**PROMES**
**Process Models for Engineering of Embedded Systems**

**Title:**
State-of-the-art report on process components

**Version** : **2.0**
**Date** : **15/10/2015**
**Pages** : **67**


**Author:**

**Jukka Kääriäinen, VTT**
**Jari Lehto, Nokia**
**Olli Kuusisto, VTT**
**Jukka Hemilä, VTT**
**Christian Manteuffel, RUG**
**Sofia Charalampidou, RUG**


The Promes Consortium consists of:

Haloila, KE-Works, Valmet Automation, Nokia Solutions and Networks, Océ Technologies, TNO ESI, University of Groningen, Vector Fabrics, VTT

**Printed on:**
15-10-2015 15:14:00

| **Status:** | | **Confidentiality:** | | |
|---|---|---|---|---|
| [ ] | Draft | [ X ] | Public | - Intended for public use |
| [ ] | To be reviewed | [ ] | Restricted | - Intended for Promes consortium only |
| [ X ] | Final / Released | [ ] | Confidential | - Intended for individual partner only |

**Deliverable ID:** **D2.2**

**Title:**

# State-of-the-art report on process components


**Summary / Contents:**

This deliverable describes state-of-the-art overview on process component practices and studies. In the deliverable the experiences of applying the process components is compared to the state-of-the-art.

# CHANGE LOG

| Vers. | Date | Author | Description |
|---|---|---|---|
| 0.1 | 03.02.2015 | Olli Kuusisto | Draft created |
| 0.2 | 07.05.2015 | Jukka Kääriäinen, Olli Kuusisto, Susanna Teppola, Jukka Hemilä | Upgrade –service SOTA, component experiences, value of component. Audit -service. Ready to send for PROMES contribution. |
| 0.3 | 28.05.2015 | Jukka Kääriäinen, Olli Kuusisto | Audit-service + Service/R&D collabotarion added. |
| 0.4 | 03.06.2015 | Jukka Kääriäinen, Jari Lehto | Component contribution: KE-Works, Nokia, RUG, VTT |
| 0.5 | 11.06.2015 | Jukka Kääriäinen, Christian Manteuffel, Sofia Charalampidou, Bas Huijbrechts, Jos van Eijndhoven | Component contribution: TNO, VF<br><br>PROMES related ITEA projects, process framework state-of-the-art (RUG) |
| 0.6 | 18.06.2015 | Jukka Kääriäinen, Jari Lehto | Document ready for internal review: Reviewers: KE-Works (Maarten Nelissen), Valmet (Antti Välimäki) |
| 1.0 | 30.06.2015 | Jukka Kääriäinen, Jari Lehto | Finalised according to review. **Deliverable v1.0 accepted and released.** |
| 2.0 | 15.10.2015 | Jukka Kääriäinen, Jari Lehto | Deliverable finalised for public release. **Deliverable v2.0 accepted and released.** |

# TABLE OF CONTENTS

# 1. INTRODUCTION

This deliverable describes the state-of-the-art overview of PROMES process components. The components that have been developed in WP2 and applied in case studies in WP3 will be studied against state-of-the-art solutions. I.e. the intention is to reflect the results of the WP2 against the state-of-the-art. State-of-the-art study covers research and solutions from scientific literature and solutions from professional publications / vendors' web pages, when applicable.

The working method in PROMES project is iterative (2 iterations)(see next figure). Deliverable D2.1 forms a basis for the work in tasks 2.1, 2.2 and 2.3. Developed solutions are described in D2.3 and stored into process repository (WP1) as well as applied and verified in the case studies (WP3). D2.3 deliverable (iteration 1 and iteration 2) is a summary of these solutions. Deliverable 2.2 documents the State-of-the-art overview on process component practices and studies. In D2.2 the experiences of applying the process components is compared to the state-of-the-art.

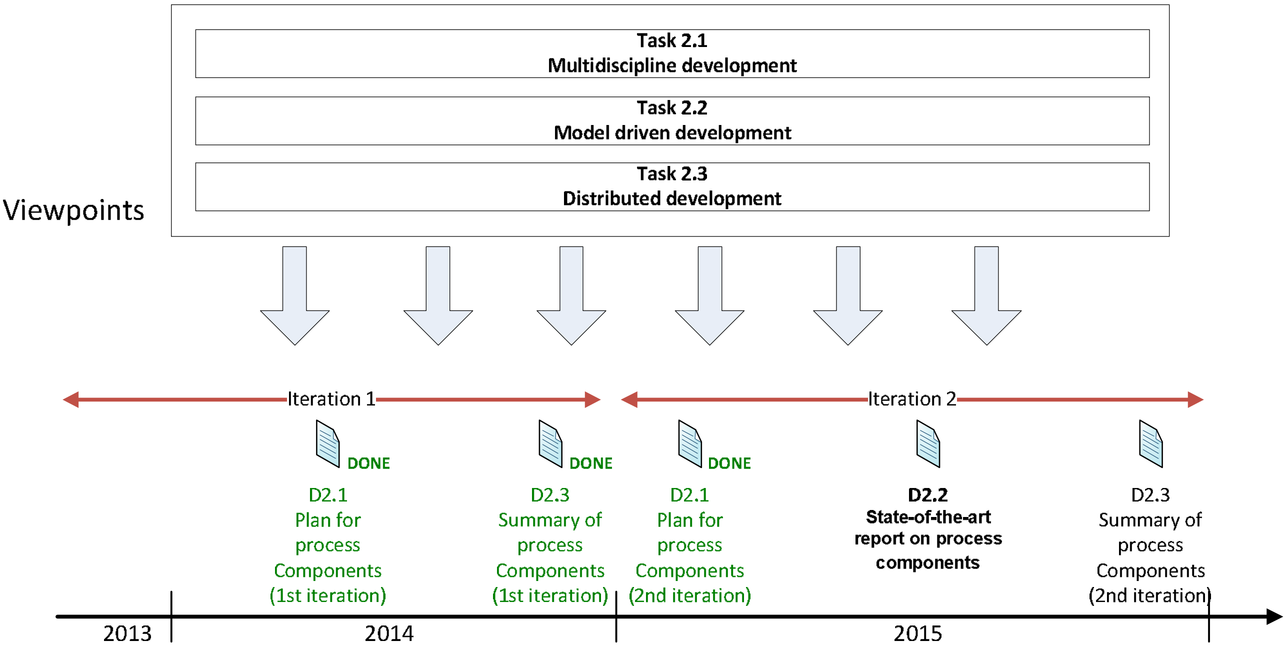


Figure. Overview of WP2 tasks and deliverables.

D2.2 deliverable (State-of-the-art report on process components) is organised as follows. Next chapter introduces the state-of-the-art studies of components. Chapter 3 presents PROMES process framework state-of-the-art. Chapter 4 discusses the ITEA projects related to PROMES process framework.

# 2. PROCESS COMPONENT STATE-OF-THE-ART STUDY

This chapter introduces the state-of-the-art studies of process components. State-of-the-art study of each component contains the following structure:

- State of the art/practice: this sub-section introduces state-of-the-art/practice based on scientific literature as well as professional publications and solutions from vendors' web pages related to process components.
- Component experiences compared with state of the art/practice: this sub-section discusses the process component compared to the state-of-the-art/practice study.
- Value/benefits of component: what kind of benefits are expected when using this component.

## 2.1 UPGRADE -SERVICE (PC-FIN-VTT-01)

### 2.1.1 State of the art/practice

**Industrial services**

In the digital economy, products and services are linked more closely to each other. The slow economic growth during recent years has boosted the development of product-related services even more – they have brought increasing revenue for the manufacturing companies in place of traditional product sales (Oliva & Kallenberg, 2003; Borcher and Karandikar, 2006). According to Frost & Sullivan (2014) end users in process industries start to outsource operation and maintenance services to system vendors or third-party service providers. They indicate that the drivers for this are the following:

– declining availability of a skilled work force
– need to concentrate on core competence – not to O&M (Operation & Maintenance) services

In today's business environment most companies offer a mix of products and services. The development of a pure product without additions or a pure service without a physical part is becoming rare. Manufacturers typically provide services which are based on the manufacturer's knowledge of their own equipment, which they can extend to similar equipment. The value and significance of knowledge-intensive business services (KIBS), including knowledge-intensive maintenance services, is steadily growing in modern industries. (Hanski et al., 2012)

Developing integrated solutions is one way for supplier companies to adapt to the changes in customers' business. Based on previous literature, Hakanen (2014) defines integrated solutions to be bundles of products and/or services that are customized to meet customer-specific needs and assumed to offer greater potential for value creation than the individual components alone. Brax & Jonsson (2009) add that they are long-term oriented, integrate the provider as part of the customer's business system, and aim at optimizing the total cost for the customer.

Next figure presents an imaginary example of industrial services with core, support and augmented services. It is in line with product-service bundle while it adds core services as new element to it.

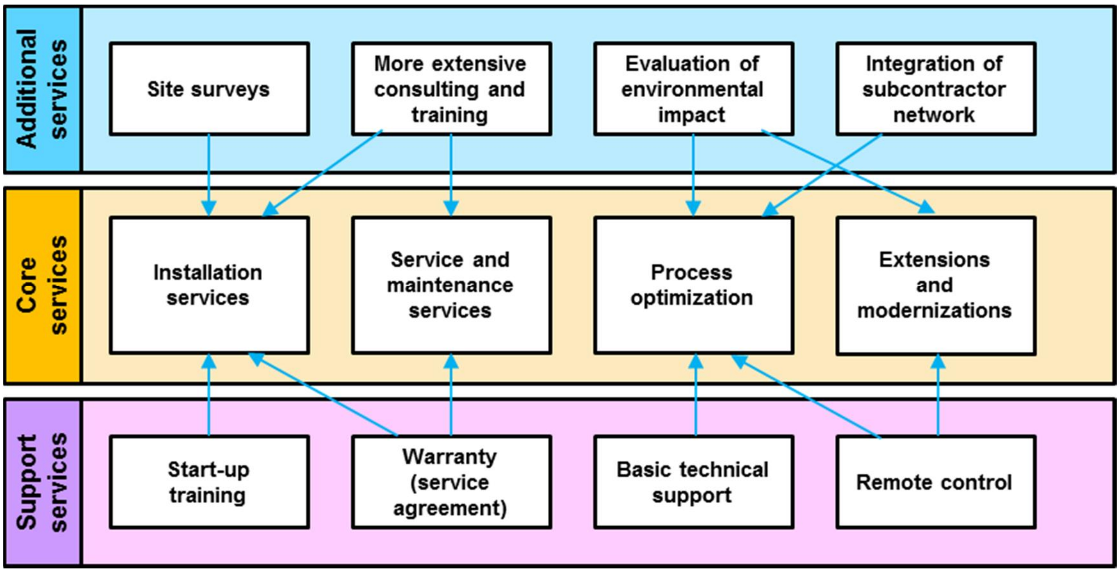


Figure. An example of industrial service offering structure (Toivonen, 2015b).

***Core services*** include services which the customer primarily seeks from supplier. In addition, ***support services***, that are imperative for the usability of core services, are needed. In most cases the offering includes also ***additional services*** that raise the service quality conception of the customer. Additional services are a flexible means to improve the overall offering, and often implementable with relatively few resources. The division into core, support and additional services shows the business focus of the supplier and clarifies the use of resources showing what is indispensable and where there is room for flexibility. It must also be noted that the division is not invariable – e.g. same service can at one time be additional and at another time core service. (Toivonen, 2015a&b)

Industrial services have been commonly thought to be just after-sales services, not dealing with the whole life cycle of products. However, the most effective way of doing service business requires lifecycle thinking. Different areas of scope on which an industrial company could focus, and which could be the main service business areas are shown in next figure. (Kalliokoski et al., 2003)

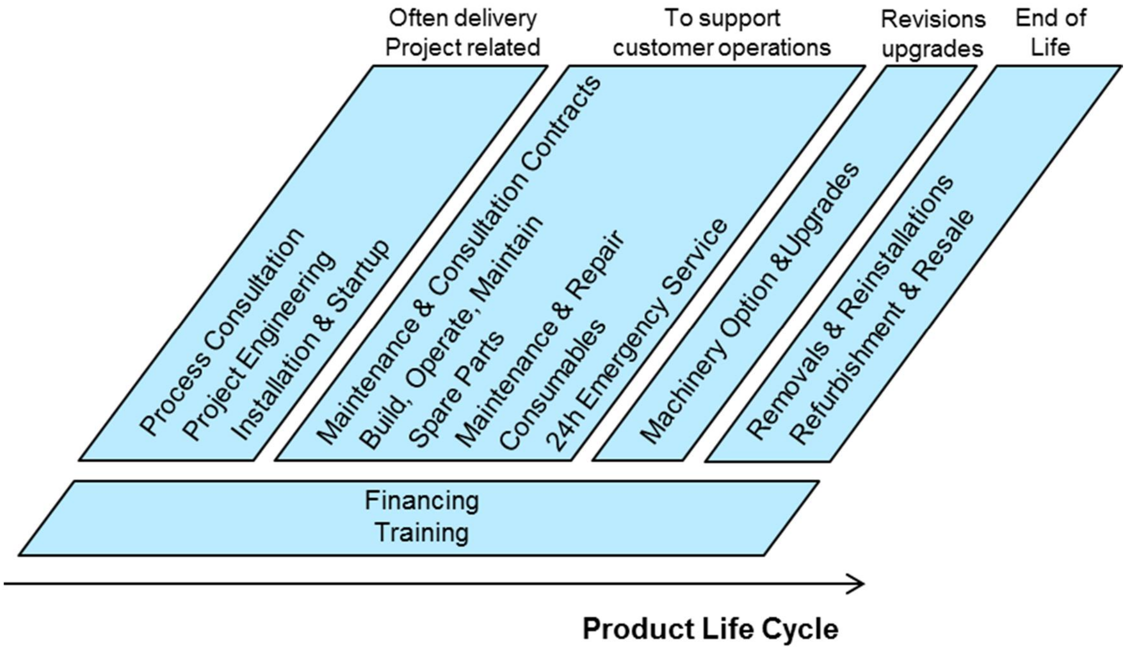


Figure. Industrial services over the lifecycle of the product. (Modified from Kalliokoski et al., 2003)

**Installed base systems and life cycle planning**

Installed base is defined in different ways. According to Oliva and Kallenber (2003) installed base is '*the total number of products currently under use*', and according to Borcher and Karandikar (2006) '*a measure of the number of units of a particular type of product or system actually in use*'. Ala-Risku (2009) states that installed base do not indicate the number of installations but it is regarded as formed by individual products. As the manufacturer itself might not be necessary the one who provides after sales service, there is also a definition where the installed base is defined as the whole set of systems/products for which an organisation provides after sales services (Dekker et al, 2010).

Providing aftermarket services to products already manufactured and delivered to customers, requires that service provider must have a good and up-to-date knowledge of the installed base of their products. According to (Borcher and Karandikar, 2006) perfect knowledge of the installed base has high value, because it serves manufacturer's immediate product sales and service but also product marketing and future product planning. Delivering the right service in a cost-effective manner requires good knowledge of the installed base, and creates significant sales opportunities for value-added services and products.

According to Borcher and Karandikar (2006) Installed base (IB) systems attempts to track down exactly where the sold products are located, who owns and /or operates them, what they are used for, under which conditions they are applied, their life cycle status, which service actions and technical changes have been performed, which parts serviced or replaced and their current technical state. As industrial goods typically have long life-cycles; they include complicated technical structures, use different suppliers of parts, require demanding service support, and have a multi-level sales processes, all of which cause that maintaining installed product information also becomes complicated. A producer of industrial goods can be connected to its end customers directly, but it is also possible that the customer itself may sell the products again, modified or unchanged, stand-alone or incorporated into some larger system. Then the sales chain to the end user may include channel of partners, distributors, system integrators and resellers. All these effect to installed base information (Borcher and Karandikar, 2006). Borcher and Karandikar (2006) presents an illustration of the different sources of information important for installed base, see the next figure.

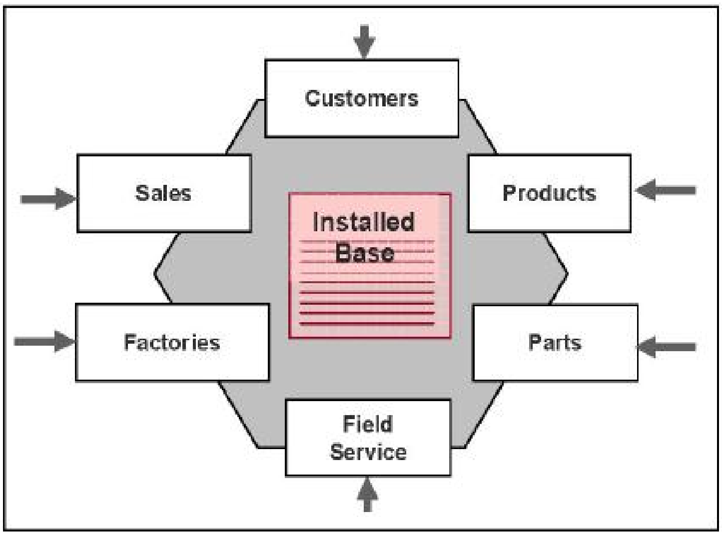


Figure: Different sources of information important for installed base (Borcher and Karandikar, 2006).

Ala-Risku (2009) divides installed base information into three main categories. These are:

- Item data, which consists of information about products.
- Location data, which consists of information about the customer site or process phase that is the target of product deliveries and service operations.
- Event data, which consists of information about service operations.

Borcher and Karandikar (2006) lists the following important information that installed base should include:

- Number of items of a particular product delivered to whom, and where in use.
- Installation site and location.
- Owner company information, responsible units/persons.
- Technical configuration of the product when it left from the manufacturer and when it was installed on the customer site.
- Life cycle status.

- The purpose of use.
- Under which conditions applied.
- Service contracts, warranty times and conditions.
- Service actions and technical changes performed, which parts have been serviced or replaced and their current technical state.

