



د. لمياء أبوزيد

# Software Evolution : TOC

---

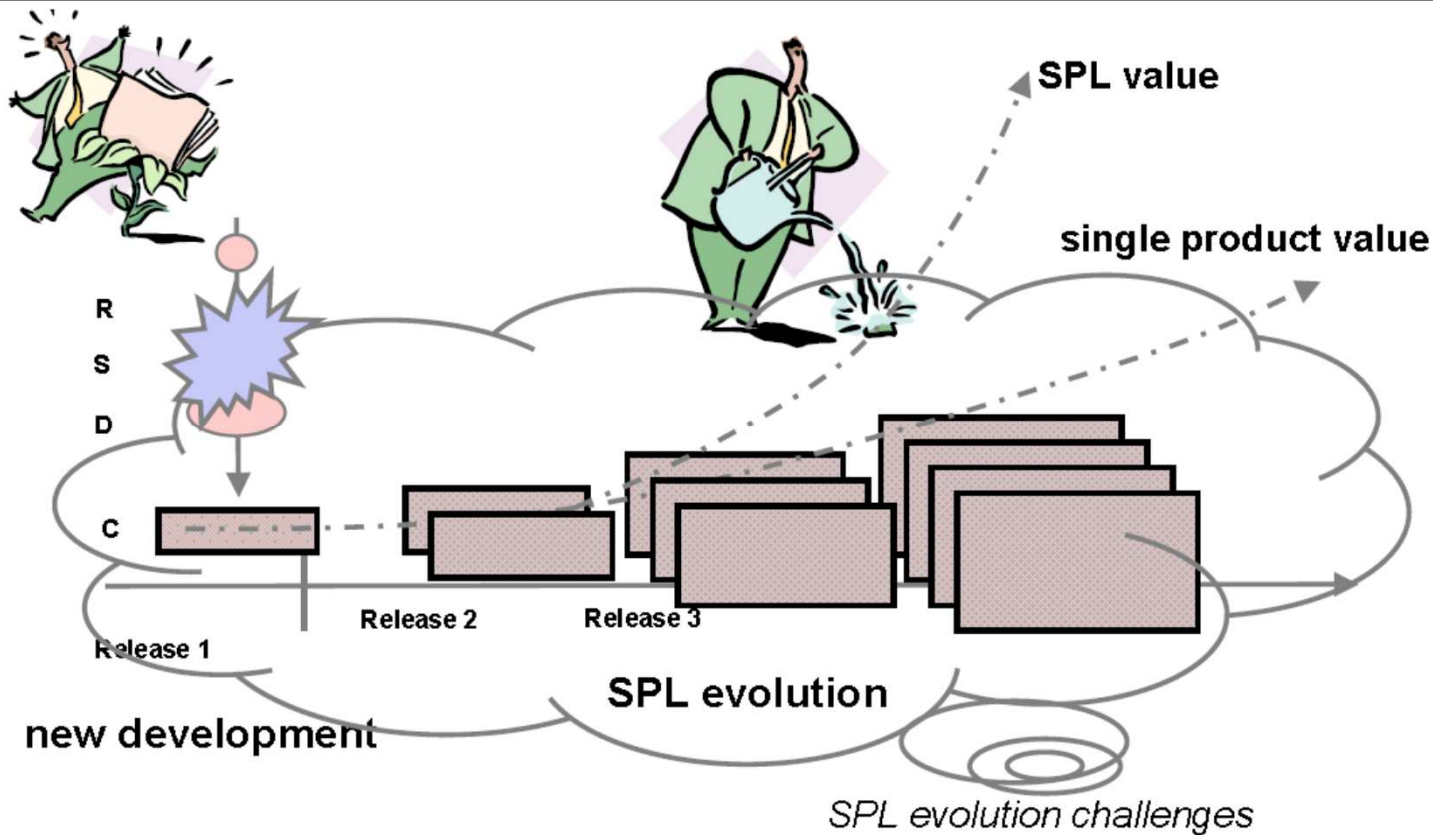
1. Introduction to Software Evolution
2. Taxonomy of Software Maintenance and Evolution
3. Evolution and Maintenance Models
4. Reuse and Domain Engineering
5. Program Comprehension
6. Impact Analysis
7. Refactoring
8. Reengineering
9. Legacy Information Systems

