Quality Assurance Aspects in Automation Systems Engineering Projects

Challenges and Lessons Learned from Industry Prototype Applications

Dietmar Winkler
Christian Doppler Laboratory “Software Engineering Integration for Flexible Automation Systems”
Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria
{*Dietmar.Winkler*}@tuwien.ac.at

Abstract— **Automation systems, e.g., hydro power plants and industrial automation systems include heterogeneous engineering disciplines, e.g., mechanical, electrical, process, and software engineering, and raise additional challenges for quality assurance activities, e.g., identifying defects in change management processes where different disciplines are involved. Our observations in industry shows various tools and data models with limitations in collaboration and data exchange that hinder effective and efficient quality assurance across disciplines where experts have to collaborate and exchange data. The Automation Service Bus (ASB) offers a middleware platform with focus on (a) integrating tools from difference disciplines and (b) bridging the gap between data models coming from different sources. Based on technical and semantic integration quality assurance across disciplines and domain borders becomes possible. In context of quality assurance and project management this paper presents selected research challenges and shows results of implemented application prototypes of an ongoing project in industry context.**

Keywords-Quality Assurance, Automation Systems Development Processes, Defect Detection, Integration Testing, Project Management

I. INTRODUCTION

Large-scale automation systems engineering projects, like hydro power plants, steel mills and large manufacturing systems include heterogeneous disciplines during design and development, construction and test, commissioning, and operation [4][18]. During the product life-cycle [4] various stakeholders from heterogeneous disciplines have to collaborate and exchange data across domain-specific tools applying domain specific data models [12]. For instance, electrical engineers focus on circuit diagrams and wiring, process engineers provide process workflows for the plant, and software engineers develop control system to observe and control the automation system under construction. Our observations showed that process engineers and domain experts focus on bridging the gap between heterogeneous environments manually to support development, commissioning (i.e., system launch at the customers site) and operation (i.e., operation and maintenance at the plant site). Note that these manual activities are error prone (e.g., caused by media breaks) and require a high effort provided by experts when linking various data models and sources manually. Because of limited interaction and collaboration support provided by individual tools (i.e., technical heterogeneity) and various applied data models (semantic heterogeneity) collaboration becomes more difficult and results in a high manual effort for synchronizing data models during the life-cycle phases [25]. Efficient automation-supported data exchange across disciplines and domain borders is an important key challenge to increase quality and decrease defects, effort and cost.

Quality assurance is a comprehensive activity across all project phases to enable the construction of high quality products. In addition results of quality assurance activities help project and quality managers in identifying defects early [25], assessing individual project states frequently, and – based on individual project states – enables product quality evaluation with respect to monitoring and controlling the engineering project [13]. Nevertheless, loosely coupled tools and data models hinder efficient quality assurance and require high manual effort for data collection, aggregation, and analysis. As a consequence, quality assurance tasks are executed less frequently and can raise quality issues. Figure 1 highlights the need for automation supported data exchange for change management and quality assurance in heterogeneous engineering environments.

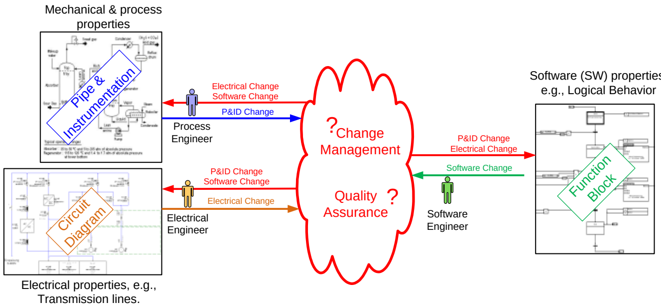


Figure 1: Changes and Defects - Challenges in heterogeneous Engineering Environments.

Change Management (CM) is a success critical issue in software and systems development projects during development and runtime [22]. CM addresses the need to respond to changes coming from individual disciplines by other related disciplines. For instance a modified sensor type (from the electrical discipline) can require modified control flows (change within the process discipline) and modified data visualization approaches (software engineering discipline), e.g., modified value ranges of the changed sensor. Changes have to be passed to all involved disciplines within a short time interval to enable addressing these changes within assigned individual models by related engineers. Thus, change management is a crucial use case for engineering projects and quality assurance. Quality assurance focuses on identifying defects in general and in change management processes specifically. Thus, we observed three main challenges:

- Quality Assurance support for early defect detection in engineering processes across domain borders [22].
- Testing signal chains from hardware sensor to software variable to address defects across disciplines in later phases of systems development [25].
- Comprehensive view on the project progress including signal change observations and project course assessments [13].

The remainder of this paper is structured as follows: Section 2 introduces to common automation systems development projects and project dashboards with focus on quality assurance and project management. Section 3 presents the research issues. Section 4 presents candidate solution approaches implemented at our industry partners. Finally, Section 5 summarizes, presents lessons learned and highlights future work.