As mentioned above in service business it is needed to have a way to track the customer's installed base of the product, but it is also important to know what else has been installed so it is possible to predict the behavior of products in that environment. There are several possible goals and objectives which can be achieved through the successful implementation of an installed base. According to the study by Borcher and Karandikar, 2006, from business and service point of view the knowhow of product installations improve customer service quality and value in use by enabling the realization of the necessary upgrade needs by a customer and possible restrictions of an upgrade, and enhance aftermarket business planning and implementation. From product development point of view installed base serves as an input for compatibility and manageability issues. In case of products with long replacement cycles backwards compatibility may be as important as pure functionality. Installed base also affect the timing at which new products and technologies should be introduced. Understanding infrastructure and changes in it can be crucial to be able to forecast the timing with which new products will take off.

Hallmans et al. (2014) study evolution from process point of view. They modify the utilisation and support stage in ISO/IEC 15288 standard by replacing them with the evolution stage where a system is not only retired and replaced but rather evolved into the next generation. They argue that using this approach changes the view of system development for this specific type of systems towards a way of incremental development, where new functions can be added at the same time as old legacy parts are replaced with functionally equivalent modules based on new hardware.

Poulsen (2014) presents an approach for the organisation and planning of upgrades and replacements. Poulsen states that "along with assessing the necessary migration activities to cope with technical deterioration, it is also important to assess new challenges and new opportunities from a business point of view". The approach comprises five steps for long-term migration planning: mobilize (understand the goals), analyze (understand the existing system), target (needs and to-be-solution), justify (preliminary business justifications based on rough cost and benefit estimation) and plan (implementation planning).

**Existing solutions in automation industry**

The aim of this section is to get an insight *how industrial automation vendors' communicate / present the upgrade needs for their customers.* The information has been collected from automation system vendors' web-pages (white papers, brochures) and, therefore, the extent of the information varies highly depending on vendor. The study does not cover maintenance services, etc. generally, but focuses to system upgrade issues. Concerning automation markets:

- The biggest players in DCS (Distributed Control Systems) markets are ABB, Emerson, Honeywell, Yokogawa and Siemens (Frost & Sullivan, 2013).
- The biggest players in automation services market for process industries are ABB, Siemens, Honeywell, Emerson and Yokogawa (Frost & Sullivan, 2014).
- However, when considering Pulp&Paper customer domain, Valmet is 3rd biggest among DCS vendors.

ABB

ABB provides a six-stage Evolution for Life program for their customers to enable manageable evolution path for automation system (Harding, 2008).

- *Stage 1:* Identify customer's business goals.
- *Stage 2:* Analysis of installed systems (configurations) and how they are affected by business goals.
- *Stage 3:* Assess ABB's system and services offering.
- *Stage 4:* Develop long- and short term technical and commercial plans.
- *Stage 5:* Develop and implement an agreed Evolution schedule.
- *Stage 6:* Continuous re-evaluation of business needs and solutions.

Evolution planning is an ongoing collaborative process between ABB and system owners where ABB analyses *customer's system status, creating short and long term roadmap for system enhancements based on customer's business needs and giving recommendations and value propositions for updates* (Morrisey, 2012). With evolution planning customer gets benefits, such as *getting insight into system risk areas from lifecycle management point of view, identifying new system functionality and solutions to extend the value of the system to meet customer's business need and known timing for evolution action (2-5 years)* that assists customer's budgeting process (upgrades do not come as a surprise)(Morrisey, 2012).

Rockwell Automation

Rockwell Automation provides installed Base Evaluation & Lifecycle Analysis service for their customers (Rockwell Automation, 2012): Step 1 – Field Collection, Step 2 – Processing, Step 3 – Delivery/presentation. This evaluation&analysis (Rockwell Automation, 2012) provides e.g.:

- Reports, with red, yellow and green coding indicating the lifecycle status of all parts
- What is installed, what is missing and what equipment is nearing the end of its critical life.
- Presentation discusses solutions that are going to be most effective in helping improve plant performance.

Other vendors

Honeywell's marketing material presents LCM (Life Cycle Management) Program as an approach to manage upgrades for customers' installed base (Honeywell, 2012). LCM will be specified according customer's needs to allow ongoing support and to extend the life of the assets in a predictable manner.

Siemens provides different service packages for different needs: standard service, maintenance service, basic life cycle service, extended life cycle service (Siemens, 2014). Furthermore, it seems that one technology that has a significant impact to automation system upgrade needs is operating systems.

### 2.1.2 Component experiences compared with state of the art/practice

Deliverable D2.3, section 3.1, describes Upgrade –service process component that has been piloted in Valmet and is under wider deployment in the company. The component is especially intended for automation industry. However, some parts of the component might be applied also in other domains that produce and maintain complex long-living customer-tailored products that comprise in-house developed or COTS HW/SW components. According to state-of-the-art/practice analysis, big players in automation system markets seem to have mature solutions for systematic upgrade planning. However, smaller automation system providers need to improve their capabilities for providing industrial services to keep their position in markets. This is huge opportunity since automation service industry is steadily growing (Frost & Sullivan, 2014). Furthermore, in literature there seems to be only few practical models or cases that explain on practical level how the upgrade –service should be done and what kind of tool support (e.g. InstalledBase, Life cycle planner –functionality, Customer extranet) is needed for that. Therefore, research and especially practical case studies are highly welcome to help smaller service providers to systematise their industrial service processes – not only Upgrade-service but also other services like Audit –service (see the next chapter). Upgrade-process was described by applying the PROMES process framework (D1.1). The framework provided easy to use description method to describe processes. Process descriptions were compact and made it easier to discuss the processes with the Valmet representatives. This method was utilised in all Industrial service – process components.

If PROMES Upgrade-service process component is compared with the approach presented in Poulsen (2014), it can be noted that our approach emphasises more the usage of IT systems during the upgrade process. The basic phases are somewhat similar starting from the identification of needs through clarification of installed system and analysing the upgrade effects. However, Poulsen addresses more implementation planning phase that includes also investment planning and project risk assessments that could be valuable also for PROMES process component.

Hallmans et al. (2014) study evolution from process point of view. The research is interesting for Upgrade –service process component since it extends ISO/IEC 15288 system development stages called utilisation/support/retirement stages with the evolution step. PROMES Upgrade –service process component is practical industrial-based service process description that can be applied on the evolution stage of Hallmans et al. (2014) approach. Furthermore, Hallmans et al. (2014) state that "*Due to the long life time and the number of generations working in parallel a large amount of legacy knowledge is needed to be able to support the system… new support engineers have to be educated on a number of different (product) generations and also customer specific implementations… Large scale systems with customer specific changes can be hard to keep track of during the long life time of the system*". All these issues are well in line with the industrial problems behind Upgrade –service process component and can be utilized in future component improvement.

### 2.1.3 Value/benefits of component

Based on workshops and interviews Valmet identified that the following benefits are expected for customer when using the process component:

- Customer can prepare for a change (scheduling, budgeting). Schedule for upgrades.
- It is easier to justify gradual small changes than expensive big-bang change.
- The reliability and availability improve.
- Opportunity to optimal upgrade path

- No surprises. Understanding the risk areas of the system that need actions.

The following benefits are expected for Valmet when using the process component:
- Knowing the status of automation system through its lifecycle.
- Opportunity to optimal upgrade path.
- Security updates on-time.
- Visualising what needs to be upgraded and when (transparency to upgrade opportunities) (possibility to justify upgrades for a customer).
- Possibility to introduce new functionality flexible step by step.
- Better service (and on time).

**References:**

Ala-Risku, T.: Installed Base Information – Ensuring customer value and profitability after the sale. Helsinki University of Technology, Department of Industrial Engineering and Management doctoral dissertation series, 2009/6. Espoo: Yliopistopaino, 2009.

Brax, S.A., Jonsson, K.,Developing integrated solution offerings for remote diagnostics, International Journal of Operations & Production Management, Vol. 29 Iss 5 pp. 539 - 56 (2009)

Borchers, H.W. and Karandikar, H: A Data Warehouse approach for Estimating and Characterizing the Installed Base of Industrial Products. International conference on Service systems and service management, IEEE, Vol. 1, pp. 53-59 (2006)

Dekker, R., Pince, C., Zuidwijk, RA and Jalil, MN.: On the use of installed base information for spare parts logistics: a review of ideas and industry practice. Research paper, Report / Econometric Institute, Erasmus University Rotterdam. Econometric Institute, 2010.

Frost & Sullivan, Analysis of the Global Distributed Control System Market, March 2013.

Frost & Sullivan, Global Automation Services Market for Process Industries, April 2014

Hakanen, Taru. Co-creation of integrated service solutions in business networks. Espoo, VTT. 95 p. + app. 71 p. VTT Science; 71. (2014)

Hallmans D., Jagemar M., Larsson S., Nolte T., Identifying Evolution Problems for Large Long Term Industrial Evolution Systems, IEEE 38th International Computer Software and Applications Conference Workshops (COMPSACW), 2014, 21-25 July 2014, Page(s):384 - 389

Hanski, J., Kunttu, S., Räikkönen, M., Reunanen, M.: Development of knowledge-intensive product-service systems. Outcomes from the MaintenanceKIBS project. VTT, Espoo. VTT Technology : 21 (2012)

Harding, D. (2008). “Evolving automation systems. Automation systems discover the meaning of life.” ABB Review. White paper.

Honeywell: Lifecycle management. Honeywell. Service brochure. 2012.

Kalliokoski, Petri; Andersson, Göran; Salminen, Vesa; Hemilä, Jukka. BestServ. Feasibility study. Final report. Kerava, Teknologiateollisuus. 61 p. (2003)

Morrisey, K. (2012). “Evolution planning considerations.” ABB Group. Presentation slides. March 26, 2012.

Oliva, R., Kallenberg, R: Managing the transition from products to services. International Journal of Service Industry Management, Vol. 14, No. 2, pp. 160-172 (2003)

Poulsen, L.: Life-cycle and long-term migration planning. InTech magazine (a publication of the international society of automation), January/February 2014, pp. 12-17, (2014)

Rockwell Automation, (2012). ”InstalledBase Evaluation.” Rockwell Automation. Service brochure.

Siemens, 2014 SIMATIC PCS 7 Lifecycle Services https://support.industry.siemens.com/dl/files/764/90000764/att_839016/v1/SIMATIC_PCS7_Lifecycle_Services_EN.pdf

Toivonen, Marja. Opportunities and challenges in Finnish service business. In Service Economy Revolution – Finlands Growth Potential. Publications of the Ministry of Employment and Economy, 2015a. pp. 68-107.

Toivonen, Marja. Palvelutuotteet ja palvelukonseptin integrointi tuotekehitykseen, Aalto PRO: Diploma in Product Development –training 28.1.2015 (2015b). (in Finnish)

## 2.2 AUDIT –SERVICE (PC-FIN-VTT-02)

### 2.2.1 State of the art/practice

Audit-service can be applied for various industrial services. Audit-service is used in automation industry to detect the status and health of the customer's product/system. Actually, typically all industrial services start with the some sort of Audit to get an overview of the customer's situation. Important part of the Audit is the identification of installed system, i.e. what kind of system configuration exists in customer's premises. Therefore, customers' Installed Base information is vital for service companies (see section 2.1.1 for Installed Base literature). Also Ala-Risku (2007) discusses the role of InstalledBase information during overall service delivery process (i.e. any product-related service basically needs Installed Base information).

Next we will provide examples what kind of Audit-services automation system vendors' offer. The information has been collected from automation system vendors' web-pages (white papers, brochures) and, therefore, the extent of the information varies highly depending on vendor. We present information from two companies that were ranked as big players in automation service industry (Frost&Sullivan, 2014): Emerson and Honeywell.

<u>Emerson</u> has different kinds of audit services for automation industry. One service is Flow Audit Service (Emerson, 2013). Service description introduces what service vendor does (i.e. Emerson) and what the customer should do (i.e. Emerson's customer) during Audit-service (See next figure).

**Flow Audit Roles and Responsibilities**

| Stage | Customer | Emerson |
|---|---|---|
| Identify scope | Define number of Flow instruments and identify the customer team | Define scope, set objectives and furnish quote |
| Office Preparation | Instrument details. P&ID information.<br>Device criticality (operations and HSE) | Collect instrument data, calibration records, configuration, service and obsolescence information |
| Field research | Advise safety, work permits as needed. Provide access to key on-site personnel | Identify "hot issues", check instruments, download parameters, take pictures, create parts list, get information on maintenance and identify training needs |
| Data analysis | Provide detailed answers to questions that may arise | Analyze configuration (range, output), calibration (zero, FCF), maintenance, obsolete products, possible replacements, inventory, training needs, etc. |
| Report | Review DRAFT report and provide comments/ feedback | Create a customer report containing: Introduction, scope, results, conclusion, discussion and recommendation |
| Presentation | Assemble key stakeholders for a formal review of the results | Present report , findings and recommendations<br>Discuss and agree on action and road map |
| Follow-up | Budget required services, parts and replacements | Provide customer with quotes for products & services outlined in the final report |

Figure: Flow Audit Service (Emerson, 2013).

Furthermore, Emerson has the tool-based solution for process or installation audit called Inspect. They state that "*Performing an audit of a process or installation might be considered to be a relatively straightforward task. However, when many different assets, processes and people are involved, the audit management can become complex*" (Emerson, 2015).

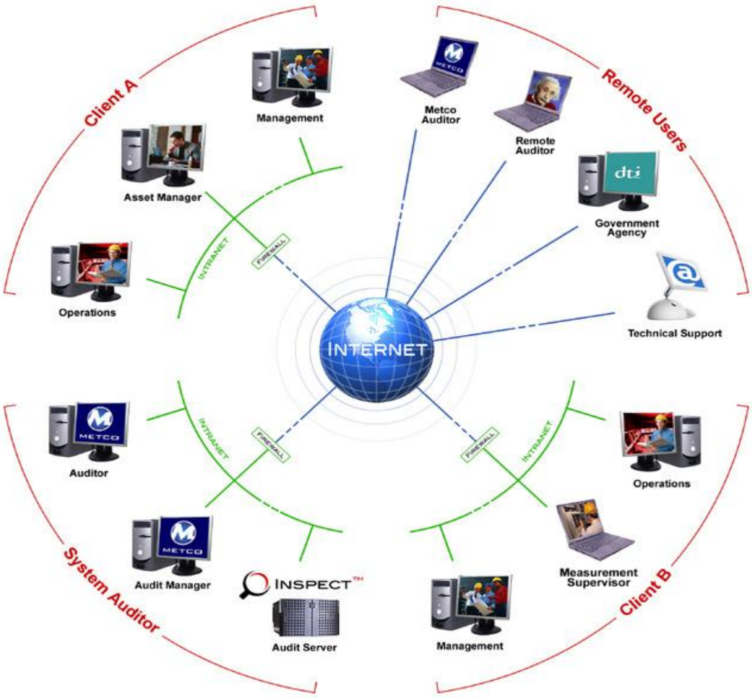


Figure: Connecting stakeholders and organisations (Emerson, 2015).

Honeywell offers Integrated Automation Assessments (IAA) with two service variants (standard, enhanced). IAA produces an audit report that is a starting point for users to develop an automation roadmap and prevent reliability issues by rectifying future obsolescence through timely migrations and upgrades. It also enables them to improve process stability by correcting control loop performance issues and increases operator effectiveness through timely identification of alarm issues. IAA process contains field and back office phases. (Honeywell, 2015)

| Integrated Automation Assessment - IAA | Standard | Enhanced |
|---|---|---|
| **1. Control System Update & Performance Audits** | **X** | **X** |
| - FTE Network Installation and Configuration | x | x |
| - Security Updates and Antivirus Verification | x | x |
| - Operating System Performance | x | x |
| - EPKS Client Server Configuration | x | x |
| - Verification of Graphics Performance | x | x |
| - Back Up and Recovery Assessment | x | x |
| **2. Technology Integration** | **X** | **X** |
| - TPS Audit | x | x |
| - System Performance Baseline | x | x |
| **3. Lifecycle Optima Evolution Path** | **X** | **X** |
| - Update on the Status of the Installed Base | x | x |
| **4. Process Control Performance Audit** | | **X** |
| **5. Alarm Benchmark** | | **X** |
| **"Standard" IAA Pricing:** | **EPKS** | **TPS** |
| **Price Per System:** | | |
| **Customers with existing SPB options in contracts:** | | |
| **"Enhanced" IAA Pricing:** | **EPKS** | **TPS** |
| **Price Per System:** | | |
| **Customers with existing SPB options in contracts:** | | |
| **Optional:** | | |

Figure: IAA service variants and their coverage. (Honeywell, 2015)

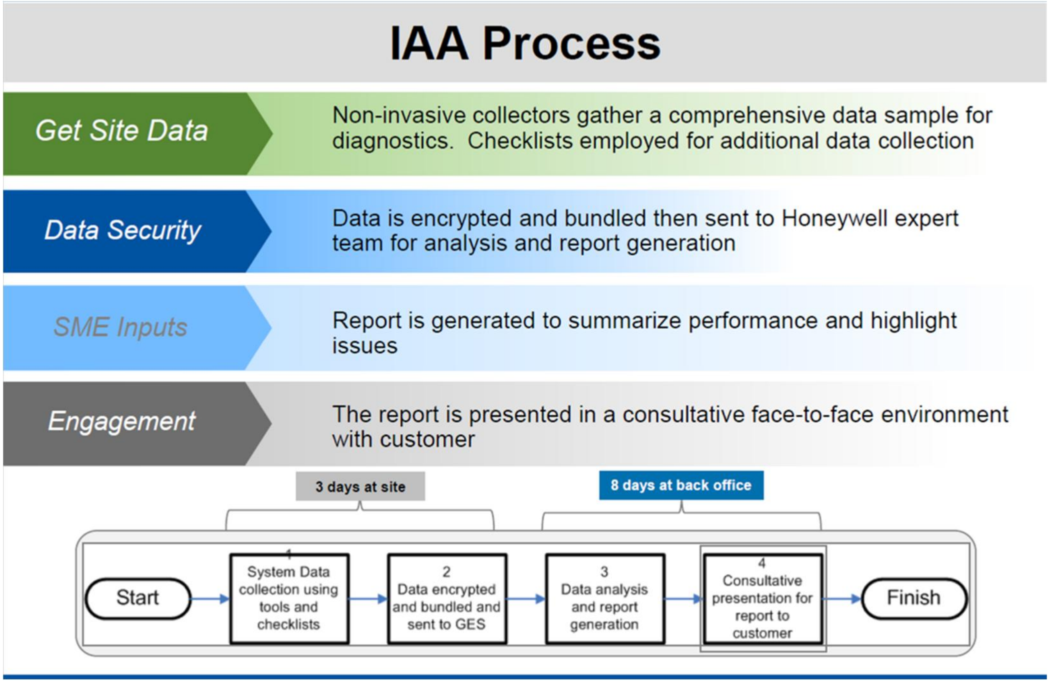


Figure: IAA process. (Honeywell, 2015)

### 2.2.2 Component experiences compared with state of the art/practice

Audit-service can be seen as more generic service if compared to e.g. Upgrade-service. Some sort of audit is needed prior to most of services and, therefore, audit may lead to upgrade, training, maintenance, consultancy, etc. Therefore, this is a potential common activity for all types of technical services. Similarly as in Upgrade-service it seems that big players in automation system markets have mature solutions for systematic audits. However, smaller automation system providers need to improve their capabilities for providing industrial services to keep their position in markets. Therefore, research and especially practical case studies are highly welcome to help smaller service providers to systematise their industrial service processes. The process component is especially intended for automation industry. However, some parts of the component might be applied also in other domains that produce and maintain complex long-living customer-tailored products.

