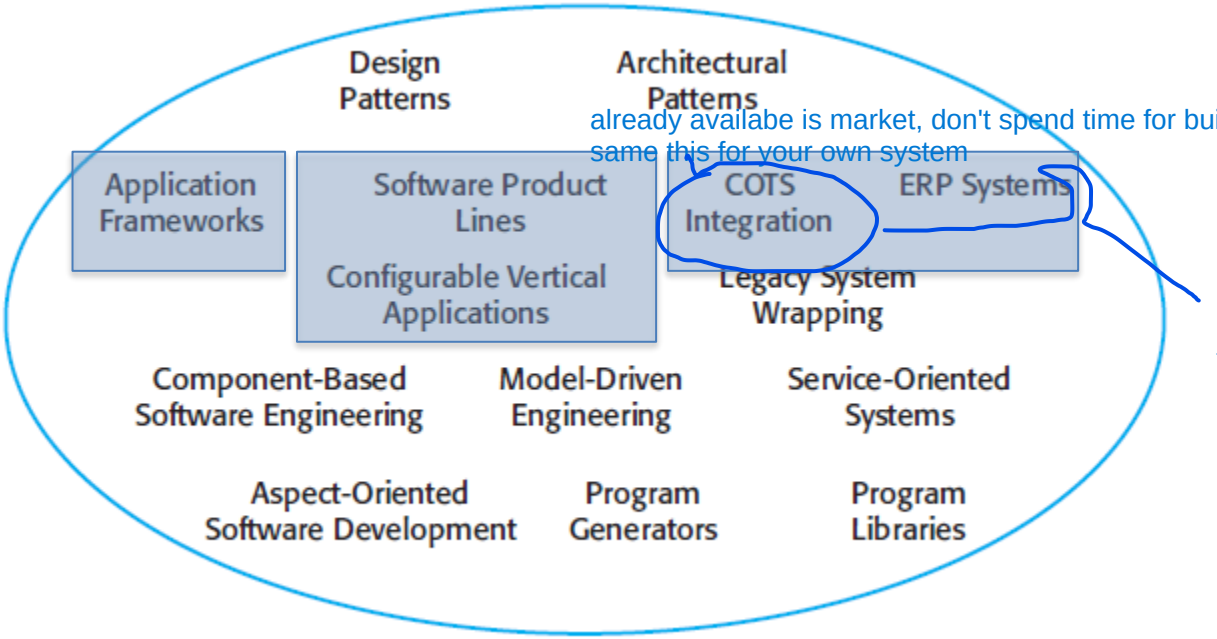
Chapter 10 – Software Reuse

Lecture 2

The reuse landscape (revisited)



Product Line: Creating a collection of similar **software** systems from a shared set of **software** assets using a common means of production.

Software product lines

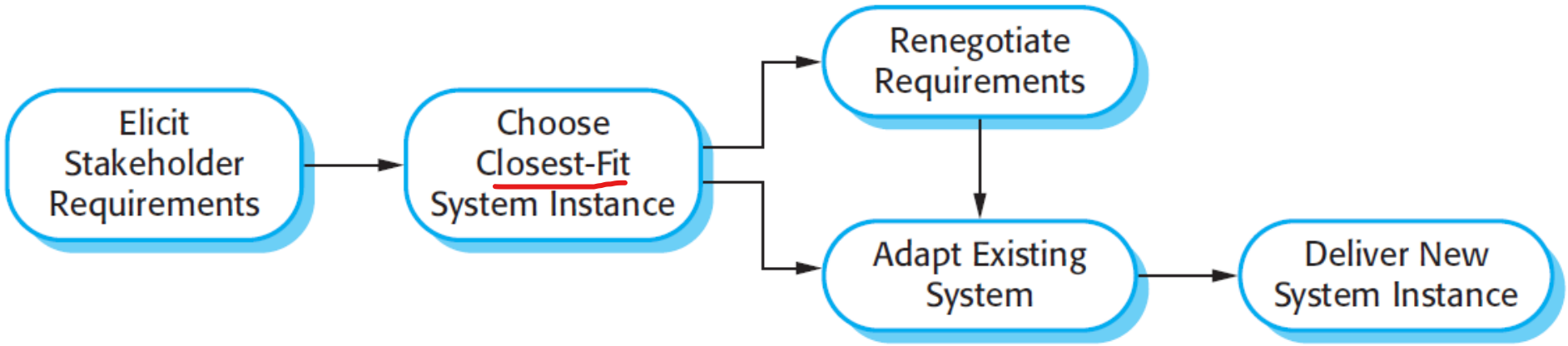
series of silimar type of softare
skeleton is same, but some variation is there based on peoduct

