Product Line Engineering at Siemens – Challenges and Success Factors
A report on industrial experiences in product line engineering
Munich, August 25th 2011

Hon. Adv. Prof. (Tsinghua) Dr.-Ing. Reinhold Achatz
Corporate Research and Technologies, Corporate Development Center
Introduction

Product Line Engineering (PLE) at Siemens

PLE Examples

Training and Qualification

Trends

Conclusion
Siemens is one of the world’s largest software companies ...

- Siemens has more than 15,000 software developers.
- More than 60% of Siemens’ business is based on software.
- Siemens spends more than 2 billion euros per year on software development.
Examples:

Automation devices

Industrial control systems

Power distribution systems

Rail systems

Medical devices
60% of our business depends on software

Functionality previously realized by mechanical or electro-mechanical systems is now being realized in software.

Many innovations are realized in software (cf. study by VW that 90% of innovations are realized in software)

Increased trend toward use of platforms & components

Large projects and large systems (hundreds of developers, millions of lines of code)

Multi-site development projects, ca. 30% of our software staff work in “low-cost” countries (source: Siemens Software Initiative, 2005)

Functional and non-functional requirements are important for our businesses e.g. real-time performance (50%), restricted hardware resources (ca. 40%), safety, etc. (source: Siemens Software Survey, 2002)

More “enhancements, customizing, maintenance” (ca. 60%) than “new” development (40%) (source: Siemens Software Survey)
Introduction

Product Line Engineering (PLE) at Siemens

- PLE Examples
- Training and Qualification
- Trends
- Conclusion
A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment and that are developed from a common set of core assets in a prescribed way.

Paul Clements and Linda Northrop, 2002

Reuse culture – reuse is planned and prescriptive at all levels of the organization

Business driven – business considerations determine core assets

Variability management – explicit management of commonalities and variability at all organizational levels and throughout all phases of the product development
Software Product Line Engineering –
Main ingredients

Core assets – set of reusable artifacts, i.e. requirements, domain models, architecture, patterns, concepts, documentation, test cases, budgets, work plans, process.

Domain – is a field of knowledge driven by business requirements and characterized by a set of concepts and terminology understood by stakeholders in that area – correlates with market (segment).

Commonality / variability analysis – the explicit identification of differences and commonalities between products / solutions on the basis of features (C/V analysis).

Prescribed reuse – support for systematically deriving products/solutions from core assets.
Software Product Line Engineering – Process

Core process of product line engineering

Domain engineering

- Scoping
- Economical analysis
- Domain system analysis / design
- Domain analysis
- Domain design
- Domain implementation / production

Domain testing

Platform

- Asset management
- Traceability
- Core Assets
- Configuration management
- Change management

Application / product engineering

- Definition
- Economical analysis
- System analysis / design
- Application / product analysis
- Application / product design
- Implementation / production

Application / product testing

Source: Pohl, Böckle, van der Linden
Product line engineering …

- … reduces overall cost of development.
- … increases quality.
- … reduces time to market.
- … reduces complexity of applications.
- … creates one look & feel within a product family.

Product Line Engineering shifts the focus from technology to business!
PLE process and organizational challenges

PLE process and organization setup enable efficient reuse through

- focusing on the chosen market segment’s needs (Requirements!)
- clear separation of responsibilities
- support for timely and economically sound and transparent decisions
- aligning parallel development processes following different goals
- fostering collaborative work instead of politics

Product Line Engineering shifts the focus from technology to business!
PLE decision criteria

Cost\textsubscript{PLE} + \sum \text{cost}\textsubscript{application} \ll \sum \text{cost}\textsubscript{application}

Cost with PLE

Cost without PLE
Without reuse, the breadth and longevity of Siemens’ product portfolios would not be possible. Mastering platforms and product lines has to be a core competence throughout all disciplines.
Product Line Engineering at Siemens

- First definitions of “reusable software” in the early 80’s
- First platform initiatives in the 90’s mainly driven by development
- Participating in EU-funded projects Esaps, Café, Families, starting in the late 90’s
- Current participation in product line and product line related projects (Indenica, Cesar, SPES)
- Active support by Corporate Research and Technologies since early 2000, especially towards product line engineering
- Joint research and consulting in requirements engineering, processes, architecture and testing
- Regular events with external and internal speakers and workshops
- High management attention for platforms
Product Line Trends at Siemens

- Increasing *interconnectivity* between product lines that span several business units
- *Long-lived* platforms that become trademarks and major marketing instruments
- *System* families: software product lines in combination with mechatronic product lines

