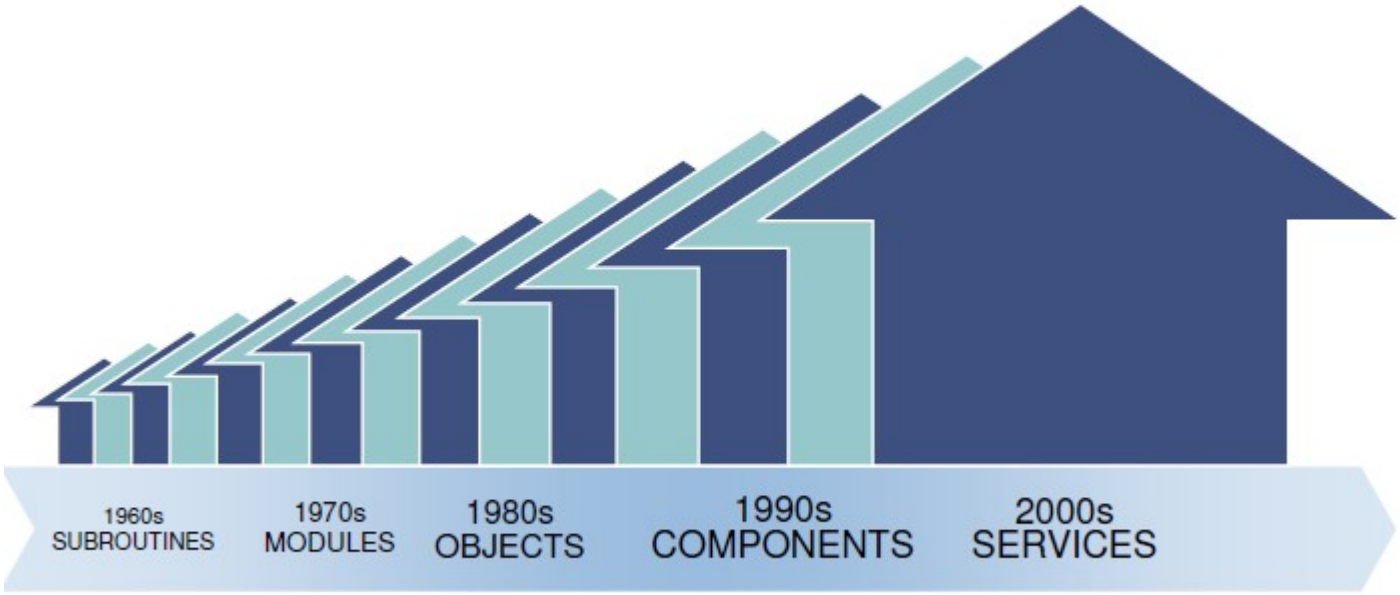
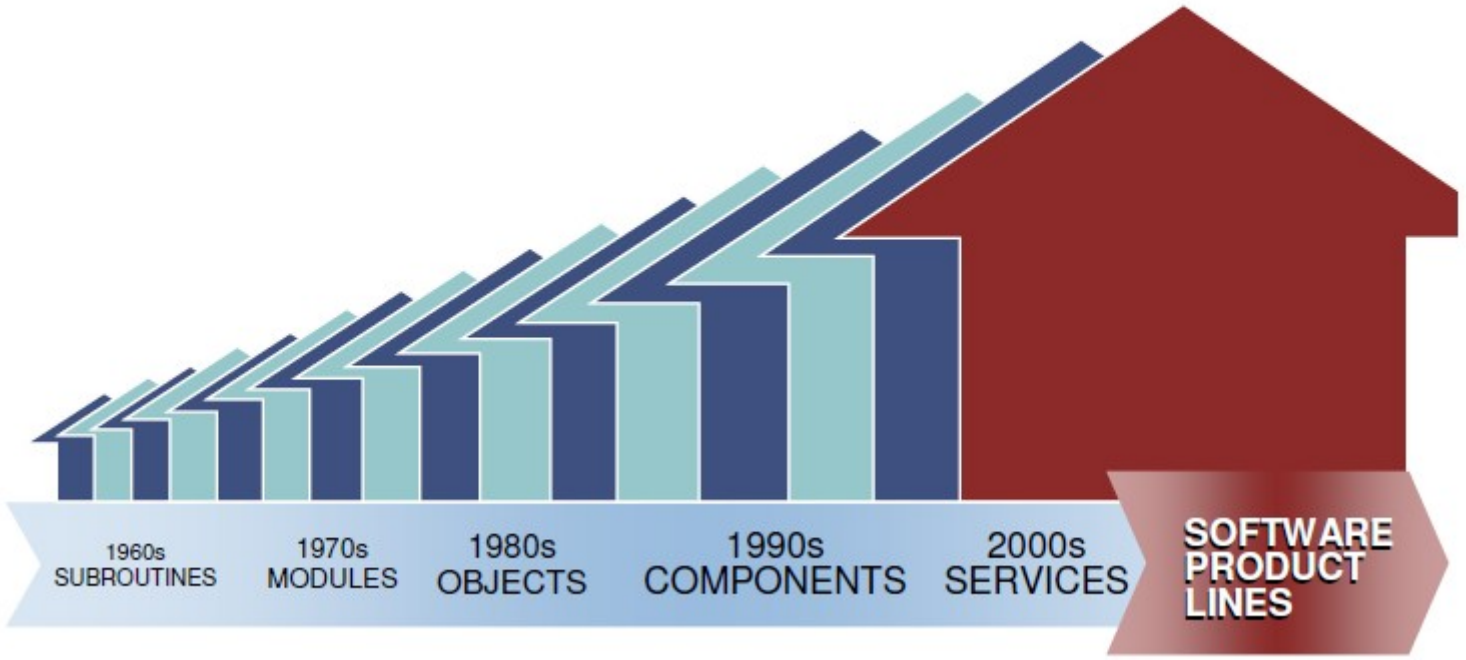
Software Product Lines: What, Why, When and How