Need for audit may come from different sources. Service contract may include periodical audit or the trigger for audit may come from customer request, vendor's internal analysis, on-site engineer, etc. Emerson's audit process description is interesting since industrial services are executed in cooperation with customer. Therefore, it is good to clearly specify what is needed from company side and what is needed from customer side. Actually this is something that should be considered for every industrial service – this will be discussed next.

The customer does many things regarding the service which the vendor does not perceive if it is not especially thought over. The process descriptions of vendors consist usually of internal process descriptions related to quality system. They should be complemented with descriptions about what is needed from the customer side before, during and after the service delivery. So-called blueprint-model is developed for this purpose and it depicts the service phases as separate visible and invisible functions. Development needs and benefits for each phase can also added to it. Blueprinting is often used e.g. by service designers. (Toivonen, 2015)

### 2.2.3 Value/benefits of component

Based on workshops and interviews Valmet identified that the following benefits are expected for customer when using the process component:

- The reliability and availability improve.
- Understanding the risk areas of the system that need actions.
- Improved process performance.
- Better usability through new features and functionality.

The following benefits are expected for Valmet when using the process component:

- Faster and more systematic analysis of customer's automation product.
- Possibility to detect business opportunities.
- Better service (and on time) due to better understanding of customer's product.
- Easier planning and follow up of life cycle activities.

**References:**

Ala-Risku, T, "Installed base information management with industrial service operations", 18th Annual Conference of Production and Operations Management Society (POMS), Dallas, TX, USA, May 4 - 7, 2007

Emerson, SOFTWARE | INSPECT, http://www2.emersonprocess.com/en-US/brands/metco/software/Inspect/Pages/INSPECT-Features.aspx (available 12.06.2015)

Emerson, FLOW AUDIT SERVICE, Service brochure, MC-001825 Rev C, 10/2013 http://www2.emersonprocess.com/siteadmincenter/PM%20Micro%20Motion%20Documents/Flow-Audit-SVC-MC-001825.pdf

Frost & Sullivan, Global Automation Services Market for Process Industries, April 2014

Honeywell, Integrated Automation Assessments (IAA), https://www.honeywellprocess.com/library/marketing/presentations/integrated-automation-assessments-overview-presentation.pdf (available 12.06.2015)

Toivonen, Marja. Opportunities and challenges in Finnish service business. In Service Economy Revolution – Finlands Growth Potential. Publications of the Ministry of Employment and Economy, 2015. pp. 68-107.

## 2.3 SERVICE-R&D COLLABORATION (PC-FIN-VTT-03)

### 2.3.1 State of the art/practice

In manufacturing companies, the product development processes are usually systematic, but the service development is often intuitive and disconnected (Hanski & al, 2012). Nonetheless, the services offered and developed should be compatible with the existing product and service portfolio. Hanski & al (2012) state that the commonly used product development models are not designed for the development of services. In the development of services, essential factors are customer-centricity and service promises to the customer (Toivonen, 2015). This often means changes in the organizational culture in traditional manufacturing companies that have focused on providing products that meet predefined specifications. According to Panesar and Markeset (2008) the development should also be iterative and flexible. The development of products and services should be defined and the systematic processes integrated (Aurich et al. 2004).

The term product-service system (PSS) appeared first in the 1990s by authors having background in the environmental viewpoint. It was then realised that the shift of focus from products that fulfil the customer need to the final customer need would lead to greater freedom in the design giving also better chances to improve sustainability. The sustainability connection of PSS is also highlighted in standards (e.g. ISO 26000, 2010). At the same time companies started to offer integrated solutions in order to meet final customer needs. Hence, the companies were able to

1. improve their innovation potential,
2. improve their position in the value chain,
3. and enhance the added value of their offerings. (Tukker & Tischner 2006)

Hanski & al (2012) gathered different definitions of product-service system (see the next table). The baseline for business development is the final functionality instead of a product that would potentially fulfil customer needs. Also, instead of taking the existing structures, routines and the position of the firm for granted, the final functionality should be provided with an open minded mindset (Hanski & al, 2012). This may enable the company to produce the offering with e.g. better use of resources and materials.

Table. Definitions of a product-service system (from Hanski & al, 2012).

| ***Author*** | ***Definition: A PSS…*** |
|---|---|
| Goedkoop et al. (1999) | is “a marketable set of products and services capable of jointly fulfilling a user’s needs” |
| Mont (2001) | is “a system of products, services, networks of actors and supporting infrastructure that is developed to be competitive, satisfy customers and be more environmentally sound than traditional business models” |
| UNEP (2002) | is “the result of an innovative strategy that shifts the centre of business from the design and sale of (physical) products alone, to the offer of product and service systems that are together able to satisfy a particular demand” |
| Tukker & Tischner (2004) | “consists of a mix of tangible products and intangible services designed and combined so that they jointly are capable of fulfilling final customer needs” |

There are very few published PSS design methodologies available, partly because the area is such new. There are some widely accepted product design methodologies and also some methodologies for service design, but the integrated view is not yet common. Nevertheless, the concept of PSS is strongly supported in academia and several practical PSS applications can be found in the literature (e.g. Tukker 2004). (Hanski et al., 2012)

Basically, when developing product-service systems (PSS), both the development of products and services should be defined and systematized and the systematic processes should then be integrated (Aurich et al., 2004). However, there is less information available how the needs of a service organization could be taken into account during product development in practice. What kind of product-service development collaboration could improve the quality and lead time of the industrial services?

The development of product-service design processes enables the companies to design solutions consisting of the best possible combination of products and services from the perspective of the customer and the supplier. The adoption of PSS design helps companies to take the services into account when developing products and vice versa. PSS design methods can decrease the lead-times of the development processes, and lead to better quality solutions as both product and service characteristics can be taken into account as early as possible in the development process. (Hanski & al., 2012)

The development of integrated product-service solutions is not without challenges (Hanski & al, 2012):

1. Product and service design are typically accomplished in different organization units.
2. Consequently services are generally planned afterwards, causing compatibility problems.
3. The business analysis has often been left for too little attention, e.g. life cycle profit evaluation or revenue logic assessment of alternative product-service systems.

The separation of service and R&D organization may cause communication problems that need to be tackled with new practices and organizational units (Lakemond & Magnusson, 2005). Lakemond & Magnusson (2005) present a case study in a company that has separated R&D and Service into own organisations and made a strategic decision to focus more to service business. The case study found the following practices that could improve the interaction between Service and R&D organisations, e.g. service organization participates to the gate-reviews from serviceability point of view, and there are external teams in both organisations representing service and R&D viewpoints.

### 2.3.2 Component experiences compared with state of the art/practice

The importance of industrial services has increased and there needs to be systematic practices/processes to support service and product development. This has been indicated also in other studies, e.g. in Lakewood & Magnusson (2005), Tukker & Tischner (2006), Aurich et al. (2004). However, there is less information available concerning how in practice the needs of the Service organization could be taken into account during product development. What kind of service/R&D collaboration could improve the quality and lead time of the industrial services? In this process component, objective is not to describe service development process but rather to try to understand and collect industrial best practices that increase the collaboration and transparency between service and R&D organizations so that customers can be better and faster serviced.

Therefore, this process component advances the state of the practice in the sector of service-R&D collaboration. The process component is especially intended for automation industry. However, some parts of the component might be applied also in other domains that produce and maintain complex long-living customer-tailored products that contain HW/SW components that are in-house developed of COTS components.

Our process component aims to describe the collaboration between Service and R&D organizations. The component has been developed in close cooperation with Valmet Automation using two cases (Upgrade-service and Audit-service) that give practical examples about the collaboration, i.e. *what the collaboration between Service and R&D organizations means in real-life industrial environment*.

The component shows that service needs has to be taken into account already in business planning phase of the product development process. Furthermore, there were roles and teams that worked between service and R&D organizations to facilitate the interaction between the organizations. The approach has some similarities to the solution that is presented in Lakewood & Magnusson (2005). Similarly their case study indicated that there were need to have organizational units that worked in between the organizations that enabled the interaction.

Based on this research, it is possible to better understand interfaces and needs between Service and R&D organizations. This research provides for other companies and research institutes that work with industrial companies the practical real-life cases how Service and R&D organizations collaborate. This component provides for other industrial companies a good ground to compare their operational environment and collaboration practices/processes with the solutions developed based on Valmet Automation and apply the collaboration practices when appropriate and applicable. For us, this study creates a basis for further research to study the service/R&D collaboration needs of the other industrial services – for instance such as preventive maintenance services, optimization services, security assessment services.

### 2.3.3 Value/benefits of component

The following benefits are expected for Valmet when using the process component:

- The voice of service organisation is taken into account in R&D that contributes to products that are easier to service and their lifecycle is coordinated.
- Better detection and understanding of lifecycle constraints in service.
- Overall transparency between R&D and Service increases.

**References:**

Aurich, J.C., Fuchs, C., DeVries, M.F. (2004). An Approach to Life Cycle Oriented Technical Service Design. CIRP Annals – Manufacturing Technology, Vol. 53, No. 1, pp. 151. ISSN 0007-8506. doi: 10.1016/S0007-8506(07)60666-0.

Cooper, R. (1993). Winning at new products: accelerating the process from idea to launch, 2nd edition. Addison-Wesley, Cambridge, MA.

Goedkoop, M., van Halen, C., te Riele, H., Rommens, P. (1999). Product service systems, ecological and economic basis. PricewaterhouseCoopers. In report: Mont, O. (2001). Clarifying the concept of product–service system. Journal of Cleaner Production 10, 2002, pp. 237–245.

ISO 26000 (2010). Guidance on social responsibility, Finnish Standards Association SFS.

Lakemond, N., Magnusson, T. Creating value through integrated product-service solutions: Integrating service and product development, Proceedings of the 21st IMP-conference, Rotterdam, Netherlands, (2005)

Mont, O. (2001). Introducing and Developing a Product-Service system (PSS) in Sweden. Lund, Sweden: IIIEE reports 2001:6, Lund University.

Panesar, S., Markeset, T. (2008). Development of a framework for industrial service innovation management and coordination. Journal of Quality in Maintenance Engineering, 14, 2, pp. 177–193.

Tukker, A., Tischner, U. (2006). Product-services as a research field: past, present and future. Reflections from a decade of research. Journal of Cleaner Production, 14, pp. 1552–1556.

Tukker, A. (2004). Eight type of product-service system: Eight ways to sustainability? Experiences from Suspronet. Business Strategy and Environment. Vol 13. Pp. 246–260.

Ulrich, K., Eppinger, S. (2003). Product design and Development. 3rd edition. New York, McGraw-Hill. 366 p. ISBN 007-247146-8.

UNEP (2002). Product-Service Systems and Sustainability. Opportunities for sustainable solutions. United Nations Environment Programme, Division of Technology Industry and Economics, Production and Consumption Branch. Paris.

## 2.4 INDUSTRIAL PROCESS (PC-FIN-VTT-05)

The value chain of embedded systems comprises the system provider, service providers as development partners or subcontractors, and tool providers for optimal leverage of development resources. Also customer interface cooperation should be improved for understanding customer needs and tailoring customer-specific solutions. This section describes the embedded systems' engineering challenges in the collaborative supplier-customer value chain. In other terms our focus is on Industrial Process, including supplier collaboration, partly the innovation activities within the manufacturing company. We do not focus in the manufacturing itself, but on the engineering and supplier collaboration challenges. This process component defines lean and agile method for industrial company to manage its manufacturing of customer specific embedded systems.

### 2.4.1 State of the art/practice

All business functions should be integrated in the development of embedded systems, not only R&D (Löfving et al. 2013). In many cases, SMEs lack the necessary resources (staff, competencies, facilities and finances) to provide entire product-service offerings themselves that their customers require. Support for SMEs' own competencies is needed from collaboration partners. SMEs are usually flexible, flat in terms of organisational structure, benefit from fast decision-making and they have close collaboration with supply chain partners. SMEs require descriptive, easy-to-use and simple development frameworks. Successful SME manufacturing companies develop an intimate relationship with their customers and are more capable than larger firms in adjusting to customer preferences (Kim et al. 2008).

Supplier collaboration is crucial in an industrial process, especially when manufacturing is outsourced or when a customer solution is created and all tailored parts should be designed and produced together with partners. There is a range of practices and operation models, but the streamlining of processes is needed in many cases. The industrial process should manage the manufacturing, tailor-made solutions realisation and quality control of wrapping machines.

A customer order is received by the sales team and then the industrial process begins. The factory sales team checks the customer order and ensure manufacturing capacity and they also coordinate the required tailor-made solutions with R&D. The industrial process includes order management, customer-specific design, manufacturing, and factory tests. Delivery of the final product is not in focus. New methods for collaboration are needed for Sales and Factory Sales in order to respond to customer needs more effectively. In this part, the following process development will be carried out:

- Production resource planning
- Supplier collaboration development
- Analysis of existing Sales and Factory Sales collaboration
- New process and operation model for tailor-made multi-discipline engineering (in cases of customer non-catalogue requests)

The aim is to enhance cooperation between the company's internal factory sales, production, supplier collaboration and also global sales organisations. Petersen et al (2005) have addressed the question of when a supplier becomes involved in the new product development process. They present at least five different phases of product development in which suppliers might be involved (see next figure). According to our case study, a strategic supplier should be integrated in every part of product life cycle to ensure effective collaboration and knowledge sharing, but also to give more responsibility to strategic suppliers.

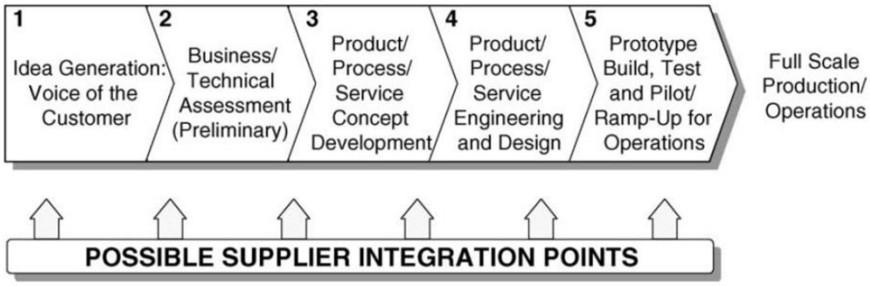


Figure. Supplier integration points (Petersen et al. 2005)

We target the next generation of operation processes in production with a faster response to customer needs and the effective utilisation of process support tools and methodologies. The process development in selected business functions is a requirement for successful business.

**Different Innovations related to industrial process**

The Oslo Manual (2005) defines four types of innovations that encompass a wide range of changes in firms' activities: product innovations, process innovations, organisational innovations and marketing innovations. Changes in the specific products/services offered to the customers correspond to product innovation, and changes to the mode by which these are created and delivered correspond to process innovation (Huang and Rise, 2012). Product innovations involve significant changes in the capabilities of goods or services, where both entirely new goods and services, and significant improvements to existing products are included. Process innovations represent significant changes in production and delivery methods. Organisational innovations refer to the implementation of new organisational methods. These may be changes in business practices, in workplace organisation or in the firm's external relations. Marketing innovations involve the implementation of new marketing methods. These may include changes in product design and packaging, in product promotion and placement, or in methods for pricing goods and services.

Increasingly companies must collaborate with their network partners in order to succeed in their innovation tasks (Landsperger and Spieth, 2011). Our research combines these approaches, collaborative networking, within product and process innovations. We have also included a partly organisational innovation viewpoint, while developing new organisational structures and outsourcing parts of product innovations for the supplier network. In this research, we do not focus on marketing innovations, which is a limitation of the study.

Although the market and economic impacts of process innovation are often considered to be significant, as the introduction of new products/services, process innovation is often downplayed in the innovation literature generally (Huang and Rise, 2012). Product innovation focuses on what is produced, while process innovation concerns itself with how existing products/ services are produced (ibid.). When managers adopt innovation strategies to introduce new processes in their organisations, they should pay attention to the utilisation of internal resources and capabilities (Huang and Rise, 2012). The organization's attention should also be drawn to external knowledge sourcing and technology acquisition, when the organization's own competences or resources are limited.

The following sub-sections define specific challenges on product, process and organisational innovations within embedded systems development within the industrial process.

*Product innovations*
In manufacturing industries, the product is still the core of the business although services are a growing area from an economics point of view. The product innovation definition in the Oslo Manual (2005) combines product and service improvements under the same heading. Product innovation is about the introduction of a good or service that is new or significantly improved with respect to its characteristics or intended uses (ibid.). In our study, the product innovation process is called Product Lifecycle Management, consisting of mechanical, electrical and software engineering processes. Product Lifecycle Management ensures embedded system functionalities and it can be installed, maintained and operated as planned. This includes significant improvements in technical specifications, components and materials, software, user friendliness and other functional characteristics. Product innovation is a continuing process when searching, developing and introducing new solutions to markets.