Product lines are shared between business units. Responsibility shifts from local business unit to higher level management. Corporate departments support with company wide knowledge sharing and qualification programs.
Introduction

Product Line Engineering (PLE) at Siemens

PLE Examples

Training and Qualification

Trends

Conclusion
Siemens Industry Example: Totally Integrated Solutions
Siemens Industry Example: Totally Integrated Solutions

Unified Product and Production Environment
Enables the digital enterprise from top floor to shop floor…

Enterprise Level
- ERP
- PLM

Management Level
- MES
- Plant Engineering

Operator Level
- SIMATIC Automation Designer
- SIMATIC IT R&D Suite
- SIMATIC IT Intelligence Suite
- COMOS Plant Engineering
- TECNOMATIX Digital Manufacturing

Siemens covers a huge range of systems and devices both in industry and process automation.
Objective: integration, common look & feel and convergence
Totally Integrated Automation Portal: Strategic Goals

TIA Portal VISION

“The open engineering framework for all Siemens automation products offering the most intuitive and efficient engineering in the automation market.”

Strategic goals for TIA Portal

- SW platform for all Siemens automation products and systems
- Best in class for intuitive and efficient engineering
- Provide open interfaces for engineering data and functions
- Ensure seamless connection to Digital Engineering
Totally Integrated Automation Portal as engineering framework for all automation tasks

Integration of Engineering-Software...

- SIMATIC Controller
  - STEP 7
  - S7-SCL
  - S7-GRAFH
  - PLC SIM

- SIMATIC HMI
  - WinCC flexible
  - WinCC
  - Options

- SIMATIC NET
  - SINEMA

- SIMOTION
  - Scout

- SINAMICS
  - StartDrive

- SINUMERIK
  - HMI PRO CS
  - SinuCom NC

... in a shared and common Software-Framework
Totally Integrated Automation Portal: Organizational Challenges

A product line integrating four business units in two divisions …

…spanning many kinds of markets from off-the-shelf-products over OEM to large solution businesses

…world wide markets with localization

…development distributed all over the world

…supported by one platform development team

… currently used in 7 integrated products and many more to come.

This requires high level management commitment to product line engineering.
Siemens Healthcare launched a domain engineering organization in the early 90’s. syngo® is a core asset base, a product line and a strong Siemens Healthcare trademark.
Starting in the 90’s:
Common Look & Feel for Siemens Imaging

syngo® architecture

Common user interface (easily made possible, thanks to joint syngo® platform)

syngo® builds on Windows platform

GG-specific applications (modules)

Impact of syngo® on R&D cost

Number of major product lines using syngo®

<table>
<thead>
<tr>
<th>Year</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994-1998</td>
<td>2</td>
</tr>
<tr>
<td>1999</td>
<td>4</td>
</tr>
<tr>
<td>2000</td>
<td>7</td>
</tr>
<tr>
<td>2001</td>
<td>10</td>
</tr>
<tr>
<td>2002</td>
<td>11</td>
</tr>
</tbody>
</table>

R&D cost

- >70' EUR of R&D cost savings on annual basis

syngo® cost

Cost for same functionality realized in GG specific platforms

1) Before capitalization

Source: Siemens Healthcare
Today:
Integrated Workflow Support for Imaging and IT

Siemens Healthcare Example: syngo® Imaging

Patient process in healthcare institutions

Soarian

The next generation Healthcare IT

syngo

Leading edge imaging supporting the clinical workflow
syngo® Imaging: Universal Access *Anywhere*.

**syngo.via** uniquely integrates imaging modalities and IT – anywhere*.

*prerequisites include: Internet connection to clinical network, DICOM compliance, meeting of minimum hardware requirements, and adherence to local data security regulations.
- Introduction
- Product Line Engineering (PLE) at Siemens
- PLE Examples

- Training and Qualification

- Trends
- Conclusion
Corporate Training and Qualification Program

Economic success of many products is highly dependent on key software knowledge and practices and architecture is seen as the most important aspect.

Qualification of software architects has highest priority for Siemens.

