Product-Focused Engineering
Process Analysis and Improvement

Stefan Biffl
Christian Doppler Laboratory CDL-Flex
Institute of Software Technology and Interactive Systems (ISIS)
Vienna University of Technology
http://cdl.ifs.tuwien.ac.at
CDL-Flex Research Background and Agenda

**TU Wien CDL-Flex**: application-oriented basic research with industry partners in line with industry trends Enterprise 2.0, the Industry 4.0 initiative in Germany, and the European Union “Horizon 2020” program.

**Context**
- Engineering organizations, business information systems
- Product development, often systems of systems
- Similar products with variations (towards product lines)
- Industry partners usually work on CMMI levels 2 to 3.

**Challenges** regarding product and process improvement

**Case Studies** and **Lessons Learned**

- Hydroelectric power plant in Foz do Iguaçu, Brazil
- Steel mill
Product-Focused Engineering Aspects

- **Typical Product** examples
  - Industrial production system automation
  - Engineering tools and information systems
  - Business application, services
- **Links between product quality** and **development processes**
- **Stakeholders** around a product development environment

---

Stakeholder Value in an Ecosystem Context

- **Repeatable product/service development** and delivery in core business areas.
- **Better cost, quality, or schedule** of software and systems engineering processes.
  - Effort and risk of development and Quality Assurance processes.
  - Improvements in sub-processes and process steps: quality, effort, duration, risk.
- Identifying and securing **mission-critical engineering know-how**.
  - Elicit the most relevant candidates for elicitation and sharing engineering know-how on best practices in the context of an organization.

VDE 3695 part 2 „Engineering of industrial plants – evaluation and optimization; subject processes“, Association of German Engineers, November 2010.
Common goal: repeatable successful product development and delivery.

Success-critical stakeholders and their goals

1. **Client**: Timely and reliable rollout of useful and affordable software.
2. **Software Manager**: Repeatable successful product evolution and delivery.
3. **Quality Manager**: Testable and assessable products and processes.
4. **Software & Systems Team**: Effective development environment & processes.
Software and Systems Engineering Companies
Selected Challenges at Industry Partners

Scope: family of products or product line, often systems of systems.

Business-level challenges

- **A1** Domain-specific **business definition** and scope of product-focused process
- **A2** **Product line parameters**: fixed and variable parameters, and their impact
- **A3** **Ecosystem stakeholders** and their interests, **value streams**

Process-level challenges

- **B1** **Continuous data collection** in a heterogeneous project environment
- **B2** **Reuse organization** of software artifact asset candidates
- **B3** **Engineering know-how**: continuous elicitation during a project & across projects

VDE 3695 process model

VDE 3695 part 2 „Engineering of industrial plants – evaluation and optimization; subject processes“, Association of German Engineers, November 2010.
Software and Systems Engineering Companies

Process Challenges at Industry Partners

- B1 **Continuous data collection** in a heterogeneous project environment
- B2 **Reuse organization** of software artifact asset candidates
- B3 **Engineering know-how**: continuous elicitation during a project & across projects

VDE 3695 process model

Market orders

VDE 3695 part 2 „Engineering of industrial plants – evaluation and optimization; subject processes“, Association of German Engineers, November 2010.
## Popular Software Process Approaches
### Matching to Challenges of Industry Partners

<table>
<thead>
<tr>
<th>Software process assessment approaches</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g., CMMI, Spice, MPS-SW/SV</td>
<td>+/</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+/</td>
<td>-</td>
</tr>
</tbody>
</table>

| Systematic software process models    | +/| +/| -  | -  | +/| -  |
| e.g., Waterfall, RUP, Spiral model, V-Modell XT |  |

| Software management approaches        | +/| -  | -  | -  | +/-| +/-|
| e.g., Agile, Lean, Kanban             |  |

| Software process improvement approaches | +/| +/| +/-| + | + | +/-|
| e.g., QIP, PDCA, **VDE 3695**, QATAM   |  |

**Legend:**
- + ... good match
- +/- ... partial match
- - ... poor match

<table>
<thead>
<tr>
<th>MPS-SV Maturity Levels</th>
<th>CMMI-SVC Maturity Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>A – In Optimization</td>
<td>5 – In Optimization</td>
</tr>
<tr>
<td>B – Quantitatively Managed</td>
<td></td>
</tr>
<tr>
<td>C – Defined</td>
<td></td>
</tr>
<tr>
<td>D – Largely Defined</td>
<td></td>
</tr>
<tr>
<td>E – Partially Defined</td>
<td></td>
</tr>
<tr>
<td>F – Managed</td>
<td></td>
</tr>
<tr>
<td>G – Partially Managed</td>
<td></td>
</tr>
</tbody>
</table>

