind of defects might remain undetached if not tested appropriately. If such a defect is uncovered, there is a need for analyzing the defect and the origin of the defect (across all involved disciplines). A weak link between different disciplines will hinder efficient analysis of defects across disciplines

---

[1] For a description of the V-Modell XT see http://www.v-modell-xt.de

and could lead to quality problems and project delays. Please note that this analysis steps are typically conducted by experts who are familiar with at least two involved disciplines. Figure 3a presents a basic synchronization step applicable in every phase of the sequential process workflow. Technical integration of tools and semantic integration of data models could help supporting synchronization across disciplines and tool borders (see Figure 3b).
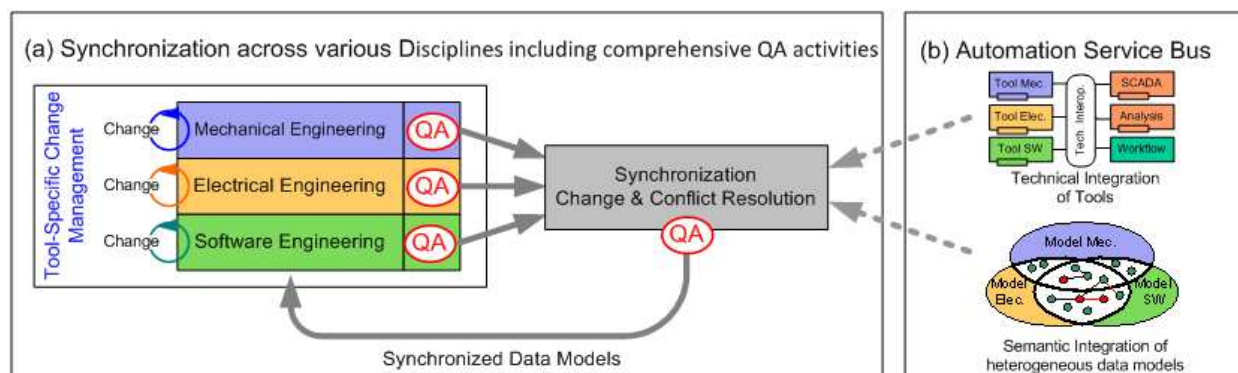

Fig. 3. Synchronization of heterogeneous disciplines with QA (Winkler *et al.*, 2011).

In current industry projects this synchronization step is conducted manually by experts. Expert knowledge is embodied in domain-specific standards, terminologies, people, processes, methods, models, and tools (Moser *et al.*, 2010b). Note that these standards typically do not support technical and semantic integration of tools and data models across disciplines. Assuming that technical and semantic gaps between different engineering experts lead to a lack of QA of artefacts and inefficient change management approaches (Schäfer *et al.*, 2007), a major challenge is to bridge the gap between heterogeneous disciplines on a technical and semantic level to enable efficient change management, quality assurance, and data collection for project monitoring and control during development, commissioning, and maintenance.

Technical and semantic integration of tools and data models across disciplines enables frequent synchronization and data exchange, supports efficient change management processes, and enables more effective and efficient QA. In addition, processes across disciplines and tools borders become observable and – as a consequence – enable effective and efficient project management (PM), project monitoring, and control. Figure 3b illustrates the basic contribution of the ASB approach for technical integration of tools and semantic integration of data models to support PM and QA more effectively and efficiently.

2.2 Quality Assurance aspects for Automation Systems development

Quality Assurance (QA) – embedded within isolated disciplines – is supported by appropriate methods and tools (Schulmeyer, 2008). Nevertheless, a key challenge is to conduct QA activities across domain and tool borders. Based on *Software Engineering Best Practices*[2] (Schatten *et al.*, 2010) specific methods from business IT software development, e.g., inspection and testing, are promising approaches for application in AS development projects.

---

[2] See http://bpse.ifs.tuwien.ac.at for additional material related to the book in English language.

2.2.1 Reviews and software inspection

Reviews and (more formal) software inspections are common and well-established QA methods in business IT software engineering to discover candidate issues and defects systematically. Depending on the configuration of the inspection process (Laitenberger *et al.*, 2000), software inspections are applicable to various types of artefacts, e.g., written text documents, models, drawings, and code. In addition, inspections are applicable in all stages of software and systems development. See (Kollanus *et al.*, 2007) for an overview on software inspection research and (Winkler, 2008) for an empirical evaluation of selected inspection variants.

Individual inspectors form an inspection team and apply a defined sequence of steps *(inspection process)* to identify defects (a) individually and (b) in teams (Biffl *et al.*, 2003). Based on individual inspection results, an aggregated team defect list is generated in an inspection team meeting based on interaction and discussions. Depending on the project scope and the problem domain, *reading techniques* support inspectors and inspection teams to focus on a certain type of defects and defect classes. Basically, reading techniques are structured approaches for reading the document under investigation systematically (Basili, 1997)(Biffl, 2001). Example reading techniques are *checklist-based reading* (inspectors apply a pre-defined and/or customized checklist), *perspective-based reading* (based on different perspectives and disciplines), and *usage-based reading* (use cases and scenarios represent the guidelines for defect detection and drive the inspection process)(Winkler, 2008). Assuming that different perspectives will lead to different defects and defect classes, perspective-based reading techniques focus on defect detection from different viewpoints on the artefact under inspection. For instance, the tester view might lead to defects regarding testability of requirements (e.g., based on the completeness of use cases), the developers might focus on a fully specified design and architecture (e.g., ability to implement requirements), and the user view might focus on end-user requirements (e.g., software solutions that must be usable in the customer domain). Usage-based reading focuses on business cases (typically described as use cases) and the value contribution of the software solution within the business domain. Therefore, this reading technique approach focuses on the most important use cases with the most valuable outcome of the software solution (e.g., based on prioritized use cases).

Analyzing the state-of-the practice at our industry partner, we identified software inspection as a candidate method for improving product quality in ASE projects. Experts analyzed overlapping areas during the synchronization process phase to identify defects in a rather unsystematic and informal way. Software inspections techniques and reading techniques (e.g., perspective based reading) can help engineers in better focusing on a certain type of defects coming from various disciplines.