II. RELATED WORK

This section summarizes related work on (a) Automation Systems Engineering (ASE) processes, (b) change management in ASE projects, and (c) project management with project cockpits/dashboards.

A. *Automation Systems Engineering Processes*

Lessons learned from business IT software development show a variety of different software processes as framework for project management support [20]. Traditional software development projects typically follow a sequential or V-model approach (e.g., Waterfall Model and V-Modell XT[1]). A lack of flexibility with respect to changes caused by customer requests (e.g., frequent changing requirements) and/or defects lead to the application of more flexible and agile process approaches, e.g., Scrum [19] or eXtreme Programming [2].

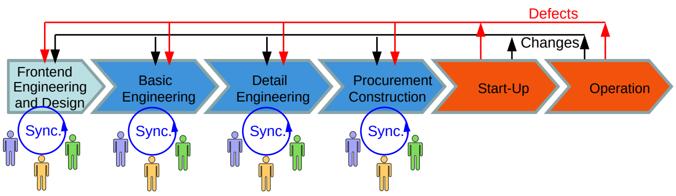


Figure 2: Sequential Systems Development Process in Automation Systems Engineering Projects.

In automation systems development projects various discipline apply heterogeneous tools and data models. Changes can occur frequent (in different disciplines) and have to be synchronized to enable a consistent engineering project. Our observations in industry projects showed a traditional engineering process approach following a (sequential) product life-cycle process model. Figure 2 presents the basic process approach at our industry partner – a large scale systems development and integration organization – including six basic steps. The individual process phases are organized strictly sequentially. A major challenge is to synchronize various disciplines within individual phases as often as possible to (a) identify defects early, (b) enable efficient handling of changes, (c) enable a consistent engineering repository involving artifacts from all related disciplines, and (d) enable a comprehensive view on the overall project progress. Today this synchronization is executed mainly manually or in a less automated way [21].

B. *Change Management Processes in ASE Projects*

Figure 3 presents the challenges of Change Management in heterogeneous environments [6][22]. Electrical engineers perform changes in electric circuit diagrams including electric planning tools and data models (1). Typically individual tools are able to handle changes and quality assurance tasks individually. Nevertheless, change management and quality assurance across systems borders and tools are not sufficiently solved. Changes have to be passed to affected engineers/disciplines to enable efficient synchronization and defect detection (2). In addition notification of changes (3) can support engineers in better understanding the individual change and its consequences, and – in case of conflicts – enable an efficient problem-solving between involved engineers.

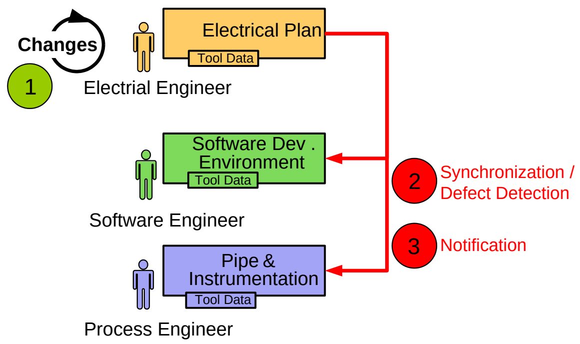


Figure 3: Challenges in Change Management in Heterogeneous Engineering Environments.

In industry practice we observed less frequent synchronization (because of a high effort by experts who a familiar with at least two disciplines), informal discussions during problem/conflict solving processes [11][22] and quality assurance tasks. Based on interviews with engineers with our industry partners we learned that a set of defects are identified during the start-up phase (commissioning) by introducing system components stepwise (including on-site quality assurance tasks). In addition we learned that common defects, e.g., missing links between (hardware) sensors and (software) variables or defective component descriptions could have been found earlier by applying a systematic and frequent synchronization

[1] See *http://www.v-modell-xt.de* for a detailed description of the V-Modell for software and systems development projects.

of different disciplines, systematic quality assurance tasks (e.g., by supporting focused reviews), and testing approaches [25] on integration test level.

C. *Comprehensive Project Management and Control*

Project and quality management, i.e., project observation and control, are key activities of project and quality managers across individual phases of systems development to capture the current project state and introduce counter measures in case of derivations in terms of resource problems and quality issues [20]. Software cockpits [16] and software project control centers [14] have been developed to enable project and quality management in business IT software projects. Main objectives of *engineering cockpits* include (a) the collection of engineering data, (b) the aggregation and analysis of the collected data, and (c) the presentation of the analysis results according to the need of project and quality managers [15][16]. Nevertheless, heterogeneity of engineering disciplines, tools, and data models include an additional challenge in establishing a comprehensive view on an automation engineering project.

Because of less transparent changes in automation system engineering projects (due to less frequent synchronization between disciplines and limited data exchange capabilities), changes and the impact of changes remain unclear and depend on the individual estimation of experts, i.e., project and quality managers. A key question of project managers focuses on the number of changes in an engineering project per time interval, project phase and the impact of/on disciplines raising changes and/or defects.