# Development-for-Reuse - Domain Engineering

---

- ❑ Domain Analysis refers to the set of activities that support defining **multiple related products** from the very beginning of the software development process.
- ❑ **Software Product Lines** (also called **Product families**) apply the concept of *product lines* defined in manufacturing to the software development process. It moves the software development from a **product-based** development to a **product line-based** development
- ❑ Examples:
  - HP has a product line for their Printers (drivers software)
  - A product Line for Content Management Systems
  - Philips has a product line for their MRI scanners
  - MS has a product line for their SQL server

# Development-for-Reuse - Domain Engineering

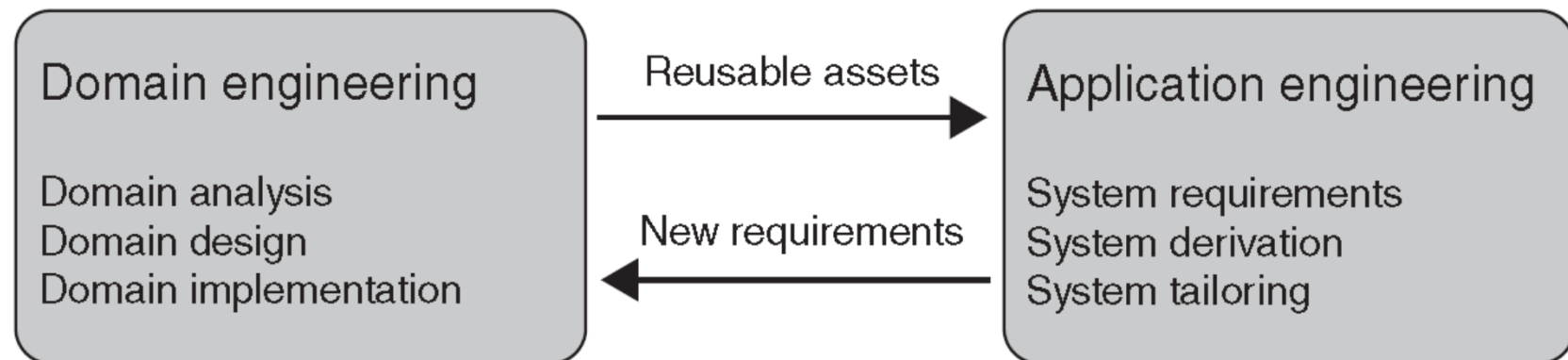


© 2004, Klocwork Inc.