2.2.2 Software and systems testing

Traditional software testing approaches focus on executing a program with the intent to identify defects (Kaner *et al.*, 1999). Nevertheless, the availability of executable code and test information (e.g., test case specification, test data, and test environments) are pre-conditions for applying software testing techniques (Meyers *et al.*, 2004). Traditional testing approaches – aligned with some V-Model approach – focus on different levels of detail within the overall

system (see Figure 4): (a) detecting defects on *component level* (e.g., applying unit tests), (b) integration tests to verify and validate the design and the architecture on *architectural level*, and (c) systems and acceptance test with focus on customer requirements *(system level)*.

Lessons learned from testing business IT software products showed the applicability of prominent basic testing techniques, i.e., Black-Box and White-Box testing techniques (Sommerville, 2007), as promising testing approaches on different levels of AS development projects. The component level focuses on testing and simulation of individual components located at isolated disciplines. Integration testing of components – aligned with the architecture – can be seen as testing across disciplines and domain borders, and acceptance testing seems to be similar to system testing and commissioning at the customers site. Figure 4 illustrates the different levels of testing AS project artefacts. Note that test cases and test scenarios can be defined early, following the Test-Driven (Beck *et al.*, 2004) or Test-First (Winkler *et al.*, 2009) approach based on agile software development, another Best-Practice learned from Business IT software development.
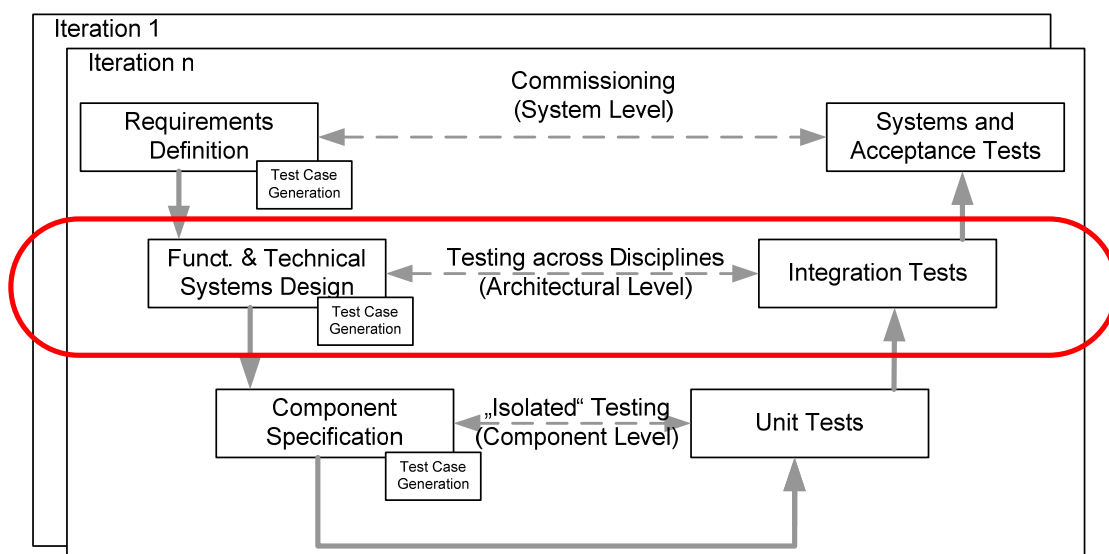

Fig. 4. Test levels in automation systems development according to the W-model (Baker *et al.*, 2008)(Winkler *et al.*, 2009).

In context of AS Black-Box can refer to the *'interfaces'* between various disciplines, e.g., wired connections at a control unit or interface to a software visualization component. Testing these interfaces refers to some kind of '*Black-Box Testing*'. The commissioning phase (comparable to system tests at the customer site), including all hardware and software components of the power plant or manufacturing system, is one of the most critical phases related to QA in the AS domain. Isolated subsystems are launched step by step with real hardware and software. Our observations at industry projects showed that defects – found during this phase – have to be detected manually by analyzing paper work (e.g., drawings) and hardware/software components. Therefore, the commissioning phase requires a very high effort by experts.

The ASB concept aims at supporting QA by enabling testing across domain borders, i.e., testing the overall system *from (hardware) sensor to (software) variables*, comparable to an integration test – well-known from testing business IT software products.

3. Research questions

Heterogeneous engineering environments suffer from weak or missing links between individual disciplines, e.g., mechanical, electrical, and software engineers, on technical and semantic level. Missing links between disciplines hinder efficient collaboration between engineers and makes PM (project observation and control), change management, and comprehensive QA more difficult, risky, and error-prone. Based on observations at our industry partners and related work, we derived the following set of research questions to improve collaboration, project and change management, and QA in heterogeneous environments across disciplines and engineering domains.

Research Question 1 (RQ1). *How can we link various disciplines on a technical and semantic level to enable efficient data exchange in heterogeneous ASE environments?* Efficient data exchange is a pre-condition for effective and efficient PM, change management, and QA. Figure 1 presented the need for collaboration regarding the overlapping areas of individual disciplines, where experts have to synchronize data (from various disciplines) manually. The first research question focuses on eliciting the common concepts, i.e., data represented in the common and overlapping areas, of related disciplines.

Research Question 2 (RQ2). *How can we support QA across disciplines in heterogeneous environments?* Quality assurance aspects can focus on identifying defects in engineering artefacts and overlapping areas of different artefacts coming from various disciplines in a heterogeneous environment. Observations at industry projects showed that these QA activities require a high manual effort provided by experts. We expect a significant improvement (in terms of reducing effort and increasing quality by means of identifying defects more effective and efficient) of QA performance. Therefore, the second research question focuses on providing mechanism to support defect detection in these overlapping areas.

Research Question 3 (RQ3). *How can we support project and quality managers in collecting and analyzing project data (from heterogeneous sources) more effective and efficient to enable continuous project monitoring and control?* Observations at industry projects revealed a high manual effort for collecting and analyzing data from different sources. Because of this high effort (conducted by experts) the project state is captured less frequent and hinder efficient and effective PM. The third research question focuses on providing a 'window to engineering data' across disciplines and domain borders to provide engineering project data tool-supported, frequently and fast.

4. Automation Service Bus for automation systems engineering projects

This section describes the basic concept of the Automation Service Bus (ASB) as a foundation for enabling tool-supported QA activities in heterogeneous environments across disciplines and tool borders (Biffl *et al.*, 2011).