III. Research Issues

Based on observations and discussions with our industry partner we identified a set of research questions:

RQ1: How can we enable data exchange between heterogeneous data sources? Efficient data exchange between tools and various data sources is the foundation for efficient change management, quality assurance, and project management.

RQ2: How can we enable efficient quality assurance across disciplines? Individual quality assurance activities are located within the individual discipline and tool. Today, these quality assurance tasks are executed by experts, who are familiar with at least two disciplines and requires a high effort. Automation supported QA across disciplines can increase QA efficiency and product quality.

RQ3: How can we enable a comprehensive view on the engineering project? Traditional project management approaches focuses on an isolated discipline within homogenous environments. We observed limitations in heterogeneous environments involving loosely coupled tools and data models in various disciplines. One key question is how we can develop a project cockpit to address cross-disciplinary engineering projects.

IV. Use Case and Industry Pilot Application

This section presents the key use case "Change Management" [22] and the pilot application at our industry partner to enable data exchange across disciplines as foundation for (a) efficient quality assurance (i.e., focused reviews and integration testing) and (b) project observation and control with the Engineering Cockpit [13].

A. *Automation Service Bus and Common Concepts*

The Automation Service Bus (ASB) has been developed as a middleware platform [4] to bridge the technical gap between various tools and the semantic gap between individual data models [5]. Technical and semantic integration of tools and data models are the foundation for comprehensive quality assurance and project monitoring and control.

Figure 4 presents the common concepts [6] represented as the virtual common data model (VCDM), aiming at bridging the gap between various and heterogeneous sources [12]. The main idea is focusing on common data sets that enable the collaboration of related disciplines. For instance, electrical engineers apply dedicated tools for constructing electrical plans (1); on the other hand side software engineers apply tools (5) for modeling software functionality, e.g., function plans (common graphical language for implementing control applications). Common concepts (3) located in a centralized data base – the engineering data base (EDB) – holds data which are relevant for both disciplines for synchronizing and collaboration purposes. Note that the common concepts are limited to a small number of data elements which are mandatory for synchronization. Other tool-specific data (not required for synchronization steps) are left within related individual tools. Data changes are passed and synchronized via transformers (T) to and from related disciplines in a bidirectional way (see (2) and (4)). Thus, this concept enables data exchange across different tools using different data models.

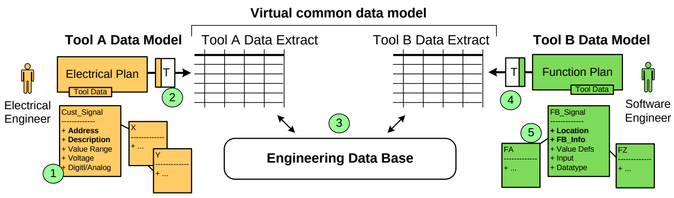


Figure 4: Common Concepts **[6]**.

Analyzing typical data models used in hydro power plants at our industry partner, we identified signals as underlying and common concepts linking various disciplines [6]. For instance, electrical engineers see signals as wires and measurable voltage levels, process engineers identify signals as input/output values of pipes and valves, and software engineers see signals as software variable used to control and visualize data within the control application. Thus, signals (as common concepts) represent a light-weight conceptual approach that can enable efficient collaboration of various disciplines and enable efficient data exchange and synchronization between heterogeneous engineering environments.

B. *Synchroniation between Various Disciplines*

Figure 2 presented a basic engineering process for constructing large-scale automation systems, i.e., hydro power plants, observed at our industry partner. Frequent synchronization of engineering artifacts (across tools and data models) can increase construction efficiency, increase quality, and reduce cost [22][25].

Figure 5 presents a basic synchronization step of individual disciplines (see Figure 5a), i.e., mechanical, electrical, and software engineering. Note that every discipline applies domain specific quality assurance activities [17] with limitations on overlapping quality assurance tasks, i.e., defect detection across disciplines, tools, and data models. Thus, a key challenge is bridging the gap between heterogeneous environments. The ASB approach enables an automation supported synchronization step, i.e., based technical integration of tools and semantic integration of data models (see Figure 5b) [25]. Note that frequent synchronization will lead to a more consistent engineering data base (EDB) applicable for all related disciplines.

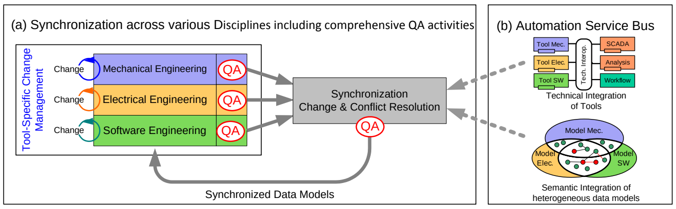


Figure 5: Synchronization between Disciplines [25].