# Development-for-Reuse - Domain Engineering

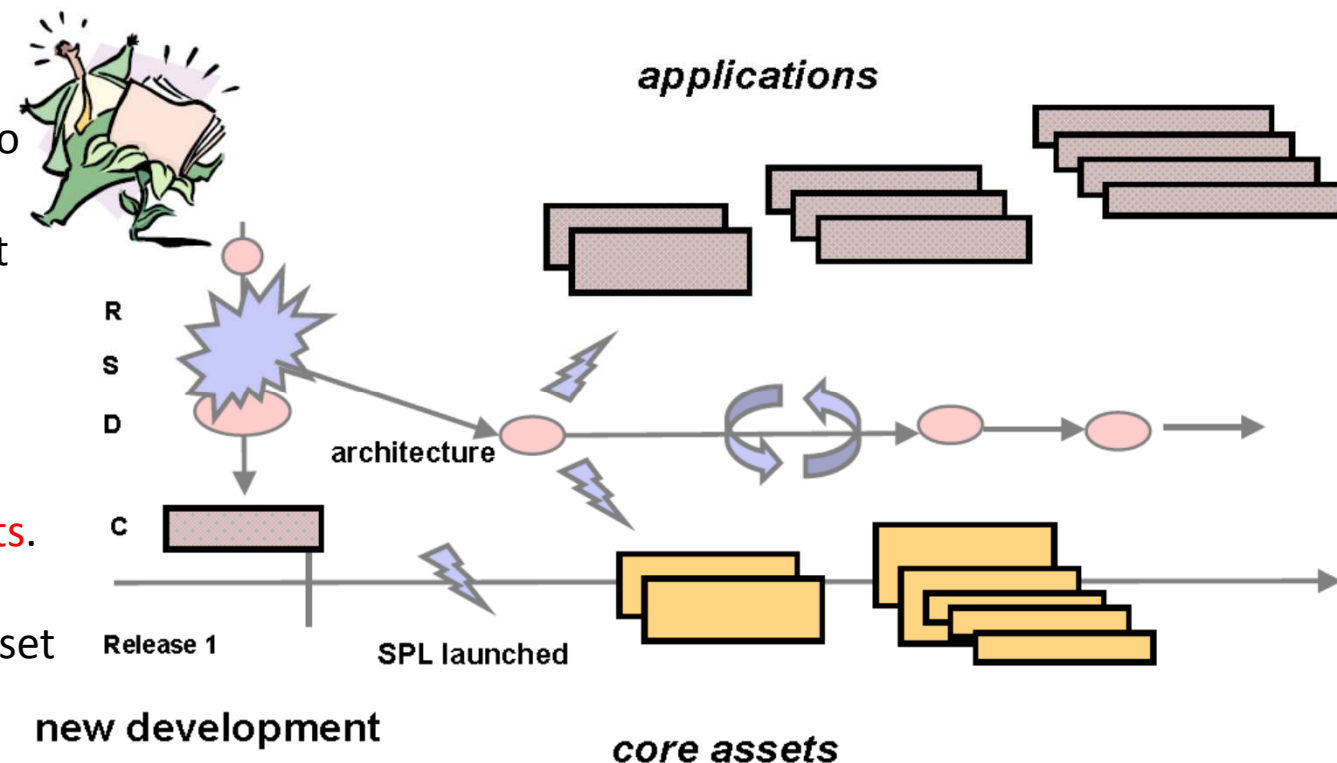
□ Domain Engineering is a process that includes the following activities:

1. Domain Analysis
2. Domain Design
3. Domain Implementation



# Development-for-Reuse - Domain Engineering

- ❑ The **product line architecture** allows it to be **configured** to produce different products meeting the needs of different customers.
- ❑ A software product line is commonly defined to consist of **a common architecture**, and **a set of reusable assets**. Together they are used in **configuring** individual products by using a different set of assets in each individual product.



# Development-for-Reuse - Domain Engineering

---

□ Domain analysis comprises three main steps:

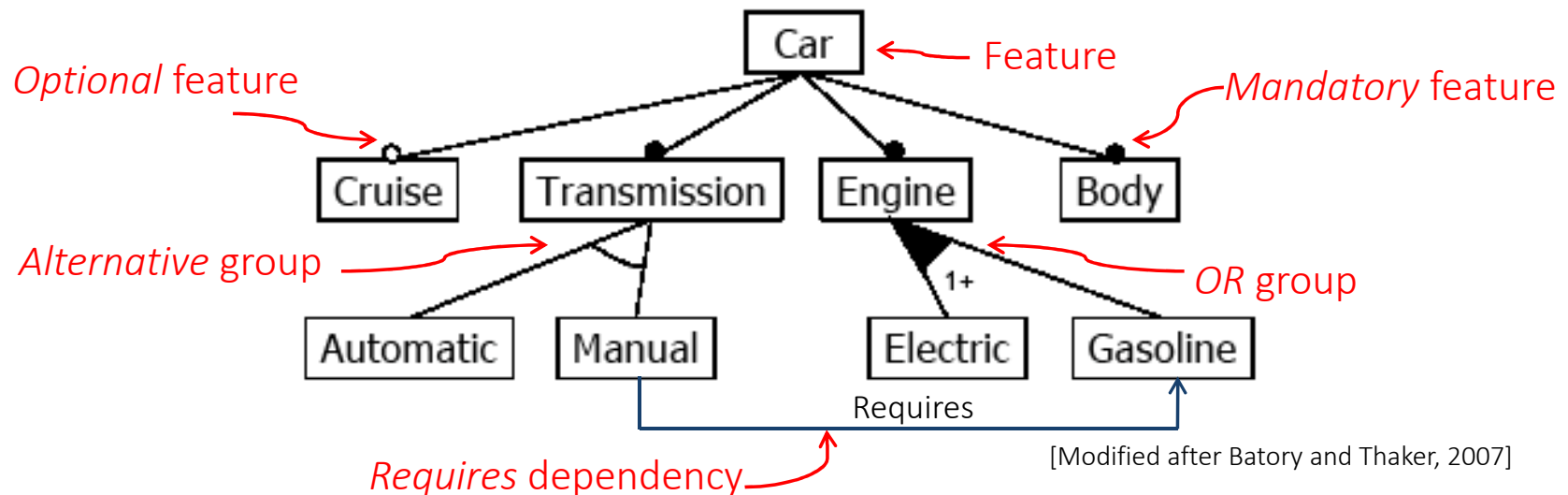
1. Identify the family of products to be constructed
2. Determine the variable and common features in the family of products
3. Develop the specifications of the product family.