Software process assessment levels

VDE 3695 process improvement process

See references for the software process approaches in the reference section.
B1: Continuous Data Collection From Engineering Environments – The Heterogeneity Issue

- Data sources in engineering environments are often heterogeneous, e.g.
  1. **Tool chains and disciplines** in systems engineering,
  2. Business software development **project consortium**,
  3. **Disciplines in game development** process.
Challenges from Heterogenity in the Engineering Process of Industrial Production Plants

1. “Engineering Polynesia”: tool islands with interfaces that do not fit seamlessly.
2. “Engineering Babylon”: engineers use project-level concepts, tools do not.

CDL “Flex Improvements” Contributions Overview

Scope: family of products or product line; systems of systems
- A1 Stakeholder interest elicitation and negotiation;
- A2 Software product lines in an organization (VDE 3695);
- A3 Software ecosystems in a business domain (SECO).

Engineering process analysis and improvement according to VDE 3695.
- B1 Data integration for process support and analysis;
- B2 Organization of reusable semi-finished products;
- B3 Eliciting and sharing engineering know-how with collective intelligence.

Lessons learned from case studies with research and industry partners
QM & Defect Detection
CDL “Flex Improvements” Case Studies with Focus On Process-Level Challenges

- **B1** Data integration for process support and analysis
- **B2** Organization of reusable semi-finished products
- **B3** Eliciting and sharing engineering know-how

1. Quality-Assured Tool Chains: **Semantic Dropbox**
2. Project Overview with the **Engineering Cockpit**
3. Early Defect Detection
4. Engineering Process Analysis
5. Reuse of **Software Artifacts** and Expert Know-How
Case Study: Quality-Assured Tool Chains: Semantic Dropbox – Context & Issues

- **Tool chains** link engineering process activities
- **Multitude of models and tools** used by engineers, management, and customers

Implementation often only as **manual activities** or **fragile constructs**, e.g., scripts.
- Issues: version management, work culture in **systems of systems environments**.
- Issue: **visibility of process information** from heterogeneous data sources.

- Effort and user friendliness for quality-controlled propagation of changes in heterogeneous software data models needs to be improved.
Case Study: Quality-Assured Tool Chains: Semantic Dropbox – Approach

The Semantic Dropbox

- provides **traceable and automated propagation of changes** between engineering tools.
- enables project participants to create **workspace folders**, and share and synchronize files in these folders with other project participants.
- **transforms data** between local representations of common concepts, so each project participant sees the representations of common concepts in his local representation format.

Mapping of local tool concepts to common project team concepts.

Typical common concepts in industrial plant engineering.
Case Study: Quality-Assured Tool Chains: Semantic Dropbox – Contribution & Improvement

Process improvements at software and systems engineering organizations

1. Domain experts can produce **traceable and secure tool chains** easily (in a few days instead of weeks).
2. Practitioners can **propagate changes** to engineering objects efficiently (in seconds instead of minutes).
3. Quality managers can **evaluate activities on engineering objects** (e.g., changes to library code blocks) automatically, even **across several projects**.
4. Project management: Clear traceability of changes to engineering plans coming from **external project partners**.

Change propagation in a heterogeneous environment.
Case Study: Quality-Assured Tool Chains: Semantic Dropbox – Lessons Learned

- **Process support** must be simple and efficient to be used regularly.
- The reduction of effort for the synchronization of signals in engineering systems of systems enables a **change in the work culture**.
- **Data integration** is the foundation for change process support and analysis.
- The **more accurate data basis** for progress and risk management facilitates **engineering process analysis** and improvement.
- Easy and reliable change propagation can have a profound impact on the **work culture**, the **engineering process**, and **product quality**.
Case Study: Project Overview with the Engineering Cockpit – Context & Issues

- **Data sets of systems-of-systems engineering groups** evolve concurrently, often without project-wide version management and progress tracking.
- Lead engineers and managers get a clear picture only shortly before project milestones, seeing risks unnecessarily late.
- In particular, **late changes** to plans are insufficiently visible to **enable the engineering process analysis for improvements**.
- Project managers need to see between milestones the overview on project progress based on current and systematically integrated data.
Case Study: Project Overview with the Engineering Cockpit – Approach