*Process innovations*
According to the Oslo Manual (2005), a process innovation is the implementation of a new or significantly improved production or delivery method, which includes significant changes in techniques, equipment and/or software. In our study, multiple IT tools are used for engineering, and a future challenge will be to harmonise tools with the supplier network but also to introduce new modern IT tools in practice. Process innovation should include the following aspects:

- Searching for and introducing new manufacturing technologies
- Analysis and the effective use of design tools internally and in the supplier network
- Effective global purchasing of materials and services

*Organisational innovations*
An organisational innovation is the implementation of a new organisational method in the firm's business practices, workplace organisation or external relations (Oslo Manual, 2005). New organisational methods in a firm's external relations involve the implementation of new ways of organising relations with other firms or public institutions, such as the establishment of new types of collaborations and integration with different stakeholders (ibid.). SMEs are usually flexible, flat in terms of organisational structure, benefit from fast decision-making and have close collaboration with supply chain partners. Successful SME manufacturing companies develop an intimate relationship with their customers and are more capable than larger firms of adjusting to customer preferences (Kim et al. 2008). SMEs are forced to focus their innovative efforts, which require the outsourcing of some areas (Landsperger and Spieth, 2011). Outsourcing of non-core areas in innovation processes has been a widely discussed topic in operation management and innovation literature (e.g. Maglio and Spohrer, 2013).

**Specialities in Silicon Valley, USA**
Our research has included visiting research period in Silicon Valley, which is a nickname for the South Bay portion of the San Francisco Bay Area in Northern California, United States. In this chapter we present some specialities from Silicon Valley area related on product engineering. Silicon Valley has the highest concentration of high-tech workers of any metropolitan area, with 285.9 out of every 1,000 private-sector workers. Thousands of high technology companies are headquartered in Silicon Valley as well as thousands of small start-ups. Statistically, only one out of a hundred start-ups succeeded in Silicon Valley, so competition is really high. Most of the best engineering and technology universities and graduate schools are located in the Silicon Valley area (see next table), so the world's best learning environment can be found in this area.

Table. *Engineering school rankings 2014*

| **World Top 100 universities for engineering and technology 2013-2014** (Times Higher Education World University Rankings, 2014) | **The top graduate engineering schools for industrial and manufacturing engineering in US, 2014** (U.S. News Best Engineering Schools 2014) |
|---|---|
| #1 Massachusetts Institute of Technology (MIT), United States | #1 Georgia Institute of Technology Atlanta, GA |
| #2 Stanford University, United States | #2 University of Michigan, Ann Arbor, MI |
| #3 University of California, Berkeley, United States | #3 Northwestern University (McCormick) Evanston, IL |
| #4 California Institute of Technology (Caltech), United States | #4 University of California, Berkeley, Berkeley, CA |
| #5 Princeton University, United States | #5 Stanford University, Stanford, CA |

Density of high tech and start-up companies, but also the level of engineering teaching and research were the main reasons for the visiting research period at San Jose.

Engineering and innovation practices include many models and approaches, which can be learned in school. Stanford University Mechanical Engineering Department is teaching different engineering methods by using Method Cards (see next figure). There are many examples of cards used to assist or provide structure to the design process, yet there has not yet been a thorough articulation of the strengths and weaknesses of the various examples (Wölfel and Merritt, 2013). Cards have been used widely by designers to make the design process visible and less abstract and to serve as communication tools between members of the design team and users (ibid.).

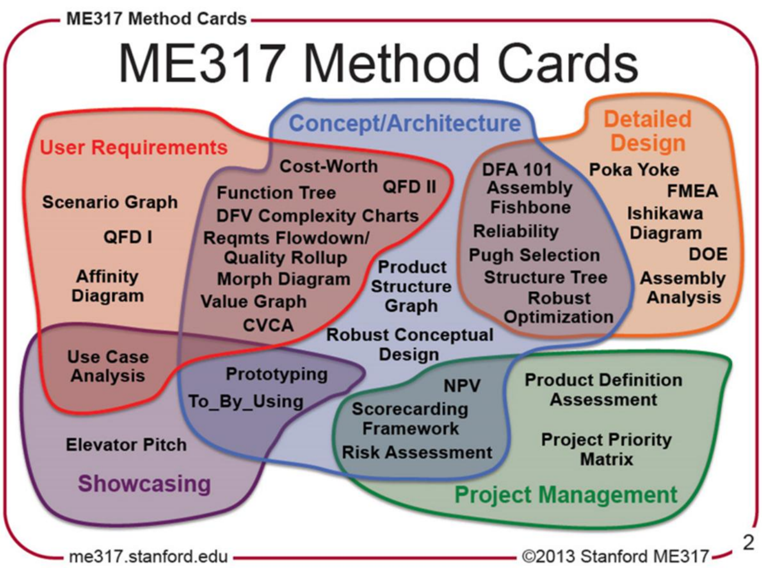


Figure. Collection of Stanford University engineering methods.

According our findings from Silicon Valley, the majority of start-ups are telling their story and convincing financiers and other stakeholders with the kind of methods Stanford University teaches.

SME manufacturers are still producing a lot of design and engineering without drawings or planning, but just in “piece of paper”. One can often hear “we have always done this way”, and it is evident that new methods are not in use or implemented in practice.

### 2.4.2 Component experiences compared with state of the art/practice

*Innovations perspective*

The challenges faced by manufacturing companies are mostly related to the product, process and organizational innovations. Manufacturing SMEs providing embedded systems should have a wide range of knowledge, resources and experience during the innovation. In practice, the resource required might not always be available internally, but might not easily be found in the markets. Partners with different backgrounds enrich the network by adding valuable competences and know-how in innovation processes. SMEs are of necessity more specialised in their activities, which increases the importance of efficient interaction with other firms and public research institutions for R&D, but also the exchange of knowledge for commercialisation and marketing activities (Oslo Manual, 2005). It has been argued that finance can be a determining factor for innovation in SMEs, which often lack internal funds to conduct innovation projects and have much more difficulty obtaining external funding than larger firms (ibid). Tekes, the Finnish Funding Agency for Innovation is the most important publicly funded expert organisation for financing research, development and innovation in Finland. Today, Tekes funding is more targeted on SMEs, rather than large companies, which in some cases brings a huge advantage for innovative SMEs in today’s turbulent market and economic situation.

Companies should understand their business processes, so that each individual realises their position and relation to other process parts. In the case of SMEs, processes are not typically well documented. Our case study shows that even SMEs might have complex processes. The generic framework for process definition helps in the realisation of process descriptions. For successful innovation activity, managers should involve all business functions, the competent resources from the organisation but also from the network. The process should start with the selection of business functions and then the development of business processes within those functions. The model also creates knowledge around the relationships between different organisational functions and collaboration partners. In our study, the case company did not have any processes described before our research. During our workshops, we discussed with experts and defined process charts. It was really valuable to have a common platform (generic framework) when describing process charts. During the process definitions, we found useless loops in the processes and then streamlined the method of collaboration within the case company. In this research, we do not focus on the marketing innovations, which is the limitation of the study. In future research, the framework should be used for process improvement. As our case company defined their processes the first time, we did not have a clear understanding of how processes will be improved when streamlining is carried out and cutting unused loops from innovation processes. The measurement and key performance indicators should be defined as a part of the process framework for better calculation of improvements.

Manufacturing firms have become increasingly interested in creating value by delivering more complex product-service offerings (Kowalkowski et al. 2013). From the academic perspective, it is important to understand the embedded systems characteristics and the meaning of collaborative engineering processes for the success of the manufacturing firms. In a collaborative environment, both supplier and customer co-development are today widely discussed in innovation, marketing and service science literature (e.g. Maglio and Spohrer, 2013). Customers are dealt with at all parts of the development, from segmentation and requirements to the sales, delivery and operation support processes. SMEs need supplier collaboration in R&D, delivery of materials, components and parts. SMEs with limited resources may have outsourced many operations; if this is the case, it is crucial for them to have close collaboration with suppliers.

### 2.4.3 Value/benefits of component

When analysing the case company's business environment, we realised that the most important business functions are: 1) product life cycle management, 2) consulting sales processes, and 3) the industrial process. Particularly in terms of collaboration with suppliers and customers, these business functions were most meaningful. We utilised the PROMES process framework for the definition of main processes within the selected business functions. The framework, its elements, instances and relationship structure helped the definition of processes. The framework provides a modularised process model that supports basic process composition activities, which include:

- Explicit support for multiple, synchronised life cycles along the value chain of organisations in order to enhance multi-organisational and multi-disciplinary development (as they are required in embedded systems), and
- Support for modern reuse-centric multi-life cycle approaches like product line engineering or model-driven development (which rely on multiple life cycles even within software development in a single organisation).

Applying the PROMES process framework will enable industry to shorten development cycles and hit market windows by reducing time-to-market. When developing processes according to the proposed framework, it makes company's processes more predictable and improves the quality and reliability of products. Overall businesses can react more quickly to changes in the market when processes, responsibilities and actors are clearly defined. Our proposed embedded systems engineering model takes these limited resources into account and therefore can be considered suitable, especially for SMEs. Due to contextual factors such as organisational culture, management culture, political issues, etc., large companies may have other requirements, which lead to the choice of other development frameworks. We found that ICT, different tools (like CAD systems) and documentation (like minutes, plans and drawings) have a great impact on how the supplier and customer interact with each other. The management of IT is critical to embedded system value chains.

**References:**

Charalampidou, S., Ampatzoglou, A., Avgeriou, P. (2014): "A process framework for embedded systems engineering", Proceedings 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2014), Verona, Italy, August 27–29, 2014.

Heath, S. (2003): "Embedded systems design". EDN series for design engineers (2 ed.). Newnes. ISBN 978-0-7506-5546-0

Huang, F. and Rice, J. (2011): "Openness in Product and Process Innovation", International Journal of Innovation Management, Vol. 16, No. 04 (2012)

Kim, K. S., Knotts, T. L., Jones S. (2008): “Characterizing viability of small manufacturing enterprises (SME) in the market”, Expert Systems with Applications, Volume 34, Issue 1, January 2008, Pages 128–134

Kruchten, P. (2000): “The Rational Unified Process: An Introduction”, Addison-Wesley Longman Publishing, 2nd Edition, Boston, MA, USA. 2000.

Landsperger, J. and Spieth, P. (2011): “Managing Innovation Networks in the Industrial Goods Sector”, International Journal of Innovation Management, Vol. 15, No. 06, pp. 1209–1241 (2011)

Maglio, P., Spohrer, J. (2013): “A service science perspective on business model innovation”, Industrial Marketing Management 42 (2013), pp. 665–670.

Oslo Manual (2005): “Guidelines for Collecting and Interpreting Innovation Data”, Third edition. OECD/EUROPEAN COMMUNITIES 2005, ISBN 92-64-4 01308-3

Alam I. and Perry, C. (2002) A customer-oriented new service development process, Journal of Services Marketing, Vol. 16, No. 6, pp.515 - 534.

Bullinger, H. J., Fähnrich, K. P., Meiren, T. (2003) Service engineering-methodical development of new service products, International Journal Production Economics. 85, pp. 275- 287.

Hemilä, J., Vilko, J., Kallionpää, E., Rantala, J. (2014) Value Creation in Product–Service Supply Networks. The proceedings of the 19th International Symposium on Logistics (ISL2014): Designing Responsible and Innovative Global Supply Chains. Ho Chi Minh City, Vietnam, 6-9th July 2014

Jensen, J. L., Rodgers, R. (2001) Cumulating the intellectual gold of case study research, Public Administration Review, Vol. 61, No. 2, pp. 235-260.

Kelly, D., Storey, C. (2000) New service development: initiation strategies, International Journal of Service Industry Management, Vol. 11, No. 1, pp. 45 - 63.

Kim, K. S., Knotts, T. L., Jones S. (2008) Characterizing viability of small manufacturing enterprises (SME) in the market, Expert Systems with Applications, Volume 34, Issue 1, January 2008, Pages 128-134

Kowalkowski, C., Witell, L., Gustafsson, A. (2013) Any way goes: Identifying value constellations for service infusion in SMEs, Industrial Marketing Management, Vol. 42, No. 1, January 2013, pp. 18-30.

Kruchten, P. (2000), The Rational Unified Process: An Introduction, Addison-Wesley Longman Publishing, 2nd Edition, Boston, MA, USA. 2000.

Löfving, M., Säfsten, K., Winroth, M. (2014) Manufacturing strategy frameworks suitable for SMEs, Journal of Manufacturing Technology Management, Vol. 25 Iss: 1, pp.7 – 26

Marsillac, E., Roh, J. (2014) Connecting product design, process and supply chain decisions to strengthen global supply chain capabilities, International Journal of Production Economics, 147(2014), pp. 317–329

Petersen, K., Handfield, R., Ragatz, G. (2005) Supplier integration into new product development: coordinating product, process and supply chain design. Journal of Operations Management 23 (2005) 371–388

## 2.5 ARCHITECTURAL KNOWLEDGE MANAGEMENT (PC-NL-RUG-01)

### 2.5.1 State of the art/practice

The importance of preserving architectural design rationale along with the structural aspects of a system has been long acknowledged [1], [2]. Nevertheless, capturing design rationale in an effective and efficient way has been an open issue ever since. In 2004, Bosch recognized that the documentation of architecture decisions and the rationale behind them play a major role in preserving design rationale [3]. Therefore, he argued that architecture decisions should be treated a first-class architecture elements and thus, should be explicitly documented. In the following years, researchers associated the explicit documentation of decisions with a multitude of benefits, such as avoiding knowledge vaporization, supporting change impact estimation, increasing system understanding, improving knowledge sharing, and facilitating architecture evaluation [4]–[6].

Despite the benefits, architecture decisions are often not explicitly documented in industrial practice. There have been attempts to close this gap between industrial practice and the benefits demonstrated by research, through a growing number of architectural knowledge management tools [7]. However, most of these tools are developed in academic, non-commercial contexts, and therefore often do not meet the expectations and demands of the industry [8].

In the following, five examples of AKM tools will be discussed: Knowledge Architect, Architecture Design Decision Support System (ADDSS), Process-centric Architecture Knowledge Management Environment (PAKME), and Archium. The presented approaches differ in the offered features, implementation style, e.g., a web application, an extension to a programming language, or client-server approach, and application context.

- The *Knowledge Architect* tool suite, proposed by Liang et al. [9], is based on a common knowledge repository, which is accessed by clients to store and retrieve architectural knowledge (AK) [10]. Amongst others, Knowledge Architect provides Word and Excel plugins to annotate AK in existing documents. The tool portfolio includes an explorer to analyse and visualize relationships between AK entities; a Word plug-in that supports capturing AK by annotating relevant text passages and using AK from the repository in the document; and an Excel plug-in that supports capturing AK from quantitative analysis models.

- *ADDSS* is a web-based tool for storing, managing, and documenting architectural decisions along with candidate architectures [11]. The tool visualizes the evolution of AK over time by capturing AK and architecture models in an iterative process. It advocates the use of mandatory and optional attributes for characterizing the design decisions and provides a combined codification-personalization strategy [7]. Furthermore, the tool provides traceability between requirements and design decisions and offers a query system to retrieve information about related requirements, decisions, and architecture artefacts stored in the tool. ADDSS also allows to generate PDF reports containing a detailed description of the architecture and the decisions [10]
- *PAKME* is also a web-based tool that has been built on top of an open source groupware platform [12]. PAKME's features can be categorized into knowledge acquisition, knowledge maintenance, knowledge retrieval, and knowledge presentation [7]. It aims at providing knowledge management support for the software architecture process. It consists of three components; web-based user interface, knowledge management, search, and reporting.

- *Archium*, proposed by Jansen et al. [13] takes a different approach by defining a component language extension for Java, which allows modelling design decisions and relating those to components. This makes the source code to act as an architectural knowledge repository and an implementation of a system at the same time.

Several studies analysed and compared architectural knowledge management approaches. In 2010, Tang et al. [7], conducted a study that compared five architectural knowledge management tools based on a framework, which represented how AK management is used in the architecture life-cycle . They found that there is no tool that supports all activities of the architecture lifecycle, i.e., analysis, synthesis, evaluation, implementation, and maintenance. The results also show that the capability to provide different perspectives on AK is limited, as the tools do not support the concept of views and viewpoints. Furthermore, it was noted that knowledge-sharing features are not fully supported by the tools yet.

This finding is also supported by an earlier study conducted by Farenhorst et al. in 2007 [14]. In this study, the authors analyzed five tools from academia and industry from a knowledge sharing perspective. The results show that AK personalization strategies are not sufficiently supported and that the tools do not provide the possibility to generate stakeholder-specific content. The latter was also identified by Tang et al. [7]

In 2009, Liang and Avgeriou [10] surveyed nine existing tools to what extent they support the following categories of use cases: consuming AK, knowledge management, intelligent support, and producing AK. They found that the tools lack of the ability to identify stakeholders, to apply decisions to produce the design (synthesis), to evaluate AK, and to support the assessment design maturity.

### 2.5.2 Component experiences compared with state of the art/practice

Parts of the process component have been validated in studies, especially the documentation subcomponent. Van Heesch et al. [15], [16] demonstrated that the effort of creating the decision views is reasonable and that the proposed decision documentation framework is particularly useful for communicating decisions and reviewing architectures. More importantly, he also found that decision view creation also has an immediate benefit for the architects during the design process, as they support a systematic reasoning process [15].

In an industrial case study, we assessed to what extent and why software architects intend to use our approach. We therefore investigated the perceived usefulness, perceived ease-of-use, and contextual factors that influence the architect's intention to use the tool. We found that the tool has a high relevance for software architects, that it increases the quality of decision documentation and that it improves productivity of architects in various ways. With respect to ease-of-use, the study identified the need for guidelines and recommendations. With respect to contextual factors, we found several factors that influence the intention to use the add-in in an industrial environment.

The experiences of applying the process component in the industrial case NL-OCE are described in D3.3. The observations so far confirms the absence of explicit approaches that allow capturing decision and design rationale in an efficient and effective way. So far, we further explored decision-sharing and decision-reuse in the context of embedded systems engineering projects.

### 2.5.3 Value/benefits of component

The following benefits from applying the process component are expected and partially confirmed:

- Preservation of important design rationale.
- Better support for communication and sharing of decisions in a multi-* environment.
- Improved maintenance and evolution through knowledge preservation and increased system understanding.
- Improved reuse of components, models and decisions due to the availability of the original design rationale.
- Improved quality of architecture decisions via a systematic reasoning process.
- Improved support for change impact analysis and architecture reviews.

