Workflow Processing and Verification for Safety Critical Engineering – Detailed Design and Integrating Feedback of Experts

Deliverable D6.2

FFG – IKT der Zukunft
SHAPE Project
2014 – 845638

Table 1 Document Information

Project acronym:	SHAPE
Project full title:	Safety-critical Human- & dAta-centric Process management in Engineering projects
Work package:	6
Document number:	6.2
Document title:	Workflow Processing and Verification for Safety Critical Engineering Detailed Design and Integrating Feedback of Experts
Version:	0.1
Delivery date:	E03/2016 (M3)
Actual publication date:	....
Dissemination level:	Private
Nature:	Technical Report
Editor(s) / lead beneficiary:	Siemens
Author(s):	Alois Haselböck, Tudor Ionscu
Reviewer(s):	WU ???

Table 2 History

Version	Changes	Authors
0.1	Created Structure	A. Haselböck

Contents

1 Introduction

This deliverable provides an overview of the main design aspects and requirements to workflow processing and verification for safety critical engineering. The requirements and design decisions where presented and discussed with experts from the rail automation department from Siemens Austria. Their feedback, which is encompassed in this deliverable, was taken into consideration throughout the development process.

The detailed design of the system is presented in Appendix A in Chap. 5.

2 Requirements to Workflow Processing and Verification

During several meetings, experts from Siemens Rail Automation defined and refined requirements on the integration of a business process management system (BPMS) into their engineering and verification process. The main input came from Mr. Wolfgang Schwaiger (head of engineering group) and Mr. Heinrich Müller (head of verification/validation group).

2.1 Requirements from Engineering

Following are the main requirements from the viewpoint of engineering.

- *Distributed engineering teams.* Rail Automation projects usually are large and complex technical projects, meaning that most often more than one engineer/teams of engineers are involved. Engineers working on a project are often located at different sites (e.g., Austria and Slovakia).
- *Parallel projects.* Usually, an engineer or a team of engineers works on several projects in parallel.
- *Different system types.* The portfolio of Siemens Rail Automation in Austria contains various different types of systems. The main one in Austria is the so-called "Anlagenbauprozess" for configuring electronic interlocking systems in Austria. For interlocking systems in countries other than Austria, a different technology and tool set is used. Other types of systems are: ETCS (European Train Control System) level 1 and 2, level crossings, etc.
- *Different project types.* The main two different variants of a project for a system are: (1) Build a new system; (2) Modify/extend an existing system.
- *Process definitions.* For each system, a strict process for engineering, verification / validation and integration is defined. This process definition is available in the form of a semi-formal documentation (usually descriptive text and Visio

drawings). Processes are embedded in a *project management process*, covering all the phases and steps of a project.

- *Process documentation.* The different process steps of a project (i.e., a process instance) - defined in a generic project definition - are currently documented manually by the engineer in a so-called project logbook and in checklist of predefined formats.
- *Engineering and verification tools.* Engineers mainly work with specialized tools. It is of crucial importance that the right tool versions and the right versions of input data files are used.
- *Work parallelization.* In principle, the different processes are sequential processes. Due to distributed work and parallel projects, many process steps are performed in parallel.
- *Conformance checks.* There are many constraints specified on the different process steps and assets (e.g., the right versions of input files). Currently, many of these checks are done at the end of a project. Doing checks as early as possible could save rollback and rework time.
- *Skills.* Engineers have different experiences and skills. The goal is to assign that engineer to a process step who is (a) available, (b) has the necessary skills, (c) preferably knows already the facility in case of a modification/extension.
- *Project communication.* Communication between team members is mainly done face-to-face and per e-mail. Partly, predefined email templates are used, e.g. for passing over engineering data from one phase of the project to the next phase.
- *Documentation of time and costs.* Documentation of time and costs of engineering/verification tasks in a project is usually done not on basis of single steps, but on process phases: How many hours are used for engineering or verifying a project? Documentation on a finer level (e.g. on the level of a single process step) could facilitate a better resource allocation and planning of future projects.
- *Data access rights.* Currently, there are no hard restrictions or tool support on who is reading or changing which data.
- *Change interdependencies.* Currently, it is very difficult to know or track interdependencies of changes at different parts of the system. This leads to challenges like: (a) change tracking, (b) change effect interdependency control, (c) communication of changes, and (d) to make all the involved people work according to the interdependency model.
- *Non-human resources.* The main, critical non-human resources are hardware and workstations in the test laboratory.

2.2 Requirements from Verification/Validation

Following are the main requirements from the viewpoint of the verification/validation process.

- *Final approval.* The main goal of the verification/validation process is the final approval of a national safety authority.
- *Verification and validation process.* Part of the overall project management plan are the verification process and the validation process. These processes come after the engineering process. Doing some of the verification/validation steps earlier could save time by finding errors earlier.
- *Process approvals.* Engineering, verification and validation processes must have several approvals, like "Verfahrenssicherheitsnachweis" and "Gutachten". An extensive and up-to-date process documentation is an important prerequisite of these approvals.
- *SIL 4.* Most of the systems built by Rail Automation must be engineered under the condition of SIL 4 (Safety Integration Level). SIL 4 is a very high level and usually cannot be achieved by making each single process step SIL 4 compliant, but the whole process is designed to result in SIL 4 systems by integrating diverse redundant engineering steps. Example: The output of an engineering tool is re-transformed to the input data and checked against them by a separate step/tool.
- *RAMS.* RAMS (Reliability, Availability, Maintainability, and Safety) are a cluster of non-functional requirements that need to be fulfilled by any safety-critical system, including railway systems. To ensure a satisfactory level for the RAMS non-functional requirements, a series of domain-specific testing and verification methods can be used. These methods must be supported by a suitable and reliable processes and comprehensive documentation. The achievement of the RAMS requirements is a prerequisite for fulfilling the requirements of the EN-standards and SIL 4.
- *Standards and norms.* The main standards are EN50126 [1] (life-cycle model), EN50128 (important for the engineering process), and EN50129 (risk analysis, safety targets, hazard rates).
- *Traceability.* Tracing of steps between input and output should be improved. Documentation of process steps is an important input for validation (usually at the end of a project).

[1] In the Bachelor's Thesis of Julia Fuchsbauer [2] - written in the course of the project SHAPE - the rules of EN50126 have been systematically extracted and categorized.

- *Process documentation.* Engineering must prove that the system has been engineered in compliance with the defined, safe engineering process. Currently, there are no automated monitoring/tracking capabilities, so the challenge is to know whether all process steps are executed correctly or not. Most of the checks are currently performed at the end of the process.
- *Delta verification.* In modification/extension projects, much verification work could be saved, if only those parts of the system data are checked which has actually been changed.

Safety integrity level (SIL) is defined as a relative level of risk-reduction provided by a safety function, or to specify a target level of risk reduction. In simple terms, SIL is a measurement of performance required for a safety instrumented function. The requirements for a given SIL are not consistent among all of the functional safety standards. In the European functional safety standards based on the IEC 61508 standard four SILs are defined, with SIL 4 the most dependable and SIL 1 the least. A SIL is determined based on a number of quantitative factors in combination with qualitative factors such as development process and safety life cycle management.

2.3 Derived SHAPE Requirements

Derived from Siemens' Rail Automation requirements we identify the following main requirements as input for our SHAPE reference architecture and tool selection (cf. Fig. 1):

- *Monitoring.* Based on a clear and standardized definition of the different process types at Rail Automation, the BPMS must be able to monitor the engineers execution of their process steps.
  - Data of all steps should be logged: Who has performed the step? How long did it take? What were the input and output artifacts? Etc. These logs can be used for process documentation, querying, learning (e.g. the skill matrix) and data analytics.
  - Deviations from the process should be avoided. In exceptional situations, where deviations are necessary, the BPMS should document deviations and guide users back to the specified process.
- *Conformance Checking.* Based on rules defined as additional input to the process definition, conformance checks should prove the validity of the process step results. Sources of such rules are:
  - Standards and regulations. E.g., the verification step must be performed by a user different to the one who did the corresponding engineering step.

  - Data integrity. E.g., the right version of tool EPOS must be used. E.g., the newest versions of the customer data files must be used.
  - Performance heuristics. E.g., if a task is assigned to a user who does this kind of tasks the first time, a second, experienced user must be assigned to that task, as well.
- *Adaptivity/Flexibility.* Although the process management system shall strictly guide the engineers through the process, the system must be able to react on exceptional situations.
  - Re-allocation of resources, e.g. if an engineer gets sick.
  - Exceptional deviations from the standard process step sequence.
- *Automation.*
  - Automatic document generation, e.g. project logbook and checklists.
  - Automated tasks, e.g. automatic check-in of process step results; automated sending of notification emails.
  - Automated computation of optimal assignments of (human or non-human) resources to tasks.

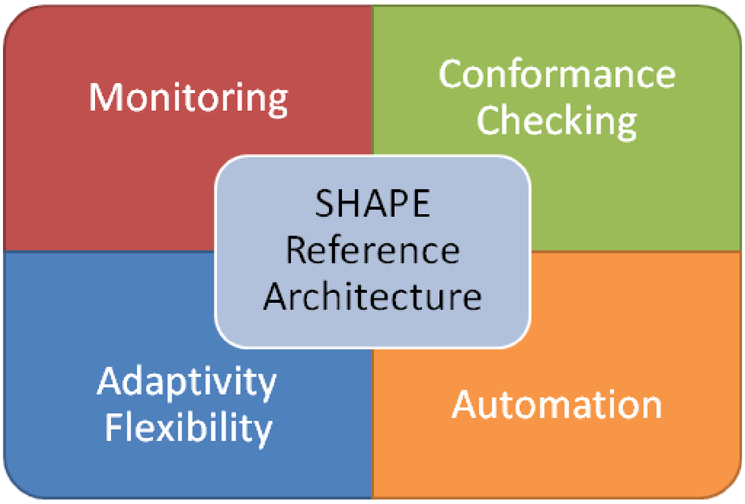


Figure 1 Main SHAPE requirements derived from customer requirements.

Please note: These are the requirements specified with high priority by our Siemens customer. There is another topic which is of high interest in SHAPE- the mining of process data from structured and unstructured sources. This topic is investigated in work package 3 in full detail and is considered in the reference architecture of SHAPE. The reason why the interest of Rail Automation in that topic is lower than in the above requirements is that derivation and specification of engineering and verification/validation processes is usually a highly sophisticated manual task, mainly involving safety aspects and SIL level considerations.

3 Tool Selection

In the Master's Thesis of Alexander Wurl [3], which is considered to be part of this deliverable, several main business process management languages and tools are evaluated. The goal was to find the best language and tool candidate for being used in the SHAPE reference architecture, having the potential to be deployed at Siemens Rail Automation after the project SHAPE.

The Business Process Model and Notation (BPMN) language [1] was selected (cf. [3], chapter 2), because it is a state-of-the-art language, supported by the most current business process engines. BPMN is a graphical representation for specifying business processes in a business process model. Business Process Management Initiative (BPMI) developed BPMN, which has been maintained by the Object Management Group since the two organizations merged in 2005. BPMN is inspired by a number of previous process languages, and nowadays BPMN is promoted and suggested as a standard for modeling and describing processes due to the benefit making process models understandable for humans as well as for machines.

Various different tools have been evaluated in [3], chapter 4:

- Bizagi
- Camunda
- jBPM
- Axon Ivy
- Oracle BPM
- Process Maker
- Softproject

The tool **Camunda** was selected to be used for prototyping in SHAPE. The main reasons are:

- Camunda is an open source BPMS based on Java. It provides a process engine architecture with a great variety of Application Programming Interfaces (APIs) that help to extend the BPMS arbitrarily. Furthermore, it is possible to redesign the whole BPMS since it provides the full source code in Java.
- Camunda is an offspring of the Alfresco BPMS, which is itself an offspring of the jBPM platform developed by JBoss. During its development, Camunda has taken the best parts from both Alfresco and jBPM, while being much more light-weight than its predecessors. Also Camunda offers more flexibility than its predecessors and provides a simple an intuitive development environment based on Eclipse and an easy to use visual BPMN editor. The coupling between the tools is loose and, for the most part, is based on the convention over configuration principle.

- The company Camunda has a close cooperation with the OMG Group41 which affects the BPMS in a way that state-of-the-art technologies and standards such as BPMN 2.0 are built-in and available for this BPMS.
- In terms of service and support, when there is no need for an open source version, Camunda offers an enterprise version with different price levels according to the needs of functionality.

Details of this evaluation can be found in Appendix B in Chap. 6 of this document.

The main process at Rail Automation, the so-calls "Anlagenbauprozess" has been prototypically modelled with Camunda.

In particular, Phase265 of the process was modelled as a BPMN process composed of several call activities (modular sub-processes). This ensures the replaceabilty and testability of every sub-process.

4 Reference Architecture

Figure 2 shows a sketch of the overall SHAPE architecture. Appendix A in Chap. 5 describes the architectural design details.

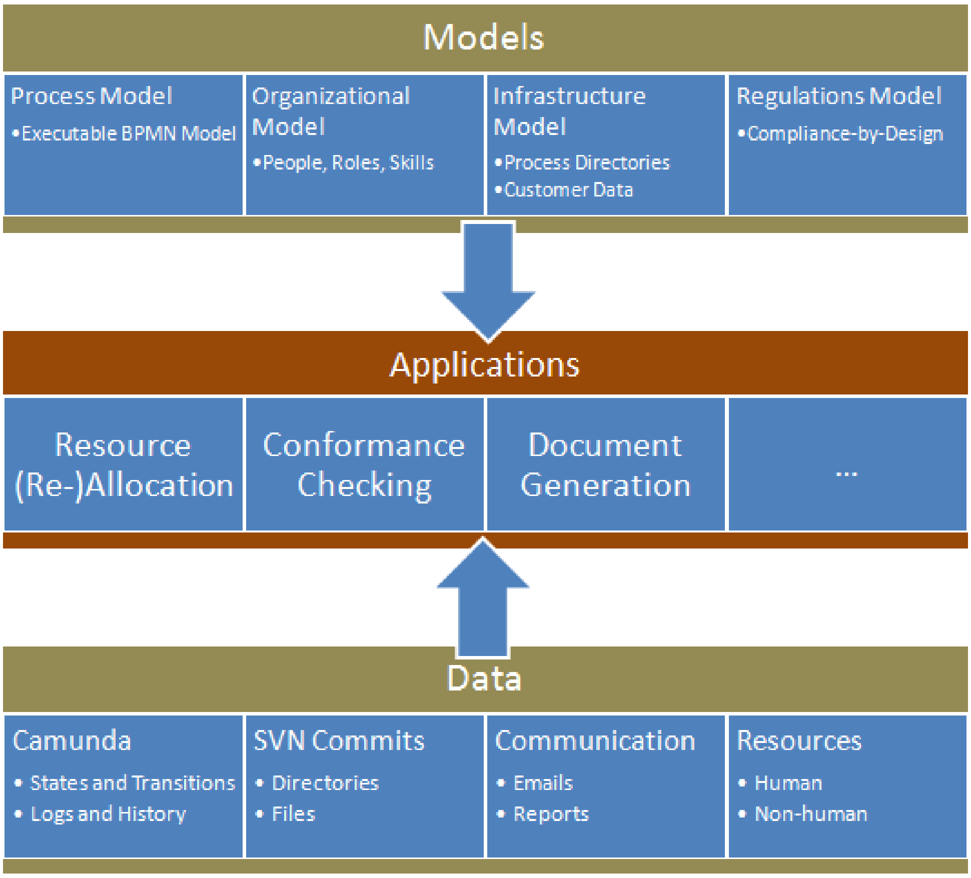
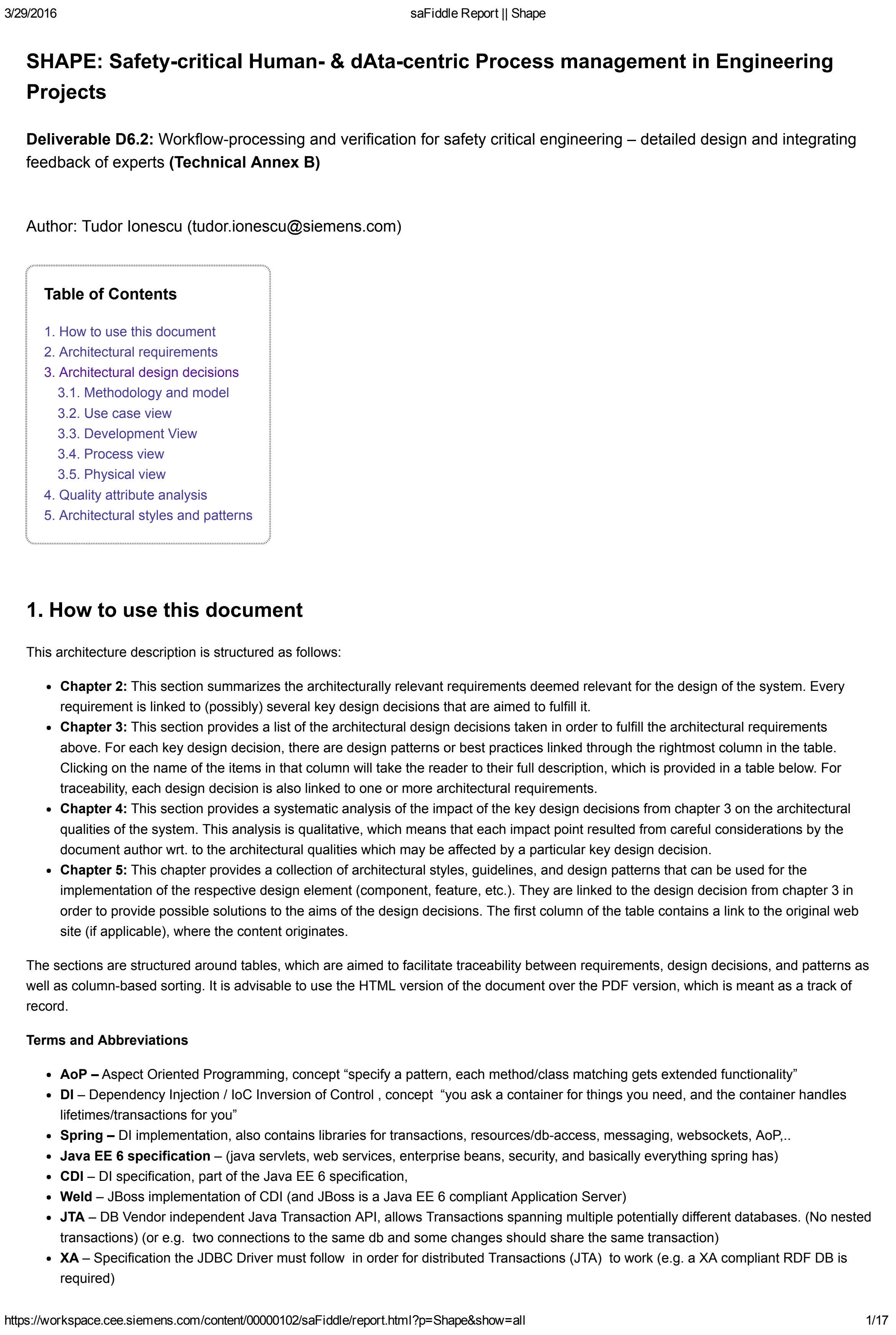