Isolated disciplines apply individual data models and tools with limitations regarding collaboration and interaction (see Figure 1). In industry projects experts bridge the gap between these data models from heterogeneous sources manually. To overcome high effort for manual synchronization and improve the quality of manual activities the Automation

Service Bus (ASB) provides an tool-supported approach to link heterogeneous sources (e.g., data models, data formats, and tools) for PM and QA (Biffl *et al.*, 2009b). In contrast to existing solutions, e.g., Comos PT[3] or EPlan Engineering Center[4], the ASB concept provides a flexible and light-weight infrastructure based on the Enterprise Service Bus (Chappell *et al.*, 2004) concept.

'Flexibility' refers to the concept to respond to changed environments (e.g., introduction of new/modified tools and data models) easily and 'light-weight' refers to reducing the effort for synchronizing different disciplines by focusing on a subset of data (common concepts) similar for all related disciplines. Note that data synchronization can be limited to these common data without considering additional domain-specific data (not relevant for synchronization). These common concepts are represented by a *virtual common data model (VCDM)* (Biffl *et al.*, 2011).
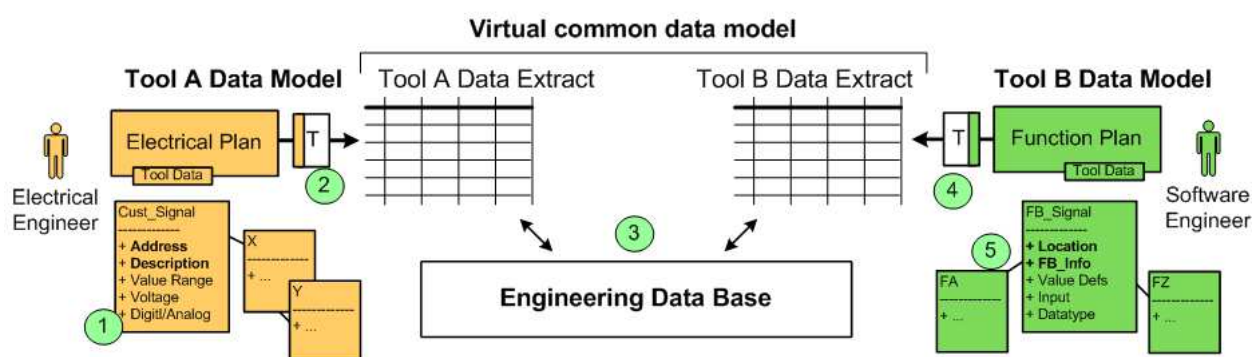

Fig. 5. Schematic overview on the virtual common data model (Biffl *et al.*, 2011).

Basically the VCDM aims at bridging this gap between heterogeneous sources. Figure 5 illustrates the VCDM and the relationship of two different tools from two different disciplines by example. Electrical engineers use tools for designing an electrical plan using defined tool data and attributes located within the (isolated) tool domain (1). On the other hand side, a software engineer (5) use specific tools for designing function plans, a common representation for the development of control applications. Both experts have to agree on a common language (the VCDM) to exchange data efficiently. In the AS domain, e.g., power plants, we observed signals as common concepts between different domains (Winkler *et al.*, 2011). For instance, a signal is represented as a voltage level of an electrical device (by the electrical engineer) and as a software variable (by the software engineer). Additional common information, e.g., hardware addresses and signal description, is used by both disciplines. The agreement of the VCDM results in a mapping table for translation purposes. Note that a transformation is required to map the VCDM to the individual tool data models (see (2) for the electrical plan transformer and (4) for the software model transformer). Finally, signal lists are passed from individual tools via transformer to the engineering data base (EDB) (3) holding the common data based on the VCDM. Therefore, signals and related

---

[3] Comos PT: http://www.automation.siemens.com
[4] EPlan Engineering Center: http://www.eplan.de/

common signal-specific information are used as foundation for efficient data exchange in AS development projects at our industry partner.

Note that this concept (a) enables interaction between related disciplines (and tools) via the VCDM and the EDB and (b) represents the foundation for PM and QA in the overlapping areas in heterogeneous environments.

5. Pilot application for Quality Assurance support in ASE Projects

This section provides a critical use case, i.e., signal change management (Winkler *et al.*, 2011), and highlights the results of a prototype implementation at our industry partner, a large hydro power plant systems integrator.

5.1 Use case: Signal change/deletion management with warning

The analysis of typical projects at our industry partner showed that an overall number of around 40,000 signal engineering objects are spread across several disciplines in a distributed and heterogeneous environment. Up to now manual synchronization processes were executed infrequently; as a consequence continuous PM and comprehensive QA become difficult and challenging.

Therefore, there is a strong need for synchronizing individual (discipline specific) signal lists more frequently to enable (a) efficient PM and (b) efficient QA across these disciplines. The VCDM enables tool-supported synchronization, efficient interaction, and data exchange between these disciplines. We identified signal change and signal deletion management as critical tasks within an engineering project in the AS domain and designed a basic workflow for signal handling (Moser *et al.*, 2010a). Figure 6a illustrates the process approach for change management and synchronizing of signal lists implemented at our industry partner. Signal deletion (i.e., an engineer removes a signal from the local tool data base) represents a critical issue because if the signal will be removed automatically, the entire signal and related automation aspects will be removed in the local data base of the participating engineer after updating the local tool data. Therefore, we implemented tool-supported notification regarding the removed signal to enable transparency of the deletion process. Figure 6b presents the extended change management process for handling deleted signals.

In detail, both processes are implemented as follows:

- *Signal Change Management (Figure 6a).* An electrical engineer modifies an existing electrical plan (1), e.g., changing the sensor type from a digital to an analogue sensor type. The local and isolated tool data base holds the modified data. The second step includes the submission (check-in) of the changed signals via the transformer (2) to the Engineering Data Base (EDB). Based on the available information in the EDB and the newly submitted signal list, a difference analysis (3) is conducted automatically (including separation of new and modified signals). The second engineer (in this example the software engineer) applies a check-out process (4) and synchronizes with his own local tool data base. If conducted frequently this concept enables a consistent data base available for all participating engineers.

