CS5030

Software Reuse

Learning objectives

- On completing this lecture and associated reading, you should
  - Understand the motivation for software reuse and the benefits and challenges associated with it
  - Be aware of the different scales in which reuse can take place and the different artefacts that can be reused
  - Understand reuse as applied to application frameworks, software product lines and application systems

Software reuse

- In many engineering disciplines, systems are designed by composing existing components that have been used in other systems
- Traditionally software engineering has been more focused on original development
- It is now recognised that a design process based on systematic software reuse will produce better software more quickly and at lower cost
- There has been a major switch to reuse-based development over the past few years

Artefact reuse in software engineering

- Reuse of non-software artefacts
  - Architecture styles, software architectures, design patterns, designs of components / sub-systems / systems, test suites, etc

- Reuse of software artefacts
  - At different scales

Reuse based software engineering

- System reuse
  - Complete systems, which may include several application programs may be reused
- Application reuse
  - An application may be reused either by incorporating it without change into other or by developing application families
- Component reuse
  - Components of an application from sub-systems to single objects may be reused
- Object and function reuse
  - Small-scale software components that implement a single well-defined object or function may be reused

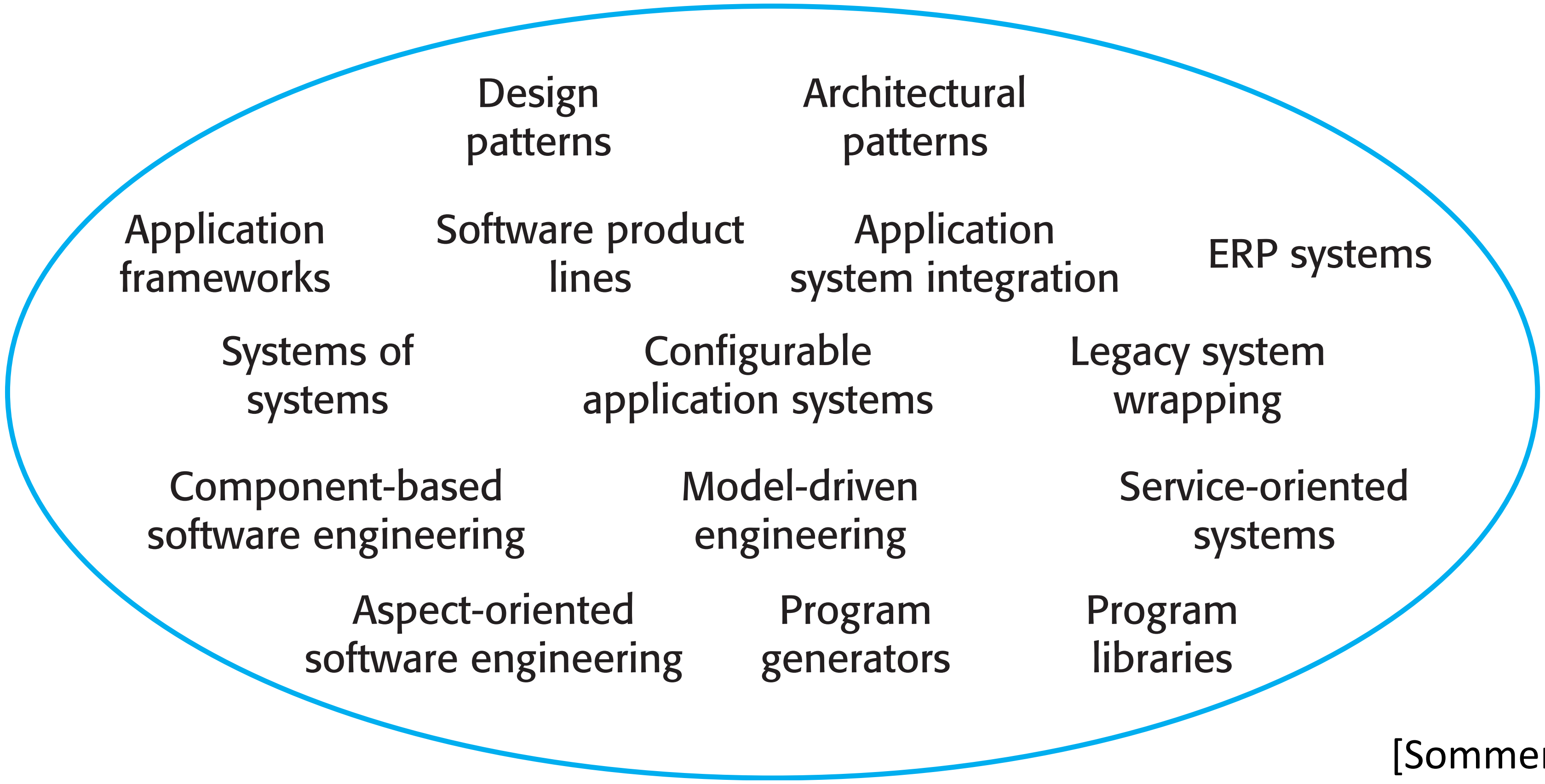
Benefits of reuse

- Faster development
- Effective use of specialists
- Increased dependability
- Lower development costs
- Reduced process risk
- Standards compliance

Challenges of reuse

- Creating, maintaining and using a component library
- Finding, understanding, and adapting reusable components
- Increased maintenance costs
- Lack of tool support
- Not-invented-here syndrome

The reuse landscape



[Sommerville, 2016]

Reuse planning factors

- Development schedule for the software
- Expected software lifetime
- Background, skills and experience of the development team
- Criticality of the software and its non-functional requirements
- Application domain
- Execution platform for the software

Application frameworks

- "..an integrated set of software artefacts (such as classes, objects and components) that collaborate to provide a reusable architecture for a family of related applications."
- Frameworks are moderately large entities that can be reused
  - Somewhere between system and component reuse
- Frameworks are a sub-system design made up of a collection of abstract and concrete classes and the interfaces between them
- The sub-system is implemented by adding components to fill in parts of the design and by instantiating the abstract classes in the framework

Classes of frameworks

- System infrastructure frameworks
  - Support the development of system infrastructures such as communications, user interfaces and compilers
- Middleware integration frameworks
  - Standards and classes that support component communication and information exchange
- Enterprise application frameworks
  - Support the development of specific types of application such as telecommunications or financial systems