□ **Feature Oriented Domain Analysis (FODA)** technique defined by Kang (1990) is used for domain analysis and modelling

□ FODA is used for the analysis of software product lines due to its ability to represent and model **commonality** and **variability** among various applications of a certain domain.

# Development-for-Reuse - Domain Analysis

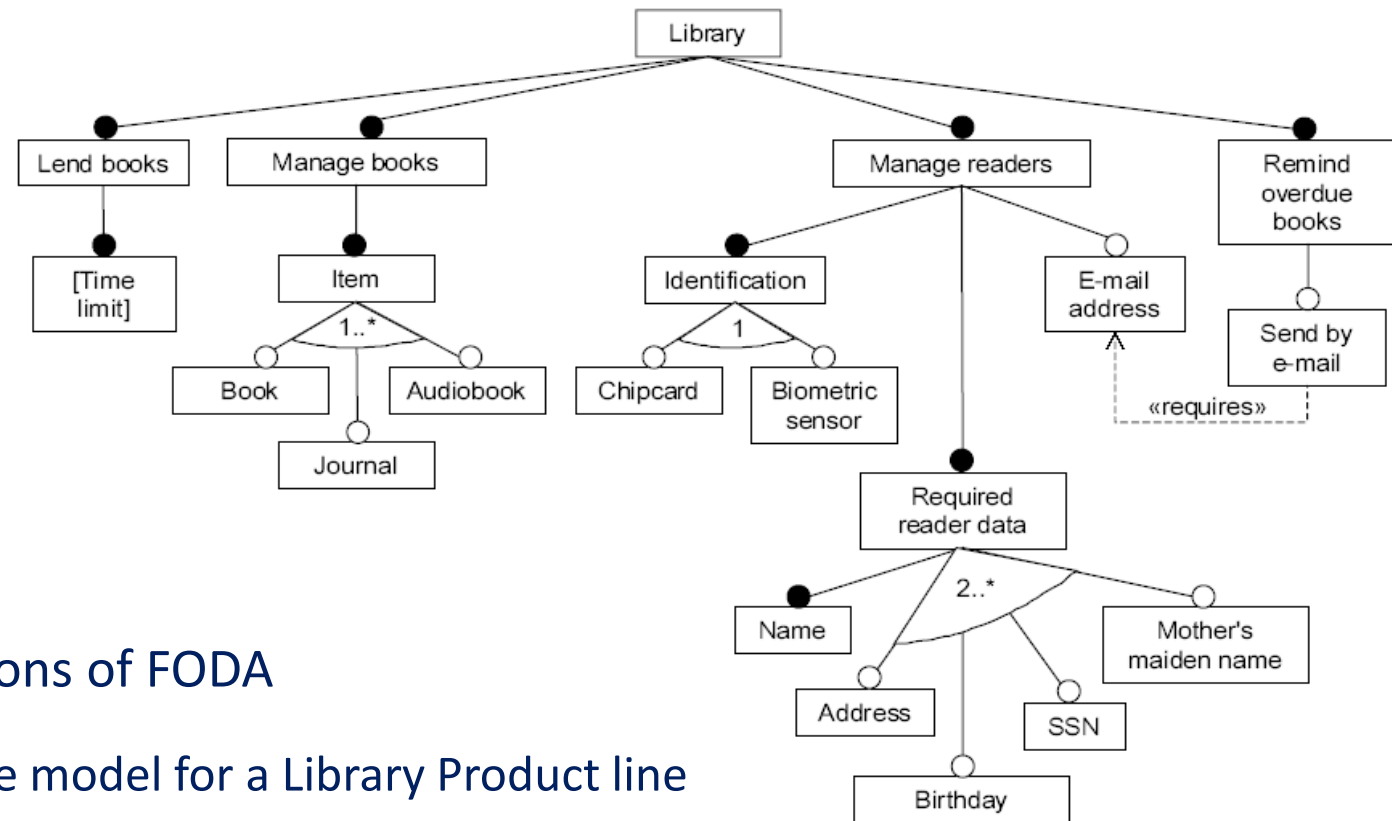
- Feature models represent the common (reusable) features of the product line in addition to their relations and dependencies showing how they contribute to its variability.