- Software product lines or application families are applications with generic functionality that can be *adapted and configured* for use in a specific context.

used for embaded system(both hardware and soft)

- A software product line is a set of applications with a common architecture and shared components, with each application specialized to reflect different requirements.
- For example, there may be a software product line for a family of printers.

Product instance development



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 frameworks and product lines

- Application frameworks rely on object-oriented features such as polymorphism to implement extensions. Product lines need not be object-oriented (e.g. embedded software for a mobile phone)
- Application frameworks focus on providing technical rather than domain-specific support. Product lines embed domain and platform information.
- Product lines often control applications for equipment.
- Software product lines are made up of a family of applications, usually owned by the same organization.

3. COTS product reuse

- A commercial-off-the-shelf (COTS) product is a software system that can be adapted for different customers without changing the source code of the system.
- COTS systems have generic features and so can be used/reused in different environments.
- COTS products are adapted by using built-in configuration mechanisms that allow the functionality of the system to be tailored to specific customer needs.
  - For example, in a hospital patient record system, separate input forms and output reports might be defined for different types of patient.

Benefits of COTS 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. However, this approach has its own risks, as I discuss below.
- 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.

Problems of COTS 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. The customer must therefore adapt their business to reflect these assumptions.
- 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.

COTS solution systems

- COTS-solution systems consist of a generic application from a single vendor that is configured to customer requirements. sometimes single product can't fulfill need, you need to take two products and use
  - For example, a COTS-solution system may be produced for dentists that handles appointments, dental records, patient recall, etc.
- Domain-specific COTS-solution systems, such as systems to support a business function (e.g. document management) provide functionality that is likely to be required by a range of potential users.

COTS integrated systems

- COTS-integrated systems involve integrating two or more COTS systems (perhaps from different vendors) to create an application system.
- You may use this approach when there is no single COTS system that meets all of your needs or when you wish to integrate a new COTS product with systems that you already use.

COTS-solution and COTS-integrated systems

COTS-solution systems	COTS-integrated systems
Single product that provides the functionality required by a customer	Several heterogeneous system products are integrated to provide customized functionality
Based around a generic solution and standardized processes	Flexible solutions may be developed for customer processes
Development focus is on system configuration	Development focus is on system integration
System vendor is responsible for maintenance	System owner is responsible for maintenance
System vendor provides the platform for the system	System owner provides the platform for the system