**References:**

[1] D. E. Perry and A. L. Wolf, “Foundations for the study of software architecture,” ACM SIGSOFT Softw. Eng. Notes, vol. 17, no. 4, pp. 40–52, 1992.

[2] P. Kruchten, “Architectural blueprints---The ``4+ 1'' view model of software architecture,” IEEE Softw., vol. 12, no. 6, pp. 42–50, 1995.

[3] J. Bosch, “Software architecture: The next step,” Softw. Archit., pp. 194–199, 2004.

[4] P. Kruchten, R. Capilla, and J. C. Dueñas, “The Decision View’s Role in Software Architecture Practice,” IEEE Softw., vol. 26, no. 2, 2009.

[5] J. Tyree and A. Akerman, “Architecture decisions: demystifying architecture,” Software, IEEE, vol. 22, no. 2, pp. 19–27, 2005.

[6] J. S. Van Der Ven, A. Jansen, P. Avgeriou, and D. K. Hammer, “Using Architectural Decisions,” in Proceedings of the 2nd International Conference on the Quality of Software Architectures (QoSA), 2006.

[7] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, and M. A. Babar, “A comparative study of architecture knowledge management tools,” J. Syst. Softw., vol. 83, no. 3, pp. 352–370, 2010.

[8] H. Koziolek and T. Goldschmidt, “Tool-Driven Technology Transfer to Support Software Architecture Decisions,” in SE14, 2013, pp. 1–6.

[9] P. Liang, A. Jansen, and P. Avgeriou, “Knowledge Architect: A Tool Suite for Managing Software Architecture Knowledge (Technical Report RUG-SEARCH-09-L01),” 2009.

[10] P. Liang and P. Avgeriou, “Tools and Technologies for Architecture Knowledge Management ,” Springer-Verlag, Berlin Heidelberg, 2009.

[11] R. Capilla, F. Nava, J. Montes, and C. Carillo, “ADDSS: Architecture Design Decision Support System Tool,” 2008, pp. 1–2.

[12] M. A. Babar and I. Gorton, “A Tool for Managing Software Architecture Knowledge,” in Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent, 2007. SHARK/ADI ’07: ICSE Workshops 2007. Second Workshop on, 2007, p. 11.

[13] A. Jansen, J. S. Van Der Ven, P. Avgeriou, and D. K. Hammer, “Tool support for architectural decisions,” in Software Architecture, 2007. WICSA’07. The Working IEEE/IFIP Conference on, 2007, pp. 1–10.

[14] R. Farenhorst, P. Lago, and H. van Vliet, “Effective tool support for architectural knowledge sharing BT - Software Architecture,” in Software Architecture, Springer Berlin Heidelberg, 2007, pp. 123–138.

[15] U. van Heesch, P. Avgeriou, and R. Hilliard, “A documentation framework for architecture decisions,” J. Syst. Softw., vol. 85, no. 4, 2012.

[16] U. van Heesch, P. Avgeriou, and R. Hilliard, “Forces on Architecture Decisions - A Viewpoint,” in Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on, 2012, pp. 101–110.

[17] C. Manteuffel, D. Tofan, H. Koziolek, T. Goldschmidt, and P. Avgeriou, “Industrial Implementation of a Documentation Framework for Architectural Decisions,” in Software Architecture (WICSA), 2014 IEEE/IFIP Conference on, 2014, pp. 225–234.

## 2.6 REFACTORING AND TRACEABILITY (PC-NL-RUG-02)

### 2.6.1 State of the art/practice

One of the basic characteristics of software is its evolving nature. During evolution, software is enhanced, modified, or adapted to new environments. As a result it tends to drift away from its original design, which in most of the cases leads to degrading quality (Mens and Tourwe, 2004). To handle this problem, software engineers need to identify parts of the source code that suffer from low quality, with the aim of incrementally restoring it. One of the most established ways to improve quality, after the development of the software, is the application of software refactorings. According to Fowler et al. (1999) refactorings are defined as transformations that improve certain quality attributes, but do not affect the external behaviour of the software. In their seminal book on refactorings Fowler et al. (1999) describe more than 70 object-oriented refactoring techniques for resolving potential bad smells.

In the software engineering literature refactorings, and the corresponding bad smells that they resolve, have been heavily studied, resulting to more than 40 papers until the end of 2014. According to two secondary studies that provide an overview of the current state of research and practice ((Mens and Tourwe, 2004) and (Zhang et al., 2011)), the main challenges in the domain of refactorings are as follows:

- Creation of methods/tools for detecting code bad smells in source code;
- Improvement of the understanding of code bad smells; and
- Suggestion/enhancement of methods/tools for refactoring code bad smells.

Additionally, Zhang et al. (2011) report the long method smell among the most commonly studied bad smells, whereas Chatzigeorgiou and Manakos (2014), suggest that long method is the most frequently occurring and persistent bad smell, based on a study regarding the application frequency of four common bad smells in practice (i.e., open source projects).

The aforementioned observations constitute long method as a bad smell that is interesting to focus on. Specifically, it would be of high interest to attempt to address the aforementioned challenges, by proposing methods that aim at the identification of long methods, as well as techniques and tools for refactoring them. One of the most common ways of resolving the long method smell is the application of the extract method refactoring (Fowler et al., 1999). Extract method is usually based on the identification of different (or at least semantically distant) functionalities that are implemented in the same method. In the literature, we have identified five studies that aim at identifying extract method opportunities from source code (Tsantalis and Chatzigeorgiou, 2011; Yang et al., 2009; Meananeatra et al., 2011; Yoshida et al., 2012; Silva at al., 2014). Tsantalis and Chatzigeorgiou, (2011) suggest an approach that uses complete computational slices (i.e., the code fragments that are cooperating in order to compute the value of a variable) as potential extract methods. Yang et al. (2009) suggest that the code of the long method should be decomposed either based on control structures (i.e. for-statements, if-statements, etc.) or code styling (i.e., blank lines in the code). The approach suggests that the composition of extract method opportunities should basically consider the size of the created method, by setting appropriate thresholds. Later the calculation of coupling metrics is used in order to rank the extract method opportunities. Meananeatra et al. (2011) and Yoshida et al. (2012) propose the decomposition of source code using abstract syntax tree (i.e., data flow and control flow graphs) and the proposition of extract method opportunities based on the calculation of complexity and/or cohesion metrics. Finally Silva at al. (2014) proposes the use of the abstract syntax tree and the creation of all possible combinations of lines within the blocks.

Finally, a very significant aspect to be discussed is the easy application of refactorings by practitioners. Taking as an example a large complex system of poor quality, which suffers from the existence of long methods, and is under the refactoring process, we can expect that the software engineers would try to identify concrete functionalities so as to extract them from the body of the long method. However, this process could be simplified by the existence of traces (i.e., links) between requirements and source code. Software traceability aims at creating relations between artefacts developed during the software lifecycle (e.g., requirement specifications, software analysis, designs, test models, source code) and aims at enhancing understandability of the software system (Spanoudakis and Zisman, 2005). In the literature, many studies can be found on software traceability. However, since the scope of this component is on the link of requirements with artefacts produced in later development phases, we are only interested in the state of research and practice on requirements traceability. Specifically, we have identified a review on traceability between artefacts using natural language processing techniques (Borg et al., 2014), which suggests that until now 32 studies have attempted to link requirements with source code artefacts. However, the majority of the proposed techniques suffer from either low recall or low precision rates (e.g., (Antoniol et al., 2002)).

### 2.6.2 Component experiences compared with state of the art/practice

This component is a novel approach for providing traceability links between functionalities (i.e., requirements) and source code (i.e., set of lines of code), which in turn leads to identifying extract method opportunities. The basis for the construction of the method is the single responsibility principle (SPR), as introduced by Martin (2003). SRP is an object-oriented principle stating that every module should have exactly one responsibility (reasons for change) and all its sub-functionalities should be aligned with it, having no or small semantic distance. (Martin, 2003). Specifically, we have developed an algorithm, which identifies fragments of code that potentially collaborate in order to provide a unique functionality, by calculating the cohesion between sequential lines of code. The approach aims at reducing the size of the initial method, by extracting cohesive fragments into new methods, improving the understandability and the quality of the code, whereas at the same time ensures the tracing of functionalities to modular pieces of code (i.e., methods). To validate the ability of the proposed algorithm to identify highly cohesive parts of a long method that concern a specific functionality, we have conducted a case study at Océ. This component advances the state of practice, since it overcomes the limitations of the existing approaches that identify extract method opportunities, which are presented below:

- (Tsantalis and Chatzigeorgiou, 2011) – The study calculates the complete computational slice of a variable, considering only cases when the variable changes value, without taking into account the lines where the variable is used. However, such lines might participate in a code fragment serving a unique functionality.
- (Yang et al., 2009) – The study depends on code styling assumptions, like the separation of code fragments, which concern unique functionalities, with an empty line. This is considered as a threat to potential failure, in cases that the assumption is not valid.
- (Yoshida et al., 2012) and (Meananeatra et al., 2011) – The studies use the abstract syntax tree to decompose source code to structural nodes. However, these structural nodes may consist of multiple statements, which cannot be further decomposed. Thus, potential extract method opportunities, which capture unique functionalities, may not be identified.
- (Silva at al., 2014) – The study suggests the identification of the exhaustive set of all possible combination of continuous lines within the syntax blocks. However, such an

approach may cause an enormous number of extract method opportunities which have not been selected based on any quality characteristic. Although most of the unique functionalities may be identified, this exhaustive tactic is not considered as optimal.

### 2.6.3 Value/benefits of component

- Better reuse, due to improved system understandability and the fact that code modules will be linked to concrete features.
- Improved quality, with respect to reusability, modularity and understandability.
- Reduced time-to-market, due to better software quality. New requirements can be more quickly implemented in a well-structured and easy to understand software.
- Cost saving, in terms of reduced effort/working hours. All three aforementioned quality improvements (reusability, modularity and understandability) are expected to simplify the work on the refactored software.

**References:**

G. Antoniol, G. Canfora, and G. Casazza, A. De Lucia, E. Merlo, "Recovering Traceability Links between Code and Documentation", IEEE Transactions on Software Engineering, 28(10), pp.970-983, October 2002

T. Mens and T. Tourwe, "A survey of software refactoring," IEEE Transactions on Software Engineering, 30(2), pp.126-139, February 2004

M. Fowler, K. Beck, J. Brant, W. Opdyke, and D.Roberts, “Refactoring: Improving the Design of Existing Code”, Addison-Wesley Professional, 1 edition. July 1999.

M. Zhang, T. Hall, and N. Baddoo, “Code Bad Smells: a review of current knowledge.” Journal of Software Maintenance and Evolution, 23(3), pp.179-202, April 2011.

A. Chatzigeorgiou and A. Manakos, “Investigating the evolution of code smells in object-oriented systems”, Innovations in Systems and Software Engineering, 10(1) , pp.3-18. March 2014.

N. Tsantalis and A. Chatzigeorgiou, “Identification of extract method refactoring opportunities for the decomposition of methods”, Journal of Systems and Software, 84 (10), pp.1757-1782. October 2011.

D. Silva, R. Terra, and M. T. Valente, “Recommending automated extract method refactorings”. In Proceedings of the 22nd International Conference on Program Comprehension (ICPC 2014), ACM, New York, NY, USA, pp.146-156. 2014.

P. Meananeatra, S. Rongviriyapanish, and T. Apiwattanapong, “Using software metrics to select refactoring for long method bad smell”, 8th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp.492-495. 17-19 May 2011.

N. Yoshida, M. Kinoshita, and H. Iida, “A cohesion metric approach to dividing source code into functional segments to improve maintainability”, 16th European Conference on Software Maintenance and Reengineering (CSMR), pp.365-370. 27-30 March 2012.

L. Yang, H. Liu, and Z. Niu, “Identifying Fragments to be Extracted from Long Methods”, Asia-Pacific Software Engineering Conference (APSEC '09), pp.43-49. 1-3 December 2009.

M. Borg, P. Runeson, and A. Ardö, "Recovering from a decade: a systematic mapping of information, retrieval approaches to software traceability", Empirical Software Engineering, 19(6), December 2014.

R.C. Martin "Agile software development: principles, patterns and practices", Prentice Hall, New Jersey. 2003.

G. Spanoudakis and A. Zisman, "Software traceability: a roadmap." In: Chang, S.K. (ed.) Handbook of Software Engineering and Knowledge Engineering, vol. 3, pp. 395-428, World Scientific, Singapore, 2005.

## 2.7 DESIGN FRAMEWORK (DF) METHOD (PC-NL-TNOESI-01)

### 2.7.1 State of the art/practice

Architecting an (embedded) system design in general is a delicate process involving multiple disciplines which are typically represented by mono-disciplinary architects. This process requires sharing information on relevant cross-disciplinary system interactions, and dealing with trade-offs between different architectural relevant system properties, up to the level that a consistent system design can be established. In practise we see for most domains that this architecting is still a pure human process predominated by informal interaction and communication (http://www.gaudisite.nl/). Such informal processes come with certain deficiencies like no proper documentation of design choices (lack of traceability), inconsistent views between the disciplines, lack of a cross-discipline quantitative validation for relevant design parameters, etc.

In other domains, typically driven from strict non-functional requirements (e.g. safety, reliability, etc.), one can observe formal approaches starting to emerge. In these domains methods such as PLM tools, SysML tools, computing software and configuration management tools start to play a role. In even more mature domains (e.g. automotive, aerospace, ..) we see domain specific methods focussing on for that domain relevant aspects, for example focussing on a standardized workflow, on a dominant discipline, or on a specific set of non-functional requirements. In general such a specialization of a method is less suitable to be applied in a different domain characterized by different focus areas.

### 2.7.2 Component experiences compared with state of the art/practice

The Design Framework is developed as a generic method (i) to support the architectural process as effective and light-weighted possible, and (ii) to fulfil the needs of system architects cooperating in a predominant multi-disciplinary design environment, especially during the shaping phases of system development. For the Design Framework method's proof of concept the following intended value proposition - positioning the method deliberately against known deficiencies in existing state of the art methods, tools, and practices - has been taken into mind.

The Design Framework method explicitly pursues:

- Promoting the communication between domain architects by enabling of (formal) exchange of information between multiple domains, including non-software related system aspects
  (.. and thereby being not really suitable for software-only domains).
- Establishing and guarding architectural consistency over multiple disciplines, explicitly from the need to provide a good system design overview.
- Supporting the use of models.
- Providing a single point of (architectural) truth of the developed system
  .. thereby becoming the single source of consistent documentation.
- Supporting model interoperability of heterogeneous models (e.g. co-simulation of multiple models within one simulation) using an approach supporting quantitative validation
  .. where the focus is on structural, not behavioural, models.
- Supporting traceability of the architectural relevant design decisions.
- Being easy to use for architects from different (non-software) backgrounds.
- Being explicitly positioned in the shaping (mostly early) architectural and system design phases

.. where architectural teams deal with a typical relative small but design-decisive set of key relationships spread over multiple disciplines.

Note that information in the remainder of this section is based on experiences with existing (commercial) tools, where tool vendors' promotional material, such as websites, white papers and brochures are used as source of information. Therefore the extent and value claim of the information varies highly depending on the source of the information. The list of discussed methods and example tools is based on a small inquire done within the industrial network of TNO-ESI which includes innovative world-leading high-tech industry players (Philips Healthcare, Philips Lighting, ASML, Thales, NXP, Océ).

**PLM tools (example: Siemens TeamCenter)**

Product Lifecycle Management (PLM) tools are focusing on providing an information management system solution integrating data, processes, business systems and, ultimately, people in predominant large-scale enterprises. Ideally these solution allow users to manage this information throughout the entire lifecycle of a product efficiently and cost-effectively, from ideation, design and manufacture, through service and disposal.

In comparison to the Design Framework method PLM tools are mainly focused on the product release processes, and communication of information between different functional parts of and/or product departments within the organization. PLM tools are typical characterized by a low user friendliness and accessibility level requiring a steep detailed learning curve. For a long time their supported processes have from its origin been document-based.

**SysML tools (example: Sparx Enterprise Architect)**

SysML is a generic systems engineering language, standardized, that is inspired by the ideas of UML. While SysML is already less extensive than UML, within the high-tech industry system engineering community it is still perceived as overly generic to support the architectural process. Furthermore, originating from UML it has a very software-discipline-oriented focus.

In comparison to the Design Framework method the SysML language is extensive and thereby rather complex (in large system models the overview could be lost easily), and low in support for the quantitative aspects. Many project examples can be found which opted for a coupling of external tools to SysML tools to compensate for this lack of good quantitative support.

**Computing software (examples: MatLab/SIMULINK, LabView, Excel, ..)**

Analytical and numerical modelling tools are often used to make quantitative relationships explicit, study them, and generate visual representations to communicate the findings. Compared to the Design Framework method they are less suited to create and keep overview of multiple models, and they do not support the development process with storing rationales, arguments, criteria, decisions.

**Configuration management tools (examples: SVN, GIT, IBM RTC, ..)**

Support for change management, versioning, and team work is also provided in detail by configuration management (CM) tools. These tools are used in software communities and very well suited for distributed software development. Although their functionality is partly overlapping the Design Framework method, their approach is too abstract, i.e., there is no intrinsic meaning that relates to architecting, and too software-discipline-oriented.

**PREEvision, an example of a domain specific solution**

PREEvision supports specification of an AUTOSAR-conformant system architecture initiated by the automotive domain. It does this by providing graphic diagrams for the software architecture and Electronic Control Unit (ECU) network as well as numerous engineering features. The system architect specifies types of application software components and compositions in a software architecture library. Graphic diagrams support the definition of software architectures and the connection of ports. Numerous Electrical/Electronical (E/E) engineering features for a convenient working process are available to the developer, such as automatic port connectors or refactorings for the distribution of existing components. The hardware networking architecture can also be modelled graphically. Mapping of software components to hardware components is supported by graphic and table views. Interrelationships between hierarchy levels in the software architecture can be represented by software system diagrams. PREEvision supports various AUTOSAR versions for import and export.

PREEvision is a proven automotive tool with the same goal as the Design Framework method. It has some of the same characteristics as SysML tools (many views grouped by modelling layers, a large language, therefore a steep learning curve), although the coupling to external tools is formally supported to some extent. In comparison to the Design Framework method this tool is too specific for the E/E development to be applied as a generic method for system development.

