Product Line Architecture
Systems

• Systems often come in families: basic, regular, professional, enterprise,…

• Can we share components?

• Is architecture useful is encouraging an approach that uses components as building blocks for systems.
Software Product Line

• A collection of software-intensive systems sharing a common, managed, set of features that satisfy the specific needs of a market segment or mission that are developed from a set of core assets in a prescribed way.
  – Product lines are a feature of many physical products (e.g. cars)
  – Useful in the context of software engineering
Software Product Lines

• Are directed by business goals in a particular application domain.
• The products in a product line share a software product line architecture.
• Products are structured by the product line architecture and are built from services and components.
• Architecture and components are the core assets used to satisfy the business goals.
• Product line leverage commonality and limit variability of the product.
Product lines spread costs over several products

- Requirements and requirements analysis
- Domain model
- Architecture and design
- Performance engineering
- Documentation
- Test cases, data, and plans
- Skills
- Processes, methods and tools
- Defect fixing
- Components and services
Benefits to the organisation

• Large-scale productivity gains
• Improve time to market
• Maintain market presence (rapidly evolving variants)
• Sustain growth
• Improved market agility
• Better use of skills
• Enable mass customisation
• Gain control of configuration
• Improve product quality
• Better predictability of cost, schedule and quality
Costs of a product line

• Architecture: flexible enough to support variation in the products
• Software components: general enough to support variability
• Test plans, cases, data: take account of variation in components
• Business cases: must operate at the level of a product family
• Project plans: generic and extensible to deal with variation
• Tools and processes: must support architecture, variation, configuration, ..
• People, skills, training: need to be skilled in architecture and product lines.
Product Line Economics

Product line summary

Core Process Activities

• Core asset development: improving the base components in terms of qualities, products they support, and architecture.

• Product development: identifying and building products to meet market need inside the product line.

• Management: monitoring and improving the processes, tools and practices.
Introducing Product Lines

• **Proactive:** Up-front investment to develop the core assets – need to know the market well (maybe have an already established set of products)

• **Reactive:** Start with one or two products and use them to generate core assets.

• **Incremental:** Develop core assets as the business need evolves.
Example: Bosch Gasoline Systems

Based on: Software Product Lines in Action, Chapter 9, Frank van der Linden, Klaus Schmid, Eelco Rommes
Company facts of Bosch Gasoline Systems

Organisational size: ~ 1,000 developers.
Starting Mode: Strategic focus, based on existing assets.

Experienced improvements:
- Reduction of calibration effort (~20%) and maintenance.
- Reduction of the resource consumption (20–30%).
- Product line definition reflecting market variance.

Business: Addressing new business challenges was a major driver. Three market segments were identified as a starting point.

Architecture: A new software architecture was developed, but assets were derived from the existing asset base.

Process: Bosch works on CMMI level 3. Systematic process engineering provided an important basis for product line development.

Organisation: A restructuring of the organisation was necessary to reflect the different roles in a product line organisation.
Bosch

• Employed 250,000 people in 2006, now 375,000
• Revenue €70.6 bn in 2015
• Gasoline systems is one of the largest divisions in 2006, employing 1000 developers.
• Every new engine controller leads to a new variant
• Thousands of controllers are in use.
Rapid Change

- Sensors: 100%, 150%
- Actuators: 100%, 140%
- Calibration parameters: 100%, 400%
- Features: 100%, 170%
- Lines of Code: 100%, 1570%
- Memory (MB): 100%, 2400%

1995 vs 2005
Goals

• Competitiveness:
  – Reduced hardware resource consumption
  – Reduced time to market for new features
• Development efficiency
  – Reuse
  – Easy configuration of software products
  – Increased planning accuracy
• Quality
  – Interface integrity
  – Reuse of core assets
• Customer needs
  – Differentiation by individual software solutions
  – Clear feature-cost mapping
Approach

• Process:
  – Consider business strategy
  – Consequences for products
  – Consequences for processes and methods
  – Consequences for tools
  – Consequences for the organisation
Architecture
Architectural Features

- Control of resource consumption e.g. memory (because of cost).
- Distributed development requires good interface management.
- Layers provide the possibility to share applications without knowing the details of particular sensors or actuators.
- Reuse goals: Applications can be used across different generations of system; “core” software is highly configurable and is reused via reconfiguration; “Vehicle functions” can be used across gasoline and diesel engines.
- Standardisation is important, Bosch play an important role in AUTOSAR: http://www.autosar.org
Architecture Standardisation

AUTOSAR Layered Architecture

Application Layer
- Application 1
- Application 2
- Application 3
- Application 4

Runtime Environment (RTE)

Application Abstraction Layer

Service Layer
- OS
  - Os
  - SchM
- Mode Management
  - ComM
  - EcuM
  - Nm
- Diagnostic
  - Dem
  - Dcm
  - FIM
- Memory
  - Nvm
  - Crc
- COM Services
  - Com
  - PdUR

Firmware
- Wdglf
- Spi, RamTst
- Gpt, Mcu, Wdg, Icu
- Port, Dio, Fwm, Adc

ECU Abstraction Layer
- CAN
  - CanSm
  - CanNm
  - CanTp
- FlexRay
  - FrSm
  - FrNm
  - FrTp
- LIN
  - LinSm
  - LinTp
- CanTc
- Can
- FrTc
- Fr
- Lin

Microcontroller Abstraction Layer

(1) based on AUTOSAR 3.x
Component Redesign

• Focussed on: reuse; simplification of calibration; resource consumption; stabilisation of interfaces (within the architecture)

• Redesign progressed by:
  – Analysing existing software inventory: features, sources of variability; relation to product line; document interdependency.
  – Concept development and design of components: simplification; configurability; architecture driven structure; document relations between features and components;
  – Baselines for variants of software components: document baselines; implement; maintain up-to-date document and implementation.
Phased Introduction

• Investigate and customise product line engineering.
• Design and pilot adequate processes and methods.
• Roll out and institutionalise in the standard development process.
Roll out

<table>
<thead>
<tr>
<th>Action</th>
<th>Purpose</th>
<th>Helpful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll out by redesign of existing platform.</td>
<td>Initial development of work products like interfaces, feature trees, overview functions, etc.</td>
<td>Product line engineering coaches.</td>
</tr>
<tr>
<td>Series of product line process workshops with middle management.</td>
<td>Understanding, acceptance and management support.</td>
<td>Management commitment.</td>
</tr>
<tr>
<td>Training program for product line engineering and architecture.</td>
<td>Understand product line engineering and internalise new methods.</td>
<td>Use of domain-specific examples.</td>
</tr>
</tbody>
</table>
Toolchains

Feature model

Features mapped to components

Component model

Requirements spec.

Tool support

Product configuration.
Toolchains

• Tools were primitive and not well adopted – strong requirements for: feature modelling; architecture documentation; interface documentation, checks and management; linkage between features and implementation; feature-based product derivation.

• Strong drive to standardise to develop tools and ease interchange between Bosch as a supplier and purchasers of Bosch products
Maturing Toolchains
Key Elements

• Management commitment to: build up product line knowledge; establishing a product line business unit to drive product line engineering; commitment to rollout.

• Process excellence: in parallel to adoption of product line Bosch to achieve Capability Maturity Model Integration level 3 (i.e defined process characterised for the organisation and is proactive)