This synchronization step addresses all systems engineering phases, typically separated by milestones and signal states. Note that signals are the common concepts used for synchronization, change management, and signal change handling.

C. Object Change Management

Change management of common concepts, i.e., signals in hydro power plants or – more general, engineering objects in other domains, is the most critical use case in engineering projects. Figure 6 presents the implemented change management workflow at our industry partner based on the common concept of change handling (presented in Figure 3) involving electrical, mechanical and process, and software engineers [22].

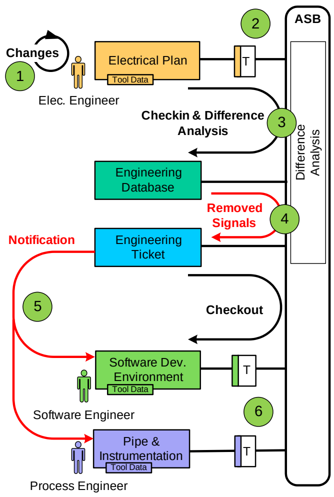


Figure 6: Object Change Management [22].

Basically, an electrical engineering executes a change in a defined project phase, e.g., during the *basic engineering* phase (see Section II-A for an overview of observed project phases at our industry partner). Note that the electrical engineer applies an electrical planning tool and modifies the underlying electrical data model, e.g., circuit diagram (1), by changing and/or removing signals (or engineering objects). The engineering data base (EDB) holds the current project data from previous phases and/or check-ins conducted by engineers coming from different disciplines. The modified signal information is passed via a defined and tool-specific transformer (T) to the ASB (2) and is compared with the current signal information located at the EDB (3). Differences are presented to an engineer, typically a systems integrator and/or project manager who are responsible for a consistent data base. Changes can be accepted or rejected by the systems integrator via web frontend.

Figure 7 presents a sample snapshot of the pilot application: three changes have been identified (e.g., highlighted in Figure 7) and can be accepted or rejected by the systems integrator. After a successful check-in of the modified data in the EDB the database holds the updated (consistent) signal information for further usage including the changes conducted (and checked-in) by the electrical engineer. Related engineers from other disciplines (e.g., software and mechanical/process engineers) can check-out the latest signal versions and update their local data models, assigned to their individual tools (6). Note that this check-out requires a corresponding and tool-specific transformer. Following this basic workflow (step 1, 2, 3, and 6), it is easy to handle new and modified signal information. Nevertheless, handling removed signals is still challenging and needs to be solved.

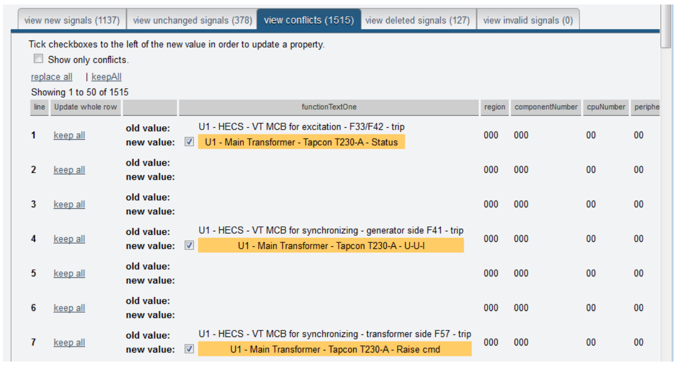


Figure 7: Prototype Implementation of Object Change Management with three changes.

Removed signals (4) are specific type of changes and require a more specific handling within an engineering project. Assuming that an electrical engineer removes defined signals, e.g., obsolete sensor signals, by applying the presented change management workflow (step 1, 2, 3, and 6), the workflow will force related tools to remove the signal in the corresponding data models as well. For instance, the signal will be removed in the function block diagram of the software engineer without any warning – the signal will disappear and cannot be used for further process steps. A more critical result of signal removal is that a connection point might become disconnected (because of a missing sensor) caused by the workflow rather than by the software expert. To overcome these “unintentional surprises” (signals will be disappear in the related data models) removed signals will initiate an engineering ticket (5), which (a) notifies related engineers about the signals which are going to be removed and (b) enable the response to these actions before checking out the modified data models. Major benefit of applying engineering tickets is that engineers can discuss and response to changes and removed signals early, i.e. during every check-in process. Note that removed signals can be rejected by the system integrator as well; an appropriate notification of the decision must also be passed to all related engineers.

The pilot application at our industry partners showed that selective notification of engineers based on removed signal information (and as a next step notification of all critical changes) turned out to be most valuable for project and quality managers to enable early discussion on selected changes and increased product quality (early identification, prediction and prevention of candidate defects caused by changes). Thus the implemented change management process is a promising starting point for applying quality assurance tasks within an engineering project in the automation systems domain.