- Collect and integrate data from engineering teams and processes
- Web application „Engineering Cockpit“ provides
  - role-specific views for participants in the engineering team
  - relevant information on current and historic project activities.
- Users specify queries in SQL to the common data basis which the „Automation Service Bus“ provides and which contains all relevant changes of data from software tools and systems in the project.
- Project participants can configure all relevant views on queries in the Engineering Cockpit and therefore always have the current view on the relevant aspects of the project status.
- Evaluation with concepts from real-world projects.
Case Study: Project Overview with the Engineering Cockpit – Contribution & Improvement

The **Engineering Cockpit** provides engineers and managers with
- a platform to organize and perform specific **tasks across domain and tools**.
- means to **collaborate efficiently within the engineering team**.
- **integrated data on project progress and risks** as soon as the engineering groups check in their local data sets to allow adjustments early.

View on data and process states across domains:
- **Which safety variables are not connected correctly** to sensors across tools?
- **Which artifacts in status „approved“ were changed in the last week?**
- **Who changed signals of the artifact „Generator“ in the last two weeks?**
Engineering Cockpit: Project Management Overview

Role-based Project View

Project Status Overview

Role-based Events

Role-based Status & Applications

Signal Overview

Project Related Stakeholders
Case Study: Project Overview with the Engineering Cockpit – Lessons Learned

- **Engineering Cockpit**: role-oriented dashboard and activity options, in particular, for project and quality management, on recent and integrated data.
- Project management: View on engineering project and process status across domains in a distributed engineering project.
- **Claim management**: trace changes back to internal or external sources.
- Quality management: engineering process risk analysis, e.g., an unexpectedly large number of changes to engineering objects late in the project.
- **Engineering Process Analysis** is basis for Engineering Process Improvement.

Project progress and risk indicators based on integrated data from engineering teams and tools.
Case Study: Early Defect Detection in Heterogeneous Environments – Context & Issues

- **Collaboration of engineers** in a heterogeneous engineering environment, e.g., electrical, mechanical, software, and process engineers.
- Use case in the Automation Systems Domain, e.g., Hydro Power Plants. Challenge of **identifying defects across several scopes of planning**.
- Loosely coupled tools (technical heterogeneity of tools) and data models (semantic heterogeneity of data models) **hinder efficient change management and defect detection**.
- Need for linking heterogeneous data models, e.g., sensors, configuration, and software variables, to **improve engineering product quality**, e.g. “end-to-end” consistency checks.

---

Case Study: Early Defect Detection in Heterogeneous Environments – Approach

- Early defect detection with **integrated data from heterogeneous data sources**.
- The **mapping** of common concepts of the domain experts in a project to their local representations in software tools facilitates the analysis of changes and conflicts.
- **Reviews focus on changes** in engineering plans ("Change-Driven Inspection").
- **Automated “end-to-end” consistency checking** and **system testing** based on early defect detection approaches and “test-first” development.
Case Study: Early Defect Detection in Heterogeneous Environments – Improvement

- **Engineering Process Improvement** with defect detection and data collection.
- Early defect detection based on integrated data from heterogeneous data sources.
- Data mapping enables **analyses across engineering data models**.
  - Basis for deriving the review scope.
  - Basis for automated end-to-end consistency checks.
- Efficient defect detection in early phases of systems development based on **semantic technologies**.

Example Query Result

- (S1, “pressure”, “mbar”, C1, V_A, “pressure”, “mbar”)
- (S2, “level”, “cm”, C5, V_C, “level”, “m”)

```
SELECT ?Electric_ID, el:desc, el:type,
?Config_ID, ?SW_ID, sw:desc, sw:type
WHERE {
  el:E_short ekb:mapsTo ?Electric_ID.
  ...
}
```
Case Study: Early Defect Detection in Heterogeneous Environments – Lessons Learned

- **Semantic data integration** helps to **mitigate defects and risks** from **inconsistent engineering plans** in distributed engineering projects.
- Linking heterogeneous data supports quality assurance experts in **focusing on most critical system parts** (e.g., based on changes).
  - Expert support for focused inspection.
  - Foundation for automating consistency checks across engineering disciplines.
- Successful application of early defect detection approaches in various domains.