Sailaja
28.03.2012

Reuse History



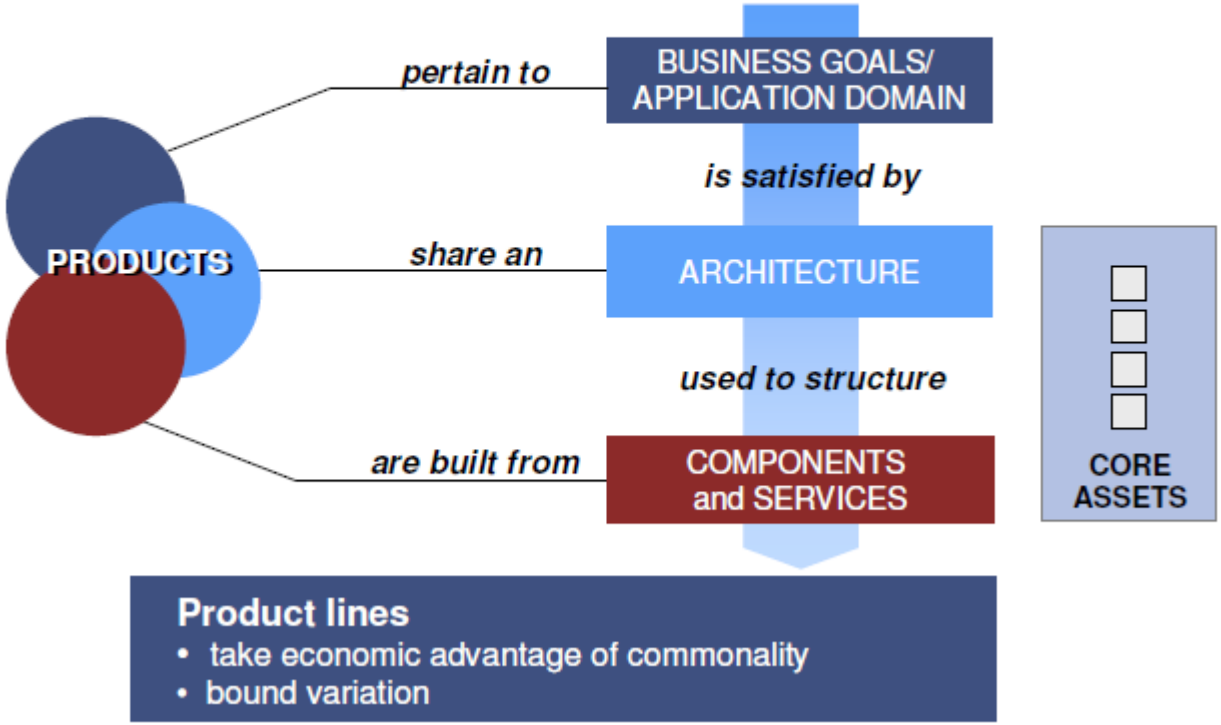
Reuse History: From Ad Hoc To Systematic