- *Signal Deletion Handling with Warning (Figure 6b)*. Locally removed signals are critical for collaboration in heterogeneous environments if signal deletion is propagated across the ASB without appropriate notification of related engineers. Note that the removed signal will disappear in the local data base after a check-out by the corresponding engineer. Note that a signal is considered to be 'removed' if the signal is stored in the EDB but the signal is missing in the new/modified signal list during the check-in process. To overcome this issue, we extended the change management process (Figure 6a) by adding tool-supported notification (Figure 6b). Similar to the signal change handling process, an electrical engineer conducts a check-in process (1) and passes the signal list via Transformer (2) and a difference analysis step (3) to the EDB. The removed signal is identified[5] (4) and results in an engineering ticket (5) including related information and contact information, which are passed to related engineers who are affected by the change/removed signals. Note that information about the involved engineers is stored in a project configuration. While handling the personalized notification the related engineers can respond to the engineering ticket and check-out (synchronize) the modifications if necessary.
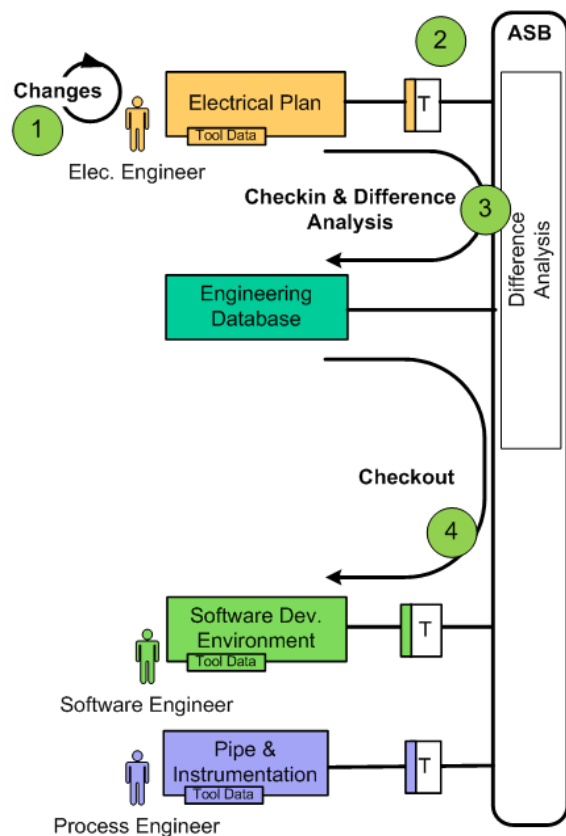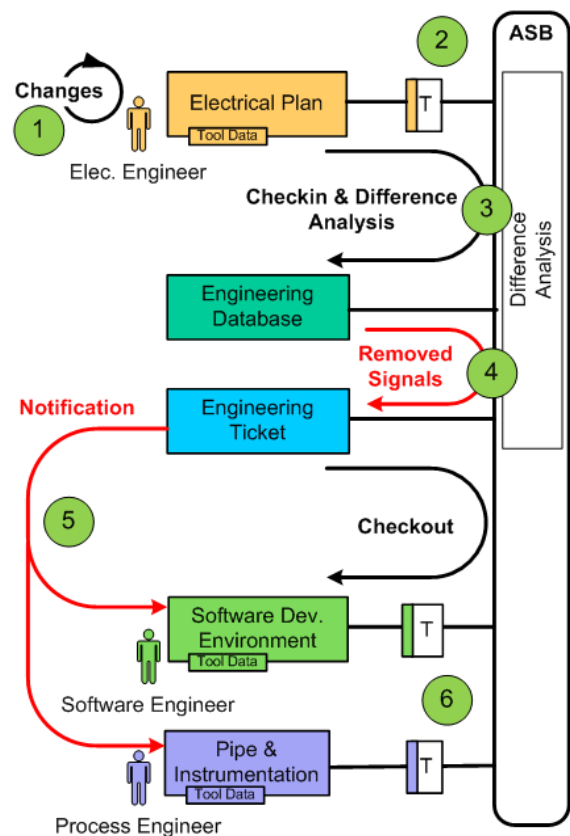

Fig. 6. Concept: Signal change (6a) and signal deletion with warning (6b).

---

[5] Note that the current scope of the check-in process, e.g., one or more engineering objects, is a mandatory pre-condition for assessing whether a signal has been removed or not.

Based on the VCDM and the signal change/signal deletion management process synchronization and data exchange between various disciplines and data models can be executed with tool support. Nevertheless, it remains open, how this concept can support QA across disciplines.

5.2 Quality Assurance in Automation Systems Engineering projects

In common ASE projects, QA activities, e.g., reviewing signal lists and/or following signal paths across several engineering documents (e.g., electrical plan, P&ID, software models) are conducted manually by experts. Because of a high number of signals (up to 40,000 signals in a common hydro power plant), these tasks are time consuming and include limitations regarding completeness and product quality. Therefore, experts use some supporting tools, e.g., macro solutions to support difference analyses. Nevertheless, these temporary solutions are created and used by individual engineers as small supporting tools. Because of these limitations it is hard to maintain and/or reuse these tools. For instance, changed environments (e.g., different projects or changed project settings) will typically require modification of the supporting tools. In addition knowledge transfer to other engineers, who are not familiar with these tools, becomes difficult.