![Diagram of engineering processes and data flow](image)

- Reviews of changes at interfaces of several engineering scopes.
- Derivation of end-to-end test cases.
Case Study: Engineering Process Analysis – Context & Issues

- Engineering process analysis and improvement according to VDE 3695 (domain and project model)
- Continuous observation and improvement of workflows and engineering processes.
- Basic Steps: (1) Definition, (2) Implementation, (3) Data collection, and (4) Workflow evaluation.

1. Workflow definition
2. Implementation
3. Data capturing
4. Workflow evaluation

Case Study: Engineering Process Analysis – Approach

- Engineering Process Observation and Analysis Framework.
- Four layers from a business perspective.
- **Process automation** supports the definition, implementation, and evaluation of **process improvements**.
- Use Case: **Continuous Integration and Test (CI&T)**.

**Continuous Integration and Test Workflow Steps:**

1. Informal CI&T Process description
2. **Transformation to a more formal representation**
3. Derivation and Implementation of rules in the workflow engine, e.g., the ASB.
4. Event Data Capturing with log files.
5. **Expected Process modeled with PRoM**
6. Evaluation of the **expected process definition** with log-file data.

Case Study: Engineering Process Analysis – Contribution & Improvement

- **Automated data collection** based on executed process steps.
- Foundation for verification and validation of workflows and processes.
  - Conformance of executed process steps with real process data (PRoM).

  ![ProM tool image]

  - **Foundation for in-depth analysis** to identify engineering process bottlenecks based on advanced engineering process analysis.

<table>
<thead>
<tr>
<th>Color</th>
<th>Level</th>
<th>Waiting Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>•</td>
<td>High</td>
<td>&gt; 2.8</td>
</tr>
<tr>
<td>•</td>
<td>Medium</td>
<td>1.8 – 2.8</td>
</tr>
<tr>
<td>•</td>
<td>Low</td>
<td>&lt; 1.8</td>
</tr>
</tbody>
</table>

Each level contains 1/3 of the whole processes

Case Study: Engineering Process Analysis – Lessons Learned

- Comprehensive and efficient engineering process observation and analysis.
- **Process model evaluation** with PRoM enables fast and efficient feedback on implemented vs. planned processes.
- Simple process and workflow evaluation as foundation for compiling processes to larger engineering process maps, e.g., agile engineering processes in research and industry projects.

![Diagram showing different stages of software development lifecycle with stakeholders and key deliverables.]

Q: How to make reuse and expert know-how sharing activities beneficial both for the project and the organization?

Limitations of typical reuse and sharing scenarios:
1. Users store software artifacts in an unstructured, incomplete manner.
2. Lack of systematic approach to store and relate artifacts.
3. No information about artifact quality, usefulness, and traceability.

This leads to …
- A “dump” of artifacts which buries valuable contributions.
- Inefficient search for available elements to build upon.
- Issue must be addressed on project-independent level.
Case Study: Reuse of Software Artifacts and Expert Know-How – Approach

Steps

1. **Identify main workflow limitations**
   of mission-critical actors,
e.g., dispersed local engineering know how.

2. **Identify artifacts of interest**, e.g. requirements, solution elements, and reorganize them in a structured pool.

3. **Actors perform contribution activity** (create, modify, review artifact) which codifies content and knowledge.

4. **Mine relevant contributions from artifacts in the pool to create behavioral triggers for actors** e.g., notifications or signals.

5. **Actors with incentive** perform contribution activities, creating a **constant flow** of new contributions and triggers.

6. Continue with step 3.
Case Study: Reuse of Software Artifacts and Expert Know-How – Contribution

Software Reuse and Sharing System

- Web platform, which orchestrates reuse and know-how sharing activities across an actor community.
- Support reuse process and improvement efforts.
- Aggregates artifacts and knowledge about them.

Key Capabilities & Improvements

1. Structured adding and storing of artifacts.
2. Consistent artifact format enriched with metadata (e.g. context, rating) and relation information between artifacts.
3. Expert know-how about quality, usefulness, and user-driven recommendations.

- Bottom-up emergent coordination.
- Recommendation: Help engineers to identify best-fitting artifacts.
Example: Github

- Collaborative, global source **code repository** platform.

1. Code repository contributions via GIT cvs.
2. Links between artifacts (e.g. repo forks).
3. Expert know-how elicitation: e.g starring, activity monitoring, developer and project discovery.