D. Quality Assurance Support with Focussed Reviews

Assuming that a hydro power-plant includes a set of up to 40.000 signals, change management and quality assurance become challenging [25]. Manual synchronization of lists of signals (and engineering objects) and defect detection in case of changes are time consuming and error prone. Thus, the automation supported synchronization process (presented in Section IV-C) based on common concepts can reduce effort for synchronization and increase project and product quality.

In context of ASB application quality assurance in heterogeneous engineering environment focuses on two important aspects [25]: (a) *individual and isolated quality assurance tasks* conducted by experts in their assigned disciplines (e.g., mechanical and process, electrical, and software engineering) and supported by isolated tool solutions; (b) quality assurance on *overlapping areas*, i.e., common concepts, where different disciplines have to collaborate and synchronize data. Figure 8 presents the focus of quality assurance (also introduced in Figure 5) by example.

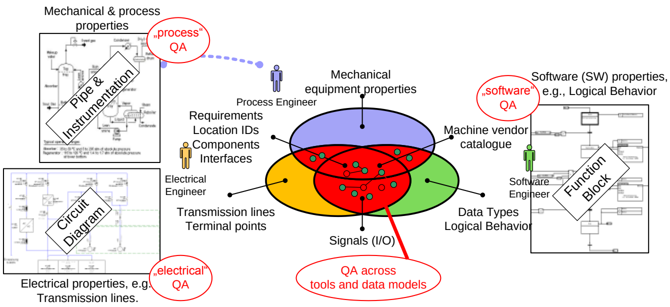


Figure 8: Focused Reviews on Common Concepts.

Highlighting the differences during signal check-in (see Figure 7 for an example) and selective notification of affected engineers by applying engineering tickets (see Figure 6 for the basic workflow) will enable focused discussions and problem-solving activities of engineers affected by the change and/or defect. Instead of discussing 40.000 signals (including an additional process step for identifying changes) the presentation of signal changes enables focusing discussions on real deviations. Note that expert estimations of our industry partners assume that typical projects include approximately 20% of signal changes along the project course. Focusing on real changes and defects will significantly reduce synchronization and defect detection effort and will increase project and product quality. Highlighted changes and the presentation of changes will reduce the number of signal comparisons and signal checks significantly and – as a consequence – will decrease effort and cost. With respect to quality assurance, reviewers and inspectors [9] can focus directly on the changes to identify defects and deviations.

Lessons learned from business IT software development and software inspection [8] as a well-established static quality assurance technique in software engineering [3][9] might lead to software inspection and reading technique [1] and reading technique variants [23] with focus on automation systems development projects in heterogeneous engineering environments. In addition, next steps can also include the analysis of engineering objects/signals and the impact of changes with respect to other related components which are not directly changed by an individual engineer, e.g., the impact on components which might have been affected by the change (but were not covered by the change request so far).

E. Integration Testing in Heterogeneous Environments

Software testing is well-known in common software engineering practice to identify defects based on executable software code [7] systematically. Based on software testing best-practices, several test levels apply, e.g., unit test, integration tests, and system and acceptance tests [10]. For instance integration tests from business IT Software development focus on testing the interfaces, integrated components and the data exchange between components. Concerning automation systems engineering projects, similar test levels are applicable [24]. Thus, we see the concept of *integration testing* (learned from business IT software development) as "comparable" to heterogeneous automation systems development, where different disciplines should be integrated and tested.

For instance, a *hardware sensor* is connected to a *systems interface* (e.g., a switchboard) and connected to a *software interface* (e.g., control application or visualization software) representing the *software behavior* [6][25]. Considering individual disciplines as components (and individually) tested via "unit tests" within the related isolated discipline, linking and testing these individual component and disciplines on architecture level might refer to integration tests including assigned and tested components and interfaces. Similar to integration tests in software engineering, integration tests in automation systems engineering projects will find different defects on architecture level which could hardly be identified by inspection and/or isolated QA activities within one discipline. Examples of candidate defects in the automation systems domain are presented in Figure 9:

- D1: Missing link between sensor (Sx) and system interface (I2). No value will be available at the software control application.

- D2: Correctly wired sensor (S2) to the system interface I3 but no link to a software variable, i.e., sensor value will not be analyzed and used for further applications.
- D3: Multiple connected sensors (S2 and S4) to I5 and V3.
- D4: Correctly wired sensor S5 to system interface I6 and wired connection at I6. Nevertheless the connection got lost on the way to the software variable, e.g., caused by a cable break.
- D5: Software Variable V5 not linked to system interface (Ix) and sensor (Sx).

Figure 9 shows related stakeholders/disciplines and interfaces [6] and illustrate highlighted defects across disciplines which cannot be identified easy during check-in processes and/or individual and local quality assurance activities.