What is a Software Product Line?

- A set of software products that **share** common features but are each **different** in some way
- Each product is produced using a set of configurable, reusable assets using the **same production process**
- The product line is aimed at a **specific** market or market segment

Software Product Lines



Why have different products?

- Customer specific requirements
- Market differentiation e.g. high vs. low-end products
- Differences between target devices
- Legal or environmental regulations e.g. emissions regulations

Example Software Product Lines

- Embedded systems:
  - Mobile e.g. games, GUIs
  - Medical e.g. diagnostic devices, pacemakers
  - Automotive e.g. engine-management systems
  - Military e.g. software-defined radio
- Web services:
  - Online marketing and stock market analytics
  - Quote management services
- Others:
  - ATMs

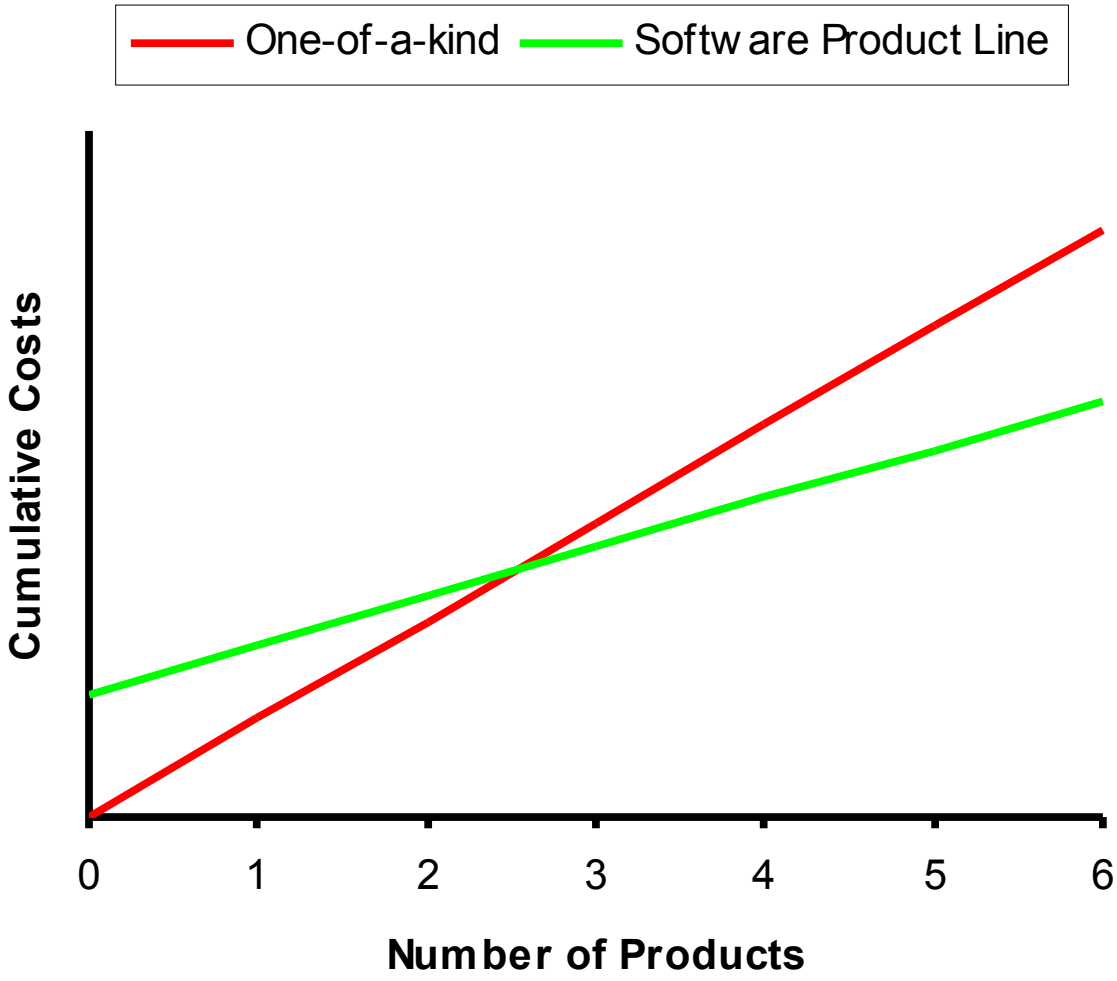
Why start a Software Product Line?