Focused inspection. A major goal of the ASB concept is to enable experts/engineers focusing on relevant and critical signals during a check-in process. Therefore, an ASB solution should be able to provide only relevant changes to the experts (for discussion and conflict solving) to decrease synchronization effort significantly. Figure 7 illustrates the focus of the QA in context of synchronizing relevant signals across disciplines.
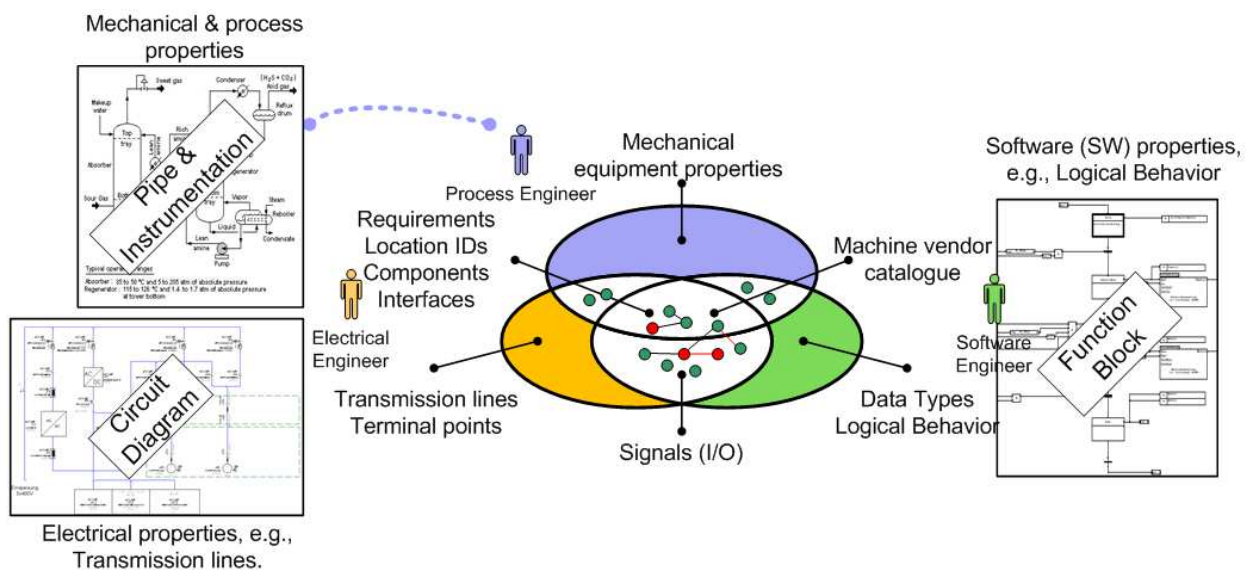

Fig. 7. Defect detection across disciplines (Biffl *et al*, 2011).

QA of signal lists includes two aspects: (a) local QA activities conducted by individual disciplines and tools limited to their application domain (not considered by the ASB approach) and (b) QA across disciplines in the overlapping areas, where experts have to synchronize signals and collaborate. Figure 7 illustrates the main aspects: the green marked

dots represent unchanged and agreed signals; the red marked dots represent changes/conflicts/removed signals which can result in major issues in related disciplines. Expert discussions have to focus on these critical changes to identify missing, wrong, or inconsistent elements (signals) or relationships. In addition the difference analysis highlights conflicts coming from changes in more than one discipline.

Figure 8 presents the results of the difference analysis after a check-in conducted by an electrical engineer. Changes are highlighted by providing the old and the modified value. Experts can use this difference check for synchronizing changes across disciplines and either accept or reject the change. Note that additional lists with focus on newly introduced signals, unchanged signals, and removed signals are presented to focus on a defined set of changes. In addition we introduced a list of 'invalid' signals to present, whether the new signal list does not confirm to given guidelines regarding data formats and structure.

Fig. 8. Screenshot: Highlighted changes at a check-in sequence.

The main advantage is that experts can focus on the changes and conduct a 'focused' inspection from different perspectives, either from the point of view of an electrical engineer, process engineer, or software engineer (as illustrated in Figure 1). Because of this tool-supported approach for data exchange, the synchronization process will be conducted more frequently and can enable the construction of 'no-surprise' systems products.

Note that future work will include a more detailed presentation of signals, including checks for consistency to enable advanced QA activities (a) within one signal and (b) across a set of signals. For instance, a check for consistency within one signal can identify missing information, e.g., a missing hardware address or incompatibility of two or more information sets; a check for consistency can find duplicate addresses or inconsistencies between similar signals within one component. Nevertheless, the difference analysis and presentation of defects will support experts in solving conflicts more effective (completeness) and efficient (within a shorter time interval).

Integration Testing. A second critical QA approach focuses on an overall consideration of signals and their connections across disciplines – comparable to (software) integration tests -

which could hardly be found by (focused) inspection. For instance, a sensor is connected to a switchboard *(System Interface)* and connected to a Software Variable *(Software Interface)* used in a control application *(Software 'Behaviour')*.
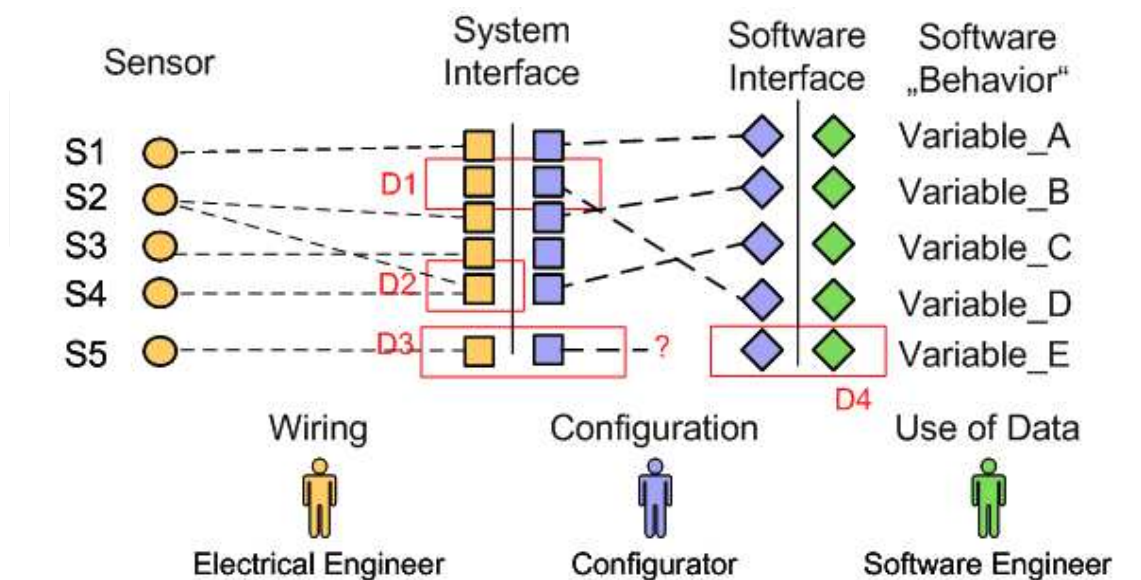

Fig. 9. Automation-supported testing across disciplines (Biffl *et al*, 2011)

Figure 9 illustrates the related disciplines and interfaces and their connection points (Biffl *et al.*, 2011). Sample candidate risks and quality issues are: (D1) Sensor data used by a software variable but without connection to a hardware sensor; (D2) multiple sensors are connected to one software variable; (D3) correctly wired sensor but no link to a software variable; (D4) Software variable not connected to a sensor data and sensors.