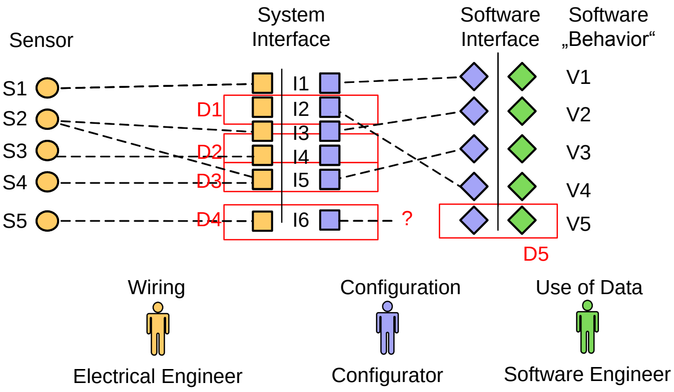


Figure 9: Testing Signal-Chains from Hardware Sensors to Software Variables.

Integration testing in automation systems development is typically embedded within the commissioning phase, where individual disciplines are linked to each other manually. Inspection and testing are applied stepwise during this ramp-up phase. Defects have to be located manually by consulting individual plans from heterogeneous sources. This manual activity, conducted by integration engineers and experts is time-consuming and require high effort and cost (e.g., tracing individual signals across the plant manually). The ASB approach enables engineer in testing plans from different sources based on the common concepts. This so-called “End-to-End” test enables engineers in testing the overall chain of signals across heterogeneous environment from hardware sensors to software variables [6]. In addition efficient navigation features have been implemented (a) to navigate to signals in various plans to see the implementation of the functional requirements (without consulting the paper work) and (b) to link runtime data during the commission phase within the affected engineering plan (e.g., presentation of runtime and/or simulation within a PDF document)[2]. Automation-supported integration testing of engineering plans from various sources and efficient navigation between affected plans can improve systems development processes and the commissioning phase significantly.

F. Project Observation and Control

Observing and controlling engineering projects are key activities of project and quality managers. Project observation in homogenous environments, e.g., business IT software projects, is supported by individual methods and tools. Software cockpits have been established as project control centers aiming at summarizing and presenting most relevant project information for project and quality managers, e.g., schedule, budget, project progress, and quality [14][16]. Note that the collection and analysis of project related data is typically limited to homogeneous engineering environments like in software engineering projects. Additional challenges arise in heterogeneous environments, e.g., in automation systems development projects, when linking data models and data from various sources.

Based on discussions with managers at our industry partner, the key challenge is to measure the impact of changes within an engineering project, i.e., identifying the number of changes per time interval and/or per project phase. Today expert estimations regarding the number of changes along the project course are the foundation for capturing change management metrics (e.g., 20%) because no detailed data regarding the real number of changes per project exists.

The ASB approach enables a more detailed view on changes by providing real data according to change management processes along the automation systems development process (see Section II-A for a detailed description of a common engineering process and Section IV-C for a change management process). Common concepts (see Section IV-A) enable efficient data exchange between various tools and data sources and are the foundation for process observation and control. Note that common concepts (i.e., signals) are used to observe and analyze the process across disciplines and tool borders and to control the next steps of the engineering process. Based on common concepts frequent synchronization between different disciplines becomes possible (see Section IV-B) and enable process analysis based on real and captured data.

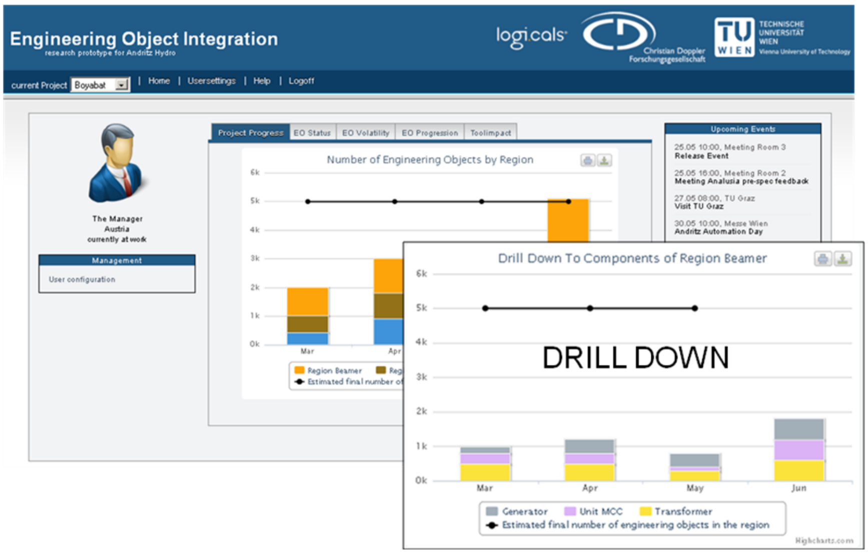
The analysis and presentation of change management data enable a more detailed view on the engineering process within an engineering cockpit. Figure 10 presents a snapshot of the implemented engineering cockpit, a “window to engineering data”, based on captured signal change management data from a real engineering project. The engineering cockpit from the perspective of managers (Figure 10a) enables an overview on the project state (visualized by the *project progress*) summarizing various groups of engineering objects (e.g., components) over time. Note that the visualization includes the current state of engineering objects (i.e., signals) per project months and the number of expected engineering objects of the completed automation system. Thus, the project progress becomes traceable and observable. A “drill-down” feature enables a more detailed view on an individual component to see subcomponents on various levels of detail.