- Can develop and support more products
- Big productivity gains
- Improved product quality
- Faster time-to-market

- … but often because **something** has to change

Why start a Software Product Line?:

Break-even analysis



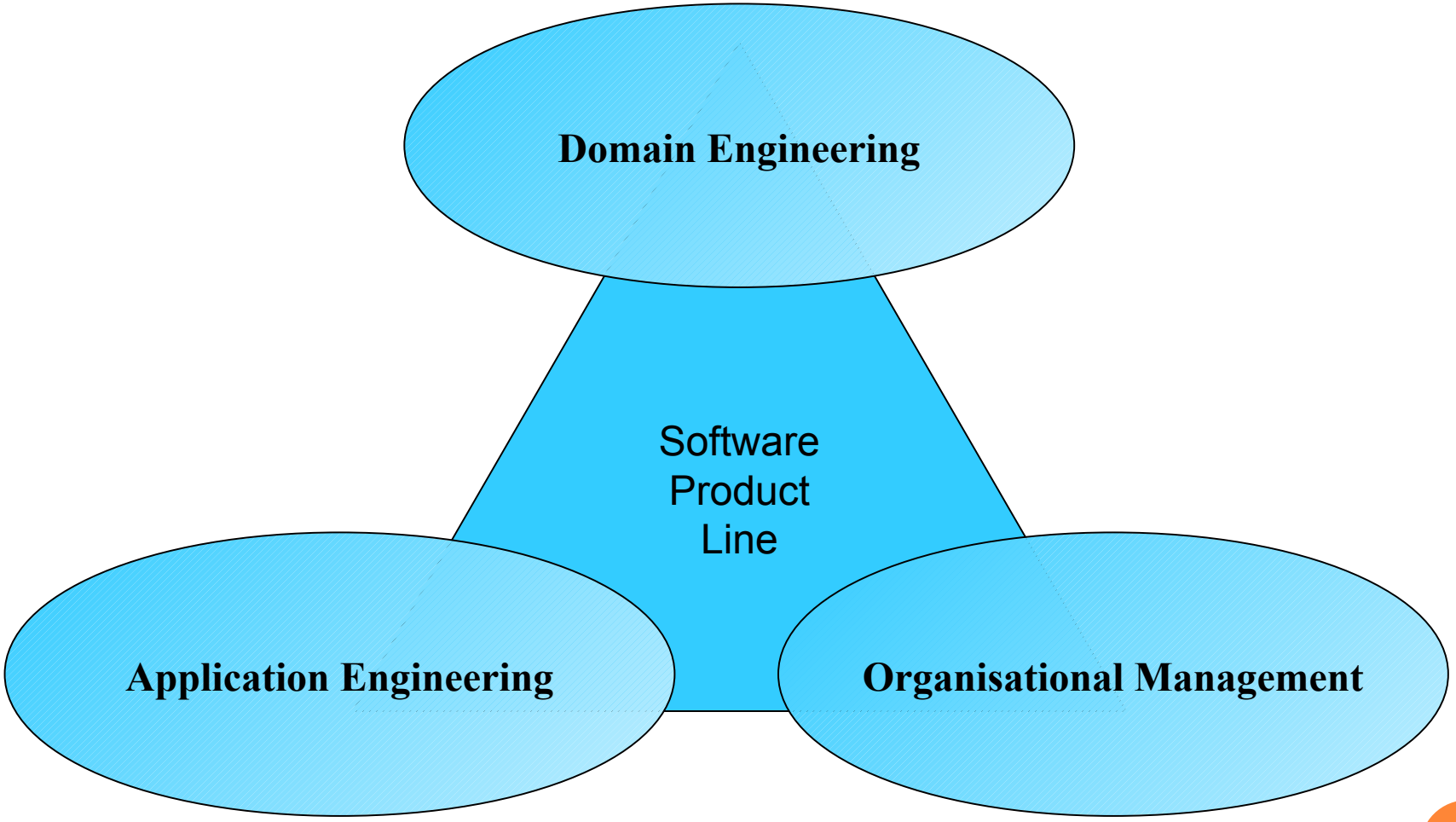
There are various ways of drawing this graph but the intention is to demonstrate that at some point investment in core reusable assets pays off.