Figure 2 Sketch from meeting on 08.02.2016

5 Appendix A: Architectural Design

10 Private Document



SHAPE: Safety-critical Human- & dAta-centric Process management in Engineering Projects

Deliverable D6.2: Workflow-processing and verification for safety critical engineering – detailed design and integrating feedback of experts **(Technical Annex B)**

Author: Tudor Ionescu (tudor.ionescu@siemens.com)

Table of Contents

1. How to use this document

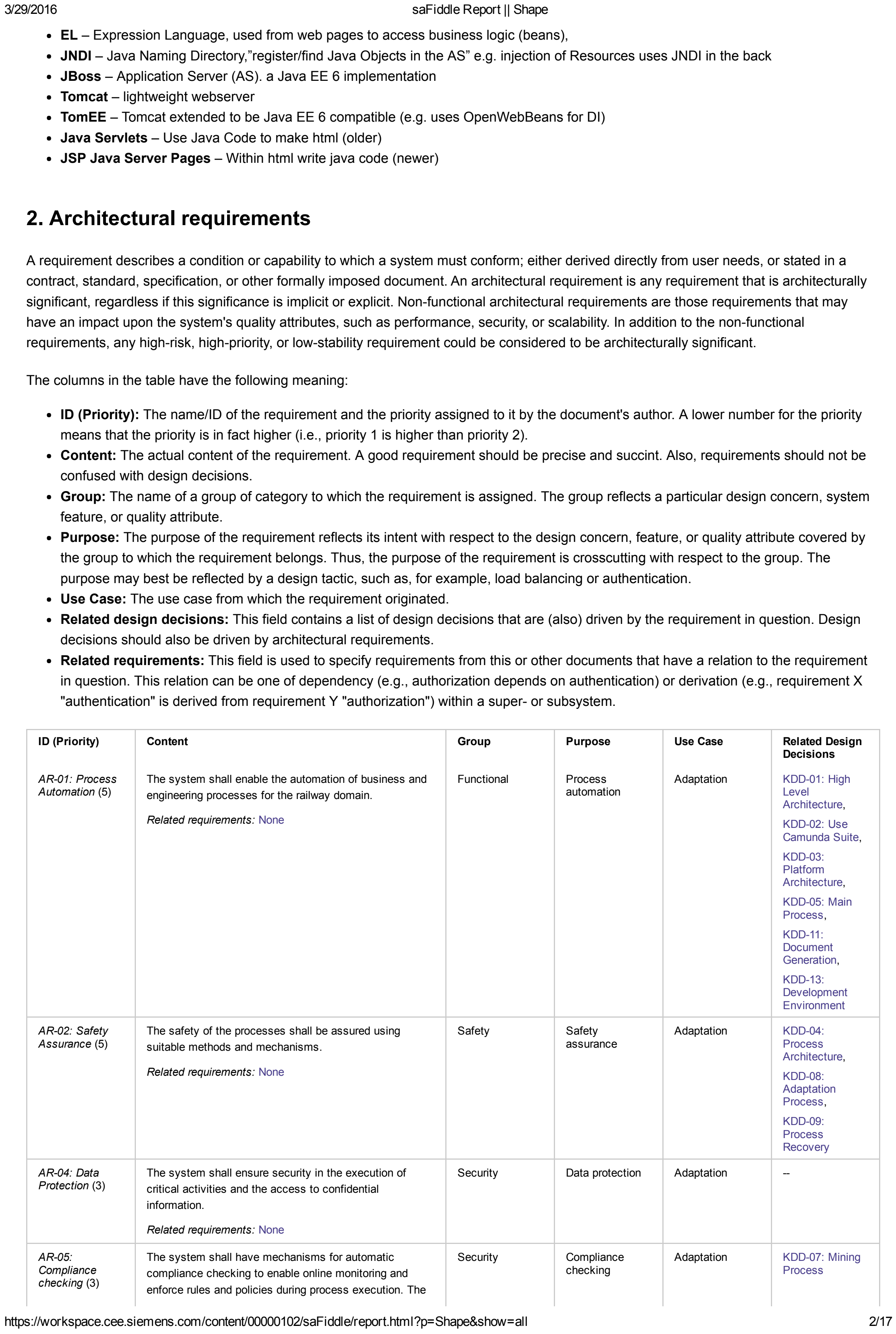
This architecture description is structured as follows:

- **Chapter 2:** This section summarizes the architecturally relevant requirements deemed relevant for the design of the system. Every requirement is linked to (possibly) several key design decisions that are aimed to fulfill it.
- **Chapter 3:** This section provides a list of the architectural design decisions taken in order to fulfill the architectural requirements above. For each key design decision, there are design patterns or best practices linked through the rightmost column in the table. Clicking on the name of the items in that column will take the reader to their full description, which is provided in a table below. For traceability, each design decision is also linked to one or more architectural requirements.
- **Chapter 4:** This section provides a systematic analysis of the impact of the key design decisions from chapter 3 on the architectural qualities of the system. This analysis is qualitative, which means that each impact point resulted from careful considerations by the document author wrt. to the architectural qualities which may be affected by a particular key design decision.
- **Chapter 5:** This chapter provides a collection of architectural styles, guidelines, and design patterns that can be used for the implementation of the respective design element (component, feature, etc.). They are linked to the design decision from chapter 3 in order to provide possible solutions to the aims of the design decisions. The first column of the table contains a link to the original web site (if applicable), where the content originates.

The sections are structured around tables, which are aimed to facilitate traceability between requirements, design decisions, and patterns as well as column-based sorting. It is advisable to use the HTML version of the document over the PDF version, which is meant as a track of record.

Terms and Abbreviations

- **AoP –** Aspect Oriented Programming, concept “specify a pattern, each method/class matching gets extended functionality”
- **DI** – Dependency Injection / IoC Inversion of Control , concept “you ask a container for things you need, and the container handles lifetimes/transactions for you”
- **Spring –** DI implementation, also contains libraries for transactions, resources/db-access, messaging, websockets, AoP,..
- **Java EE 6 specification** – (java servlets, web services, enterprise beans, security, and basically everything spring has)
- **CDI** – DI specification, part of the Java EE 6 specification,
- **Weld** – JBoss implementation of CDI (and JBoss is a Java EE 6 compliant Application Server)
- **JTA** – DB Vendor independent Java Transaction API, allows Transactions spanning multiple potentially different databases. (No nested transactions) (or e.g. two connections to the same db and some changes should share the same transaction)
- **XA** – Specification the JDBC Driver must follow in order for distributed Transactions (JTA) to work (e.g. a XA compliant RDF DB is required)



- **EL** – Expression Language, used from web pages to access business logic (beans),
- **JNDI** – Java Naming Directory,"register/find Java Objects in the AS" e.g. injection of Resources uses JNDI in the back
- **JBoss** – Application Server (AS). a Java EE 6 implementation
- **Tomcat** – lightweight webserver
- **TomEE** – Tomcat extended to be Java EE 6 compatible (e.g. uses OpenWebBeans for DI)
- **Java Servlets** – Use Java Code to make html (older)
- **JSP Java Server Pages** – Within html write java code (newer)

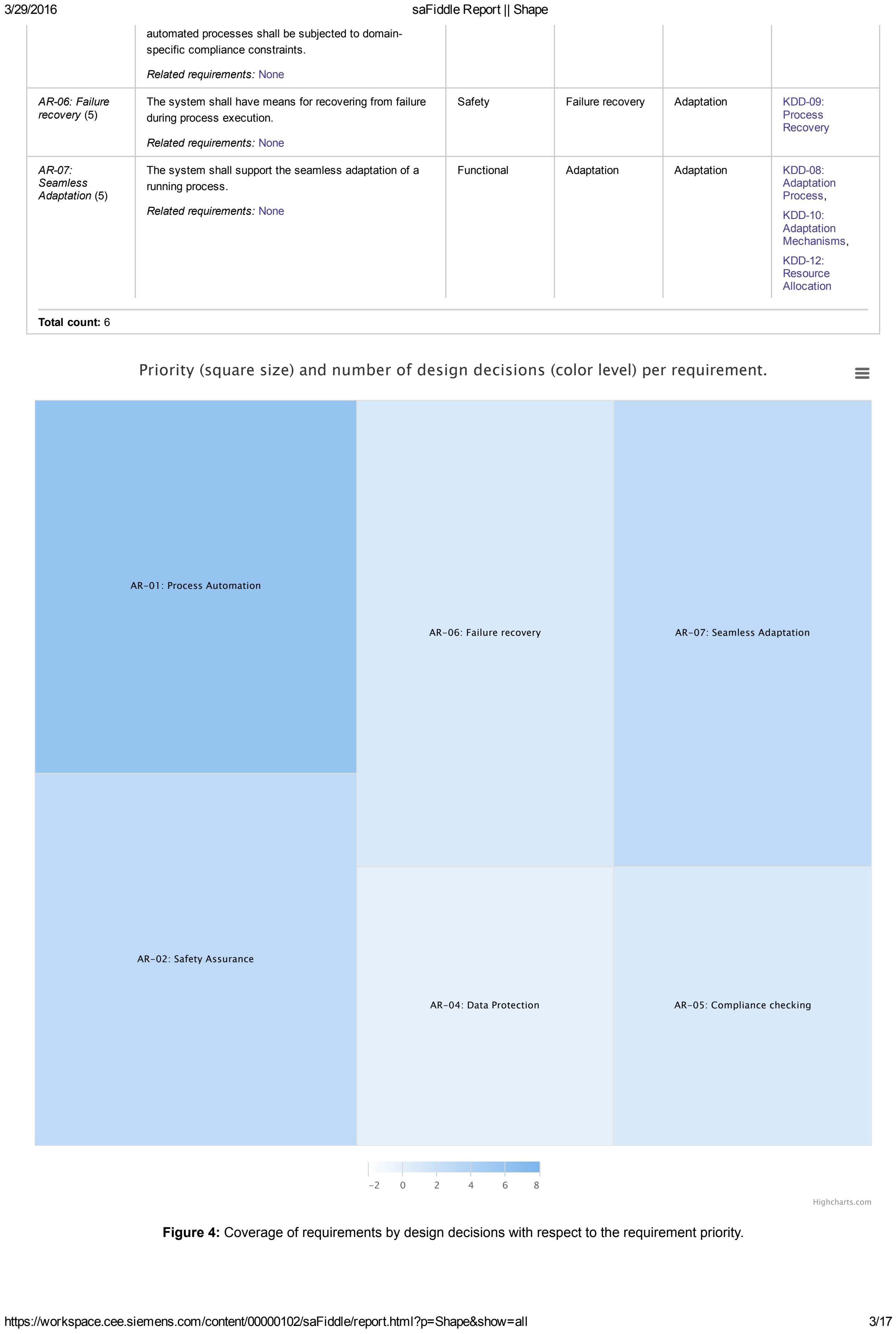
2. Architectural requirements

A requirement describes a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document. An architectural requirement is any requirement that is architecturally significant, regardless if this significance is implicit or explicit. Non-functional architectural requirements are those requirements that may have an impact upon the system's quality attributes, such as performance, security, or scalability. In addition to the non-functional requirements, any high-risk, high-priority, or low-stability requirement could be considered to be architecturally significant.

The columns in the table have the following meaning:

- **ID (Priority):** The name/ID of the requirement and the priority assigned to it by the document's author. A lower number for the priority means that the priority is in fact higher (i.e., priority 1 is higher than priority 2).
- **Content:** The actual content of the requirement. A good requirement should be precise and succint. Also, requirements should not be confused with design decisions.
- **Group:** The name of a group of category to which the requirement is assigned. The group reflects a particular design concern, system feature, or quality attribute.
- **Purpose:** The purpose of the requirement reflects its intent with respect to the design concern, feature, or quality attribute covered by the group to which the requirement belongs. Thus, the purpose of the requirement is crosscutting with respect to the group. The purpose may best be reflected by a design tactic, such as, for example, load balancing or authentication.
- **Use Case:** The use case from which the requirement originated.
- **Related design decisions:** This field contains a list of design decisions that are (also) driven by the requirement in question. Design decisions should also be driven by architectural requirements.
- **Related requirements:** This field is used to specify requirements from this or other documents that have a relation to the requirement in question. This relation can be one of dependency (e.g., authorization depends on authentication) or derivation (e.g., requirement X "authentication" is derived from requirement Y "authorization") within a super- or subsystem.

ID (Priority)	Content	Group	Purpose	Use Case	Related Design Decisions
AR-01: Process Automation (5)	The system shall enable the automation of business and engineering processes for the railway domain. *Related requirements:* None	Functional	Process automation	Adaptation	KDD-01: High Level Architecture, KDD-02: Use Camunda Suite, KDD-03: Platform Architecture, KDD-05: Main Process, KDD-11: Document Generation, KDD-13: Development Environment
AR-02: Safety Assurance (5)	The safety of the processes shall be assured using suitable methods and mechanisms. *Related requirements:* None	Safety	Safety assurance	Adaptation	KDD-04: Process Architecture, KDD-08: Adaptation Process, KDD-09: Process Recovery
AR-04: Data Protection (3)	The system shall ensure security in the execution of critical activities and the access to confidential information. *Related requirements:* None	Security	Data protection	Adaptation	--
AR-05: Compliance checking (3)	The system shall have mechanisms for automatic compliance checking to enable online monitoring and enforce rules and policies during process execution. The	Security	Compliance checking	Adaptation	KDD-07: Mining Process



	automated processes shall be subjected to domain-specific compliance constraints. *Related requirements:* None				
AR-06: Failure recovery (5)	The system shall have means for recovering from failure during process execution. *Related requirements:* None	Safety	Failure recovery	Adaptation	KDD-09: Process Recovery
AR-07: Seamless Adaptation (5)	The system shall support the seamless adaptation of a running process. *Related requirements:* None	Functional	Adaptation	Adaptation	KDD-08: Adaptation Process, KDD-10: Adaptation Mechanisms, KDD-12: Resource Allocation

Total count: 6

Priority (square size) and number of design decisions (color level) per requirement.

Figure 4: Coverage of requirements by design decisions with respect to the requirement priority.

Diagram Explanation

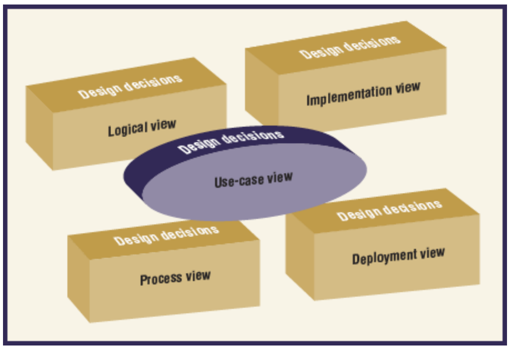
Figure 4 quantifies the number of design decisions taken with respect to each requirement's priority. In decision-oriented software architecture, each design decision is driven by one or several requirements having a stakeholder-defined priority. The diagram thus helps readers to see which requirements are targeted by more design decisions than others. The number of design decisions targeting a particular requirement is reflected by the color level of the squares. This metric can be evaluated in relation to the requirement's priority. The rationale for doing this is that high priority requirements should always be targeted by at least one design decision even in the early architectural design phase of the project.

3. Architectural design decisions

This chapter describes the key design decisions that are meant to fulfill the architectural requirements defined in the previous chapter. For each key design decision, there are design patterns or best practices linked through the rightmost column in the table. The full description of these patterns is provided in chapter 7.

3.1 Methodology and model

The description of the software architecture is based on the 4+1 Views Model [4+1 Views]. Architectural views are used to describe software architectures from the vantage point of different stakeholders in the development lifecycle, whereby each view may address different cross-cutting design concerns.