Figure 10b illustrates the *volatility of engineering* objects (i.e., signals) e.g., the number of new, removed and modified signals per project months. This metric enables project assessments and evaluations of the engineering objects within a project. A high number of changes (modified and removed signals) might indicate improvement options of the engineering

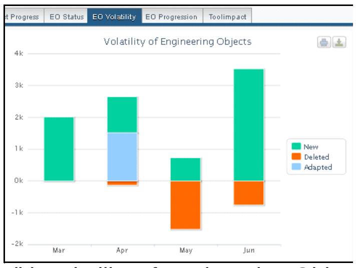
---

[2] See *http://cdl.ifs.tuwien.ac.at/en/node/29* for a detailed description of these use cases and screen casts.

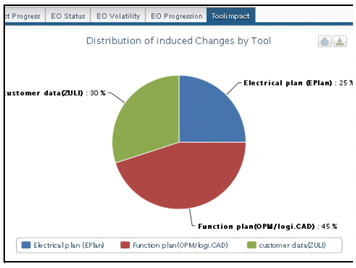
process because some components might have been reused during system development (including some required changes). Nevertheless, a more details investigation of project managements will be necessary to fully understand the signal change results; the engineering cockpit provides a visualization of the captured data and metrics. The initiator of the change, i.e., a tool or engineer, might be an indicator for improving individual disciplines and the application of components. Thus, the engineering cockpit provides a view on the *tool impact of changes* (Figure 10c).



(a) Engineering Cockpit Overview with „Drill-Down“



(b) Volatility of Engineering Objects



(c) Tool Impact of Changes

Figure 10: Project Observation with an Engineering Cockpit Prototype.

V. SUMMARY AND LESSONS LEARNED

Development projects of large scale automation systems, e.g., manufacturing plants, steel mills, and power plants, typically include a large and heterogeneous group of stakeholders who participate in systems engineering projects. These stakeholders come from different disciplines, e.g., mechanical and process engineering, electrical engineering, and software engineering, using different tools (technical heterogeneity) and data models from various sources (semantic heterogeneity). Nevertheless, heterogeneous tools and data models hinder (a) efficient collaboration between disciplines, (b) effective and efficient quality assurance activities across disciplines and domain borders, and (c) make a comprehensive project and process observation and control more difficult. The Automation Service Bus (ASB) provides a middleware platform that aims at bridging the technical and semantic gap between tools and data models from various disciplines.

Based on previous publications (e.g., [4]) this paper summarized the basic concepts of the ASB approach and illustrates a prototype application at an industry partner to improve collaboration, quality assurance, and project observation and control in a real world application context.

RQ1: How can we enable data exchange between heterogeneous data sources? Efficient data exchange approaches between data models from heterogeneous sources are the foundation for (a) efficient quality assurance activities and (b) a comprehensive process observation capabilities across tools and domain borders. To overcome semantic heterogeneity we introduced the *virtual common data model (VCDM)* that bridges the gap between different sources [5][6]. Based on various tool data models (from various sources) the VCDM holds the common concepts, e.g., signals, used for collaboration and data exchange based on an agreed subset of data elements.

RQ2: How can we enable efficient quality assurance across disciplines? Common concepts represent overlapping information areas (see Figure 8) where engineers have to collaborate and exchange data. Furthermore, these common concepts are used for synchronizing disciplines along the engineering process. Note that frequent synchronization supports change management and defect detection. In common industry practice every discipline applies individual quality assurance activities. In addition, the ASB enables two novel quality assurance approaches for defect detection: focused inspection and end-to-end integration testing. Focused inspection enables the identification of defined defects in the overlapping information areas where two or more disciplines collaborate by applying common concepts. The end-to-end test enables semi-automated testing of the signal chain from hardware sensors to software variables.

RQ3: How can we enable a comprehensive view on the engineering project? Based on the VCDM process observations we introduced an Engineering Cockpit [13], aiming at (a) providing stakeholder related data derived from the engineering project databases and (b) enabling control of project steps based on the analysis results [25]. In addition to process related information and metrics, project context information and quality data might be useful to support project and quality managers in better assessing and controlling the automation systems project.

Lessons Learned & Future Work. Based on discussion with our industry partners and our experiences in developing business IT software products, we see the technical integration of tools and semantic integration of data models as very promising approaches to support different types of projects, where different stakeholders and disciplines are involved, and/or heterogeneous engineering environments can be observed.

Future work of this ongoing research project will focus on (a) different aspects of common concepts (in other engineering domains), (b) automation supported process modeling, observation, and validation, and (c) enhanced and automation-supported quality assurance methods.