Documented case studies in the literature show that between 2 & 3 products can give a break even on investment.

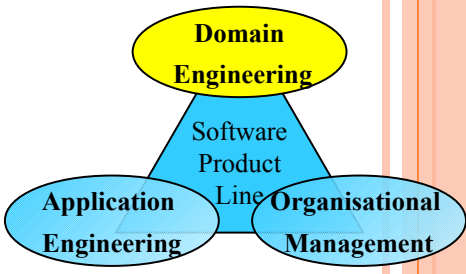
Investment in reusable assets need not occur up front. Incremental adoption is viable.

What is a Software Product Line?:

Three Essential Activities



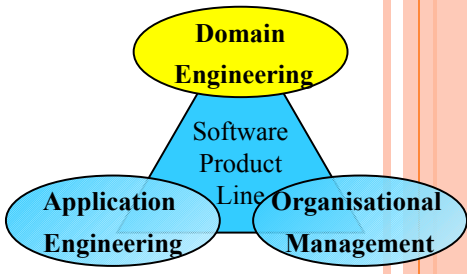
Domain Engineering



- Creates the core assets that are reused by multiple products
- Core assets can include:
  - The product line scope definition
  - The product line architecture
  - Reusable software components
  - Tools to support application engineering

Domain Engineering:

Defining Product Line Scope

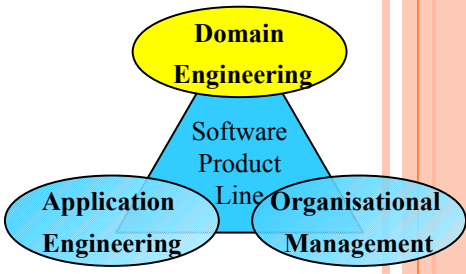


- Decide which products and features will be supported in the product line:

1. List **candidate** products and features
2. Assess which features are common and which vary between products
3. Decide which features (and products) will be supported
    - Optimise the product portfolio

Domain Engineering:

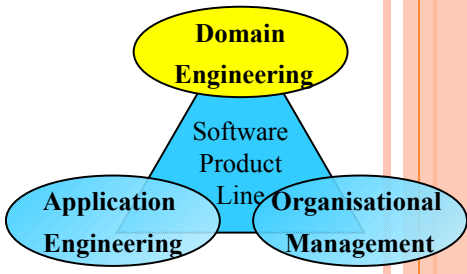
Finding Products and Features

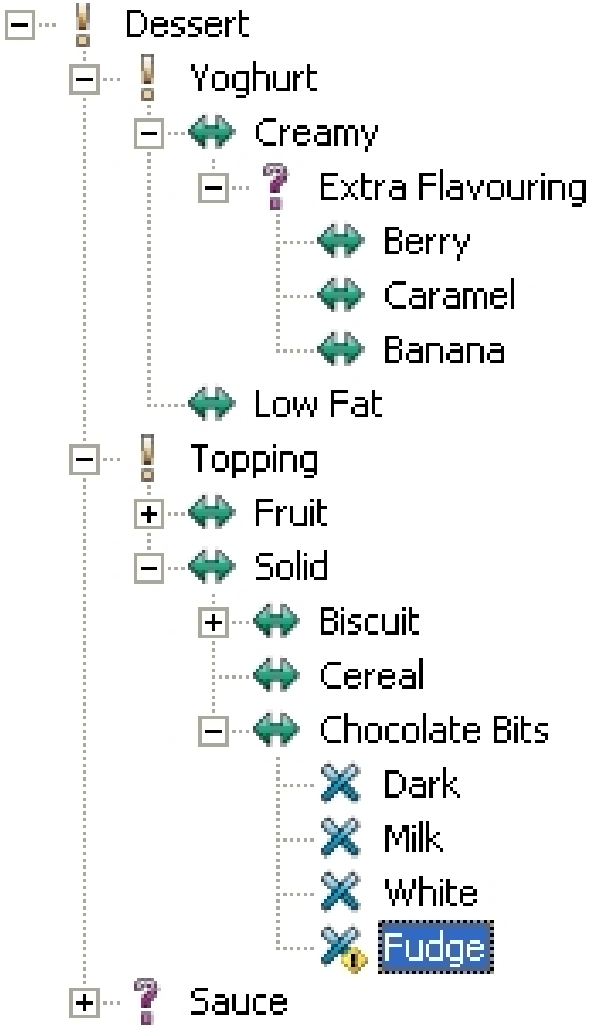


- Talk to people (product managers, architects, customers)
- Revisit any existing documentation
- Prepare a Product Line glossary
- Clarify feature relations with a *feature model*
  - A model showing the common and variable features
  - Also shows dependencies between features

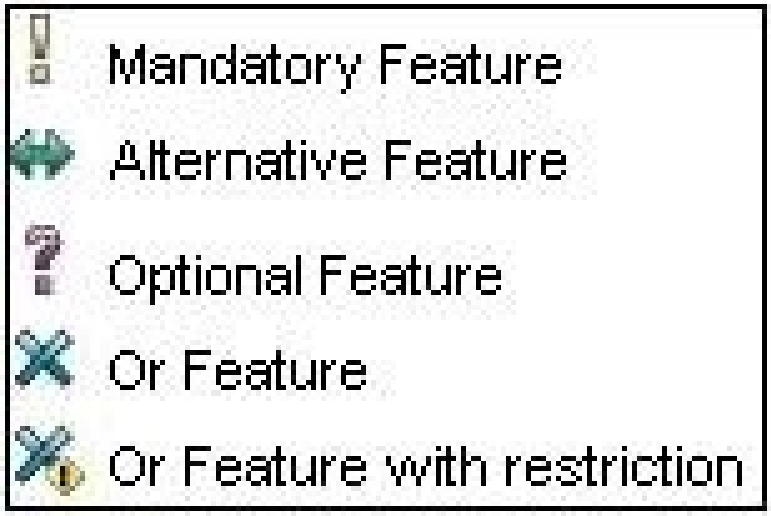
Domain Engineering:

Sample Feature Model



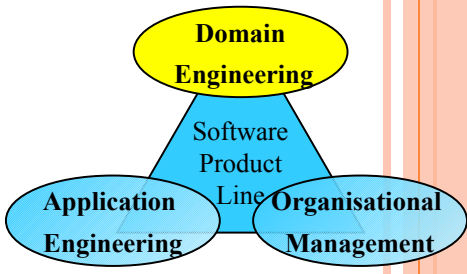


Adapted version, obscures product from which data was taken.