Levels & roles

Class A Project:
- High complexity (platform / product line)
- High degree of innovation
- Big business impact, high risk
- Cross-functional, distributed structure, big team

Class B Project:
- Moderate innovation in technology, medium risk
- Medium business impact, medium teams

Class C Project:
- Enhancing known technology and requirements, low risk
- Single site development, small teams
Competence spider for software architects (SSWA and SWA)

Competence level scale
- Basic
- Advanced
- Expert

Testing and quality
- Quality assurance
  - Test processes and methods

Software processes
- System development
  - SW development processes

Leadership
- Project management
  - Social skills

Business and strategy
- Business case understanding
- Global development
- Legal issues

Requirements engineering
- Product management
- Requirements engineering

Software architecture and development
- Product line engineering
  - Architecture and design
- Configuration management
  - SW design methods

Software Architect (SWA)
Senior Software Architect (SSWA)
Senior Software Architect (SSWA) program

Some content details

- Software Architecture and Development
  - Platform- & product-line engineering / architecture
  - Domain & system scoping / modeling
  - Iterative, risk-driven, requirements-driven, test-driven design & development, test-first as design method
  - Strategic & tactical design
    - Architecturally significant functional and non-functional operational requirements
    - Happy day and rainy day scenarios for functional requirements
  - Design for usability
    - Enforce the architecture vision
      - Communicating the architecture to stakeholders
      - Mentoring & coaching
      - Avoiding architectural drift
    - Conflict management
      - Participants learn how to use different conflict models to handle conflicts
      - Participants solve relevant challenges in their practical work

- Social Skills and Leadership

- Business Processes and Strategy

- Requirements Engineering

- Software Architecture and Development

- Testing and Quality
A qualification program for software architects requires more than training

Elements of the software architect qualification programs

- Setting standards for software development
- Fostering best practice sharing and experience exchange
- Assuring practical experience for critical projects

Qualification (courses and on-the-job training)

Software architects' network

Certification
<table>
<thead>
<tr>
<th>Guiding Principles of the Software Curriculum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Architecture as well as the continuous governance of it is the key throughout the whole lifecycle as well as across releases.</td>
</tr>
<tr>
<td>2. Build on existing basis where feasible (from technical and business perspective) and be able to recognize when such reuse is not suitable.</td>
</tr>
<tr>
<td>3. Avoid unnecessary technological platform development and use technical standards and products available on the market.</td>
</tr>
<tr>
<td>4. The product (lifecycle) manager in product/system business and the project manager in project/solution business is and must act as the owner of the main requirements and quality characteristics.</td>
</tr>
<tr>
<td>5. Pay particular attention to specifying, testing, and realizing non-functional requirements (NFRs), often overlooked but are extremely important.</td>
</tr>
<tr>
<td>6. Be prepared and able to handle changing requirements, but be aware about the risk of late changes.</td>
</tr>
<tr>
<td>7. Synchronize well across the technical disciplines (software, mechanics, electronics).</td>
</tr>
<tr>
<td>8. Work together truly as a team, avoid “silo” thinking, be willing and able to speak and understand the other roles and disciplines.</td>
</tr>
<tr>
<td>9. Work iteratively and test-driven, foster defect prevention from the beginning, and strive to identify and resolve technical and business risks early. Getting real and early feedback, both from customer and from realization team, is essential.</td>
</tr>
<tr>
<td>10. Structure the system to avoid unnecessary complexity, and to actively enable and support multi-site development.</td>
</tr>
<tr>
<td>11. Strive for transparency and base decisions on clear business / technical (not political) reasons.</td>
</tr>
<tr>
<td>12. Do not underestimate the importance of soft skills, these can be particularly important for convincing and motivating.</td>
</tr>
</tbody>
</table>
- Introduction
- Product Line Engineering (PLE) at Siemens
- PLE Examples
- Training and Qualification
- Trends
- Conclusion
Trend: PLE and Agile Development

Agile development is de facto standard, also for core asset development.

Challenges

- Stable product line architecture to allow parallel development of core assets and products
- Scoping for resolving conflicts between several product interests
- Consequent variability management
- Test automation for variability testing

Advantage

- “Do the simplest thing that could possibly work” for minimizing complexity
- Continuous stream of platform deliveries with a relevant and meaningful set of features
- Reduced risk for product development
Trend: Inner Source

Managing cross business unit assets with inner source, to increase speed and share platform ownership.

Common Artifacts
- Collective Code Ownership
- Product and Test Code

Governance
- Guidance & Rules for Consistency and Continuity
- Defined Quality Gates

Shared Responsibility
- Collaborative Development
- Protect Functionality via Smoke Tests
- Continuous Regression Testing and “ility” Trending
Introduction

Product Line Engineering (PLE) at Siemens

PLE Examples

Training and Qualification

Trends

Conclusion
Conclusion

To be able to meet the challenges of today's product lines,

we …

… invest in research on product line engineering,

… initiate and support platforms, integrating PLE in our business strategies and portfolio vision,

… train architects, product and test managers on PLE-related topics and build best practice sharing networks.
Thank you!

Thanks to Christa Schwanninger, Christian Lescher and their colleagues from Siemens Software Initiative for the support in preparing the presentation!