**Arcadia / Capella, an(other) example of a domain specific solution**

The Capella ecosystem is a field-proven modelling solution offering an environment with a high added-value for engineers working on system, software and hardware architectures. At the centre of this ecosystem is a graphical modelling workbench supporting the Arcadia engineering method. Arcadia mainly focuses on functional analysis, (complex) architecture definition and early validation. It provides a model-based approach for defining, analysing and validating architectures in a proven workflow of exactly 5 successive phases. Each phase has well-defined objectives, from the identification of operational concepts and processes to the design of the organic architecture and the definition of subcontracting items. Arcadia emphasizes multi-viewpoint trade-off analysis.

Arcadia is mainly dedicated on safety, security, productivity and quality properties and relies heavily on ULM and SysML as underlying methods, so has a bias towards software based solutions. Compared to the Design Framework methods it is less generic and too (safety & security) domain oriented to be applied as a generic method for system development. Whereas the method and its supporting ecosystem are currently widely deployed in the Thales Group (original initiator of the Arcadia method), it has not been adopted outside the Thales realm.

### 2.7.3 Value/benefits of component

The Design Framework method is explicitly positioned as a kind of 80% solution: it is low in cost (in terms of tool costs and simplicity of usage) and solves most (but on purpose not all details of the) challenges on the road of system development.

The main unique value propositions for the Design Framework method are:

- Focus on usability: simplicity, ease-of-use, generic expressivity by introducing as few concepts as possible, short learning curve, architectural-centric (create and share views focussing on the interactions and models driven from relevant architectural concerns).
- Extendibility and interoperability: as solution focuses on integrating models from other tools instead of enforcing a propriety prescribed way of modelling.

- Versatility with respect to ways of working: no prescribed process or workflow restricting architectural creativity, adheres to both a model-based and/or experimental approach, not biased for any design discipline, platform, etc.
- Transparency and traceability: provides insight in the design decisions that matter the most, tries to capture the informal side of the architectural process without posing any formal way of working or thinking.

## 2.8 MODEL CREATION METHOD (PC-NL-TNOESI-02)

### 2.8.1 State of the art/practice

Choosing whether models play a sufficient role, and need to be created, is in general a process method that falls under the responsibility of the system architect. It requires the architect to make a deliberate choice based on a trade-off on costs, lead time, possibility to re-use an existing model, possibility to re-use the model in the future, etc. In practise for most architects this is nothing more than applying common sense as can be observed in leading system architecting processes (http://www.gaudisite.nl/). Within mature, typical large-scale product development organization, one can observe it is explicitly formalized and/or enforced in a mostly propriety way of working, or even based upon a well-accepted design philosophy.

### 2.8.2 Component experiences compared with state of the art/practice

The model creation method is merely a formalization of architectural common sense. However in practice one can see that similar methods are anchored into propriety, mostly project based, product development processes of mature industrial settings where it is not uncommon to have architectural milestones built in. In well-spread design philosophies, for example lean six sigma, one can observe multiple well-defined hooks for the decision process to create of models from a specific business need.

### 2.8.3 Value/benefits of component

By formalization of the responsibility of, and rationale behind (when, how), the model creation method will contribute to a more mature (transparent, traceable, consistent) approach for doing product development.

## 2.9 "Y-CHART" (FOR PERFORMANCE ANALYSIS) METHOD (PC-NL-TNOESI-03)

### 2.9.1 State of the art/practice

Within industry the most widespread approach to analyse performance challenges is still based on performing measurements, and making simple quantitative models based on a back-of-envelope or zero-order analysis (typically by a spreadsheet approach using methods as Excel). In more mature settings one can observe worst case performance analysis based on simulations. A good example of the latter is the POOSL based approach (http://poosl.esi.nl/, http://www.es.ele.tue.nl/she/) which has an industrial track record. Unfortunately such simulations do not guarantee that the worst case is truly identified.

In the academic world there is a scale of research conducted on (academic) methods that yield more trustworthy results for certain aspects of understanding performance insights. This includes methods such as model checking, queueing networks, max-plus algebra / real-time calculus, etc. Examples are Uppaal (http://www.uppaal.org/), SDF3 (http://www.es.ele.tue.nl/sdf3/), Modular Performance Analysis (MPA), all Worst Case Execution Time (WCET) techniques, SystemC, etc.

As rule of thumb the industry only cares about getting a large variety of quantitative results of 'reasonable' quality very fast, and the academia use more formal methods because they give high quality results and facilitate unambiguity.

### 2.9.2 Component experiences compared with state of the art/practice

Y-chart modelling, being more a paradigm than a technique, is supported to some extent by different widely available methods and tools: AADL, Daedalus, MPA, Ptolemy II, MARTE. Where other widely available generic methods and tools leave an Y-chart modelling approach open for the user (POOSL, Uppaal, UML, SystemC) some of the most widely available methods and tools hardly provide support to perform a proper Y-chart modelling approach (OPNET, Matlab-Simulink).

Furthermore within the Y-chard method itself multiple design variations can be observed (Lapalme, 2009), each with its own advantages and inconveniences.

The Y-chart method connects to a wide range of other performance analysis methods. Following table gives an overview of the most popular methods and their application positioned against the Y-chart method. Each method has its own advantages and deficiencies.

| Method | Accuracy | Analysis Context | Metric Types | Scalability | Tool(s) |
|---|---|---|---|---|---|
| Discrete Event Simulation | estimation & observation | dynamic transient & steady state | best, average, worst | high | POOSL, AADL, Simulink |
| Closed Formula / First Principle Models | exact | | | low | Excel, Mathematica |
| Queuing Networks | exact & approximation | dynamic steady state | average | medium | OPNET |
| Dataflow Analysis | exact & approximation | static, dynamic transient & | (best), worst, average | medium | SDF3 |

| | | | | | |
|---|---|---|---|---|---|
| | | steady state | | | |
| Interface (Protocol) Verification | exact | | | low | |
| Property Checking | exact | dynamic steady & transient state | | low | |
| Timed Model Checking | exact & approximation | dynamic steady & transient state | | low | Uppaal, CADP, SDF3 |
| Markov Chain Analysis | exact | | average | medium | |
| Bayesian Network Analysis | exact & approximation | | average | medium | |
| Monte Carlo Analysis | estimation | | average | high | |
| Statistical Analysis | estimation & observation | | average | high | Excel |
| Numerical Analysis / Finite Element Methods | exact | | | low | |
| Rate-Monotonic Analysis | | | | medium | |
| Network Calculus / Real-Time Calculus | exact | | best, worst | medium | MPA, SymTA/s |
| Theorem Proving | exact | | | low | |

When comparing in general following aspects of the Y-chart challenges still have not come to a mature level within approaches used in industrial contexts:

- Most approaches do not depict how to extract application and platform models out of legacy systems. In practice the required separation of concerns between variable application and platform aspects is difficult to formalize because the domain specific application side is often vague to not documented.
- Calibration is rarely addressed. Using measurements (for example as depicted for process method PC-NL-TNOESI-04) is most of the time just one of the options. In general other methods such as static code analysis or statistical analysis are not considered.

### 2.9.3 Value/benefits of component

The Y-chart method is a modelling paradigm for performance analysis with variable application and/or platform parts that can be applied in the design analysis phase of a system’s lifecycle. It heavily relies on the separation of the model that represents the application functionality on one side (the ‘what’), and the model that represents the platform resources on the other side (the ‘how’). Next to the application and platform models the mapping, representing the deployment and configuration, including scheduling and arbitration mechanisms, plays a critical role.

The benefit of the method is that variation over different platforms can be independently studied from different mappings as well as from alternative application implementations. It is especially valuable when multiple alternatives, for which the models differ only partly, need to be compared with each other.

**References:**

J. Lapalme, B.D. Theelen, N. Stoimenov, J.P.M. Voeten, L. Thiele, and E. Aboulhamid. "Y-Chart Based System Design: A Discussion on Approaches." Bookchapter in: Nouvelles approches pour la conception d'outils CAO pour le domaine des systems embarqu'es, (pp. 23-56), Montreal, Canada: Universite de Montreal, 2009
ftp://ftp.ics.ele.tue.nl/pub/users/btheelen/Publications/Papers/Lapalme09.pdf

## 2.10 COLLECTING EXECUTION TIMES METHOD (PC-NL-TNOESI-04)

This process method was not introduced due to its scientific relevance but mainly to demonstrate the process components composability and variability. As it has no scientific relevance it is excluded from the state-of-the-art analysis.

## 2.11 KE-CHAIN (PC-NL-KEW-01)

### 2.11.1 State of the art/practice

On the most basic level, a workflow management system is any system that allows its users to setup, execute, monitor, and optimize different workflows. Nowadays, there exist plenty of workflow management systems that serve different purposes, provide various features, and are based on different workflow languages. Two categories of workflow management systems can be distinguished in the engineering field: those that focus on task- or business workflows, and those that focus on simulation- or computational workflows.

Within the former category an additional distinction can be made between Business Process Management (BPM) suites and PDM/PLM platforms. Here, PDM stands for Product Data Management and PLM for Product Lifecycle Management. An important difference between BPM- and PDM/PLM-based systems is that the former focus mainly on the workflow of a process and view the data structure of the underlying process as a derivative, whereas the latter systems are more data-centric. The differences between these two types of systems will be explored in more detail in the following sections.

Whereas BPM- and PDM/PLM software systems focus mainly on the level of business processes, simulation- or computational workflow management systems focus on a more technical level. A prime example of such management systems are Process Integration and Design Optimization (PIDO) platforms. Such platforms are mainly used to automate engineering workflows that use different software tools, such as FEM software packages and CFD packages. The added value from using these platforms comes from the fact that they both allow the different engineering tools to communicate without intervention from the engineer, and let the user automate the workflow.

Next figure gives an overview of the different workflow management systems that were discussed in this section, their (desired) respective features, and their relation.

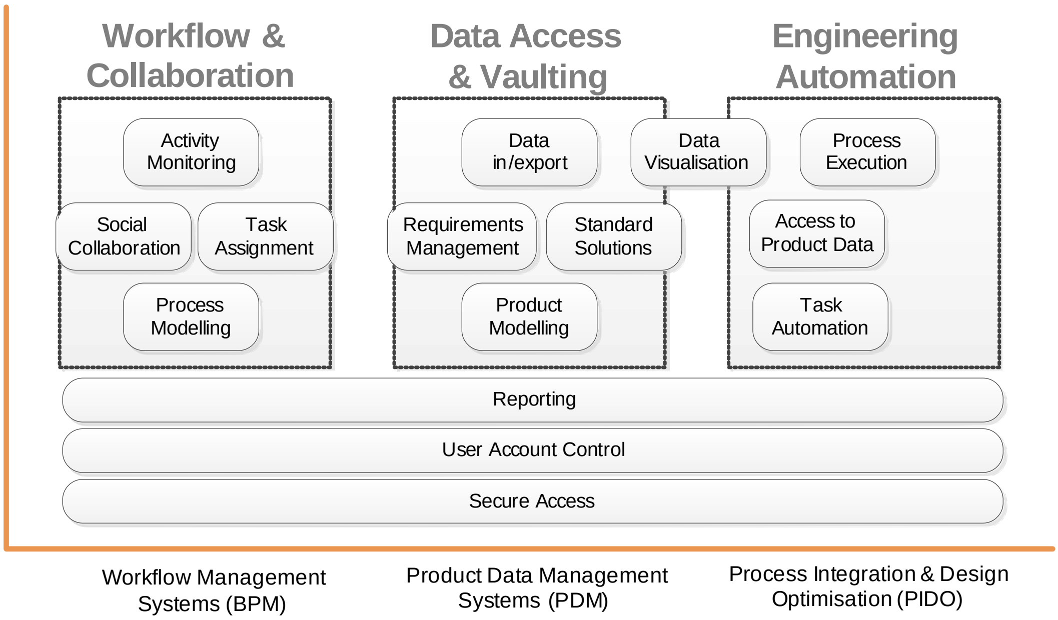


Figure: Overview of the different workflow management systems presented in this section.

**BPM systems**

The discipline Business Process Management (BPM) takes care of all tasks needed to define, monitor, communicate, implement, and optimise business process workflows. The main business objectives that BPM aims to achieve are typically

- To drive back-office efficiency
- To automate business processes
- To increase worker efficiency

On a high level, the typical functionalities offered by a BPM suite can be divided in different categories: process design, analysis, execution, monitoring, collaboration, and optimization. These points will be addressed individually below.

- **Process design** regards the identification of 'as-is' processes as well as the design of 'to-be' processes. Good design reduces the number of problems over the lifetime of the process. Whether or not existing processes are considered, the aim of this step is to ensure that a correct and efficient theoretical design is prepared. The proposed improvement could be in human-to-human, human-to-system, and system-to-system workflows, and might target regulatory, market, or competitive challenges faced by the businesses. The existing process and the design of new process for various application will have to synchronise as such will not affect the business in major outage. The business as usual is the standard to be attained when design of process for multiple systems is considered
- The **Analysis** functionality allows the users of the BPM system to perform analyses on their processes. An example could be an analysis of the effect of a change in certain properties of the raw materials used in a production process on the final product, or an analysis of the effects of a fictional delay in a certain part of the workflow.
- The **Execution** phase. In their early days, BPM systems mainly focussed on the automation of repeatable, standardized (document-driven) business processes. Recently, however, efforts have been made to extend the use of BPM systems to workflows that inherently require human intervention in parts. An example of the use of BPM systems in such combined workflows would be the automated assignment of human-oriented tasks to the best-suited engineer, based on availability, knowledge, and skills.
- **Monitoring** concerns the (real-time, offline, or other) tracking of the different processes, tracking of tickets, tasks, deliverables, and progress. Since monitoring allows the business to remain up-to-date with respect to its current activities, and allows the business to collect information that can be used to improve its future processes, monitoring is a crucial task within any BPM suite.
- **Communication**. As engineering challenges become more multidisciplinary and complex, and as concurrent engineering becomes ever more important, so does communication between different stakeholders in the engineering workflow become more important. Typical ways of communicating information within a BPM suite include linking, messaging (either in-app or via email), and commenting.
- In **Optimization**, information from the Analysis and Monitoring tools is used to optimize processes.

Besides these specific functionalities, a typical BPM system will contain the following, fundamental offerings

- **Integration**. BPM systems might vary between human-centric and integration-centric, where the latter puts more emphasize on the integration of applications and consolidation of existing IT tooling. Regardless of the emphasis, all systems offer a certain degree of integration, be it on data level, application level, or higher up on a business- or cross-organizational level. The more advanced BPM systems offer cloud-based platforms that ultimately support process execution across the supply chain.

- **Permission system and authentication**. Every suite has a permission system allowing user account control, includes techniques such as user authentication and single sign-on. The differentiators for the intelligent systems are role-based and customized views, tasks, and configuration of processes

Some examples of common BPM suites include ARIS enterprise, Appian, Bizago Go, and Pegasystems.

Although current BPM suites come with a plethora of tools, they offer neither support for measuring the critical (engineering) performance indicators in product development processes, nor integration of simulation and calculation activities or a suitable library to manage the mathematical and scientific engineering rules. While BPM systems are used to streamline document based processes on a case-by-case basis, they do not provide support for the dynamic nature of engineering, and fail to take aspects like feedback loops induced by changes and errors, repetitive tasks, bulk data, authorization steps for the release data and data adapters to enable integration with third party systems into account.

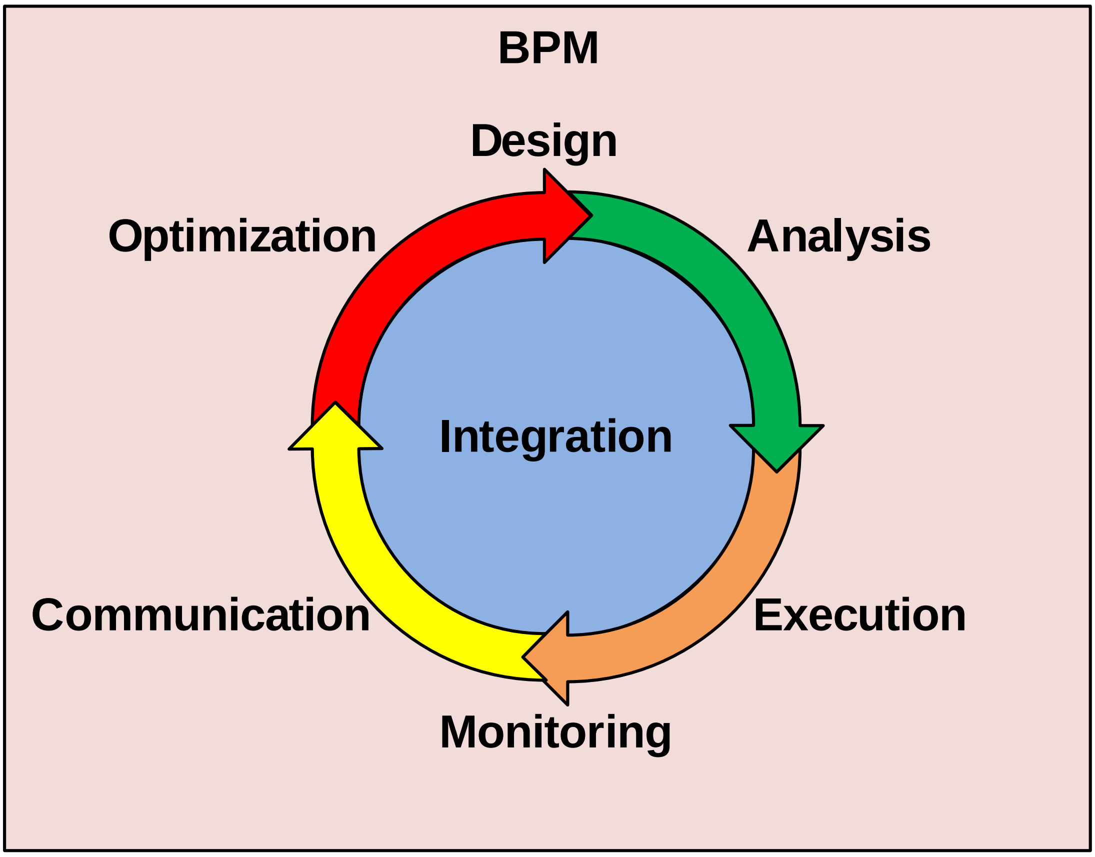


Figure. Overview of features and phases in typical BPM suites

**PDM/PLM**

Product Lifecycle Management (PLM) concerns the central management of all data that plays a role in the entire lifecycle of a product, as well as the technology that is required to create, access, and edit this information and knowledge. Besides containing the core technical data related to a product, a PLM may also contain 'secondary' information, such as supplier alternatives, customer feedback, pricing information, and any other information acquired or needed during the product's lifecycle. Furthermore, this lifecycle ranges from the initial conceiving of the product idea, up to the recycling or disposal of the product.