---

**Analyze forks and contributions**

**Mark and build upon**

**Connect with developers**

**Explore new projects**
Case Study: Reuse of Software Artifacts and Expert Know-How – Lessons Learned

- Reusable assets are a well-structured basis for product-focused development.
- The collective intelligence (CI) environment enables better coordination of QA activity and recommendation.
- Users in other engineering teams can better filter and are better aware about relevant assets based on context and quality ratings.

Success/Risk factors

- **Commitment** of management and domain experts.
- Selection of suitable “artifacts of interest.”
- CI environment that is easy to use and adapt.
- **Calibration** of CI system to load assets.

- **Risk:** If CI system is not well integrated in daily workflow of users, the CI system will not be used. Example: use of a Wiki to document all “relevant engineering know-how” without considering CI systems success & risk factors.
Software Artifact Reuse and Engineering Know-How Elicitation and Sharing – Outlook

- How to make reuse and knowledge sharing activities beneficial both for the project and the organization?
  - Make better actor behavior easier and more rewarding.
  - Advanced data and process analysis capabilities.
  - Establish knowledge management for mission-critical know-how.

- Integrated Process Improvement Approach
  - Combination of both software architecture and process improvement.
  - Enhanced, bottom-up workflows based on the orchestration of CI system design and process improvement activities.

- Going ecosystem: Extending CI systems beyond organization borders.
  - Acquire knowledge and reuse software artifacts created by external partners and communities.
Lessons Learned on Product-Focused Engineering Process Analysis and Improvement

Challenges with families of products and systems-of-systems environments

- Heterogeneity of local data models in systems and engineering tools,
- Different speeds of processes and teams,
- Overview and control are hard to keep.

B1 Data integration for process support and analysis

- Data integration is the basis for automating continuous focused data collection.
- Continuous data collection enables engineering process analysis.
- Engineering process analysis prepares process improvements.

B2 Reusable semi-finished products

- Product-focused engineering: Product vision towards product line or ecosystem development.

B3 Eliciting and sharing engineering know-how

- Collective Intelligence design approach seems promising and should be investigated in a variety of application contexts.
Summary – Product-Focused Engineering Process Analysis and Improvement

- Stakeholders want repeatable and flexible product development.
- **Software process approaches** are helpful to provide an overview on best practices and on engineering process improvement areas.
- Industry trends such as **Enterprise 2.0**, the **Industry 4.0** initiative in Germany, and the **European Union “Horizon 2020”** program emphasize the investigation of new approaches such as **cognition and intelligent support for workers** in complex work spaces **systematic design of collective intelligence systems** for specific applications.

- Research **case studies with industry partners** discussed solution approaches, which were empirically evaluated in several application domains with heterogeneity.

- Visit us online at http://cdl.ifstuwien.ac.at
Backup Slides
Collective Intelligence – Overview & Example

Collective Intelligence (CI)
- Phenomenon: Group intelligence that emerges from collaboration, collective action and competition of many individuals.
- Sociology, biology, business, computer science.

CI and IT
- Achieved by hybrid systems in which humans and computers interoperate and complement each others' capabilities.
- Highly effective collection and distribution of knowledge
- Crowdsourcing, Social Web/Media, Social/Cognitive/Human Computing

Examples
- Github (code repositories)
- TopCoder (coding contests)
- Stackoverflow (questions & answers)
Collective Intelligence Systems Research

Our Principles
1. Enabling organizations to **reorganize work in new ways**.
2. **Hybrid human-computer** systems that harness the “wisdom of crowds”.
3. Support approaches, which propagate to **build on work already being done by others** instead of **reinventing the wheel**.

Our Take
- Foundation research on new kinds of **software architectures**.
  - Analysis of existing CI systems.
  - Design and develop novel CI systems in yet unaddressed domains.
- Knowledge management for **mission-critical know-how**.
  - Effective and cost-efficient **elicitation and sharing** of distributed and dispersed know-how.
- Process Improvement
  - Coordinated **bottom-up**, emergent **workflow mechanisms** and process risk management.
References

Product-Focused Engineering Processes
- VDE 3695 part 2 „Engineering of industrial plants – evaluation and optimization; subject processes“, Association of German Engineers, November 2010.

Building Engineering Bodies of Knowledge

Collective Intelligence
Software Process Approaches