- Several extensions to FODA have been defined.



# Development-for-Reuse - Domain Analysis



## Example on extensions of FODA

- Riebisch et al. feature model for a Library Product line

# Development-for-Reuse - Domain Design

---

□ Domain Design has two main activities .

1. Develop a **generic software architecture** for the family of products under consideration
2. Develop a plan to **create individual systems** based on reusable assets.

□ The design activity emphasizes a **common architecture** of related systems. The common architecture becomes the **basis for system construction and incremental growth**.

□ The design activities are supported by **architecture description languages (ADLs)**, and **interface definition languages (IDLs)**

# Development-for-Reuse – Domain Implementation

---

- ❑ Domain implementation involves the following activities:
  1. Identify **reusable components** based on the outcome of domain analysis;
  2. Acquire and **create reusable assets** by applying the domain knowledge acquired in the process of domain analysis and the generic software architecture constructed in the domain design phase;
  3. Catalogue the reusable assets into a **component library**.
- ❑ **Development, management, and maintenance** of a **repository of reusable assets** make up the core of domain implementation.

# Development-with-Reuse – Application Engineering

---

- ❑ Application engineering refers to **development-with-reuse** process, it is the phase of **product development** and is complementary to domain engineering.
- ❑ Application engineering composes specific application systems by:
  1. **Reusing existing assets** (defined in domain engineering )
  2. **Developing** any new components that are needed
  3. **Reengineering** some existent software
  4. **Testing** the overall system.
- ❑ Similar to the standard practices in software engineering it begins by **eliciting requirements, analyzing the requirements, and writing a specification.**