These types of defects across disciplines could not be found easily during signal check-in. In addition a manual inspection whether a sensor is wired and connected to the desired variable is a complex and time-consuming activity. Therefore, tracing of signals across disciplines is a valuable approach for supporting experts in their field in better identifying defects. Based on the VCDM (where all signals are available) defined queries focus on a certain class of defects, e.g., whether all sensors are wired to software variables, and deliver a result set of signals across disciplines where this assumption is not given. Experts can focus on the designated signal traces to check whether the traces (and the connection between sensor, switchboard, and software variable) are correct. See (Biffl *et al.*, 2011) for a detailed description and prototype application of the 'End-to-End-Test'.

5.3 Project management support

Observing and monitoring the current project state is a key requirement in ASE projects. Because of the heterogeneity of disciplines and the involvement of various stakeholders in deriving the current project state becomes challenging. Our observations at industry partners showed that distributed project data have to be collected and analyzed manually including a high effort. Therefore, the project state is captured infrequently and on request.

Based on a VCDM the ASB approach enables tool-supported data collection, analysis and visualization by providing project data to relevant stakeholder, e.g., to engineers or to the project manager. We developed an engineering cockpit (Moser *et al.*, 2011) to support PM and QA in ASE projects. Figure 10 presents a prototype of an engineering cockpit to observe changes and the impact of changes in an automation system engineering projects. Major components of this cockpit are (a) role specific views on engineering data and activities, (b) team awareness, and (c) status data regarding the project based on signals as common concepts. Figure 10 presents a snapshot of an engineering project at the industry partner. The data presentation section focuses on the project progress, i.e., the number of signals and the corresponding state of the signal, (e.g., in work, released, changed) per project phase and over time. Selecting defined data sets lead to a drill-down, i.e., a more detailed view on the engineering data within the selected scope. The example shows a selected set of components and the already implemented signals as wells as the expected number of signals for completing the components. Note that the engineering cockpit enables the presentation of data (signals) across disciplines and tool borders and enables a comprehensive view on the engineering project.

See (Moser *et al.*, 2011) for a more detailed description of the engineering cockpit prototype. Based on the prototype implementation of the engineering cockpit in an ongoing research project, we will also address additional information and activities with respect to provide a central starting point for PM and QA in AS development projects.
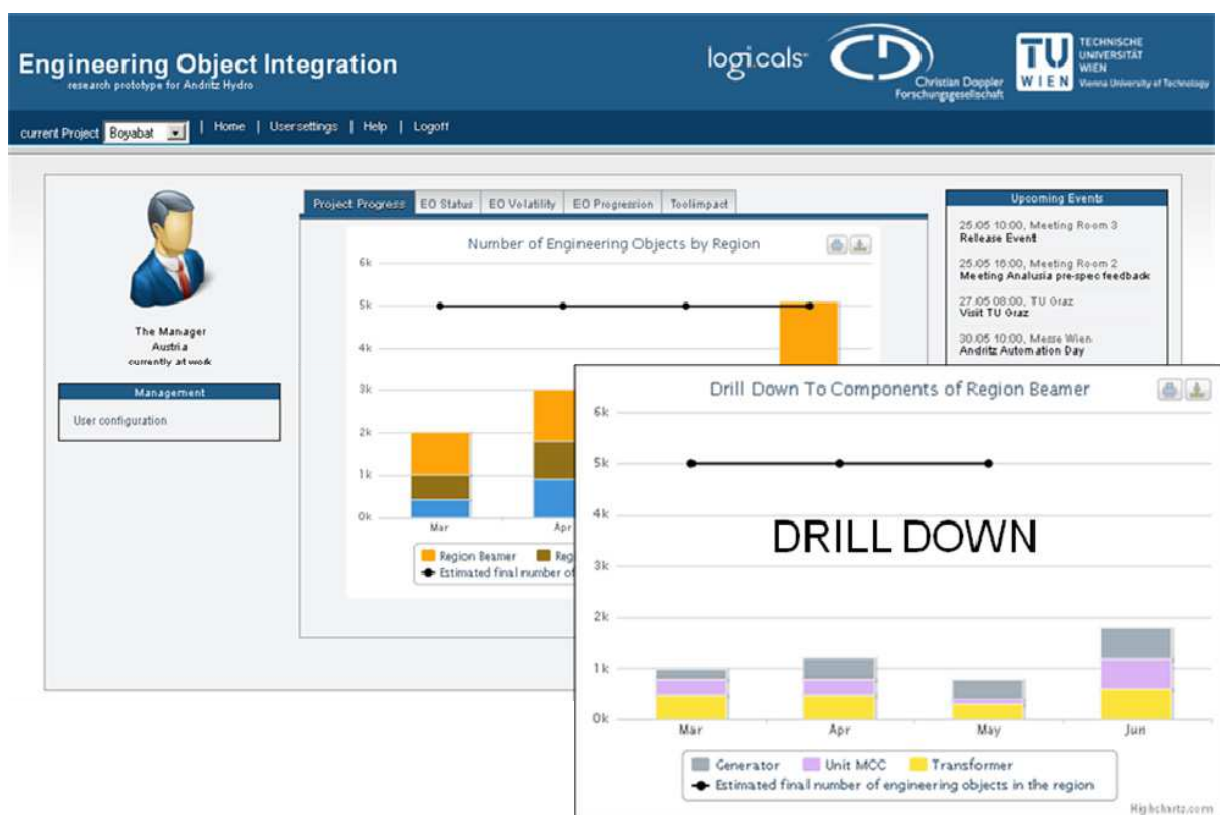

Fig. 10. Project observation with the engineering cockpit (Moser *et al.*, 2011).

6. Conclusion

This chapter presented a snapshot of an ongoing research project (CDL-Flex[6]) summarizing QA aspects for automation system development projects.

6.1 Summary and lessons learned

The development of large-scale AS, e.g., manufacturing systems and power plants, involve various disciplines, e.g., electrical, mechanical, and software engineers, who have to collaborate to construct high quality systems. Collaboration between disciplines and tool borders is a success-critical issue in AS development projects. Typically, disciplines are isolated using individual tools (technical heterogeneity), data models (semantic heterogeneity), and adapted development processes (process heterogeneity). These different levels of heterogeneity make collaboration more difficult and more risky. In addition distributed and heterogeneous data models hinder effective and efficient QA and PM. Efficient synchronization, collaboration, and QA mechanisms are required to support systems development projects. Up to now synchronization and QA across disciplines requires domain experts (familiar in at least two related disciplines), who perform these overlapping tasks manually. These manual activities require a high effort and include defects, which can be avoided by tool-supported mechanisms.