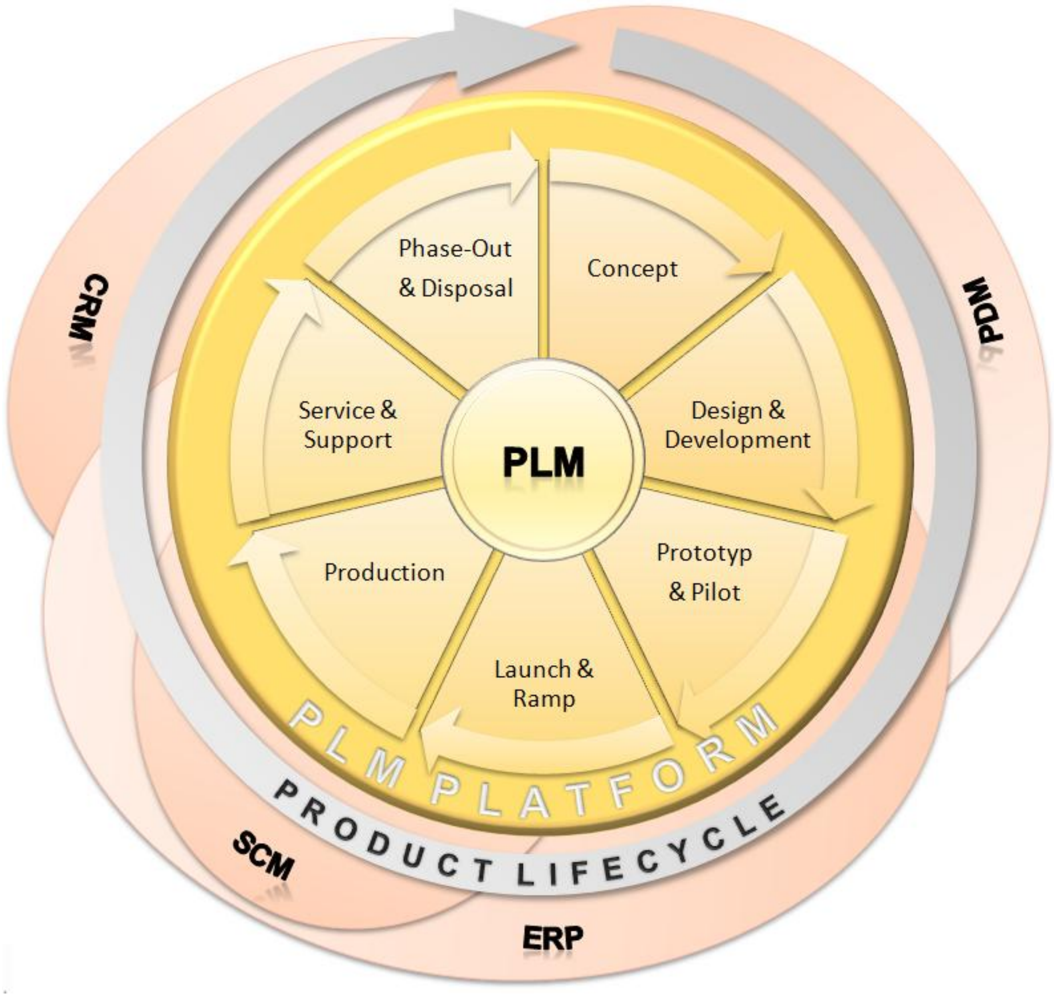


Figure. Overview of features and phases in typical PLM platforms.[1]

[1] Source: https://www.imi.kit.edu/english/209_368.php (author: Professor Dr. Dr.-Ing. Jivka Ovtcharova)

PLM is currently the most comprehensive family of IT applications to support the complete life cycle of the engineering aspects of product development. PLM features an integration of different applications, like Product Data Management (PDM) and visualization tools for Computer Aided Design and Engineering applications (CAD, CAe) files. PDM applications capture product development information, mainly catering to the data defined in Digital Mock-Ups (DMU). For authoring information, many different IT systems are used, of which CAD is the most prominent one. As a consequence, PDM applications focus on CAD-centric product information, and feature limited support for other information types. In that sense, they can be viewed as a subset of PLM. However, PDM applications can also be used as a stand-alone solution if the required information is limited to information only directly pertaining to the CAD representation of the product.

Due to the broad scope of PLM software, the functionality of different packages varies. However, typical recurring functionalities in PLM software include:[2]

- Manage design- and process documents
- Construct and control bill of materials
- Provide an electronic file repository
- Include built-in and custom part and documents metadata (‘attributes’)
- Identify materials content for environmental compliance
- Permit item-focused task assignments
- Enable workflow and process management for approving changes
- Control multi-user secured access, including ‘electronic signature’
- Export data for downstream ERP systems

Because PLM software is required to integrate various different software packages, and because the information stored needs to be consistently accessible for (at least) the entire lifecycle of the product, PLM software typically uses standard data formats to store information (such as the STEP format, based on the ISO 10303-21 norm). Examples of PLM software packages include Dassault systemes Enovia, Siemens teamcenter, PTC Windchill and Autodesk 360.

On the downside, many PLM applications are proprietary and closed systems. They have proven to be inflexible in the interoperability with alternative third party applications and feature limited integration support for open interface standards. They also offer limited support for Knowledge Management, disabling effective reuse of knowledge in internal processes, workflows and applications via Knowledge Based Engineering (KBE) techniques.

### PIDO

PIDO (Process Integration and Design Optimization) suites are used to combine different software tools in a workflow in order to find some design optimum of a product. The user of the PIDO suite typically gives a collection of parameters that can be varied as input (together with the underlying models), and the generated output is (some subset of) the design space of the product. PIDO is often a single task in a larger business workflow.

[2] Source: http://www.product-lifecycle-management.info/what-is-plm/plm-function.html

In a given (multidisciplinary) engineering problem, process integration is required because many different software packages are used in determining the optimal solution (for instance CAD software, FEM analysis, Labview, numerical routines). By default, these packages cannot exchange information without human intervention. Therefore, an additional communication layer is required in order to automate the simulation- and optimisation workflow. This layer is supplied by the PIDO software.

Standard components that can be included in workflows in PIDO suites are input-/output files (typically human-readable ascii files, Excel files, etc.), and components that run external scripts or software packages. Standard connectors that can be included in these workflows include for-loops, switches, and if-statements. This allows the user to create flexible and iterative workflows, including feedback loops.

Some analyses that can be performed in PIDO packages include the evaluation of the workflow for a single value of the input parameters, evaluation for a collection of input parameters, design-of-experiments analysis, optimizations, and reliability analysis. Examples of commercially available PIDO software include ModelCenter (Phoenix Integration Inc.), Optimus (Noesis solutions), ModeFRONTIER (Esteco) and pSeven (DATADVANCE, Inc.)

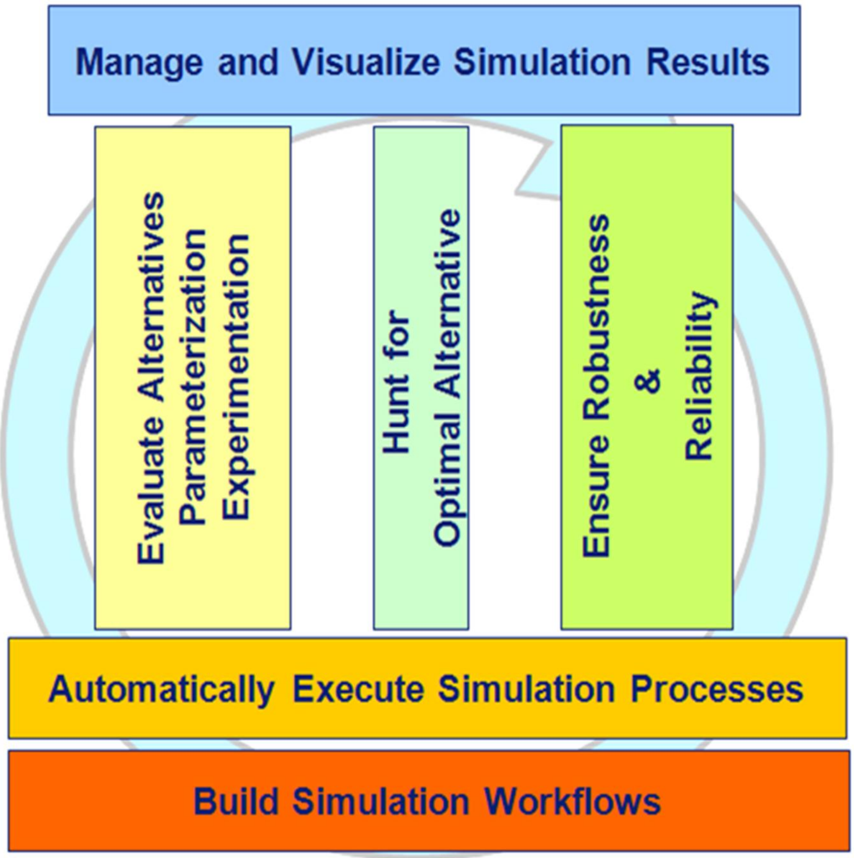


Figure. Overview of features and phases in typical PIDO platforms[3]

[3] Source: http://www.sigmetrix.com/products/optimus-process-integration-design-optimization-software/

One of the disadvantages of present-day simulation workflow packages is that they lack explicit context on their position in the business workflow and the influence of their results of the process. Systems that support hybrid workflows consisting of both human and simulation activities as characteristic for engineering are non-existing.

### 2.11.2 Component experiences compared with state of the art/practice

In the previous sections it has become clear that each of the discussed state-of-the-art workflow management systems has certain drawbacks. BPM suites do not offer an integrated solution that allows the engineer to work on the level of technical processes and tasks, nor do they allow for the monitoring of critical technical performance indicators or do they provide support for the specific, dynamic nature of engineering. PDM/PLM systems are often proprietary, closed systems that are inflexible in the use with third-party software. Furthermore, they offer little support for knowledge management. PIDO suites, finally, work only on the level of simulation- and optimization workflows. As a consequence, they lack the connection to other parts of the business workflow.

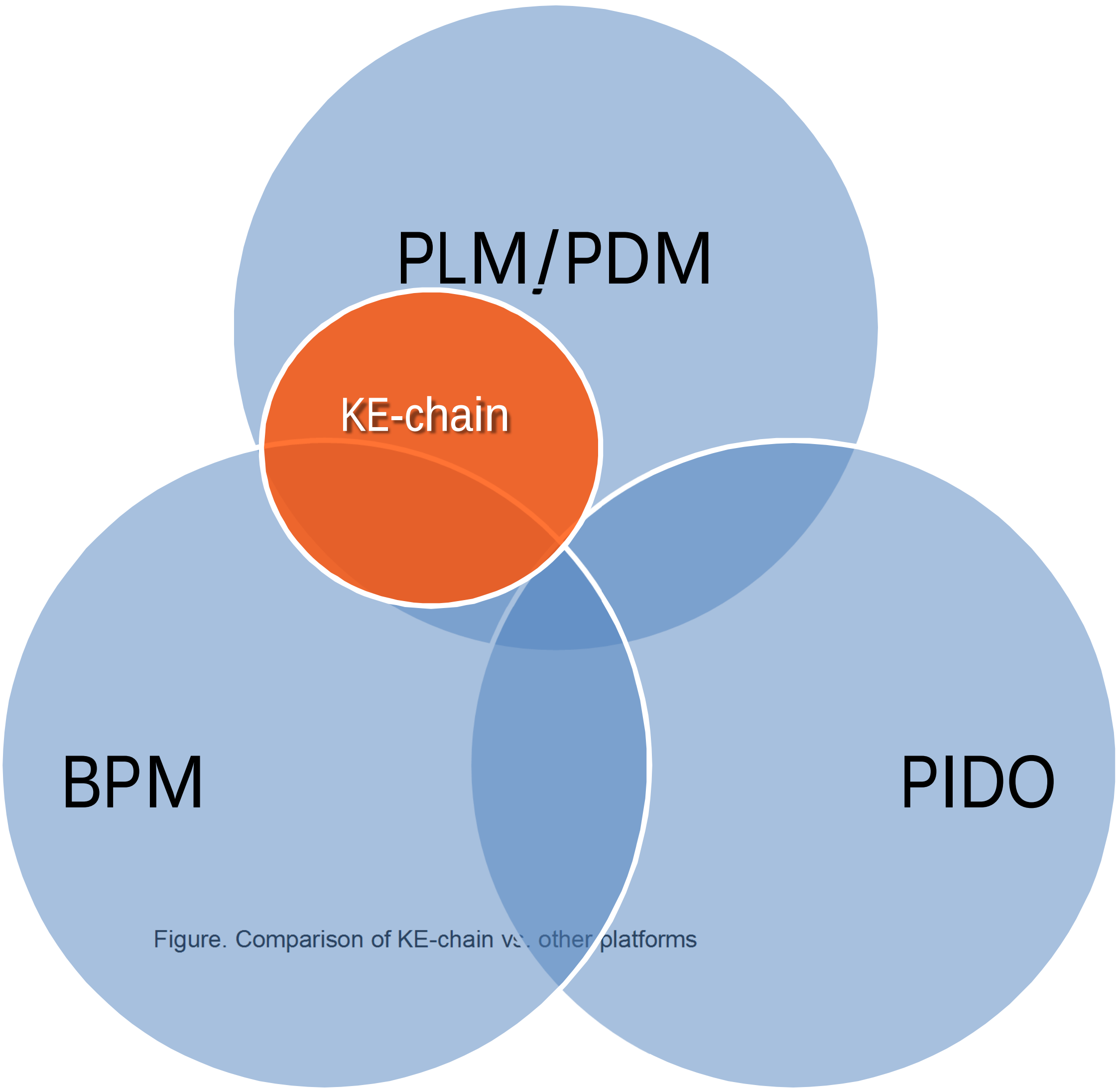


Figure. Comparison of KE-chain vs. other platforms

To address the issues described above, the introduction of a web-based Engineering Workflow Management system that combines the major aspects of the three aforementioned systems is proposed. In short, the system will

- Be able to describe lean & common engineering processes. The generated workflows should include fully prepared tasks, address required input and output information (both on process and product level) for these tasks and its relation to other aspects of the business workflow, allow the integration with specified third-party tools and methods, and have the ability to do task assignment.
- Increase engineering process control by coupling requirements to process- and product information and incorporating task change and progress management. This requires the definition of an underlying product model and the connection between this model and the workflow.
- Allow a business to reuse information through a knowledge library for processes and product parts and materials in order to decrease the setup and execution time of projects
- Be a single point of entry in a multi-disciplinary, multi-site and multi-organisational environment

To facilitate a lean and controlled engineering process, the web-based Engineering Workflow Management system KE-chain has been expanded (and will continue to be expanded) with (i) graphical product and process modellers (WP1), (ii) an engineering templates library and (iii) a requirements traceability module (WP2).

### 2.11.3 Value/benefits of component

The process components in the area of workflow management should lead to more control, less rework, better resource use, high use of standardization and knowledge reuse. This should lead to reduced time-to-market and cost saving. Other consequences should be an improved quality due to improved communication & information integration, more (integrated) use of engineering automation, standardization and incorporation of lessons learned.

## 2.12 SOFTWARE PERFORMANCE ANALYSIS (PC-NL-VF-01)

### 2.12.1 State of the art/practice

The functionality of industrial systems is increasingly determined by embedded software. The quality of such embedded software has therefor become crucial for a correct and safe behaviour of industrial systems. In many cases, such correctness and safety aspects can be human life threatening, e.g. in case of robots, airplanes, cars, etc. As already mentioned above in section 2.6 ”refactoring and traceability”, software code tends to have a long lifetime, becomes adapted and upgraded for renewed application, where the updates and modifications typically causing a degradation of its quality.
In practice, software programmers have very little knowledge about the execution performance of their code, basically due to the complexity of modern computer architectures and the complexity of the code compilation process. Furthermore, if code gets moved for (re-) application onto new target machine architecture, original assumptions on performance or even basic functional correctness might become invalid and should be re-assessed.

New (software) functional safety standards such as ISO 26262 are considered state-of-the-art for the development of high-integrity software. This standard requires the use of a software development workflow that relies on thorough validation and verification of the created software. Software validation tools are an indispensable part of such a workflow.

Unfortunately, most software today is still developed without such validation, leading to user experiences with unreliable and unsatisfactory systems, and/or to huge costs for system manufacturers for call-backs of in-the-field products.

### 2.12.2 Component experiences compared with state of the art/practice

For safety-critical ISO 26262 directed software development processes, most software validation tools (workflow components) aim for –merely- functional correctness. However, todays growing software complexity and increased data processing requirements more and more hits performance limitations. Therefore, software validation should cover performance validation as essential property next to functional correctness. The integration of performance and functional validation in a software development process is still immature. The PROMES software performance analysis component aims to research and evaluate such integration in a software development process. This would allow capture both performance and functional correctness issues in a well-structured environment.

### 2.12.3 Value/benefits of component

Tests of applying this validation method are being done both with PROMES project partners as well as with a few selected external parties. Initial feedback is that the developed validation tools are still hard to apply in a standard development process, but do provide essential feedback on performance and functionality issues that could not be obtained by any other method.

## 2.13 CREATE PROCESS RENEWAL (PC-FIN-NOKIA-01)

### 2.13.1 State of the art/practice

Product development of complex systems differs from currently used static process oriented work. Workers in complex systems development also invent new ways-of-working to deal with not earlier defined challenges. It is difficult to specify a stable set of tasks or procedures for dynamic or unanticipated situations.