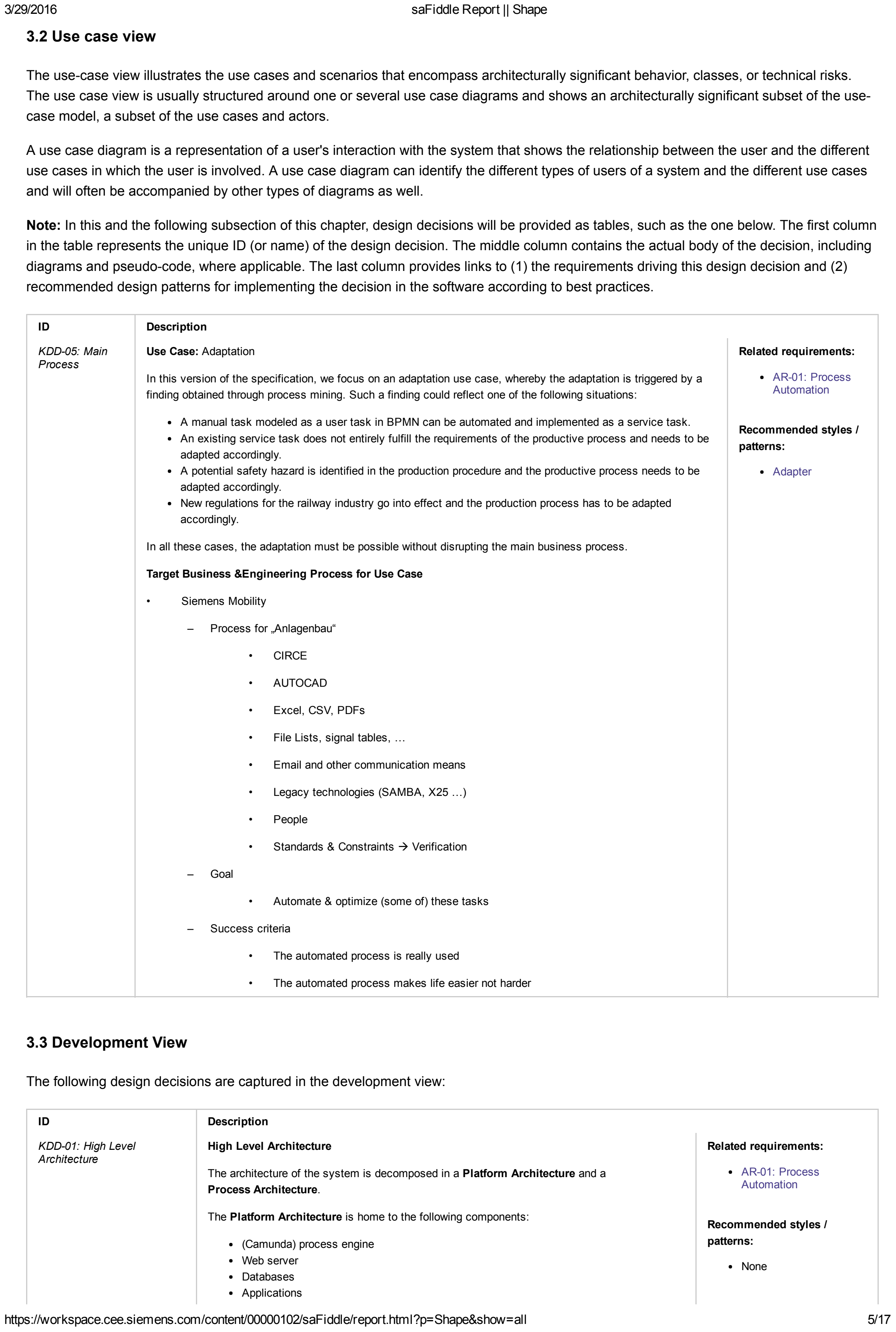
The following architectural views will be used to describe the architecture of the system [4+1 Views]:

- **Scenarios:** The description of an architecture is illustrated using a small set of use cases, or scenarios which become a fifth view. The scenarios describe sequences of interactions between objects, and between processes. They are used to identify architectural elements and to illustrate and validate the architecture design. They also serve as a starting point for tests of an architecture prototype. This view is also known as **use case view**.
- **Development view:** The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML component diagram to describe system components. UML diagrams used to represent the development view include package (or component) diagrams.
- **Logical view:** The logical view is concerned with the functionality that the system provides to end-users. UML diagrams used to represent the logical view include class diagrams, communication diagrams, and component diagrams.
- **Process view:** The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behavior of the system. The process view addresses concurrency, distribution, integrators, performance, and scalability, etc. UML diagrams to represent process view include activity diagrams and sequence diagrams.
- **Physical view:** The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. This view is also known as the deployment view. UML diagrams used to represent physical view include the deployment diagram.

Starting from selected use cases (i.e., Use Cases Ecosystem, Benchmarks, and Grid Planing, including sub-use cases), the architecturally relevant requirements were derived and presented in chapter 2. These requirements serve as drivers for architectural design decisions (ADD). Design decisions are aimed at fulfilling the architectural requirements and are classified in the document by referring to one of the aforementioned architectural views. This approach corresponds to the decision view in software architecture [Decision View]:

- **Decision view:** A “decision view” provides a useful addition and a complement to more traditional sets of architectural views and viewpoints; it gives an explanatory perspective that illuminates the reasoning process itself and not solely its results. This decision view documents aspects of the architecture that are hard to reverse-engineer from the software itself and that are often left tacit.

14 Private Document



3.2 Use case view

The use-case view illustrates the use cases and scenarios that encompass architecturally significant behavior, classes, or technical risks. The use case view is usually structured around one or several use case diagrams and shows an architecturally significant subset of the use-case model, a subset of the use cases and actors.

A use case diagram is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Note: In this and the following subsection of this chapter, design decisions will be provided as tables, such as the one below. The first column in the table represents the unique ID (or name) of the design decision. The middle column contains the actual body of the decision, including diagrams and pseudo-code, where applicable. The last column provides links to (1) the requirements driving this design decision and (2) recommended design patterns for implementing the decision in the software according to best practices.

ID	Description	
KDD-05: Main Process	**Use Case:** Adaptation In this version of the specification, we focus on an adaptation use case, whereby the adaptation is triggered by a finding obtained through process mining. Such a finding could reflect one of the following situations: • A manual task modeled as a user task in BPMN can be automated and implemented as a service task. • An existing service task does not entirely fulfill the requirements of the productive process and needs to be adapted accordingly. • A potential safety hazard is identified in the production procedure and the productive process needs to be adapted accordingly. • New regulations for the railway industry go into effect and the production process has to be adapted accordingly. In all these cases, the adaptation must be possible without disrupting the main business process. **Target Business &Engineering Process for Use Case** • Siemens Mobility – Process for „Anlagenbau“ • CIRCE • AUTOCAD • Excel, CSV, PDFs • File Lists, signal tables, … • Email and other communication means • Legacy technologies (SAMBA, X25 …) • People • Standards & Constraints → Verification – Goal • Automate & optimize (some of) these tasks – Success criteria • The automated process is really used • The automated process makes life easier not harder	**Related requirements:** • AR-01: Process Automation **Recommended styles / patterns:** • Adapter

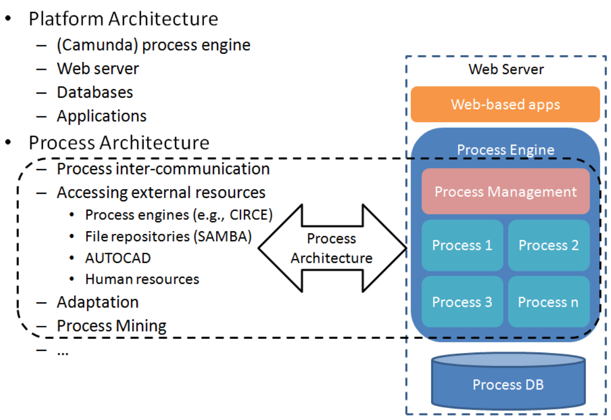
3.3 Development View

The following design decisions are captured in the development view:

ID	Description	
KDD-01: High Level Architecture	**High Level Architecture** The architecture of the system is decomposed in a **Platform Architecture** and a **Process Architecture**. The **Platform Architecture** is home to the following components: • (Camunda) process engine • Web server • Databases • Applications	**Related requirements:** • AR-01: Process Automation **Recommended styles / patterns:** • None

The **Process Architecture** concerns the functional aspects of the process:

- Process inter-communication
- Accessing external resources
- Integration of external process engines (e.g., CIRCE)
- Access to external file repositories (SAMBA)
- Access to email services (e.g., automatic sending of notification emails)
- Access to file versioning repositories (e.g., automatic check-in/out of files in/from SVN or GIT)
- Consistency checks on file contents (e.g. on XML or Excel files)
- Coordination of human resources
- Adaptation of the main process
- Monitoring and mining of the main process
- ...



- Platform Architecture
  - (Camunda) process engine
  - Web server
  - Databases
  - Applications
- Process Architecture
  - Process inter-communication
  - Accessing external resources
    - Process engines (e.g., CIRCE)
    - File repositories (SAMBA)
    - AUTOCAD
    - Human resources
  - Adaptation
  - Process Mining
  - ...

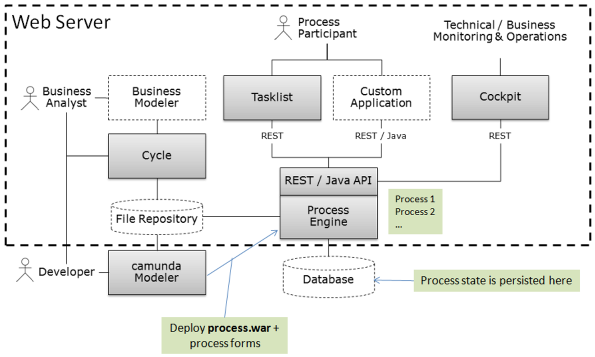
ID

KDD-02: Use Camunda Suite

Description

Use the Camunda Suite as a Process Automation Platform

The following diagram illustrate how different actors having different roles in the project interact with the Camunda suite during development and deployment.



Components:

Process Implementation and Execution

- camunda engine - The core component responsible for executing BPMN 2.0 processes.
- REST API - The REST API provides remote access to running processes.
- Spring, CDI integration - Programming model integration that allows developers to write Java Applications that interact with running processes.

Process Design

- camunda modeler - A modeler plugin for eclipse that allows developers to design & refactor processes inside their IDE.
- camunda cycle - Enables BPMN 2.0 based Roundtrip between Business and IT parties involved in a project. Allows to use any BPMN 2.0 modeling tool with camunda BPM.

Process Operations

Related requirements:

- AR-01: Process Automation

Recommended styles / patterns:

- None

- camunda engine - JMX and advanced Runtime Container Integration for process engine monitoring.
- camunda cockpit - Web application tool for process operations.

Human Task Management

- camunda tasklist - Simple web application demonstrating how the process engine task API can be used.

Pros:

+ Addresses all aspects of process management & execution

+ High-quality documentation

+ Sound software architecture (high design pattern density)

+ Uses state of the art technology (JAX-RS, AngularJS, Spring, jQuery, etc.)

+ Many supported environments (Tomcat, JBOSS, Glassfish …)

+ Flexibility, extensibility (Community extensions)

Cons:

- Is it really lightweight?

- A constellation of tools → Temptation to use them all → Complexity

- Infrastructure needs → Web servers, database servers → Administrators + Programmers

- Real flexibility or exhaustiveness?

Bottom line:

+ Pros seem to dominate cons from a software architecture perspective

ID

KDD-11: Document Generation

Description

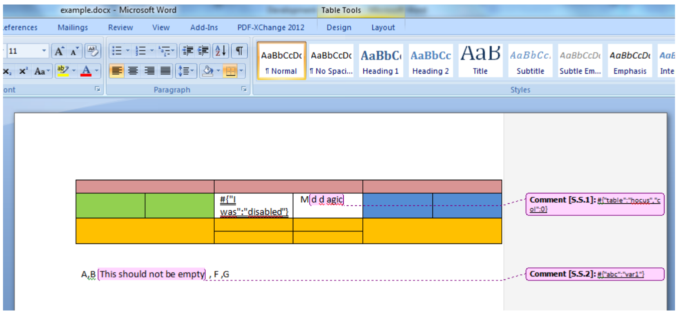
Automated Checklist Generation

During Phase265, engineers need to fill out a series of check lists. These tasks are ideal for automation using Camunda and BPMN. Checklists are usually Word documents, which the engineers currently need to fill out manually. Using Apache POI and BPMN user tasks, the system implements automated document generation of these checklists.

Implementation Details

The Document Generator is implemented as a Camunda execution listener and can be triggered after each user task. This can be specified in the BPMN editor by editing the Listeners (Add execution listeners) of each user task.

Example.docx (Template)



#{"I was":"disabled"}

M d d agic

Comment [S.S.1]: #{"table":"hocus","col":0}

A,B This should not be empty , F ,G

Comment [S.S.2]: #{"abc":"var1"}

- The template can be partly filled out and includes annotated regions and placeholder comments of the form:

- #{"varname":"varvalue"} -- after the # there must be a valid JSON object or array

The use can define custom functions for modifying the document based on the placeholders defined in the template.

NamedReference("pvar",map)

Takes a hashmap (**map**) containing key-value pairs (e.g., process variables), which replace the placeholder in the template document. The **"pvar"** parameter is called the *scope* of the search.

For example:

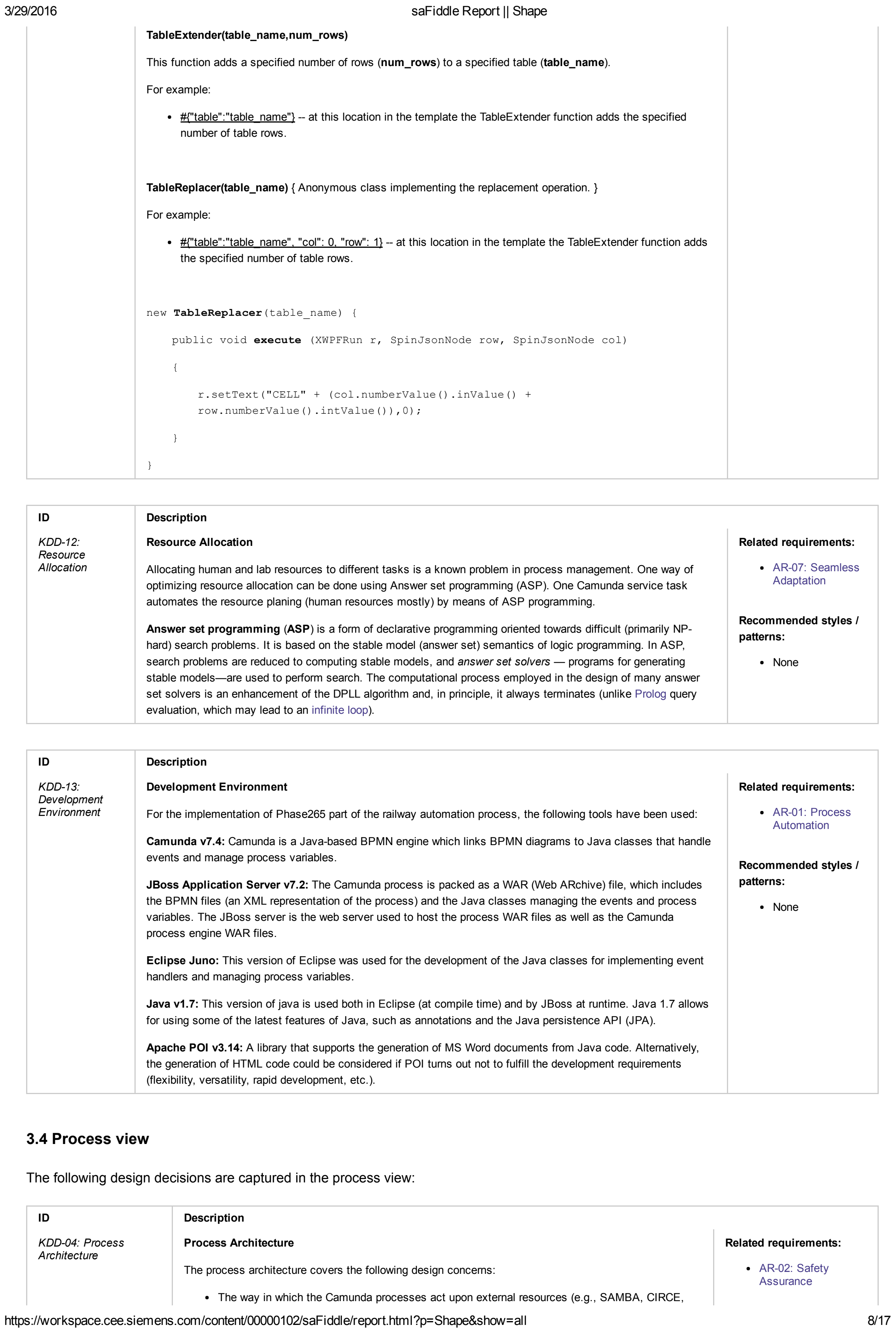
- #{"pvar":"some_process_variable"} will be replaced by the value associated to the key "some_process_variable" in the **map** paramter.

Related requirements:

- AR-01: Process Automation

Recommended styles / patterns:

- None



TableExtender(table_name,num_rows)

This function adds a specified number of rows (**num_rows**) to a specified table (**table_name**).

For example:

- #{"table":"table_name"} -- at this location in the template the TableExtender function adds the specified number of table rows.