Domain Engineering:

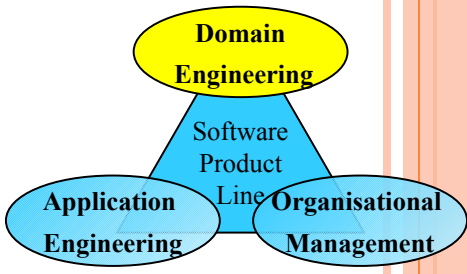
Product Line Architecture



- Must cater for **all** required product line features
  - **All** common features
  - **All** variable features
  - **All** product-specific features
- Supports common features using standard development techniques
- Supports variable features through specified *variation points*
- Respects product-specific features

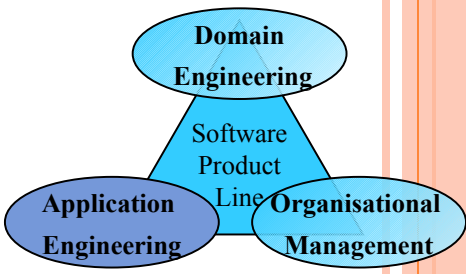
Domain Engineering:

Component Provision Options



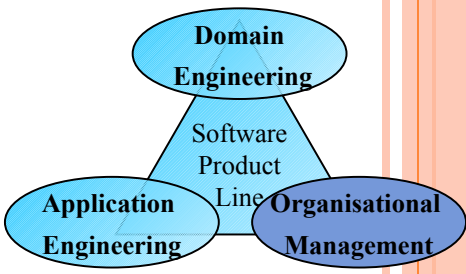
- Convert what you already have
  - Refactor, wrap
- Buy components off-the-shelf or get someone else to build them for you
  - Check for compatibility and for long-term support
- Build components from scratch
  - Remember - reusable software is a bit harder to develop

Application Engineering



- Determine application requirements
- Select and configure core assets (instantiate the architecture)
- Create and integrate product-specific assets
- Test, Document, Deploy

Organisational Management



- Starting the product line
  - Specify the business goals
  - Resource the product line effort
- Running the product line
  - Planning, managing risk
- Institutionalising the product line
  - Communicating the product line culture
  - Improving the product line

More complex than in one-of-a-kind settings.

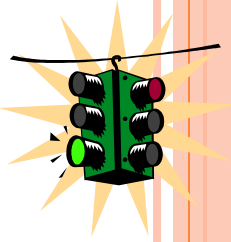
When to start a Software Product Line

- Start **only** when you have

Product Line Potential

- ✓ All **essential** criteria
- ✓ Some **supporting** criteria
- ✓ Few **exclusion** criteria

Product Line Potential:
Essential Criteria

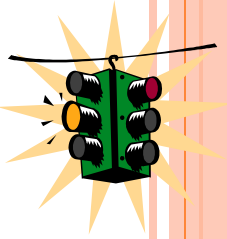


Necessary conditions for you to benefit

from PLs.

- More than one product is developed or planned to be developed
- Products have common functional requirements
- Products have common non-functional (quality) requirements

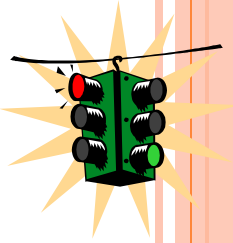
Product Line Potential:
Supporting Criteria



Indicate you probably would benefit from PLs.