company or individual that sells goods or services to customers

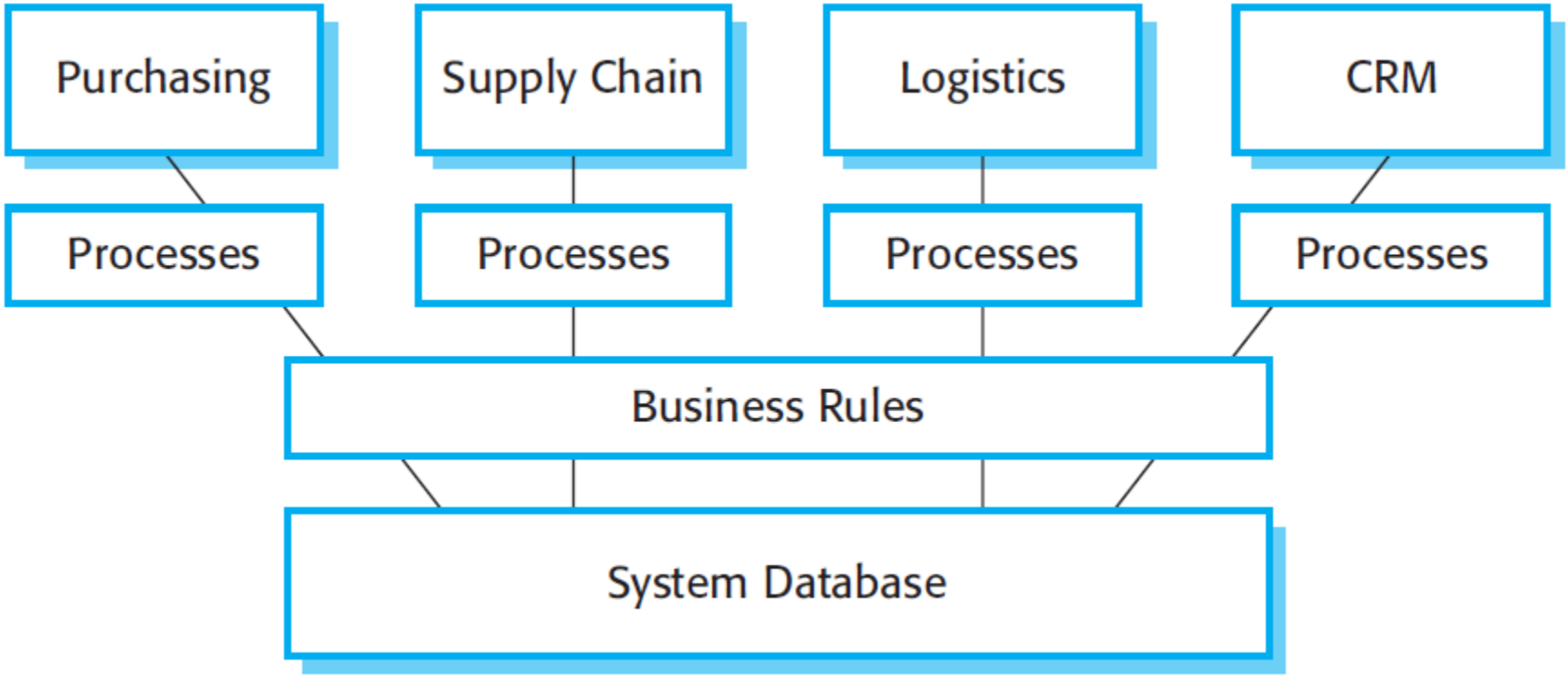
COTS system integration problems

- Lack of control over functionality and performance
  - COTS systems may be less effective than they appear
- Problems with COTS system inter-operability
  - Different COTS systems may make different assumptions that means integration is difficult
- No control over system evolution
  - COTS vendors not system users control evolution
- Support from COTS vendors
  - COTS vendors may not offer support over the lifetime of the product

ERP systems

- Enterprise Resource Planning (ERP) systems are examples of large-scale COTS reuse.
- ERP system is a generic system that supports common business processes such as ordering and invoicing, manufacturing, etc.
- These are very widely used in large companies - they represent probably the most common form of software reuse.
- The generic core is adapted by including modules and by incorporating knowledge of business processes and rules.

The architecture of an ERP system



CRM (Customer Relationship management): is any tool, strategy, or process that helps businesses better organize and access customer data.

ERP architecture

- A number of modules to support different business functions.
- A defined set of business processes, associated with each module, which relate to activities in that module.
- A common database that maintains information about all related business functions.
- A set of business rules that apply to all data in the database.

Class Work Exercises (revisited)

Q. Most desktop software, such as word processing software, can be configured in a number of different ways. Examine the software that you regularly use and list the configuration options for that software. Suggest difficulties that users might have in configuring the software. If you use Microsoft Office or Open Office, these are good examples to use for this exercise.

Key points

- Software product lines are related applications that are developed from a common base. This generic system is adapted to meet specific requirements for functionality, target platform or operational configuration.
- COTS product reuse is concerned with the reuse of large-scale, off-the-shelf systems. These provide a lot of functionality and their reuse can radically reduce costs and development time. Systems may be developed by configuring a single, generic COTS product or by integrating two or more COTS products.
- Enterprise Resource Planning systems are examples of large-scale COTS reuse. You create an instance of an ERP system by configuring a generic system with information about the customer's business processes and rules.
- Potential problems with COTS-based reuse include lack of control over functionality and performance, lack of control over system evolution, the need for support from external vendors and difficulties in ensuring that systems can inter-operate.

Reference: Software Engineering by Ian Sommerville, Pearson Education, Inc., publishing as Addison-Wesley