TableReplacer(table_name) { Anonymous class implementing the replacement operation. }

For example:

- #{"table":"table_name", "col": 0, "row": 1} -- at this location in the template the TableExtender function adds the specified number of table rows.

```
new TableReplacer(table_name) {

    public void execute (XWPFRun r, SpinJsonNode row, SpinJsonNode col)

    {

        r.setText("CELL" + (col.numberValue().inValue() +
        row.numberValue().intValue()),0);

    }

}
```

ID	Description	
KDD-12: Resource Allocation	**Resource Allocation** Allocating human and lab resources to different tasks is a known problem in process management. One way of optimizing resource allocation can be done using Answer set programming (ASP). One Camunda service task automates the resource planing (human resources mostly) by means of ASP programming. **Answer set programming** (**ASP**) is a form of declarative programming oriented towards difficult (primarily NP-hard) search problems. It is based on the stable model (answer set) semantics of logic programming. In ASP, search problems are reduced to computing stable models, and *answer set solvers* — programs for generating stable models—are used to perform search. The computational process employed in the design of many answer set solvers is an enhancement of the DPLL algorithm and, in principle, it always terminates (unlike Prolog query evaluation, which may lead to an infinite loop).	**Related requirements:** • AR-07: Seamless Adaptation **Recommended styles / patterns:** • None

ID	Description	
KDD-13: Development Environment	**Development Environment** For the implementation of Phase265 part of the railway automation process, the following tools have been used: **Camunda v7.4:** Camunda is a Java-based BPMN engine which links BPMN diagrams to Java classes that handle events and manage process variables. **JBoss Application Server v7.2:** The Camunda process is packed as a WAR (Web ARchive) file, which includes the BPMN files (an XML representation of the process) and the Java classes managing the events and process variables. The JBoss server is the web server used to host the process WAR files as well as the Camunda process engine WAR files. **Eclipse Juno:** This version of Eclipse was used for the development of the Java classes for implementing event handlers and managing process variables. **Java v1.7:** This version of java is used both in Eclipse (at compile time) and by JBoss at runtime. Java 1.7 allows for using some of the latest features of Java, such as annotations and the Java persistence API (JPA). **Apache POI v3.14:** A library that supports the generation of MS Word documents from Java code. Alternatively, the generation of HTML code could be considered if POI turns out not to fulfill the development requirements (flexibility, versatility, rapid development, etc.).	**Related requirements:** • AR-01: Process Automation **Recommended styles / patterns:** • None

3.4 Process view

The following design decisions are captured in the process view:

ID	Description	
KDD-04: Process Architecture	**Process Architecture** The process architecture covers the following design concerns: • The way in which the Camunda processes act upon external resources (e.g., SAMBA, CIRCE,	**Related requirements:** • AR-02: Safety Assurance

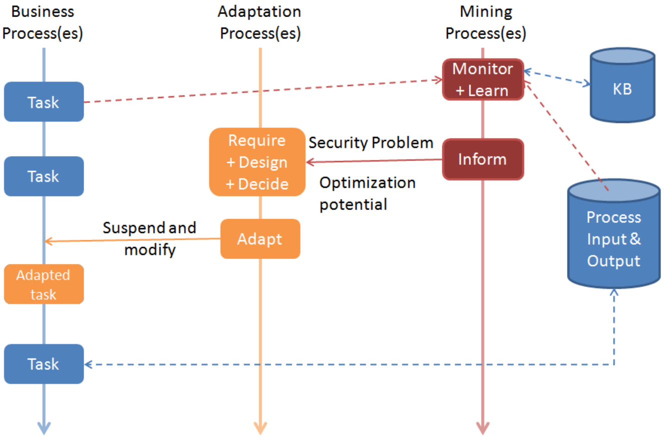
AUTOCAD, …)

- Data handling, communication, synchronization, …
- The number of types of processes needed to implement process monitoring, mining, and adaptation

Proposed Solution:

Separate application concerns from process monitoring, mining, and adaptation using three different processes:

- A use case-based process (henceforth denoted **Main Productive (Business) Process**)
- An **Adaptation Process** which deals with the runtime adaption of the main process
- A **Monitoring and Mining Process** that monitors and mines the process database informs the adaptation processes



Recommended styles / patterns:

- None

ID

KDD-07: Mining Process

Description

Process Monitoring and Mining

The mining process aimes at the following aspects of safety-critical processes:

– Process compliance constraints

– Structured querying of streaming event data

– Ensure security in the execution of critical activities and the access to confidential information

– Mechanisms for automatic compliance checking to enable online monitoring and enforce rules and policies during process execution

The monitoring and mining process runs in parallel with the productive business processes. Its main task consists of mining the unstructured process data. The results (insights) are given either in natural langage or an RDF format:

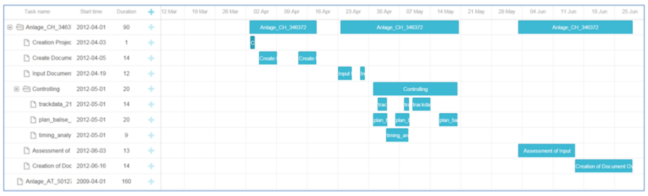
- Natural language >> Expert-driven adaptation (for safety and/or mission critical processes)
- RDF format (JSON-LD) >> Automated adaptation (for non-critical processes)

An insight can be anything that can potentially lead to a new requirement to the main process. For example, they could also show under which circumstances different key performance indicators (KPI) may exceed their designated "safe" range. Insights are used to inform the adaptation process.

Example Insight

Mining task: Look into GIT (versioning system) logs produced by the process

Result: Reverse engineered Gantt chart. The method for creating this Gantt chart is described in more details in the following publication: *BALA, S., CABANILLAS, C., MENDLING, J., ROGGE-SOLTI, A., POLLERES, A. (2015). Mining Project-Oriented Business Processes. Lecture Notes in Computer Science 9253, 425-440.*



Task name	Start time	Duration
Anlage_CH_3463	2012-04-01	90
Creation Projec	2012-04-03	1
Create Docume	2012-04-05	14
Input Documen	2012-04-19	12
Controlling	2012-05-01	20
trackdata_21	2012-05-01	14
plan_balise_	2012-05-01	20
timing_analy	2012-05-01	9
Assessment of	2012-06-03	13
Creation of Doc	2012-06-16	14
Anlage_AT_5012	2009-04-01	160

Insight: Process can be optimized (see D3.2 for details)

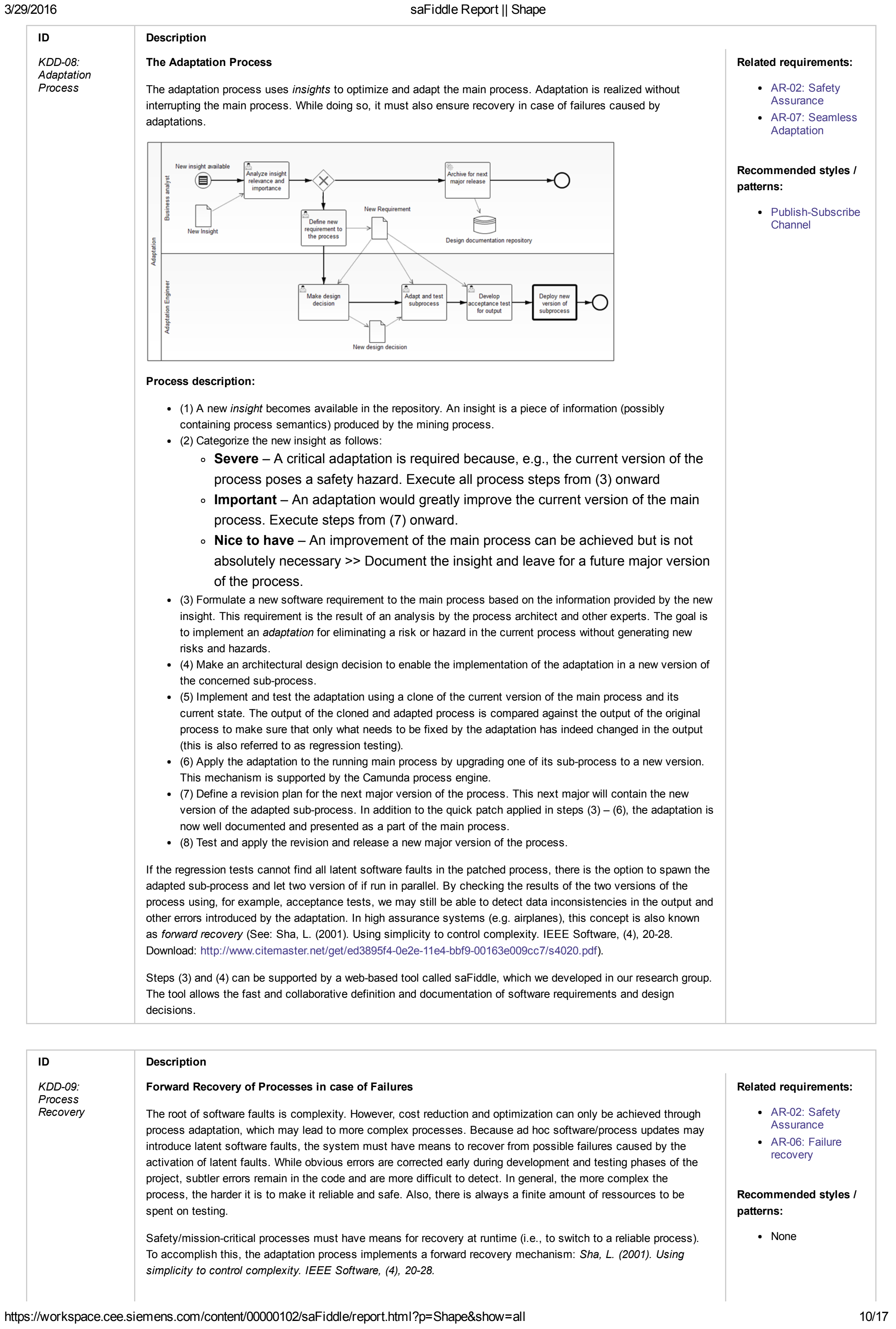
Derived requirement: Adapt the processes accordingly.

Related requirements:

- AR-05: Compliance checking

Recommended styles / patterns:

- None



ID

KDD-08: Adaptation Process

Description

The Adaptation Process

The adaptation process uses *insights* to optimize and adapt the main process. Adaptation is realized without interrupting the main process. While doing so, it must also ensure recovery in case of failures caused by adaptations.

Adaptation
Business analyst
New insight available
Analyze insight relevance and importance
New Insight
Archive for next major release
New Requirement
Define new requirement to the process
Design documentation repository
Adaptation Engineer
Make design decision
Adapt and test subprocess
Develop acceptance test for output
Deploy new version of subprocess
New design decision

Process description:

- (1) A new *insight* becomes available in the repository. An insight is a piece of information (possibly containing process semantics) produced by the mining process.
- (2) Categorize the new insight as follows:
  - **Severe** – A critical adaptation is required because, e.g., the current version of the process poses a safety hazard. Execute all process steps from (3) onward
  - **Important** – An adaptation would greatly improve the current version of the main process. Execute steps from (7) onward.
  - **Nice to have** – An improvement of the main process can be achieved but is not absolutely necessary >> Document the insight and leave for a future major version of the process.
- (3) Formulate a new software requirement to the main process based on the information provided by the new insight. This requirement is the result of an analysis by the process architect and other experts. The goal is to implement an *adaptation* for eliminating a risk or hazard in the current process without generating new risks and hazards.
- (4) Make an architectural design decision to enable the implementation of the adaptation in a new version of the concerned sub-process.
- (5) Implement and test the adaptation using a clone of the current version of the main process and its current state. The output of the cloned and adapted process is compared against the output of the original process to make sure that only what needs to be fixed by the adaptation has indeed changed in the output (this is also referred to as regression testing).
- (6) Apply the adaptation to the running main process by upgrading one of its sub-process to a new version. This mechanism is supported by the Camunda process engine.
- (7) Define a revision plan for the next major version of the process. This next major will contain the new version of the adapted sub-process. In addition to the quick patch applied in steps (3) – (6), the adaptation is now well documented and presented as a part of the main process.
- (8) Test and apply the revision and release a new major version of the process.

If the regression tests cannot find all latent software faults in the patched process, there is the option to spawn the adapted sub-process and let two version of if run in parallel. By checking the results of the two versions of the process using, for example, acceptance tests, we may still be able to detect data inconsistencies in the output and other errors introduced by the adaptation. In high assurance systems (e.g. airplanes), this concept is also known as *forward recovery* (See: Sha, L. (2001). Using simplicity to control complexity. IEEE Software, (4), 20-28. Download: http://www.citemaster.net/get/ed3895f4-0e2e-11e4-bbf9-00163e009cc7/s4020.pdf).

Steps (3) and (4) can be supported by a web-based tool called saFiddle, which we developed in our research group. The tool allows the fast and collaborative definition and documentation of software requirements and design decisions.

Related requirements:

- AR-02: Safety Assurance
- AR-07: Seamless Adaptation

Recommended styles / patterns:

- Publish-Subscribe Channel

ID

KDD-09: Process Recovery

Description

Forward Recovery of Processes in case of Failures

The root of software faults is complexity. However, cost reduction and optimization can only be achieved through process adaptation, which may lead to more complex processes. Because ad hoc software/process updates may introduce latent software faults, the system must have means to recover from possible failures caused by the activation of latent faults. While obvious errors are corrected early during development and testing phases of the project, subtler errors remain in the code and are more difficult to detect. In general, the more complex the process, the harder it is to make it reliable and safe. Also, there is always a finite amount of ressources to be spent on testing.

Safety/mission-critical processes must have means for recovery at runtime (i.e., to switch to a reliable process). To accomplish this, the adaptation process implements a forward recovery mechanism: *Sha, L. (2001). Using simplicity to control complexity. IEEE Software, (4), 20-28.*

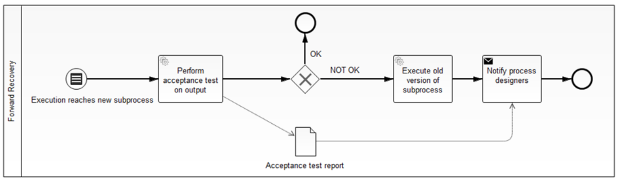
Related requirements:

- AR-02: Safety Assurance
- AR-06: Failure recovery

Recommended styles / patterns:

- None

The following diagram depicts the forward recovery process:


Forward Recovery
Execution reaches new subprocess
Perform acceptance test on output
OK
NOT OK
Execute old version of subprocess
Notify process designers
Acceptance test report

Process dynamics:

- Detect faulty behavior using an acceptance test or safe output region check
- Keep/derive an analytic relationship between the outputs of the core and adapted versions of the subprocesses, e.g.
    - DIFF: the modified process should only produce an output that is measurable and differentiable with respect to the old version of the process
- Camunda @runtime subprocess deployment mechanism

Forward Recovery

The forward recovery method works as follows:

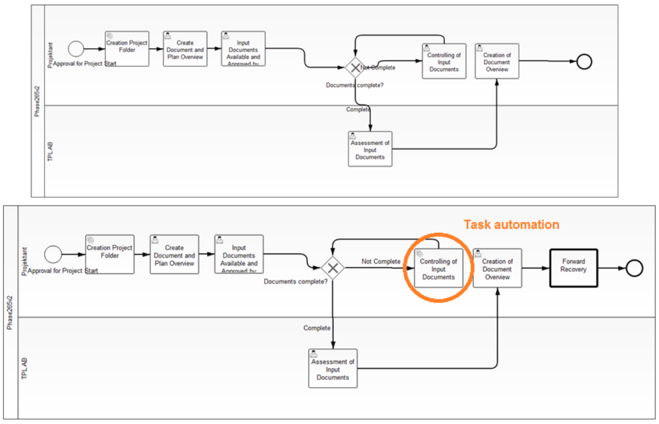
Prerequisite: The forward recovery method uses a simple and reliable core process that provides the essential service. This process is decomposed in smaller subprocesses, (idealy) such that each process deals with only one aspect of the workflow. This represents a decomposition of the process into adaptable subprocesses.

Tactic: Only one subprocess gets to be adapted at a time. The old version of the process denoted version_1 (core) is thereby replaced by an adapted version_2 of the subprocess being adapted. The core version_1 of the subprocess is kept in stand-by in case the adapted version fails.

Detection mechanism: An acceptance test checks the output of the adapted subprocess after every run using an acceptance test. The acceptance test should be able to detect any abnormalities w.r.t. the expected results of the adaptation.

Recovery: If such an abnormality is detected the acceptance test must fail and the system will automatically switch back to the core version_1 of the subprocess. The state of the process is recovered up to the point of the task that failed. The user is asked to redo that task (and any other tasks depending on it) as required in the version_1 of the subprocess.

The following diagram shows an example of a subprocess, which has been adapted. The adaptation consists of the automation of a task.


Phase265v2
Projektant
TPL AB
Approval for Project Start
Creation Project Folder
Create Document and Plan Overview
Input Documents Available and Approved by
Documents complete?
Not Complete
Complete
Controlling of Input Documents
Creation of Document Overview
Assessment of Input Documents
Task automation
Forward Recovery

ID

KDD-10: Adaptation Mechanisms

Description

How to Implement Adaptation in Camunda?