- Assets already reused between products
- Market demand for multiple products
- Recognition that something needs to change e.g. due to quality or complexity issues

Product Line Potential:
Exclusion Criteria



Indicate you probably wouldn't benefit from PLs.

- Software plays a small part in the overall product
- Market instability
- Unpredictable technological change

How to start a Software Product Line

- Assess your current state
- Prepare a business case
- Select your basic approach
  - Extractive <-> Proactive
- Reorganise?

There are a range of possible approaches. These could be considered to be extreme opposites.

How to start a Software Product Line:

Assessing your current state

- What do you have already?
- Who do you have?
- What do you know?
- How much can you spend?
- How long can you wait?
- What feels right?

How to start a Software Product Line:

Preparing a business case

- Specify the goals of the product line
  - e.g. faster time-to-market, more variants
- Document the expected costs, benefits and risks of several alternative ways of meeting the goals
- Different stakeholders will have different perspectives - make the business case personally relevant to them

How to start a Software Product Line:

Choose your approach - Extractive

- Base product line on existing products
- Identify what is common and what varies
- Create one copy of the common pieces of software
- Create one copy of each variable piece of software
- Generate a product by merging the common piece of software with selected variable pieces

- Existing architecture may not meet future

How to start a Software Product Line:

Choose your approach - Proactive

- Begin by creating your core assets
  - Product line scope definition
  - Product line architecture definition
  - Reusable software components
  - Other core assets
- Run a pilot project
  - New product
  - Prototype / Toy product
- Migrate all products to product line

How to start a Software Product Line:

Reorganise?

- Align the organisation to the product line architecture (*q.v.* Conway)
- Alternative models:
  - Development Unit
  - Business Units
  - Core Asset Unit

Shaping the Organisation:

Development Unit

A single group develops all core assets and products

- ✓ Good for small organisations
- ✓ Simplest model
- ✓ May already be how things work
- Take care to match staff with appropriate work

Shaping the Organisation:

Business Units

Multiple groups develop both core assets and products

- ✓ Good for mid-sized organisations
- Evolution and cost-sharing of core assets an issue
- ✖ Conflicts between units can occur
- ✖ Business units tend to add product-specific features
  - ✖ Risk of product line degradation

Shaping the Organisation:

Core Asset Unit

A single group develops all core assets and multiple groups develop products

- ✓ Good for large organisations
- ✓ Overview of entire product line leads to better core assets
- ✗ Core asset unit may lose product focus
- ✗ Core asset unit can be a bottleneck

Success Factors

- High-level leadership
- Domain experience
- Architectural vision
  - *"Although you have a brilliant architecture, beware of the power of a horde of developers running wild. They need continuous guidance and support."* - Anders Heie, Nokia
- The ability to say **no** to your customers … **and** to your product managers

Pitfalls

- *Pit Diggers*
  - *"Some pits exist naturally, others are dug"* - Jim Dager, Cummins Inc.
- Testing
- Not-invented here
- Staff churn
- Lack of support from conventional development tools

In summary

- In a Software Product Line, products are built through **systematic reuse** of a set of configurable, reusable assets
- Product Lines can yield significant business benefits e.g. productivity, quality, time-to-market
- The adoption barrier for Product Lines needn't be high, incremental adoption is viable

Software Product Line books

- **Adopting and Evolving a Product Line Approach**, by Jan Bosch, Addison Wesley 2000
- **Generative Programming: Methods, Tools and Applications**, by Krzysztof Czarnecki and Ulrich W. Eisenecker, Addison Wesley 2000
- **Software Product Lines: Practices and Patterns**, by Paul Clements and Linda Northrop, Addison Wesley 2002
- **Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools**, by Jack Greenfield, Keith Short, Steve Cook, and Stuart Kent, Wiley 2004
- **Software Product Line Engineering: Foundations, Principles and Techniques** by Gunther Bockle, Klaus Pohl, Frank van der Linden, Springer 2005