This chapter introduced to the Automation Service Bus (ASB), a flexible middleware platform with focus on the technical integration of tools and the semantic integration of data models in heterogeneous engineering environments (Biffl *et al.*, 2009). Technical and semantic integration is the foundation for bridging the gap between disciplines in the AS project and enable effective and efficient PM (project observation and control) and QA across disciplines and tools borders.

Research Question 1 (RQ1). *Enabling efficient data exchange in heterogeneous ASE environments.* To overcome the semantic gap between heterogeneous data models and data we introduced the Virtual Common Data Model (VCDM). The VCDM aims at bridging the gap between heterogeneous data models by (a) identifying common concepts (e.g., signals) and (b) map isolated data sources to a common Engineering Data Base (EDB) as a central repository for exchanging data between various sources efficiently. The VCDM is the foundation for synchronizing data from various sources efficiently.

Based on our observation and discussions with our industry partner we developed the signal change / signal deletion handling process (Winkler *et al.*, 2011). This process approach – embedded as a central use case for AS development projects – enables systematic data exchange between various stakeholders (across a centralized EDB) based on common concepts (signals), leaving additional discipline-specific data within the specialized domain. Therefore, this ASB approach is considered as a light-weight approach for handling heterogeneous engineering environments.

Research Question 2 (RQ2). *Support for QA across disciplines in heterogeneous environments.* Isolated disciplines and processes include individual QA activities with focus on defect detection within one (isolated) discipline. Nevertheless, QA across disciplines is still

---

[6] See CDL-Flex website http://cdl.ifs.tuwien.ac.at

challenging. Software inspection and testing (more specifically, integration testing) are promising candidates for cross-disciplinary QA activities. Based on the results of a difference analysis (part of the signal change/deletion handling process) domain experts can focus on selected and critical subsets of engineering objects (signals) for conflict resolution and defect detection (focused inspection). Focused inspection enables defect detection in the overlapping areas (where engineers from different disciplines have to collaborate) from different perspectives with respect to detecting defects. Nevertheless, a comprehensive view on the signal (from sensor to variable) is still challenging. Therefore we introduced the End-to-End test to focus on signal traces to identify different classes of defects, e.g., whether all sensors are connected to a switchboard and if all software variables are connected to data values for further operations.

Research Question 3 (RQ3). Support for project and quality managers in collecting and analyzing project data (from heterogeneous sources) more effective and efficient to enable continuous project monitoring and control. Isolated and heterogeneous data source also hinder efficient PM because – similar to synchronization and QA – data from various sources have to be captured and analyzed manually and on request. The VCDM as data source of common data from different sources is the foundation for a comprehensive view on engineering data across disciplines. We introduced the Engineering Cockpit (Moser *et al.*, 2011) providing the 'Window to engineering data' aiming at (a) providing stakeholder related data derived from the engineering project data bases and (b) enabling control of project steps based on the analysis results.

Nevertheless the presented prototype implementations are a starting point for supporting AS development projects in an ongoing research project.

6.2 Future work

Based on the results from the research project and after discussing them with our industry partner we identified a set of future work aspects related to QA and PM.

- *Common concepts.* Signals have been identified as common concepts in the domain of hydro power plant systems integration. Because of generalization opportunities of the ASB approach, we plan to investigate additional application domains to include a wider range of common concepts.
- *Process observation and control.* The signal change/signal deletion handling processes have been considered as very critical processes in the application domain. Nevertheless, future work will include the consideration of more and more complex processes. Next steps will also include tool-supported definition, implementation and evaluation of processes and process results (Sunindyo *et al.*, 2010).
- *Focused inspection.* This work highlighted the applicability of (perspective-based) inspection for inspecting the overlapping areas of related disciplines. Nevertheless, a more detailed inspection approach has to be developed and evaluated to enable the applicability in industry context.
- *Consistency.* Future work also includes checks for consistency during the check-in processes (a) within one signal and (b) across signals based on pre-defined rules for signal data sets. The goal is to identify deviations early in the development process, i.e., during the specification and design phase of the engineering project.

- *End-to-End-Test*. First results of a prototype evaluation showed benefits of the 'integration test' across disciplines. Nevertheless open issues will focus on different defect types and how to address them properly. In addition, empirical studies are necessary to evaluate the effectiveness and efficiency of end-to-end test approaches.

Finally, empirical studies in industry context are necessary to evaluate the presented concepts (a) in academic and (b) industry environments for the validation of the ASB approach and related industry applications.

7. Acknowledgements

8. References

(Baker et al, 2008) Baker, P., Zhen, R. D., Gabrowksi, J., & Oystein, H.. *Model-Driven Testing: Using the UML Testing Profile*, Springer, 2008.

(Basili, 1997) Basili, V. *Evolving and Packaging Reading Techniques*, Journal of Software and Systems, 38(1), pp3-12, July 1997.

(Beck et al, 2004) Beck, K., Andres, C. *Extreme Programming Explained - Embrace Change*, Addison-Wesley, 2004.

(Biffl, 2001) Biffl S. *Software Inspection Techniques to support Project and Quality Management*, Shaker, 2001.

(Biffl et al, 2003) Biffl S., Halling M.: *Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams*, IEEE Transactions of Software Engineering 29(5), pp 385-397, 2003.

(Biffl et al., 2009a) Biffl, S, Schatten A. & Zoitl A. (2009). *Integration of Heterogeneous Engineering Environments for the Automation Systems Lifecycle* Proceedings of the IEEE Industrial Informatics (INDIN) Conference, Cardiff, UK, June 2009.

(Biffl et al, 2009b) Biffl, S., Sunindyo, W.D., and Moser, T. *Bridging Semantic Gaps Between Stakeholders in the Production Automation Domain with Ontology Areas*, Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE), Boston, USA, 2009.