Lever 1: http://docs.camunda.org/latest/guides/user-guide/#process-engine-process-engine-concepts-suspending-process-instances

- Modify process variables while process is suspended

Lever 2: http://docs.camunda.org/latest/guides/user-guide/#process-engine-process-instance-modification-process-instance-modification-by-example

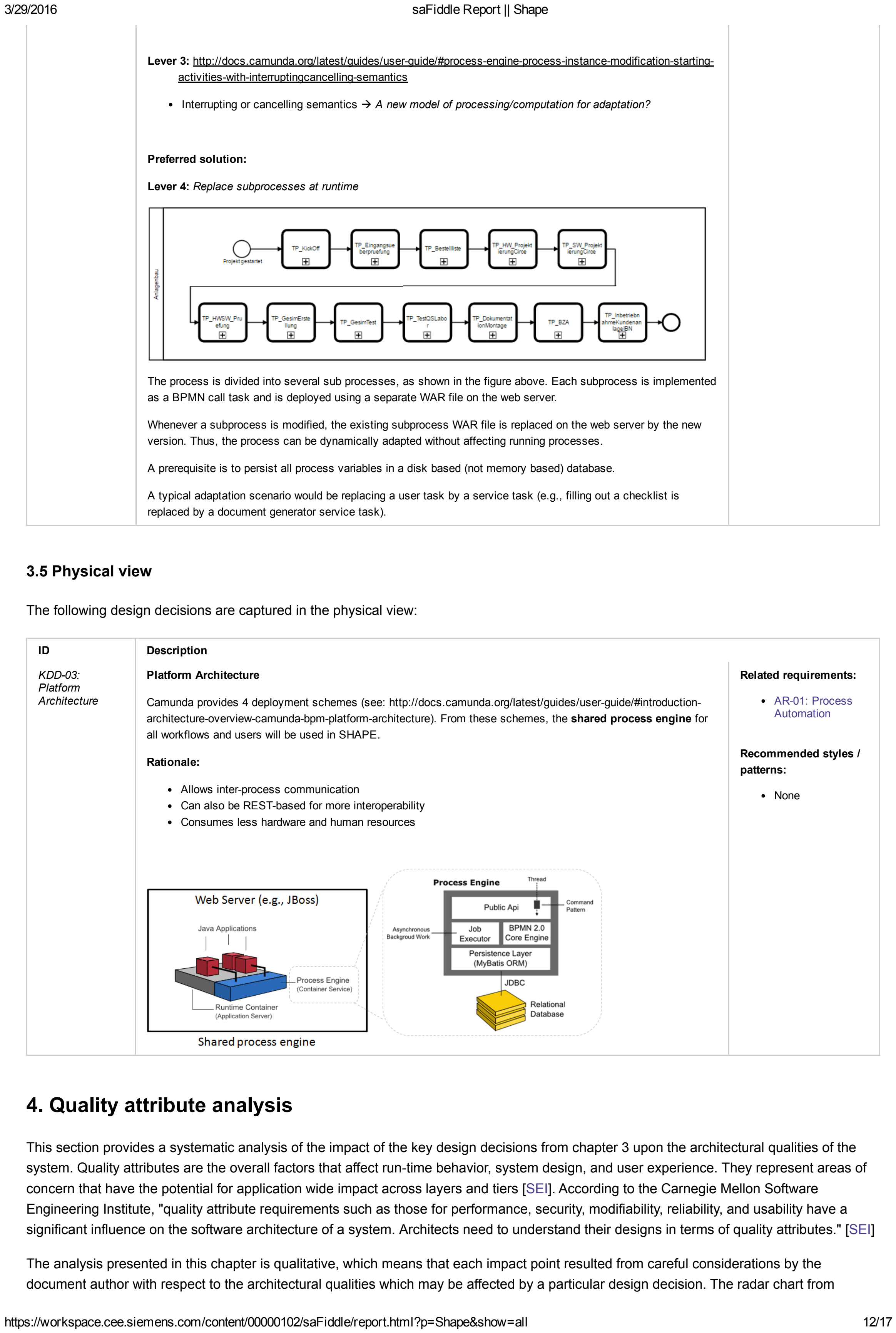
- AOP (Aspect-oriented programming)
- COP (Context oriented programming)

Related requirements:

- AR-07: Seamless Adaptation

Recommended styles / patterns:

- None



Lever 3: http://docs.camunda.org/latest/guides/user-guide/#process-engine-process-instance-modification-starting-activities-with-interruptingcancelling-semantics

- Interrupting or cancelling semantics → *A new model of processing/computation for adaptation?*

Preferred solution:

Lever 4: *Replace subprocesses at runtime*

The process is divided into several sub processes, as shown in the figure above. Each subprocess is implemented as a BPMN call task and is deployed using a separate WAR file on the web server.

Whenever a subprocess is modified, the existing subprocess WAR file is replaced on the web server by the new version. Thus, the process can be dynamically adapted without affecting running processes.

A prerequisite is to persist all process variables in a disk based (not memory based) database.

A typical adaptation scenario would be replacing a user task by a service task (e.g., filling out a checklist is replaced by a document generator service task).

3.5 Physical view

The following design decisions are captured in the physical view:

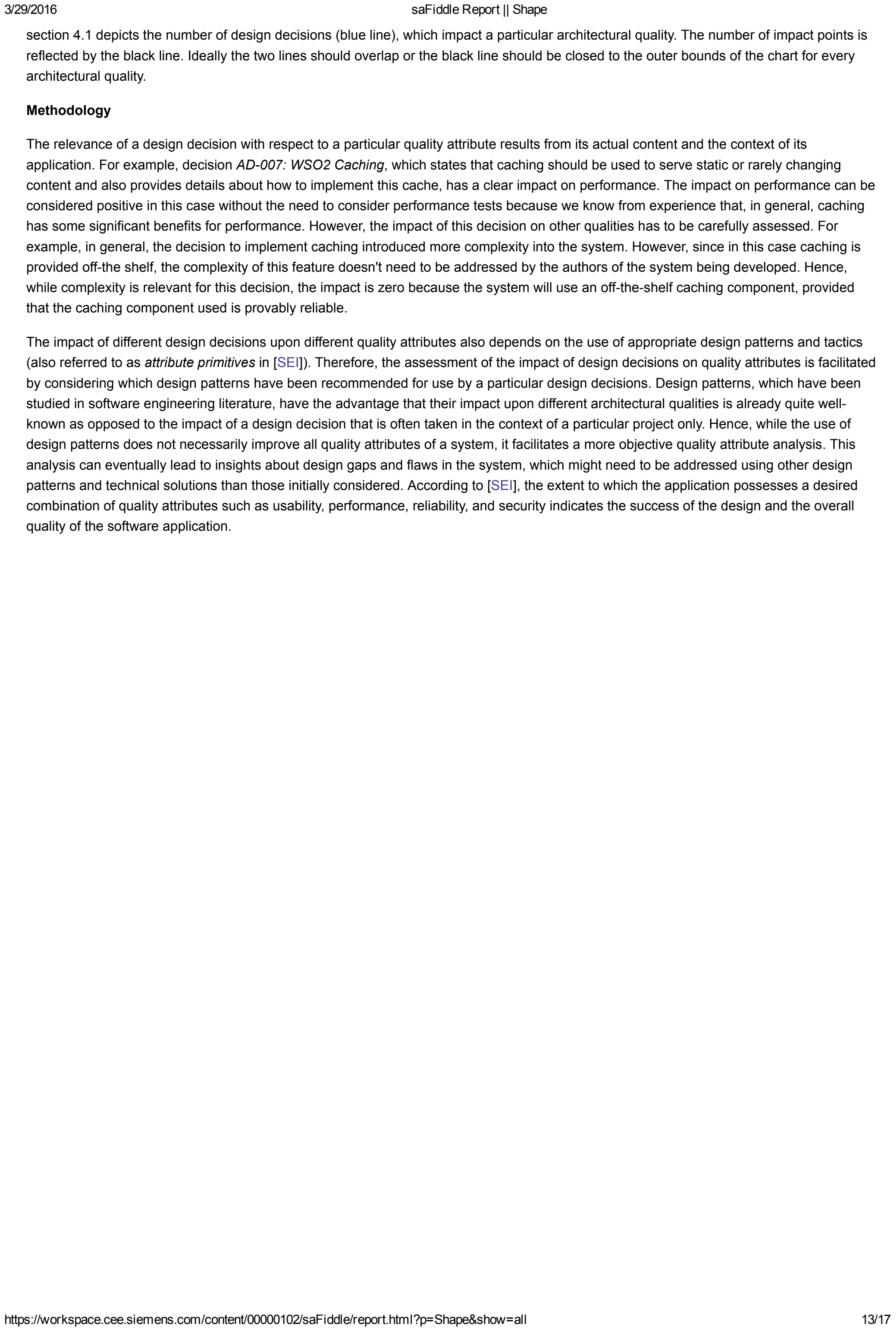
ID	Description	
KDD-03: Platform Architecture	**Platform Architecture** Camunda provides 4 deployment schemes (see: http://docs.camunda.org/latest/guides/user-guide/#introduction-architecture-overview-camunda-bpm-platform-architecture). From these schemes, the **shared process engine** for all workflows and users will be used in SHAPE. **Rationale:** • Allows inter-process communication • Can also be REST-based for more interoperability • Consumes less hardware and human resources Shared process engine	**Related requirements:** • AR-01: Process Automation **Recommended styles / patterns:** • None

4. Quality attribute analysis

This section provides a systematic analysis of the impact of the key design decisions from chapter 3 upon the architectural qualities of the system. Quality attributes are the overall factors that affect run-time behavior, system design, and user experience. They represent areas of concern that have the potential for application wide impact across layers and tiers [SEI]. According to the Carnegie Mellon Software Engineering Institute, "quality attribute requirements such as those for performance, security, modifiability, reliability, and usability have a significant influence on the software architecture of a system. Architects need to understand their designs in terms of quality attributes." [SEI]

The analysis presented in this chapter is qualitative, which means that each impact point resulted from careful considerations by the document author with respect to the architectural qualities which may be affected by a particular design decision. The radar chart from

22 Private Document

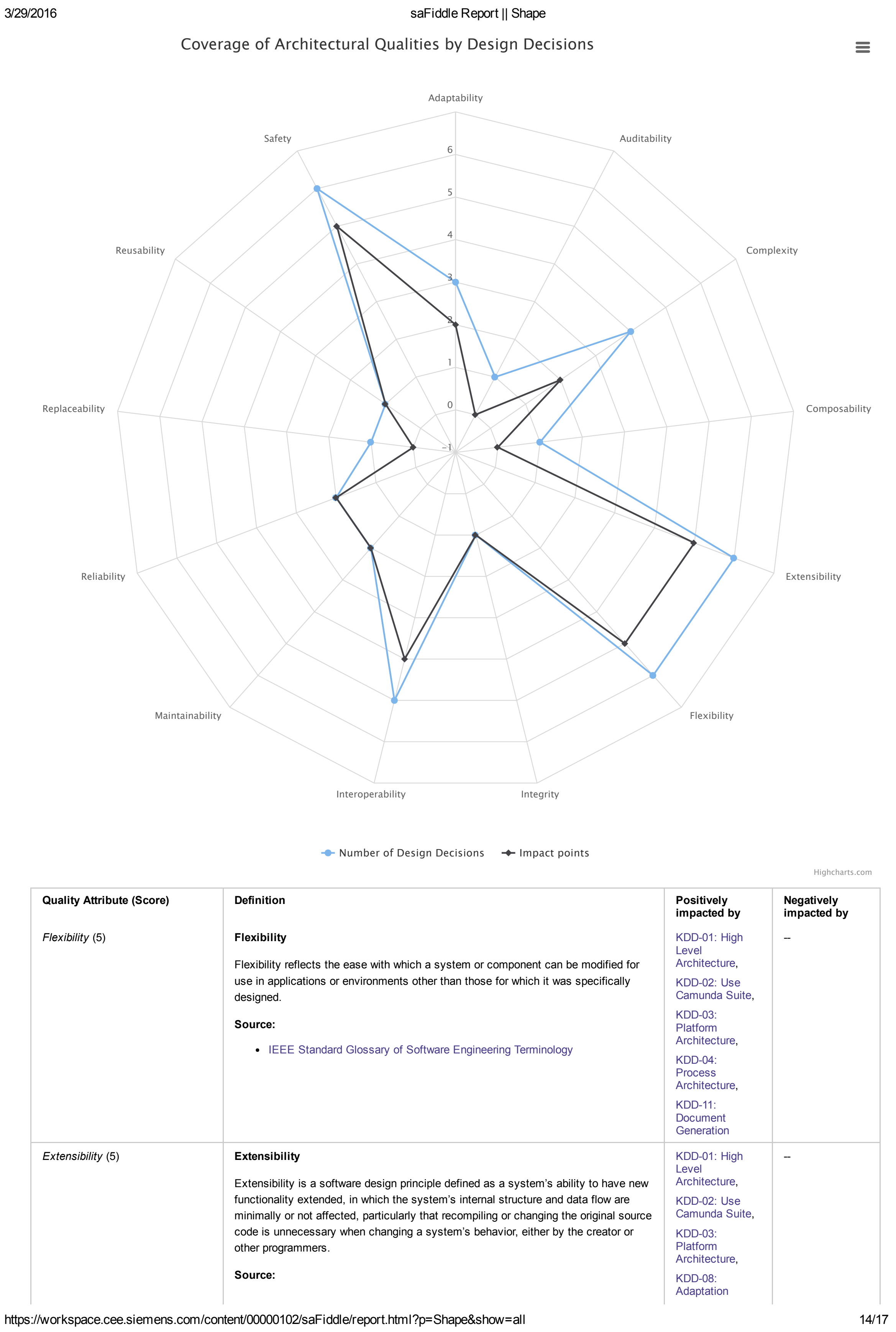


section 4.1 depicts the number of design decisions (blue line), which impact a particular architectural quality. The number of impact points is reflected by the black line. Ideally the two lines should overlap or the black line should be closed to the outer bounds of the chart for every architectural quality.

Methodology

The relevance of a design decision with respect to a particular quality attribute results from its actual content and the context of its application. For example, decision *AD-007: WSO2 Caching*, which states that caching should be used to serve static or rarely changing content and also provides details about how to implement this cache, has a clear impact on performance. The impact on performance can be considered positive in this case without the need to consider performance tests because we know from experience that, in general, caching has some significant benefits for performance. However, the impact of this decision on other qualities has to be carefully assessed. For example, in general, the decision to implement caching introduced more complexity into the system. However, since in this case caching is provided off-the shelf, the complexity of this feature doesn't need to be addressed by the authors of the system being developed. Hence, while complexity is relevant for this decision, the impact is zero because the system will use an off-the-shelf caching component, provided that the caching component used is provably reliable.

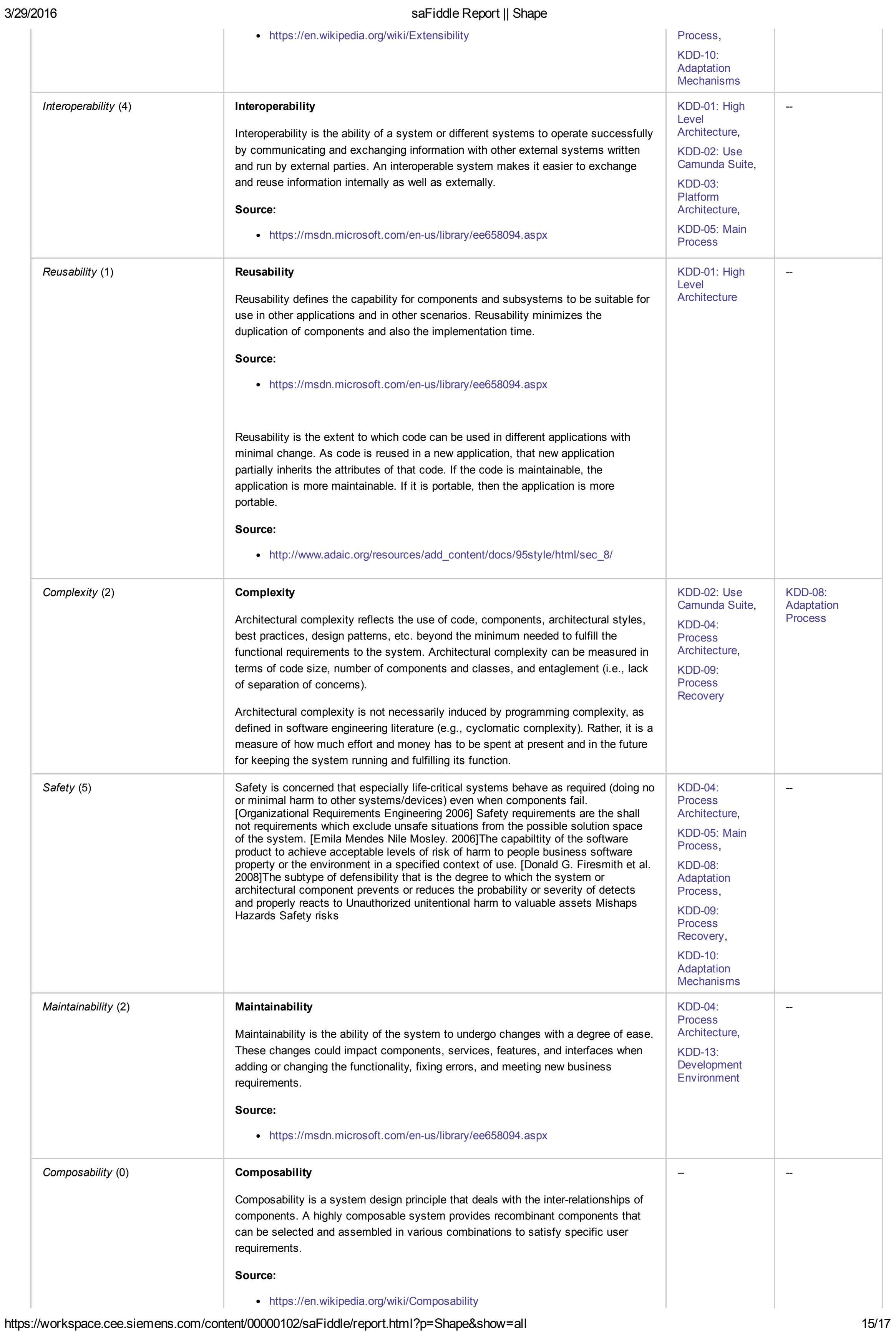
The impact of different design decisions upon different quality attributes also depends on the use of appropriate design patterns and tactics (also referred to as *attribute primitives* in [SEI]). Therefore, the assessment of the impact of design decisions on quality attributes is facilitated by considering which design patterns have been recommended for use by a particular design decisions. Design patterns, which have been studied in software engineering literature, have the advantage that their impact upon different architectural qualities is already quite well-known as opposed to the impact of a design decision that is often taken in the context of a particular project only. Hence, while the use of design patterns does not necessarily improve all quality attributes of a system, it facilitates a more objective quality attribute analysis. This analysis can eventually lead to insights about design gaps and flaws in the system, which might need to be addressed using other design patterns and technical solutions than those initially considered. According to [SEI], the extent to which the application possesses a desired combination of quality attributes such as usability, performance, reliability, and security indicates the success of the design and the overall quality of the software application.