To overcome this challenge [Naikar 2005] proposes that a feasible level of abstraction of task is used to bring in information that explains why and what for the task results are done and used. There is a target to increase the understanding of the worker instead of showing the context dependent practice (this may work with simple systems and mechanical work.)

Nokia process component of work management system focuses on decision and decision-oriented process modelling with the purpose to make the decision visible. It then explains what information is needed to make that decision. On the other hand decision points or milestones are regarded as basic building block of processes and processes themselves compose networks. Downstream decisions are divided into decision criteria, which are documented in such a detail that the worker in principle can perform their tasks. Tasks are seen as information production activities to satisfy the decision making needed of the corresponding decision-makers.

In Nokia component we define the model of process networks, processes, milestones and decision criteria extended with process communication and signalling structures as well as roles. Based on the model in Nokia environment we can generate an application, where the process networks and their components can be instantiated as used process descriptions. Further on all the data included in to the structures defined in the criteria can also be instantiated as project work products.

### 2.13.2 Component experiences compared with state of the art/practice

The model of the process network structure is modelled and a DSL (Domain Specific Language) model of that domain is created. Next the DSL model is populated with Nokia processes and the user interfaces are created for the defined roles. The approach follows closely the decision oriented process approach. The application for process developers is also created. All the work products are modelled and analyst and designer user interfaces are created. The environment allows analysts and designers to fill in their analysis and design results into the repository.

There are no big differences between the SoTA approach and the usable environment. The environment also supports on-line multiuser work in geographically distributed user base.

### 2.13.3 Value/benefits of component

The component will be piloted in fourth quarter of 2015 for the first time, so we can only give expectations here. The main benefit will be the self-reflectiveness of the environment meaning that when a change is made it is propagated all over the system immediately. In practice this means that when a process developer (decision criteria owner) changes the description of the criterion it is propagated to the end user immediately and is effective when a new work is started. Earlier started tasks used the version they initialised when the task was created. The benefit is that there is no delay caused by the bureaucracy of development activities and thus the process is “more” up-to-date.

Another clear benefit is the way the environment combines the work situation and the process description. Both are all the time visible to the worker as well as the task status and maturity. In practice this means that the worker can see which data is produced for which criterion and why.

We do not list all the benefits we expect but one more is worthwhile mentioning. The repository provides change based version history, which means that everything – everything – is recorded into the database. This allows us to analyse and compare histories of how the results are produced. At the moment we do not have these analysis functions, but obviously Big Data techniques will be used.

**References:**

[Naikar 2005] Naikar, N., Hopcroft, R., Moylan, A.: Work Domain Analysis: Theoretical Concepts and Methodology. DSTO-TR-1665 System Sciences Laboratory: Edinburgh, Australia (2005).

## 2.14 EARLY VALIDATION (PC-FIN-NOKIA-02)

### 2.14.1 State of the art/practice

Early validation has been proven to provide substantial cost savings for large scale complex systems [Vasconcelos et al 2004]. For many years, the main methods for early validation have been the usage of formal methods [Bäumer et al 2006] or prototypes [Palladio 2014]. Applying formal methods usually consists in describing the system under validation in a specific formal language, and then utilising some model checker or theorem prover to prove that the system respects some desired properties. These approaches vary widely, either in the supported formalism e.g. ranging from logical declarative formalisms to complex behavioural models [Visser et al 2002] , or in the degree of industrial applicability e.g. approaches range from models generated directly from the code [Jensen et al 2007] to abstract graph- based structures completely disconnected from the actual code [Bomsdorf et al 1996] . Nowadays, formal methods are used in very specific cases, such as the application related to human safety, where costs of a failure are above any estimation.

Prototyping techniques usually involve either building an actual simpler version of the target system to validate or building a model of the system to validate its usability and functional requirements. Fast prototyping methods with model-driven development (e.g. Web Ratio [Web Ratio 2014], Mendix [Mendix 2014] can be considered as good examples of lightweight methods for early validation. Additionally, various researches and standards have used MDD for early validation. For example, the MADES project enables early validation of timing and clock constraints by making use of the UML MARTE standard [Mades 2014] , [OMG 2011] , while in [AUTOSAR], an AUTOSAR based approach for early and iterative validation of timing behaviour of automotive systems was carried out. Most recently, the agile movement promoted the test-driven development paradigm, where from the early stages of the software project, the test become a functional specification and a means of validation [Fraunhofer 2014]. The tests are developed iteratively and follow the evolution of the software.

The limitations in scalability of these methods and the effort required to formally define specifications hinder their applicability in the industry. Furthermore, the languages underlying formal methods are far from being generally understood by regular software engineers. Producing these models also has a cost, such as in describing external libraries, frameworks, systems that interact with the system under study or in gathering metric information for analysis purposes. Concerning prototypes, the main challenge lies in the fact that they usually only focus on a specific part of the system, as compared to the system in whole. Thus, conclusions taken from a prototype may not apply to the final system.

### 2.14.2 Component experiences compared with state of the art/practice

Nokia approach differs with conventional early validation systems in the sense that we use base lining techniques to define and compose the work context. On top of the base line new features are specified in a way that resembles transparent layer on top of the base line. All the base line structure and objects are available, changeable and even they can be deleted. The specification work thus modifies and extends the earlier baseline. On the other hand several features can be made visible on top of the base line and their dependences and conflict are then visible (they can also be alarmed because the tool environment can use rules to detect conflicts). Finally when all the feature for the next release are specified and all conflicts resolved the new base line can merged. And the cycle starts again.

### 2.14.3 Value/benefits of component

The main expected benefit is the early validation benefit, which is aimed to find faults as early as possible and thus reduce conflicts of faulty specification, fault finding and correction work in short. The of fault increases more the later it is found.

**References:**

[Vasconcelos et al 2004] Vasconcelos, W. et al. Rapid Prototyping of Large Multi-Agent Systems Through Logic Programming. Annals of Mathematics and Artificial Intelligence 41, 2-4, 135-169. 2004

[Bäumer et al 2006] Bäumer, D. et al. User interface prototyping—concepts, tools, and experience. ICSE '96. 532-541. 1996

[Palladio 2014] Palladio Simulator. 2014. Available at: http://www.palladio-simulator.com

[Visser et al 2002] Visser, W. et al. Model Checking Programs. Automated Software Engineering, 203-232. 2002

[Jensen et al 2007] Jensen, K. et al. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer, 213-254. 2007

[Bomsdorf et al 1996] Bomsdorf, B. and Szwillus, G. Early Prototyping Based on Executable Task Models. CHI '96, 254-255. 1996

[Web Ratio 2014] Web Ratio: High productivity Web and Mobile App platform. 2014. Available at: http://www.webratio.com

[Mendix 2014] Mendix Platform for Enterprise Apps. 2014. Available at: http://www.mendix.com

[Mades 2014] MADES FP7 Project. 2014. Available at: http://www.mades-project.org/

[OMG 2011] Object Management Group (OMG). Modeling and Analysis of Real-Time and Embedded Systems (MARTE) profile V1.1. 2011. Available at: http://www.omg.org/spec/MARTE/1.1/PDF .

[AUTOSAR] Automotive Open Sytem Architecture (AUTOSAR): http://www.autosar.org

[Fraunhofer 2014] Fraunhofer ESK. Validation of timing behavior in automotive systems. 2014. Available at: http://www.esk.fraunhofer.de/en/projects/MoCi_Timing_Analysis.html

## 2.15 DSR: DOMAIN SPECIFIC DATA IN A REPOSITORY (PC-FIN-NOKIA-03)

### 2.15.1 State of the art/practice

Nokia collects information and experiences on applying mega-modeling principles [Vignaga 2013] , [ECLIPSE], in product line R&D work. The case study includes many DSL models [Tolvanen 2012], of product at logical level, SW architecture, SW design and testing. Other modeled domains are source information for e.g. feature analysis like telecommunication standards, company strategy, customer information etc. Their import definitions are DSL models. Other DSL models cover process artifacts, their export definitions are models, including connection to other possible tool environments. One more DSL domain is actual work, its history and related actors.

The final system is composed of DSL models created based on R&D needs and domains are relatively independent from each other. However, hyperlinks can be used to link over the DSL boarders. The way-of-working is shown by views, which show often combined parts from several domains. The views compose actually the application level for the users.

The focus of the use case is in testing and testing related data modeling and its utilization in the process. Nokia is creating a domain specific model of testing. The model will contain product data, requirements data, data about the SUT and additional data needed for testing such as special values or status if formation for preparing the SUT for specific test cases.

### 2.15.2 Component experiences compared with state of the art/practice

To enable a self-reflective system the SoTA requirements must be followed very carefully. This has been done while implementing the repository application by the vendor. The tool include stack of meta layers and domain layers in purpose to be able to product all kind of model and corresponding languages. The tool environment is following the SoTA principles highly accurately.

### 2.15.3 Value/benefits of component

The benefits of the repository are low cost creation and use of ICT applications, proper information management and distribution functionality and unique access to all data. These properties actually generate more benefits when people start to used and collect data.

**References:**

[Vignaga 2013] Vignaga, A. et al. Typing artifacts in megamodeling. Software and System Modeling. 12(1): 105-119. 2013

[ECLIPSE] Eclipsepedia. Teneo - Model-Relational mapping for EMF. Available at: http://wiki.eclipse.org/Teneo

[Tolvanen 2012] Tolvanen JP.: Domain-Specific Modeling Languages for Embedded System Development, ESWEEK workshop on metamodeling and code generation (2012)

# 3. PROMES PROCESS FRAMEWORK STATE OF THE ART

To the best of our knowledge, in the literature there are no process frameworks in the domain of embedded systems. However, as related work we have considered the IEEE International Standard [2] that provides guidelines on defining system level lifecycle processes, and a specialization of the Rational Unified Process (RUP) [3] for Systems Engineering (RUP-SE)[4].

First, the 15288:2008 IEEE Standard [2] defines a set of default process element instances (actual descriptions of each instance) that can be used in the engineering of hardware / software systems. More specifically, the standard defines four process groups, i.e. Agreement Processes, Organizational Project-Enabling Processes, Project Processes and Technical Processes, in which specific processes are classified. Then, for each process the Standard defines the purpose, the outcomes, and the activities and tasks. In addition to that, the Standard states that new processes can be added to the process groups in order to extend the proposed system development lifecycle.

However, the proposed default activities are quite generic and do not provide guidelines in terms of how they can be applied in specific process instantiation. In addition to that, the Standard does not explicitly provide a configuration mechanism for removing or adding an activity from the predefined set of processes. Furthermore, the sequence of the performed activities, i.e. iterative or waterfall, is not described. Thus, the Standard only provides the third part of process frameworks definition [1], i.e. a repository of process element instances and omits the first two parts (meta-model and usage guidelines), since they are considered out of its scope and intend.

Second, the RUP-SE, is a derivative of the Rational Unified Process, which makes use of the RUP Process Framework. RUP-SE instantiates new artifacts and applies modifications of default RUP disciplines and roles, so as to support the creation of such artifacts in system level [4]. The process is organized in four phases (inception, elaboration, construction and transition), which are split in multiple iterations. In each phase, multiple activities are performed. The activities are organized in workflows, and each workflow is assigned to a different effort during each phase, represented in the form of a hump. Both RUP and RUP-SE are controlled iterative processes that reflect all the benefits of iterative processes, like the possibility to achieve a robust architecture by applying iterative corrections, refining the engineering process and mitigating early any potential risks [3, 4]. The main point of advancement of the PROMES Process Framework, with respect to RUP-SE and 15288:2008 IEEE Standard, is its explicit support for addressing the aforementioned challenges of embedded systems, namely multi-disciplined and multi-lifecycle engineering in a multi-organizational and multi-site context.

**References**:

[1] D. G. Firesmith and B. Henderson-Sellers, “The OPEN process framework: An introduction”, Addison-Wesley Pearson Education, London, UK, 2002

[2] ISO/IEC/IEEE 15288:2008 Standard for systems and software engineering - system life cycle processes, IEEE Computer Society, 31 January 2008

[3] P. Kruchten, “The Rational Unified Process: An introduction”, Addison-Wesley Longman Publishing, 2nd Edition, Boston, MA, USA. 2000.

[4] Rational Software, “Rational unified process for systems engineering – RUP SE1.1”, A Rational Software White Paper, TP 165A, 5/02, 2001.

# 4. EVALUATION OF ITEA PROJECTS RELATED TO THE PROMES PROCESS FRAMEWORK

The following sub-sections introduce PROMES framework related projects. Each project is shortly introduced and the relation to PROMES is discussed.

## 4.1 IDSEALISM

Full Title: Integrated & Distributed Engineering Services framework for MDO
Status/Call: In Progress. ITEA 2 Call 8
Website: not yet available

*Description:*
IDEaliSM aims to change the product development process by enabling the continuous integration of distributed and highly specialized development teams. The project will deliver a new distributed, flexible and service-oriented development framework for multidisciplinary design and optimization that is capable of integrating people, processes and technology.

*Relation to PROMES:*
The project explicitly addresses multi-site and multi-disciplinary characteristics. However, the project has been recently started. Hence, its results cannot be reused yet.

## 4.2 FLEXI

Full Title: Flexible Global Product Development and Integration
Status/Call: Completed ITEA 2 Call 1
Website: http://www.flexi-itea2.org/index.php

*Description:*
The FLEXI project provides a flexible, rapid and agile approach for software development. The goal of the project is to achieve moving "from idea to product in 6 months", when applied on a company level.

*Relation to PROMES:*
- Look into multi-site and multi-organization characteristics
- Focus only on one discipline (software)
- Possibility of reusing results for PROMES on process component level

## 4.3 TWINS

Full Title: Optimizing hw-sw Co-design flow for software intensive systems
Status/Call: Completed ITEA Call 8
Website: https://web.archive.org/web/20111010092931/http://www.twins-itea.org/

*Description:*

The TWINS project aims at improving the co-design of products in which hardware and software are tightly integrated. As outcome of this project BoKE was created, i.e. a Body of Knowledge for HW/SW co design. BoKE is a public searchable repository (website) of solutions and experiences that can assist companies to find the appropriate solution for their hardware/software co-design problem. BoKE contains industrial verified software/hardware co-design solutions that can be tailored to specific domains and situations.

*Relation to PROMES:*
- Focus on multi-disciplinary characteristics
- BoKE does not instantiate components of a process framework. Therefore it is more difficult to identify components and how those components interact.
- TWINS is unavailable. The BoKE website has not been maintained.
- BoKE solutions can possibly be reused on component level.

## 4.4 MARTES

Full Title: Model driven approach to Real-Time Embedded System Development
Status/Call: Completed ITEA Call 7
Website: www.martes-itea.org

*Description:*
The main focus of MARTES is to introduce how to use UML and SystemC in combination for systematic model-based development of real-time embedded systems. As outcome of the project the MARTES UML Profile has been developed (http://www.omgmarte.org), while also the interoperability of several software engineering tools has been improved.

*Relation to PROMES:*
- Provides a common way of modeling both hardware and software aspects of RTES (to some extent related to multi-disciplinarity).
- Possibility of reusing components that address multi-disciplinary development (e.g. as artifact or tool)

## 4.5 MERLIN

Full Title: Embedded Systems Engineering in Collaboration
Status/Call: Completed ITEA Call 6
Website: http://www.merlinproject.org
https://web.archive.org/web/20140824171706/http://www.merlinproject.org/

*Description:*
The project focuses on the enhancement of collaboration in embedded systems engineering and software engineering. During this project a repository with recipes for multi-site development was developed. The repository is available at http://www.sameroomspirit.org/index.php/Main_Page

*Relation to PROMES:*
- The wiki approach could be adapted for the PROMES needs, based on the insights obtained from VTT

- The collaboration items and solutions are less formally defined compared to the PROMES process components (only the collaboration item contains a *role* attribute).
- The solutions in the wiki can be reused on a component level.

## 4.6 OPEES

Full Title: Open Platform for the Engineering of Embedded Systems
Status/Call: Completed. ITEA 2 Call 3
Website: www.opees.org/

*Description:*
Open engineering platform aims at speeding up adoption of innovative technologies for dependable/critical software-intensive embedded systems.The basic goals of the project are: a) to develop open source tools for embedded systems, and b) to settle a community and build the necessary means and enablers to ensure long-term availability of innovative engineering technologies in the domain of dependable / critical software-intensive embedded systems.

*Relation to PROMES:*
Among other goals, the project aims at providing

- an open technical repository.
- a set of processes and guidelines for tools/components maturation, verification & qualification

However, comparing these goals to the PROMES project, the OPEES repository only concerns tools instead of process components in general, so the PROMES repository will include a bigger variety of reusable components. Similarly, in OPEES, the terms processes and guidelines refer only to tooling, while in PROMES we use the term process to refer to any process related to embedded systems development, and we provide guidelines of how to design a process using a set of elements proposed by the PROMES process framework.

## 4.7 PRISMA

Full Title: Productivity in Collaborative Systems Development
Status/Call: Completed. ITEA 2 Call 2.
Website: www.prisma-itea.org/
https://web.archive.org/web/20111024235007/http://www.prisma-itea.org/

*Description:*
The goal of the PRISMA project is to improve productivity of collaborative systems development through improved development technologies.

*Relation to PROMES:*

- Focus on globally distributed (multi-site) development of software systems
- Use of the SameRoomSpirit Wiki: addition of industrial experiences and best practices in the form of reported processes, methods and practices aiming to meet the globally distributed system development challenges (multi-site challenge)
- Possible reuse of the wiki approach, due to the insights obtained from VTT
- The collaboration items and solutions are less formally defined compared to our process component (only the collaboration item contains a *role* attribute).

- The solutions in the wiki can be reused on a component level.

## 4.8 EUROSYSLIB

Full Title: European Leadership in System Modeling and Simulation through advanced Modelica Libraries
Status/Call: Completed. ITEA 2 Call 1
Website: www.eurosyslib.com

*Description:*
The ultimate objective of EUROSYSLIB is to make Modelica the de-facto standard language for embedded system modelling and simulation. In order to support this major product lifecycle management effort, the EUROSYSLIB consortium, composed of 18 European partners, is committed to delivering a large set of high-value, innovative modelling and simulation libraries based on the freely available Modelica object-oriented modelling language.

*Relation to PROMES:*
- Focus on multi-disciplinary engineering
- Aims at knowledge capitalization and re- use
- The library components provided could be reused in the PROMES repository.