(Biffl et al., 2011) Biffl, S., Moser, T., Winkler, D.: Risk Assessment in Multi-Disciplinary (Software+) Engineering Projects. International Journal of Software Engineering and Knowledge Engineering (IJSEKE), Special Session on Risk Assessment, Volume 21(2), March 2011.

(Chappell et al., 2004) Chappell D.A.: *Enterprise Service Bus: Theory in Practice*, O'Reilly Media, ISBN: 978-0596006754, 2004.

(Gamp, 2008) GAMP 5. *Good Automated Manufacturing Practice*. International Society for Pharmaceutical Engineering (ISPE), 2008.

(Hametner et al, 2011) Hametner, R., Kormann, B., Vogel-Heuser, B., Winkler D., Zoitl, A., *Test Case Generation Approach for Industrial Automation Systems*, Proceedings of the 5th International Conference on Automation, Robotics and Applications (ICARA), Wellington, New Zealand, December 2011.

(Kaner et al, 1999) Kaner, C., Falk, J., Hguyen, H.Q.: *Testing Computer Software, Wiley, 1999,*

(Kollanus et al, 2007) Kollanus, S., & Koskinen, J., *Survey of Software Inspection Research: 1991-2005*, Working Paper WP 40, University of Jyväskylä, 2007.

(Laitenberger et al, 2000) Laitenberger, O., DeBaud J.M. *An encompassing life cycle centric survey of software inspection*, Journal of Software and Systems (JSS), 50(1), pp5-31, 2000.

(Meyers et al., 2004) Meyers, G.J., Sandler, C., Badgett, T. & Thomas, T.M. *The Art of Software Testing*, 2nd edition, John Wiley & Sons, 2004.

(Moser et al., 2010a) Moser, T., Waltersdorfer, F., Zoitl, A. & Biffl. *Version Management and Conflict Detection Across Heterogeneous Engineering Data Models*. Proceedings of the 8th IEEE International Conference on Industrial Informatics (INDIN), Osaka, Japan, July, 2010.

(Moser et al., 2010b) Moser, T., Biffl, S., Sunindyo, W.D & Winkler, D.: *Integrating Production Automation Expert Knowledge across Engineering Stakeholder Domains.* Proceedings of the 4th Interational Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Krakow, Poland, February 2010.

(Moser et al., 2011) Moser, T., Mordinyi, R., Winkler, D., & Biffl, S. *Engineering Project Management using the Engineering Cockpit: A collaboration platform for project managers and engineers*. Proceedings of the 9th International Conference on Industrial Informatics (INDIN), Lisbon, Portugal, July 2011.

(Sage et al., 2009) Sage, A.P., Rouse, W.B. *Handbook of Systems Engineering and Management*, 2nd Edition; Wiley, 2009.

(Schäfer et al, 2007) Schäfer, W. & Wehrheim, H. *The Challenges of Building Advanced Mechatronic Systems*, Proceedings of the Intenational Conference on Software Engineering (ICSE), Future of Software Engineering, Washington DC, USA, 2007.

(Schatten et al, 2010) Schatten, A., Biffl, S., Demolsky, M., Gostischa-Franta, E., Östreicher, T., & Winkler D., *Best-Practice Software-Engineering : Eine praxisorientierte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen*, Spektrum Akademischer Verlag, 2010.

(Schulmeyer, 2008) Schulmeyer, G.G. *Handbook of Software Quality Assurance.* 4th edition, Artech House Inc, 2008.

(Schwaber, 2004) Schwaber K.: *Agile Project Management with Scrum*, Microsoft Press, ISBN: 978-0735619937, 2004.

(Sommerville, 2007) Sommerville, I. *Software Engineering*, 8th Edition, Addison Wesley, 2007.

(Sunindyo et al., 2010) Sunindyo W.D., Moser T., Winkler D., & Biffl S.: *Foundations for Event-Based Process Analysis in Heterogeneous Software Engineering Environments*, Proceedings of the 36th Euromicro Conference, Software Engineering and Advanced Applications (SEAA), Lille, France, September 2010.

(Winkler, 2008) Winkler. D. *Improvement of Defect Detection with Software Inspection Variants: A Large-Scale Empirical Study on Reading Techniques and Experience*, VDM, May 2008.

(Winkler et al, 2009) Winkler, D., Biffl, S., & Östreicher, T. *Test-Driven Automation – Adopting Test-First Development to Improve Automation Systems Engineering Processes.* Proceedings of the 16th European Software & Systems Process Improvement and Innovation (EuroSPI) Conference, Madrid, Spain, September 2009.

(Winkler et al., 2011) Winkler, D., Moser, T., Mordinyi, R., Sunindyo, W.D. & Biffl, S. *Engineering Object Change Management Process Observation in Distributed Automation Systems Projects*, Proceedings of the 18th European Systems & Software Process Improvement and Innovation Conference (EuroSPI), Roskilde, Denmark, June 2011.
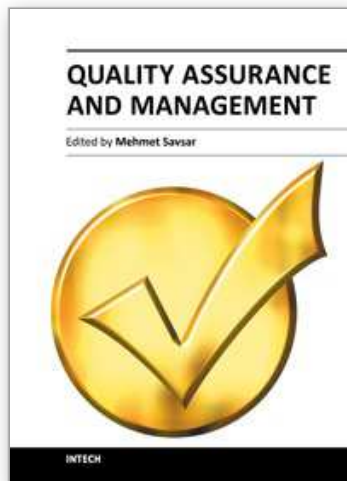
Quality Assurance and Management

Edited by Prof. Mehmet Savsar

The purpose of this book is to present new concepts, state-of-the-art techniques and advances in quality related research. Novel ideas and current developments in the field of quality assurance and related topics are presented in different chapters, which are organized according to application areas. Initial chapters present basic ideas and historical perspectives on quality, while subsequent chapters present quality assurance applications in education, healthcare, medicine, software development, service industry, and other technical areas. This book is a valuable contribution to the literature in the field of quality assurance and quality management. The primary target audience for the book includes students, researchers, quality engineers, production and process managers, and professionals who are interested in quality assurance and related areas.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Dietmar Winkler and Stefan Biffl (2012). Improving Quality Assurance in Automation Systems Development Projects, Quality Assurance and Management, Prof. Mehmet Savsar (Ed.), ISBN: 978-953-51-0378-3, InTech, Available from: http://www.intechopen.com/books/quality-assurance-and-management/improving-quality-assurance-in-automation-systems-development-projects