ACKNOWLEDGMENT

This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. We also want to thank the domain experts at our industry partners for their valuable insight into engineering projects and their feedback on the prototypes presented in this paper.

REFERENCES

[1] Basili V.: “Evolving and Packaging Reading Techniques”, JSS, 38(1), pp3-12, 1997.

[2] Beck K., Andres C. “Extreme Programming Explained - Embrace Change”, Addison-Wesley, 2004.

[3] Biffl S.: “Software Inspection Techniques to support Project and Quality Management”, Shaker, 2001.

[4] Biffl S., Schatten A., Zoitl A.: “Integration of Heterogeneous Engineering Environments for the Automation Systems Lifecycle”, Proc of INDIN, Cardiff, UK, 2009.

[5] Biffl S., Sunindyo W.D., Moser T.: “Bridging Semantic Gaps Between Stakeholders in the Production Automation Domain with Ontology Areas”, Proc of the 21st SEKE Conference, Boston, USA, 2009.

[6] Biffl S., Moser T., Winkler D.: “Risk Assessment in Multi-Disciplinary (Software+) Engineering Projects”, Int. Journal of Software Engineering and Knowledge Engineering (IJSEKE), Special Session on Risk Assessment, Volume 21(2), March 2011.

[7] Kaner C., Falk J., Hguyen H.Q.: “Testing Computer Software”, Wiley, 1999.

[8] Kollanus S., Koskinen J.: “Survey of Software Inspection Research: 1991-2005”, Working Paper WP 40, University of Jyväskylä, 2007.

[9] 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.

[10] Meyers G.J., Sandler C., Badgett T., Thomas T.M.: “The Art of Software Testing”, 2nd edition, John Wiley & Sons, 2004.

[11] Moser T., Waltersdorfer F., Zoitl A., Biffl S.: „Version Management and Conflict Detection Across Heterogeneous Engineering Data Models”, Proc of the 8th Int. Conf on Industrial Informatics (INDIN), Osaka, Japan, 2010.

[12] Moser T., Biffl S., Sunindyo W.D., Winkler D.: “Integrating Production Automation Expert Knowledge across Engineering Stakeholder Domains”, Proc of the 4th CISIS Conference, Krakow, Poland, 2010.

[13] Moser T., Mordinyi R., Winkler D., Biffl S.: “Engineering Project Management using the Engineering Cockpit: A collaboration platform for project managers and engineers”, Proc of the 9th Int. Conf on Industrial Informatics (INDIN), Lisbon, Portugal, 2011.

[14] Münch J., Heidrich J.: “Software project control centers: concepts and approaches”, Journal of Systems and Software, 70, (1-2), pp. 3-19, 2004.

[15] Neumann R., Zbrog F., Dumke R.: “Cockpit Based Management Architectures”, in Abran A., Braungarten R., Dumke R.R., Cuadrado-Gallego J.J., and Brunekreef J. (Eds.): “Software Process and Product Measurement”, Springer, pp. 87-100, 2009.

[16] Ramler R., Beer W., Klammer C., Wolfmaier K., Larndorfer S.: “Concept, Implementation and Evaluation of a Web-Based Software Cockpit”, Proc. 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 385-392, 2010.

[17] Sage A.P., Rouse W.B.: “Handbook of Systems Engineering and Management”, 2nd Edition; Wiley, 2009.

[18] Schäfer W., Wehrheim H.: “The Challenges of Building Advanced Mechatronic Systems”, Pros of ICSE, Future of Software Eng., Washington DC, 2007.

[19] Schwaber K.: “Agile Project Management with Scrum”, Microsoft Professional, ISBN 978-0735619937, 2004.

[20] Sommerville I.: “Software Engineering”, 8th Edition, Addison Wesley, 2007.

[21] Sunindyo W.D., Moser T., Winkler D., Biffl S.: “Foundations for Event-Based Process Analysis in Heterogeneous Software Engineering Environments”, Proc of 36th Euromicro Conference, SEAA, France, 2010.

[22] Winkler D., Moser T., Mordinyi R., Sunindyo W.D., Biffl S.: “Engineering Object Change Management Process Observation in Distributed Automation Systems Projects”, Proc of the 18th EuroSPI Conference, Denmark, 2011.

[23] Winkler D.: “Improvement of Defect Detection with Software Inspection Variants : A Large-Scale Empirical Study on Reading Techniques and Experience”, VDM, 2008.

[24] Winkler D., Biffl S., Östreicher T.: “Test-Driven Automation – Adopting Test-First Development to Improve Automation Systems Engineering Processes”, Proc. of the 16th EuroSPI Conference, Madrid, 2009.

[25] Winkler D., Biffl S.: “Improving Quality Assurance in Automation Systems Development Projects”, in “Quality Assurance”, INTEC Open Publishing, 2012 (accepted for publication).