Software product lines

- A software product line is a set of applications with a common architecture and shared components, with each application specialised to reflect different requirements

- Product line purpose: business and/or engineering

- Adaptation may involve
  - Component and system configuration
  - Adding new components to the system
  - Selecting from a library of existing components
  - Modifying components to meet new requirements

Software product line - composition

- Core components
- Configurable application components
- Specialised application components

Product line architectures

- Architectures must be structured so that different sub-systems are separate and can be modified
- The architecture should separate entities and their descriptions
  - Higher levels in the system access entities through descriptions rather than directly

Product line specialisation

- Platform specialisation
  - Different versions of the application are developed for different platforms
- Environment specialisation
  - Different versions of the application are created to handle different operating environments e.g. different types of communication equipment
- Functional specialisation
  - Different versions of the application are created for customers with different requirements
- Process specialisation
  - Different versions of the application are created to support different business processes

Examples of product lines

- Can you think of any that you might have used?

Application system reuse

- An application system is a software system that can be adapted for different customers without changing the source code of the system
- Application systems have generic features and so can be used/reused in different environments
- Application systems are adapted by using built-in configuration mechanisms that allow the functionality of the system to be tailored to specific customer needs

Benefits of application system reuse

- As with other types of reuse, more rapid deployment of a reliable system may be possible
- It is possible to see what functionality is provided by the applications and so it is easier to judge whether or not they are likely to be suitable
- Some development risks are avoided by using existing software
- Businesses can focus on their core activity without having to devote a lot of resources to IT systems development
- As operating platforms evolve, technology updates may be simplified as these are the responsibility of the COTS product vendor rather than the customer

Challenges of application system reuse

- Requirements usually have to be adapted to reflect the functionality and mode of operation of the COTS product
- The COTS product may be based on assumptions that are practically impossible to change
- Choosing the right COTS system for an enterprise can be a difficult process, especially as many COTS products are not well documented
- There may be a lack of local expertise to support systems development
- The COTS product vendor controls system support and evolution

Key points

- There are many ways to reuse software - from the reuse of classes and methods in libraries to the reuse of complete application systems
- There are advantages and challenges of software reuse
- There are many possibilities for software reuse including application frameworks, software product lines and application systems