# Application Engineering - Product Instance Development

---

## ☐ Elicit stakeholder requirements

- Use existing family member as a prototype

## ☐ Choose closest-fit family member

- Find the family member that best meets the requirements

## ☐ Re-negotiate requirements

- Adapt requirements as necessary to capabilities of the software

## ☐ Adapt existing system

- Develop new modules and make changes for family member

## ☐ Deliver new family member

- Document key features for further member development

# Application Engineering - Product Line Configuration

---

## □ Design time configuration

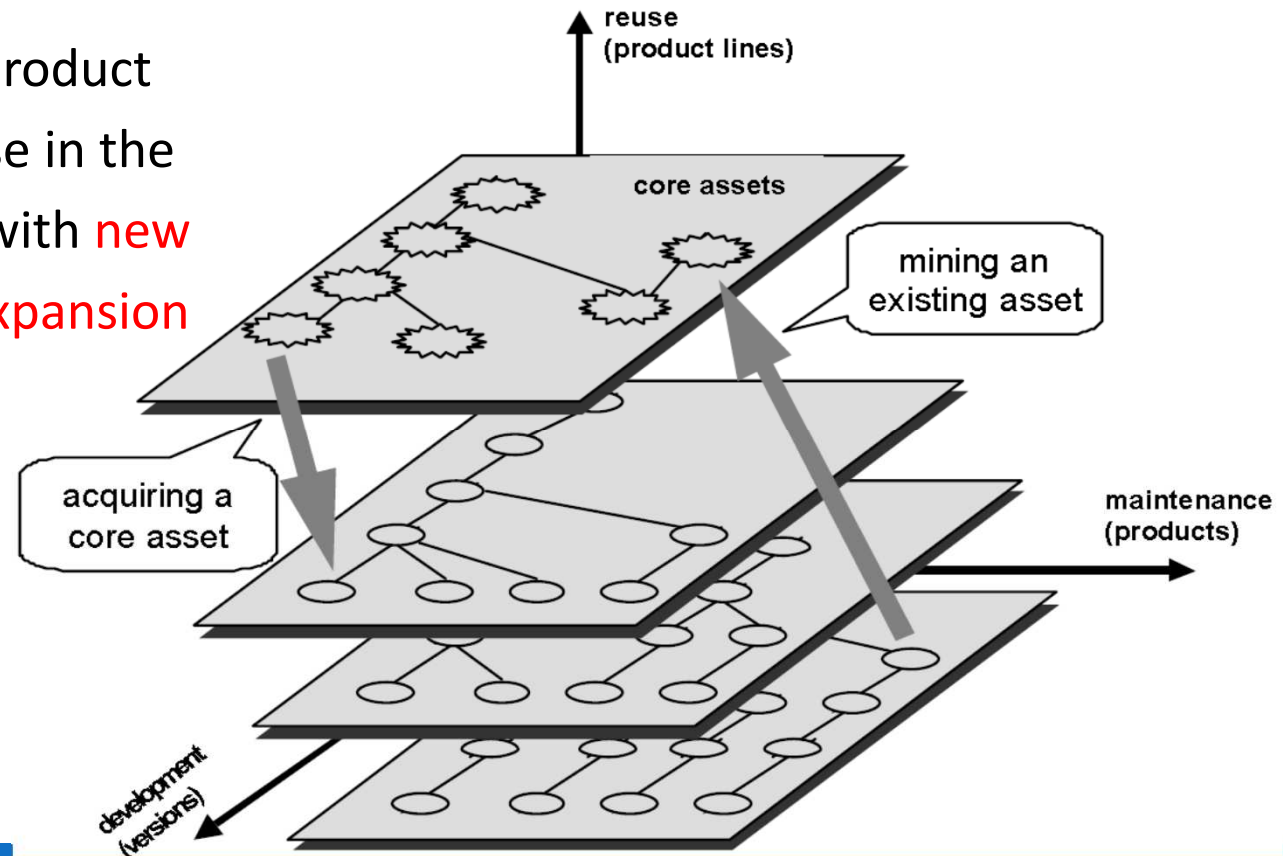
- The organization that is developing the software modifies a common product line core by developing, selecting or adapting components to create a new system for a customer.

## □ Deployment time configuration

- A generic system is designed for configuration by a customer or consultants working with the customer. Knowledge of the customer's specific requirements and the system's operating environment is embedded in configuration data that are used by the generic system. E.g. Drupal

# Product Line Evolution

- The overall evolution of the product line is expressed by the increase in the number of planes, associated with **new products**, as well as with the **expansion of the core assets plane**.



# Domain Engineering Approaches

---

The following domain engineering approaches reported in literature:

- ❑ Draco - a basis for **Generative programming**
- ❑ Domain Analysis and Reuse Environment (DARE) **Oracle**
- ❑ Family oriented Abstraction, Specification, and Transportation (FAST) - **Alcatel-Lucent**
- ❑ Feature-Oriented Reuse Method (FORM) – **extends FODA**
- ❑ Komponentbasierte Anwendungsentwicklung (KobrA) - **component-based application development**
- ❑ Product line UML-based software engineering (PLUS) - **UML based PL approach**
- ❑ Product Line Software Engineering (PuLSE) - **PL development with product-centric focus**
- ❑ Koala - **Philips Corporation**
- ❑ Reuse driven Software Engineering Business (RSEB) - **Use-case-driven reuse method based on UML**



# Assignment

---

- ❑ In your opinion what will allow us to easily configure Product lines? (i.e. what will allow for seamless product configuration/derivation )