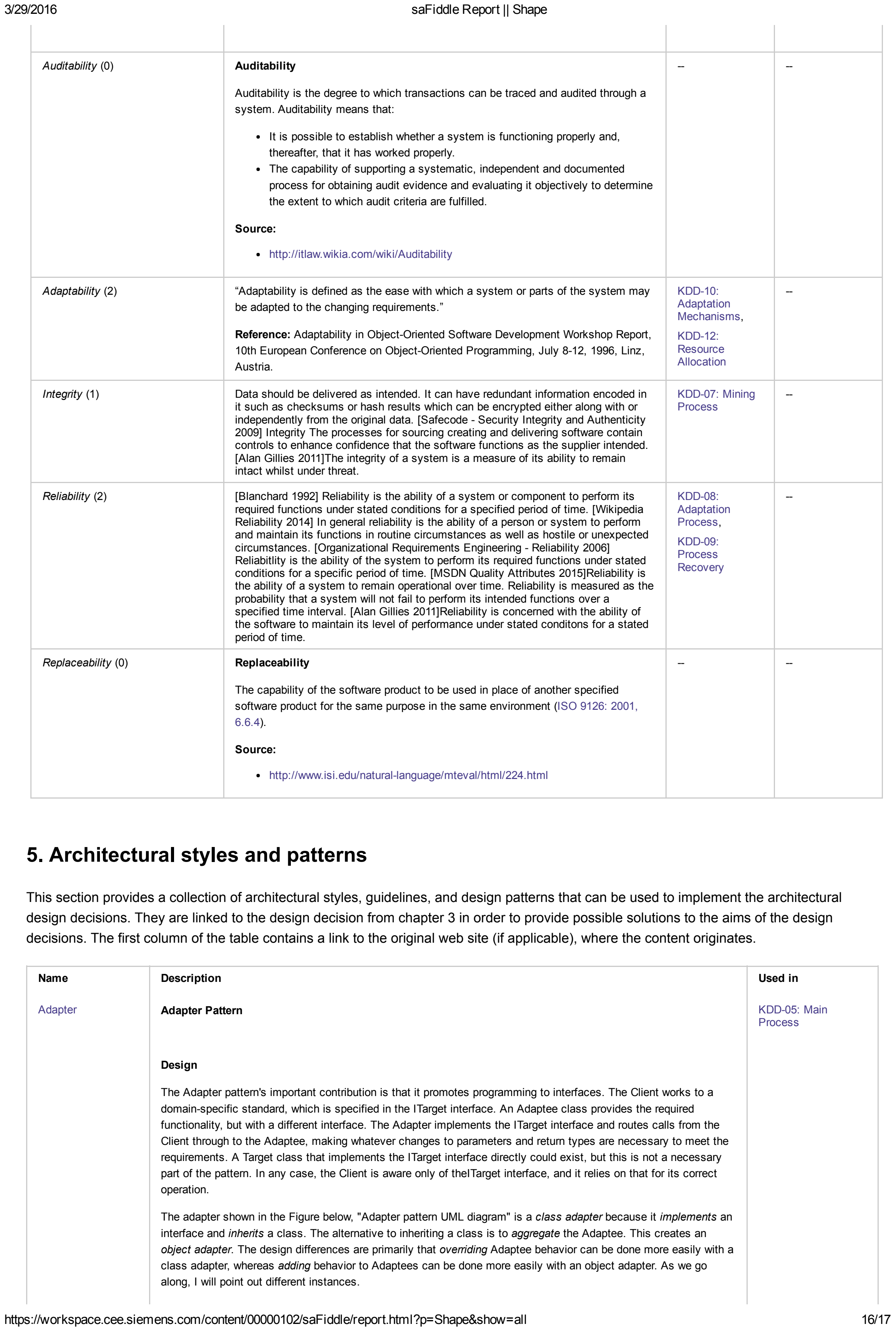
Coverage of Architectural Qualities by Design Decisions

Quality Attribute (Score)	Definition	Positively impacted by	Negatively impacted by
Flexibility (5)	**Flexibility** Flexibility reflects the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed. **Source:** - IEEE Standard Glossary of Software Engineering Terminology	KDD-01: High Level Architecture, KDD-02: Use Camunda Suite, KDD-03: Platform Architecture, KDD-04: Process Architecture, KDD-11: Document Generation	--
Extensibility (5)	**Extensibility** Extensibility is a software design principle defined as a system's ability to have new functionality extended, in which the system's internal structure and data flow are minimally or not affected, particularly that recompiling or changing the original source code is unnecessary when changing a system's behavior, either by the creator or other programmers. **Source:**	KDD-01: High Level Architecture, KDD-02: Use Camunda Suite, KDD-03: Platform Architecture, KDD-08: Adaptation	--

24 Private Document



	- https://en.wikipedia.org/wiki/Extensibility	Process, KDD-10: Adaptation Mechanisms	
Interoperability (4)	**Interoperability** Interoperability is the ability of a system or different systems to operate successfully by communicating and exchanging information with other external systems written and run by external parties. An interoperable system makes it easier to exchange and reuse information internally as well as externally. **Source:** - https://msdn.microsoft.com/en-us/library/ee658094.aspx	KDD-01: High Level Architecture, KDD-02: Use Camunda Suite, KDD-03: Platform Architecture, KDD-05: Main Process	–
Reusability (1)	**Reusability** Reusability defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios. Reusability minimizes the duplication of components and also the implementation time. **Source:** - https://msdn.microsoft.com/en-us/library/ee658094.aspx Reusability is the extent to which code can be used in different applications with minimal change. As code is reused in a new application, that new application partially inherits the attributes of that code. If the code is maintainable, the application is more maintainable. If it is portable, then the application is more portable. **Source:** - http://www.adaic.org/resources/add_content/docs/95style/html/sec_8/	KDD-01: High Level Architecture	–
Complexity (2)	**Complexity** Architectural complexity reflects the use of code, components, architectural styles, best practices, design patterns, etc. beyond the minimum needed to fulfill the functional requirements to the system. Architectural complexity can be measured in terms of code size, number of components and classes, and entaglement (i.e., lack of separation of concerns). Architectural complexity is not necessarily induced by programming complexity, as defined in software engineering literature (e.g., cyclomatic complexity). Rather, it is a measure of how much effort and money has to be spent at present and in the future for keeping the system running and fulfilling its function.	KDD-02: Use Camunda Suite, KDD-04: Process Architecture, KDD-09: Process Recovery	KDD-08: Adaptation Process
Safety (5)	Safety is concerned that especially life-critical systems behave as required (doing no or minimal harm to other systems/devices) even when components fail. [Organizational Requirements Engineering 2006] Safety requirements are the shall not requirements which exclude unsafe situations from the possible solution space of the system. [Emila Mendes Nile Mosley. 2006]The capabiltity of the software product to achieve acceptable levels of risk of harm to people business software property or the environment in a specified context of use. [Donald G. Firesmith et al. 2008]The subtype of defensibility that is the degree to which the system or architectural component prevents or reduces the probability or severity of detects and properly reacts to Unauthorized unitentional harm to valuable assets Mishaps Hazards Safety risks	KDD-04: Process Architecture, KDD-05: Main Process, KDD-08: Adaptation Process, KDD-09: Process Recovery, KDD-10: Adaptation Mechanisms	–
Maintainability (2)	**Maintainability** Maintainability is the ability of the system to undergo changes with a degree of ease. These changes could impact components, services, features, and interfaces when adding or changing the functionality, fixing errors, and meeting new business requirements. **Source:** - https://msdn.microsoft.com/en-us/library/ee658094.aspx	KDD-04: Process Architecture, KDD-13: Development Environment	–
Composability (0)	**Composability** Composability is a system design principle that deals with the inter-relationships of components. A highly composable system provides recombinant components that can be selected and assembled in various combinations to satisfy specific user requirements. **Source:** - https://en.wikipedia.org/wiki/Composability	–	–



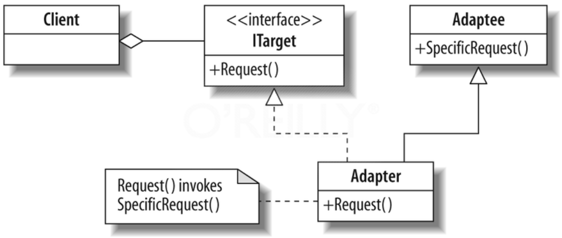
Auditability (0)	**Auditability** Auditability is the degree to which transactions can be traced and audited through a system. Auditability means that: - It is possible to establish whether a system is functioning properly and, thereafter, that it has worked properly. - The capability of supporting a systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which audit criteria are fulfilled. **Source:** - http://itlaw.wikia.com/wiki/Auditability	--	--
Adaptability (2)	“Adaptability is defined as the ease with which a system or parts of the system may be adapted to the changing requirements.” **Reference:** Adaptability in Object-Oriented Software Development Workshop Report, 10th European Conference on Object-Oriented Programming, July 8-12, 1996, Linz, Austria.	KDD-10: Adaptation Mechanisms, KDD-12: Resource Allocation	--
Integrity (1)	Data should be delivered as intended. It can have redundant information encoded in it such as checksums or hash results which can be encrypted either along with or independently from the original data. [Safecode - Security Integrity and Authenticity 2009] Integrity The processes for sourcing creating and delivering software contain controls to enhance confidence that the software functions as the supplier intended. [Alan Gillies 2011]The integrity of a system is a measure of its ability to remain intact whilst under threat.	KDD-07: Mining Process	--
Reliability (2)	[Blanchard 1992] Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time. [Wikipedia Reliability 2014] In general reliability is the ability of a person or system to perform and maintain its functions in routine circumstances as well as hostile or unexpected circumstances. [Organizational Requirements Engineering - Reliability 2006] Reliabitlity is the ability of the system to perform its required functions under stated conditions for a specific period of time. [MSDN Quality Attributes 2015]Reliability is the ability of a system to remain operational over time. Reliability is measured as the probability that a system will not fail to perform its intended functions over a specified time interval. [Alan Gillies 2011]Reliability is concerned with the ability of the software to maintain its level of performance under stated conditons for a stated period of time.	KDD-08: Adaptation Process, KDD-09: Process Recovery	--
Replaceability (0)	**Replaceability** The capability of the software product to be used in place of another specified software product for the same purpose in the same environment (ISO 9126: 2001, 6.6.4). **Source:** - http://www.isi.edu/natural-language/mteval/html/224.html	--	--

5. Architectural styles and patterns

This section provides a collection of architectural styles, guidelines, and design patterns that can be used to implement the architectural design decisions. They are linked to the design decision from chapter 3 in order to provide possible solutions to the aims of the design decisions. The first column of the table contains a link to the original web site (if applicable), where the content originates.

Name	Description	Used in
Adapter	**Adapter Pattern** **Design** The Adapter pattern's important contribution is that it promotes programming to interfaces. The Client works to a domain-specific standard, which is specified in the ITarget interface. An Adaptee class provides the required functionality, but with a different interface. The Adapter implements the ITarget interface and routes calls from the Client through to the Adaptee, making whatever changes to parameters and return types are necessary to meet the requirements. A Target class that implements the ITarget interface directly could exist, but this is not a necessary part of the pattern. In any case, the Client is aware only of theITarget interface, and it relies on that for its correct operation. The adapter shown in the Figure below, "Adapter pattern UML diagram" is a *class adapter* because it *implements* an interface and *inherits* a class. The alternative to inheriting a class is to *aggregate* the Adaptee. This creates an *object adapter.* The design differences are primarily that *overriding* Adaptee behavior can be done more easily with a class adapter, whereas *adding* behavior to Adaptees can be done more easily with an object adapter. As we go along, I will point out different instances.	KDD-05: Main Process

Adapter pattern UML diagram



The purpose of the ITarget interface is to enable objects of adaptee types to be interchangeable with any other objects that might implement the same interface. However, the adaptees might not conform to the operation names and signatures of ITarget, so an interface alone is not a sufficiently powerful mechanism. That is why we need the Adapter pattern. An Adaptee offers similar functionality to Request, but under a different name and with possibly different parameters. The Adaptee is completely independent of the other classes and is oblivious to any naming conventions or signatures that they have. Now, let's consider the roles in the pattern:

ITarget

The interface that the Client wants to use

Adaptee

An implementation that needs adapting

Adapter

The class that implements the ITarget interface in terms of the Adaptee

Request

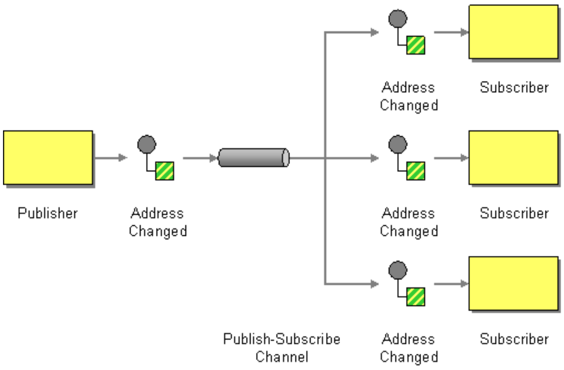
An operation that the Client wants

SpecificRequest

The implementation of Request's functionality in the Adaptee

Publish-Subscribe Channel

How can the sender broadcast an event to all interested receivers?



Send the event on a *Publish-Subscribe Channel*, which delivers a copy of a particular event to each receiver.

A *Publish-Subscribe Channel* works like this: It has one input channel that splits into multiple output channels, one for each subscriber. When an event is published into the channel, the *Publish-Subscribe Channel* delivers a copy of the message to each of the output channels. Each output channel has only one subscriber, which is only allowed to consume a message once. In this way, each subscriber only gets the message once and consumed copies disappear from their channels.

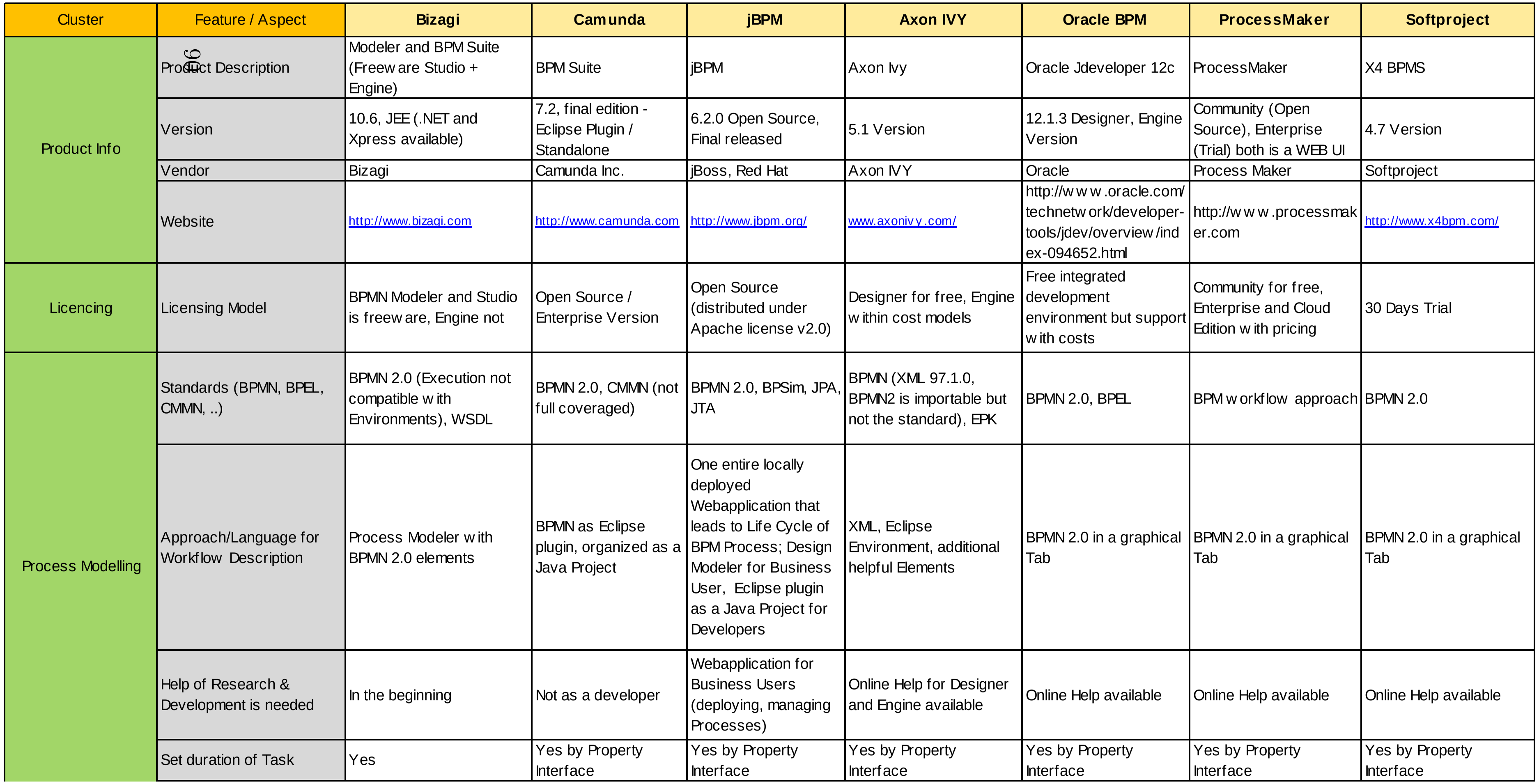
Source:

- http://www.enterpriseintegrationpatterns.com/patterns/messaging/PublishSubscribeChannel.html#1428741072

KDD-08: Adaptation Process

6 Appendix B: BPMS Tool Evaluations

The following evaluation table is taken from the Master's Thesis of Alexander Wurl [3], chapter "Appendix A - The Evaluation of BPMS".



Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Product Info	Product Description	Modeler and BPM Suite (Freeware Studio + Engine)	BPM Suite	jBPM	Axon Ivy	Oracle Jdeveloper 12c	ProcessMaker	X4 BPMS
	Version	10.6, JEE (.NET and Xpress available)	7.2, final edition - Eclipse Plugin / Standalone	6.2.0 Open Source, Final released	5.1 Version	12.1.3 Designer, Engine Version	Community (Open Source), Enterprise (Trial) both is a WEB UI	4.7 Version
	Vendor	Bizagi	Camunda Inc.	jBoss, Red Hat	Axon IVY	Oracle	Process Maker	Softproject
	Website	http://www.bizagi.com	http://www.camunda.com	http://www.jbpm.org/	www.axonivy.com/	http://www.oracle.com/technetwork/developer-tools/jdev/overview/index-094652.html	http://www.processmaker.com	http://www.x4bpm.com/
Licencing	Licensing Model	BPMN Modeler and Studio is freeware, Engine not	Open Source / Enterprise Version	Open Source (distributed under Apache license v2.0)	Designer for free, Engine within cost models	Free integrated development environment but support with costs	Community for free, Enterprise and Cloud Edition with pricing	30 Days Trial
Process Modelling	Standards (BPMN, BPEL, CMMN, ..)	BPMN 2.0 (Execution not compatible with Environments), WSDL	BPMN 2.0, CMMN (not full coveraged)	BPMN 2.0, BPSim, JPA, JTA	BPMN (XML 97.1.0, BPMN2 is importable but not the standard), EPK	BPMN 2.0, BPEL	BPM workflow approach	BPMN 2.0
	Approach/Language for Workflow Description	Process Modeler with BPMN 2.0 elements	BPMN as Eclipse plugin, organized as a Java Project	One entire locally deployed Webapplication that leads to Life Cycle of BPM Process; Design Modeler for Business User, Eclipse plugin as a Java Project for Developers	XML, Eclipse Environment, additional helpful Elements	BPMN 2.0 in a graphical Tab	BPMN 2.0 in a graphical Tab	BPMN 2.0 in a graphical Tab
	Help of Research & Development is needed	In the beginning	Not as a developer	Webapplication for Business Users (deploying, managing Processes)	Online Help for Designer and Engine available	Online Help available	Online Help available	Online Help available
	Set duration of Task	Yes	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Modelling	Model is Importable	XML (without graphical positioning), XPDL, Visio	BPMN 2.0 (graphical positioning included	BPMN 2.0 (graphical positioning included	Other versions or plain BPMN2 XML files might be imported anyway, but they are not fully supported (e.g. positions of tasks).	BPMN 2.0 (graphical positioning included	BPMN 2.0 (graphical positioning included	Through BPMN 2.0. Process models from tools such as GBtec, MID Innovator, Signavio or Aris can be read with the BPM Designer and processed further. Thus, previously static process descriptions can be technically implemented after being imported.
	Model is exportable	XML (with Bizagi-Tags), XSD, XPDL, BPMN 2.0 (with bizagi tags), Visio (loss of positioning information), Image, Publishing to Web, Sharepoint and Wiki	XML, BPMN 2.0 and project (.war, .jar, .ear)	XML, BPMN 2.0 and project (.war, .jar)	The internal version in the exported XML file is 97.1.0.	XML, BPMN 2.0 and project (.war, .jar)	BPMN 2.0	BPMN 2.0, projects, PDF
	Support for easy & quick modelling	yes, Microsoft Interface	Eclipse environment	Within jBPM or Eclipse environment	Axon.ivy is based on the Eclipse platform. So when you start Axon.ivy Designer you launch an Eclipse workbench.	Similar to Eclipse but own Environment	Quick modelling in a Webbrowser	Easy handling
	Definition of procedural Facts	BPMN 2.0 and internal property configuration	BPMN 2.0, properties are set in XML form	BPMN 2.0, properties are set in XML form	BPMN (XML 97.1.0)	BPMN 2.0, properties are set in XML form	BPM workflow definitions	BPMN 2.0, properties are set in XML form

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Modelling	Definition of organizaional Rules	Internal Property configuration in the tool	BPMN 2.0, properties are set in XML form; Constraints, e.g. RAL	BPMN 2.0, properties are set in XML form	BPMN (XML 97.1.0)	Busines Rules are an extra Category but within the BPMN	Departments can form hierachic trees. Each department has to name responsible person who can not be part of another department. ProcessMaker has a simple to use, embedded business rules engine which drives the logic behind the process. At each decision gateway, business analysts can build in logic to determine how a process should flow down one particular path instead of another in the business	Business rules are defined graphically or by rule description and executed by the integrated rules engine.
	Model is validated	through internal validation checker, making sure the process passes through all the sequence flows	Depends on the editor: Eclipse Plugin does not support a validation check, Signavio does.	Validation Button in the Designer with visible corrections, Processes can also be checked manually with the RuleFlow Process Class	No Validation in times of designing but afterwards in the simulation of the process (is not the engine!)	Automatic validation and simulation	Advanced Debugger (Enterprise Version)	The automated testing of structured data such as EDIFACT messages, invoices, contracts or other documents is possible without any difficulty. Graphical process debugging; Further testing tools to validate processes against test schemes
Process Realization	Exceptions designable	via property configuration	By BPMN 2.0 Error	By BPMN 2.0 Error	By BPMN Error Event	By BPMN 2.0 Error	BPM elements	By BPMN 2.0 Error
	Define Notification on Process Events	yes (e.g. Email)	Implementation of a Task Listener required	Yes (e.g. Email)	User mail notification can be configured on the Axon.ivy Engine in the Engine Administration UI or in the Workflow UI applications	Email, User, SMS, Voice, IM	Yes (e.g. Email)	Yes (e.g. Email)
	Illustrate Data structure and authorization	via internal property configuration	Via BPMN 2.0	Via BPMN 2.0	Via BPMN style			Via BPMN 2.0

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Realization	Support for easy/quick Programming	Programming in sense of Editing Expressions, nothing else. No Java etc. supported	Based on Java EE	Based on Java EE	In the Designer, projects may only exist in one version at a given point of time. Projects are created and organized inside an Eclipse workspace. On the Designer, the workspace corresponds to the application on the engine what makes it easy to switch for programming.	Based on Java EE	Interface for the definition of business rules but it's necessary to complete a formular (DynaForm): fields of formular are defining the application of the business (conditions). Once definced, it is to be integrated in the BP like a trigger or derivated rule of the BPs.	Based on Java EE with a easy switch between Design phase and Implementation
	Definition of Monitoring Design	WEB UI set by Bizagi Configurations	WEB UI deployed from a Java Project. Pre-defined design, but extensible	Web UI leads through BPM Life Cycle: Project explorer with Authoring, Data Modeler for including external Data, Form Modeler creating Masks for User, Dashboard for out of the box report for process instances, e.g. total amount of order that come into the system	WEB UI deployed from a Java Project. Pre-defined design, but extensible	WEB UI deployed from a Maven project. Predefined design, but extensible	WEB UI with a pre-defined design	The X4 ControlCenter provides functions for testing, monitoring, evaluating and documenting business processes. Also service level agreements deposited within the process model can be monitored in real-time during the process execution.
	Export of Process Data	XML with Bizagi-Tags	Via BPMN 2.0	Via BPMN 2.0,	XML, Axon Ivy Archive (.iar)	Export whole project with BPMN 2.0 files included	Via XPDL	Via BPMN 2.0
	Adapters to connect to External System	ERP, CRM, Legacy	Spring framework etc.		Enterprise Service Bus	File Adapter, Java API	No	SAP, MS-Sharepoint, MongoDB, ..
	Quality Assessment of Implementation	validation of processes; simulation shows if Gateways are synchronized, Messages are sychronized, Decisions probabilities are correctly assigned, Routing behaves as expected, all tokens have ended	There is no validation except from plugins, Unit Test in the Maven environment	Unit testing in the Maven environment	Testing via simulation. Test roles can be defined (they are not equal to the real roles).	Beside Validation and Simulation an Extension for JUnit Tests can be downloaded	JavaScript validates input fields	through validation and testing

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Integration of Systems	Interaction with external Data sources	A integration layer enables external Applications (API, Connectors) and external Systemes (ERP, CRM, Legacy). Possible for .NET and JEE. For JEE components are registered	Java Query API, REST Query API, Native Queries (own SQL), Custom Queries, SQL Queries; Distribution for Tomcat, JBoss and Glassfish. In Enterprise version: IBM Websphere AS, Oracle WLS, XSLT Template Engine.	In the Web UI; Form Modeller, Data Modeler: jar., war. etc. uploadable	Sharepoint Add-On over Web Services, access to all functions of Sharepoinot and vice versa, SAP	SQL databases can be configured	The Business Rules Enginer, Advanced Dashboards, Custom Case List Builder, Case Archive/Restore Case Archive \& Restore, SLA Manager Service Level Agreements Manager, Simple Reporting, Advanced LDAP/AD Sync, Batch Routing from Inbox, Advanced Performance Monitor Dashboards, ProcessMaker Monitor & Multi-Tenancy Management, and Microsoft Outlook Microsoft Outlook Connector	The X4 Enterprise Service Bus (X4 ESB) connects IT systems and provides services for a service-oriented architecture (SOA). It enables synchronous and asynchronous processing, messaging, the integration of distributed resources or the transactional processing. Due to its extreme scalability, the X4 Enterprise Service Bus can handle very large amounts of data and meets the highest demands in terms of availability.
	Interaction with Infrastructure Systems e.g. User administratrion, HR-Systems etc.	HR processes are offered within the tool: Offborading, Onboarding, Recruitment	Through API, IdentityService which allows access to the management of user and groups can also be used in conjunction with services such as LDAP.	Pluggable human task service based on WS-HumanTask	Defining Roles, Import and Export Roles in XML-Definition	Extra Organization Overview that also can fix holiday etc	LDAP	Human workflows with user interactions and task management with X4 Activities - also available as mobile version

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Integration of Systems	Interaction with external Applications	SOA layer is implemented to provide the processes funtionality for the ESB (Enterprise Server Bus)	Java API; From the ProcessEngine, you can obtain the various services that contain the workflow/BPM methods. ProcessEngine and the services objects are thread safe. So you can keep a reference to 1 of those for a whole server. Runtime Container Integration; an approach offers access to the Process Engine Service. Available connector for calling web services (SOAP and REST) from the process, languages (XML, JSON, etc.), combination with ESB.	Data Modeller (Drools is a Business Rules Management System (BRMS) solution) is responsible for including applications, Java Clients with remote URL, WS, REST, jBPM Executor allows executing jobs	Enterprise Service Bus provides Routing of messages, transformations of protocols, connections scalability of platforms, (in)dependend deployment, Enterprise Integration Patterns, etc. Call element is waiting for external Events. Interfaces allow access to Java Bean. DROOLS, ESB, REST	Either via Java API or BPMN Data Sources	Via REST API, OAuth 2.0 for authorization	Via Enterprise Service Bus. Also to Web 2.0 (Facebook, Twitter)
Process Execution	Correct Execution in Person's abscence / presence	Before launching, User Interface is defined. A security module allows to define a schema of permissions. Correct handling of defined process operations if individual people are not available.	Since the project is deployed and then started it runs along the defined process.	Since the project is deployed and then started it runs along the defined task-todos.	Since the project was tested through the simulation it then will be deployed and started.	Since the project was tested through the simulation it then will be deployed and started.	Since the project was started it runs on the server	Since the project was tested through the simulation it then will be deployed and started.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Execution	Information and Help for End-User	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	Web UI guided by RuntimeManager that involves KieSession and TaskService, manages easily Session managment strategies (Singleton session, Session per request, Session per process instance), Deployment, REST, JMS, Java Client	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help
	Organisation's changes during Runtime	as far as concerning changes for in the WEB UI, they are pre-defined	Changable in Web UI during Runtime, e.g. as Admin	Changable in Web UI during Runtime, e.g. as Admin	In the Role Editor	Changable in Web UI during Runtime, e.g. as Admin	In terms of delegation of tasks	Changable in Web UI during Runtime, e.g. as Admin
	Correct Execution of defined Processes	assessment through validation	Testing process definition, process application, your application with other deployments of services, End-to-end integration test including all external systems; Unit Testing, using Mocks, using Arquillian	Parallelism, Symmetric (Load balancing, High Availability), Timer Service using Quartz, Inject services into your Context Dependency Injection environment	The simulation of the process gurantees for the correct execution since there are no errors detected	The Validation and Simulation gurantees a correct execution	ProcessMaker provides DynaForms, or "Dynamic Forms", which are the familiar forms, that can be designed in ProcessMaker to interface with the user while running a case	The automated testing of structured data such as EDIFACT messages, invoices, contracts or other documents is possible without any difficulty.
	Possibilities to start Process Instances	One Run-Button to execute	The process engine is a java library responsible for executing BPMN 2.0 processes and workflows. It is deployed, it is accessible via WEB UI where the process can be startet.	The RunTime Manager will start through Deployment of the REST UI	The program start element allows to start a process by a trigger from embedded external Java code. This opens a possibility to integrate an Axon.ivy application into other applications and systems.	Deploy the Maven Project, run .jar	Entry Task is fixed on Entry form	The RunTime Manager will start through Deployment of the Web UI

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Execution	Interdependency of Processes	Constraints in security module allow a schema of permissions.	process engine supports multi-tenancy: Users should choose the model which fits their data separation needs.	JAXB + JSON is responsible for passing important custom data back and forward to the execution server.	The simulation should notify processes that may be erroneous to each other.	The simulation should notify processes that may be erroneous to each other.		
	Possbilities to access from different channels/devices	Yes (e.g. Login, also on mobile device)	Any Webbrowser	Any Webbrowser	Any Webbrowser, CMS	Any Webbrowser, CMS	Any Webbrowser except Internet Explorer, mobile devices	Any Webbrowser, CMS
Resource Allocation / Scheduling	How human resources are represented	Allocations are done based on user properties which are defined int the Organization Module. Likewise permissions to create and administrate a process and its components are granted according those properties.	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. In the TaskManager User profiles can be created within an Administration account who owns access rights	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. In the Administration field User profiles can be created including the assignment to groups.	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. External Systems like LDAP AD, Novell map roles.	Organization Editor archives human resources	In the User Management. Inside: create roles, groups and departments.	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. In the ControlCenter human resources are listed

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Internal representation of organizational Models	An organization in Bizagi stores the information related not only to the organizational structure of the members of a company and the definition of their characteristics (position, areas,groups), but also the characteristics that make them unique on a team and allow them to be active members in the Processes of the application or applications (roles, skills, geographic location).	There is no hierachy. In the admin area User and Groups can be defined. User can be added to Groups which can also be seen as Role. In the Admin area permissions on each User and Group can be defined. RAL (Resource Assignment Language) can be used to define organizational constraints.	In the BPMN2 Modeler, the Diagram Wizard helps you decide how to document your process-driven organizational models. Organizational model includes naming of company and assigning a repository to it. There is no hierarchy for humans within.	Organizations are represented in through the BPMN notation via the Pool with its lanes. DROOLS BRMS is already integrated.	Organization Editor enables to create and edit the components within an organization. Organizations are composed of roles, organizational chart, holidays, calendars. Organizations are defined at the project level. You can export organizational information to be used within other projects. You cannot create organizational charts, calendars, or holidays using Business Process Composer. You can define roles and assign them to swimlanes. Organization information is not carried over when a project is deployed to runtime.	Departments can be structured in a hierarchical tree. It cannot be exported.	The user, groups, roles and rights is administrated by the Control Center by the administrator.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Administration and Execution of individual Tasks in the System	Work allocation; Bizagi automatically evaluates the allocation rules defined for each Task and selects one or more users that meet the given conditions from the user's list. Guided by a wizard, allocation tool includes load optimization algorithms and deals with delegates and working calendars.	User and Groups can be assigned to tasks. In the administration, authorization of User and Groups can be defined which may play a role on executing a tasks. Spring integration has a special feature for deploying resources. In the process engine configuration, you can specify a set of resources. When the process engine is created, all those resources will be scanned and deployed. There is filtering in place that prevents duplicate deployments. Only in case the resources have actually changed, new deployments will be deployed to the engine database.	Web UI allow instance based view of Tasks. Each Node has a RuntimManager with ist KieSession & TaskService. No matter how many request are sent to the server, they are all processed in parallel. User and Groups can be assigned to tasks. In the administration, authorization of User and Groups can be defined which may play a role on executing a tasks.	Web UI allow instance based view of Tasks. It is possible to administrate tasks and its executin roles.	The implementation of user tasks requires you to define a Human Task. You can use an existing Human Task or define a new one. If your project contains Human Tasks, then they automatically appear in the business catalog under the HumanTasks predefined module. Individual Tasks with ist functions will be set in the BPMN model. They can be controlled in the WEB UI. Human tasks are independent from BPMN processes. If you terminate a BPMN process while it runs a user task, the associated human tasks keeps running independently. If the process instance leaves the user task before the human tasks is completed, the human task continues running and can you can still access it. This is	Roles are used to collect permissions that define a particular function within ProcessMaker, according to a particular scope. A roles is basically just a collection of permissions that defines a function; basically ProcessMaker manage 3 types of roles PROCESSMAKER_ADMIN, PROCESSMAKER_MANAGER and PROCESSMAKER_OPERATOR, these roles have the permissions to: access to the ADMIN menu, access to the Users tab, amongst others.	Business processes frequently involv interaction with users. With X4 Activities, web-based workflow solutions can be implemented within a very short time. Reusable solution components for typical use cases such as operation lists or release workflows allow a quick and easy adjustment to customer-specific requirements. The process logic is defined in X4 processes and provided by X4 ReST as a stateless resource.

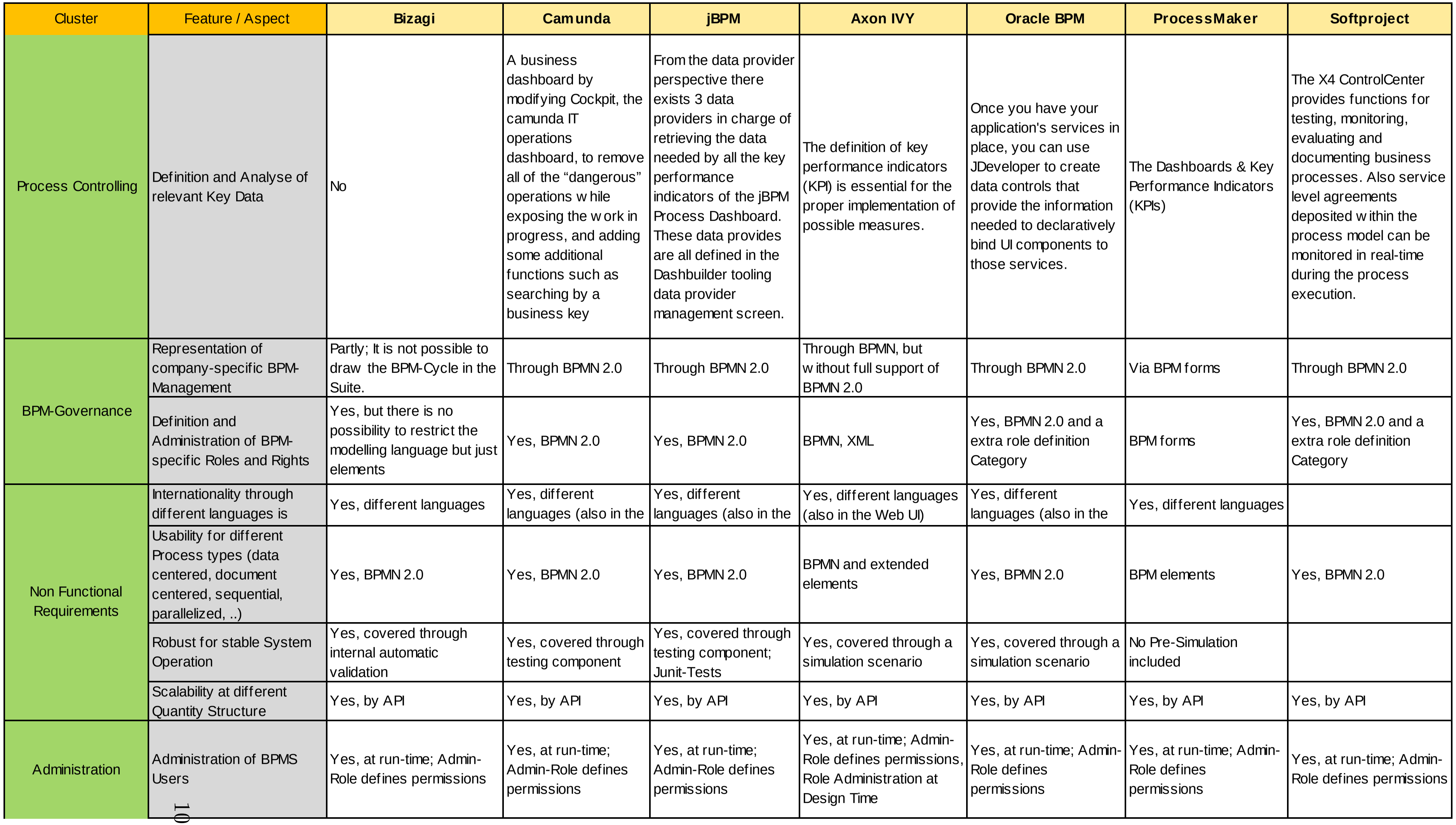
Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	100 User can be assigned to a group	Definition of User Groups in assignations. Bizagi helps to organize the modeling process by categorizing each of the rules according to its use. This feature helps the user when associating each of the rules in a specific situation by only listing the rules that correspond to the category being used.	It is clear that user and group assignments are quite cumbersome for use cases where the assignment is more complicated. To avoid these complexities, custom extensions on the user task are possible. assignee attribute: this custom extension allows direct assignment of a user task to a given user. candidateUsers attribute: this custom extension allows you to make a user a candidate for a task. candidateGroups attribute: this custom extension allows you to make a group a candidate for a task.	The Class org.jbpm.task.OrganizationalEntity contains the subclasses Group and User which enables assigning Users to a Group in Java. In the Web UI User can be added to groups.	Roles and users are always configured per application.	In the Organization Editor Users can be grouped	Individual users, groups, departments, roles. Tasks can only be assigned to users or groups. Users can represent assigned roles.	In the ControlCenter

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Scheduling mechansim is given	Bizagi offers a new template Recruitment and Selection Process to assist companies in this process. The Recruitment and Selection Process automates and reduces the time scale in actions such as scheduling and collecting the results of psych technical tests, assigns interviews, updating the list of candidates, etc. Finally provides the possibility of controlling and monitoring the performance of the process, through indicators that can be created using Query Forms and Bizagi's tools like BAM.	Not for humans but for events; The Camunda process engine includes a component named the Job Executor. The Job Executor is a scheduling component, responsible for performing asynchronous background work. Consider the example of a Timer Event: whenever the process engine reaches the timer event, it will stop execution, persist the current state to the database and create a job to resume execution in the future. A job has a due date which is calculated using the timer expression provided in the BPMN XML.	With the integration of OptaPlanner (http://www.kiegroup.org/, http://www.optaplanner.org/): OptaPlanner optimizes business resource usage.	AXON.TAM supports the process of identifying employees' and candidates' potential and skills, so you can recruit and leverage key talents efficiently. This module also helps match employees to the growing needs of your organization.	Oracle Enterprise Scheduler Jobs; Definition of the constraints within which jobs may run, based on factors such as system resources.		ControlCenter schedules process execution with X4 Scheduler. ControlCenter archives human resources as well. Additional: http://quartz-scheduler.org can be activated in the admin area.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Delegation of Tasks is supported	Work allocation is the fifth step of the Process automation Wizard where Performers are defined for each Activity of your Process. Performers are the users that have the qualities to be assigned to activities. Each Task created for end user interaction requires definition that will allow Bizagi to allocate the correct users within your organization. Bizagi automatically evaluates the allocation rules defined for each Task and selects one or more users that meet the given conditions from the user's list. Only these users will have access to work on the Activity allocated to them.	camunda Tasklist is an end-user friendly web app to manage and complete your user tasks. It supports both direct assignment of tasks to concrete users as well as role based task lists containing tasks that users can claim, delegate and unclaim.	Tasks can be assigned to individuals or to groups, instances can be clustered;	Tasks can be assigned to individuals or to groups.	Tasks can be assigned and delegated in the WEB UI	It allows to delegate tasks by Ad-hoc decision (User/assigned group can re-delegate)	Tasks can be assigned and delegated in the WEB UI The integrated user and rights management allows users, groups, roles, and rights for any Activities application to be defined. The web-based management interface is intended for administrators who want to manage users and permissions for their X4 Activities applications. A connection to an LDAP server is also possible.
	Changing and Controlling of Process Instances during Runtime	No, Before starting the process	RunTimeService is a service which provides access to Deployments, ProcessDefinitions and ProcessInstances.	Context Dependency Injection environment enables injection in an application via the RuntimeManager	Via Ivy Script	In the predifined possibilities it is possible to control in the WEB UI	Roles can be assigned by different Users	In the predifined possibilities it is possible to control in the WEB UI
	Repository	Integrated Repository Management tools to better support the complex mishmash of legacy systems and BPMS components.	It is also possible to access the BPMN model instance by the process definition id using the Repository Service (API).	Organization - Repository - Project, Source Control (Git for version control), Deployment (kjar (knowledge jar = all related artifacts), Maven), Clustered environment	BRMS based repository, Deployment spezialised databases, Sharepoint, ..	SQL Server	MySql, PostgreSQL	Own X4 Repositories

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Ability to extend Human Resource e.g. via API	No, Just fixing special attributes in the property section	If you connect the Camunda BPM platform with the LDAP identity service you have read-only access to the users and groups. Create new users and groups via the LDAP system, but not in the admin application. The OptPlanner (Human Resource Planner) provided by JBoss can also be embedded.	jBPM also includes a so-called human task service, a back-end service that manages the life cycle of these tasks at runtime. The jBPM implementation is based on the WS-HumanTask specification. Note however that this implementation is fully pluggable, meaning that users can integrate their own human task solution if necessary.	Rules Engine: DROOLS, ESB (http://www.mulesoft.org) and REST	User specific properties can be defined either in an LDAP or defined outside of LDAP inside BPM Workspace. In most situations, it might not be possible to extend LDAP to specify process specific user extension attributes.	LDAP, REST API	LDAP or using Adapter Development Kit (ADK)
	Export of Organizational Structure	It is possible to share the Extended Attributes that you created in one model and use then in other models. This allows you to maintain a uniform standard in your documented processes by always utilizing the information in the same manner.	The BPMN 2.0 file does not provide full information about the organizational structure. There is no given export for the organizational structure.	The BPMN 2.0 file contains parts of the organizational information	The BPMN 2.0 file contains the organizational information	Organizations are defined at the project level. You can export organizational information to be used within other projects. Not exportable out of the system but during runtime: access vie Identity Service	No	The BPMN 2.0 file contains the organizational information
	Changing of tasks and processes in Case of Exceptions	Only in pre-defined terms	They can be suspended, changed or deleted by the ProcessEngine API	Availability: In case one Node is down, the Session will be restored	As defined before with the exception handlling (partly through BPMN), also controlable in the Engine	The status of a process can be changed in the deployed WEB UI	The status of a process can be changed in the deployed WEB UI	The status of a process can be changed in the deployed WEB UI
Process Controlling	Examination of individual Tasks	No	With the enterprise version the actual status of tasks is traceable	Through instance based Monitoring	Through instance based Monitoring	Through instance based Monitoring		Through instance based Monitoring

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Controlling	Examination of concrete Process Instances	No	With the enterprise version the actual status of processes is traceable	Through instance based Monitoring	Through instance based Monitoring	Through instance based Monitoring		Through instance based Monitoring
	Detection of divergent and problematic Process Instances	Covered through validation	A lab project, the camunda BPM workbench, a debugging tool that allows inspection of the runtime state of processes alongside the process model, allowing breakpoints to be set in the process model (rather than in code). A console interface allows for interrogation and updating of the process variables as the developer steps through the process. The process model is displayed using the bpmn.io viewer.	In case of problems the UI will monitor this. At design time they will also be displayed.	In case of problems the UI will monitor this.	In case of problems the UI will monitor this.	In case of problems the UI will monitor this.	In case of problems the UI will monitor this.
	Traceability of Procedure of concrete Process Instances	No reports of any tasks	Java Logging and history view in the Web App	Logging as an option in the connection section	Forum, frequently serviced, Chat, Blog	History is visible to all processes and tasks also of individuals	The Simple Reports plug-in allows ProcessMaker administrators to create Report Tables based on any data involved in a process and grant permission for select users to view and export the report directly from their ProcessMaker inbox.	Automated documentation of business processes, services, adapter configurations and transformations



Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Controlling	Definition and Analyse of relevant Key Data	No	A business dashboard by modifying Cockpit, the camunda IT operations dashboard, to remove all of the “dangerous” operations while exposing the work in progress, and adding some additional functions such as searching by a business key	From the data provider perspective there exists 3 data providers in charge of retrieving the data needed by all the key performance indicators of the jBPM Process Dashboard. These data provides are all defined in the Dashbuilder tooling data provider management screen.	The definition of key performance indicators (KPI) is essential for the proper implementation of possible measures.	Once you have your application's services in place, you can use JDeveloper to create data controls that provide the information needed to declaratively bind UI components to those services.	The Dashboards & Key Performance Indicators (KPIs)	The X4 ControlCenter provides functions for testing, monitoring, evaluating and documenting business processes. Also service level agreements deposited within the process model can be monitored in real-time during the process execution.
BPM-Governance	Representation of company-specific BPM-Management	Partly; It is not possible to draw the BPM-Cycle in the Suite.	Through BPMN 2.0	Through BPMN 2.0	Through BPMN, but without full support of BPMN 2.0	Through BPMN 2.0	Via BPM forms	Through BPMN 2.0
	Definition and Administration of BPM-specific Roles and Rights	Yes, but there is no possibility to restrict the modelling language but just elements	Yes, BPMN 2.0	Yes, BPMN 2.0	BPMN, XML	Yes, BPMN 2.0 and a extra role definition Category	BPM forms	Yes, BPMN 2.0 and a extra role definition Category
Non Functional Requirements	Internationality through different languages is	Yes, different languages	Yes, different languages (also in the	Yes, different languages (also in the	Yes, different languages (also in the Web UI)	Yes, different languages (also in the	Yes, different languages	
	Usability for different Process types (data centered, document centered, sequential, parallelized, ..)	Yes, BPMN 2.0	Yes, BPMN 2.0	Yes, BPMN 2.0	BPMN and extended elements	Yes, BPMN 2.0	BPM elements	Yes, BPMN 2.0
	Robust for stable System Operation	Yes, covered through internal automatic validation	Yes, covered through testing component	Yes, covered through testing component; Junit-Tests	Yes, covered through a simulation scenario	Yes, covered through a simulation scenario	No Pre-Simulation included	
	Scalability at different Quantity Structure	Yes, by API	Yes, by API	Yes, by API	Yes, by API	Yes, by API	Yes, by API	Yes, by API
Administration	Administration of BPMS Users	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions, Role Administration at Design Time	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Administration	106 Administration of Processes and Deployment	Yes, at run-time; table of activities	Yes, at run-time; table of activities. The Cockpit also shows in design pattern where the actual token is	Everthing guided by the REST Web UI	Guided by the Engine, when started	Everything guided in the WEB UI	Everything guided in the WEB UI	Everything guided in the WEB UI
	Self-Administration and Administration of End-User	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile
Setup	Instalability / Complex Installation	Difficult Installation of Server Environment	Simple to deploy with Maven	Seam, Spring, OSGi,	Designer and Engine serperate; Designer uses Eclipse environtment. Engine is a single or clustered application server based on Java	JDeveloper is not difficult to install but needs many computational resources for Windows.	Designer and Engine as Web UI	One-click installation – Less effort for new installation and update. Easy start and less development effort thanks to extensive examples and integration and process patterns
	Client; Adjustment of End-User Workplace	Yes, automatically through deployment	Yes, automatically through deployment	Yes, automatically through deployment	Windows 8, 9 interface	Yes, automatically through deployment	Nothing locally, just in the WEB UI	Yes, automatically through deployment
	Portal; Specified Organization Configuration for visible End-User's Process Start	Depends on configuration: .Net, JEE, Xpress (smal usage)	After Deployment, access via Web App	Sharepoint add-On	Sharepoint add-On	After Deployment, access via Web App	No	After Deployment, access via Web App
	Platform	.Net, JEE	Java EE	Jboss, Java EE	independent	JEE	independet (Web UI)	JEE
	Server; Adjustment of BPMS-Backend	Microsoft SQL Server and Oracle.	MySQL, Oracle, MariaDB, IBM DB2, PostgreSql, MS Server, H2	db2, h2, mysql, postgresql, sqlserver	Microsoft, Oracle, DB2, PostregreSQL	Oracle SQL	MySQL, PostgreSQL	Databases; SQL, MongoDB
Support	Frequently serviced	Yes	Yes		Yes	Yes	Yes	Yes
	Handbook is available	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Quality of Handbook is up	Online Handbook	Online Handbook	Online Handbook	Online Handbook	Online Handbook	Online Handbook	Online Handbook
	Online Developer Community is set	Forum, frequently serviced	Forum, frequently serviced	Forum, frequently serviced, Chat, Blog	Forum, frequently serviced, Chat, Blog	Forum, frequently serviced	Forum, frequently serviced	Forum, frequently serviced
	Further Online Resources (Handbook, Docu, Tutorial,	Workshops	Workshops, Online Demo of Cockpit	Workshops, Demo version	Workshops, Demo version	Workshops, Demo version	Wiki	Workshops, Demo version

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Maturity	Product Maturity and Support	ranked at #1 BPMS Fraunhofer Institute Study 2013, although it is rather based on comfort than on programming	Powerful and flexible developing framework for business applications. Especially for developer it offers innovative, powerful features. The pricing model is flexible and allows a low cost entry.	jBPM strives for the "Zero-Code-BPM"-ideal. Powerful features are missing.	Powerful and easy-to-use BPMS. BPMN2 is not fully supported - for companies not very important, big Clients, one of the Best of Fraunhofer study 2014	Very powerful Application Development Framework with the ability to develop BPM systems. User has to get used to the maturity of possibilities. Ranked at the top of Fraunhofer study 2014	Quick and simple illustration of BPM Workflows in a WEB UI. But not very powerful since there is no great compatibility with other systems.	One of the leading players. Emphasize lies on integration of services and managing them with a powerful organization concept. Also documentation service with export in several formats (Word, Excel) included. Designer based on BPMN 2.0
Usage	Used within Siemens	No	No	No	No	No	Yes	Yes

References

1 T. Allweyer. *BPMN 2.0: Introduction to the Standard for Business Process Modeling*. Books on Demand, 2010.

2 Julia Fuchsbauer. How to manage processes according to en50126. Bachelor thesis, Vienna University of Economics and Business, July 2015.

3 Alexander Wurl. Formalisms and tools to describe and monitor engineering processes. Master's thesis, Vienna University of Technology, January 2016.