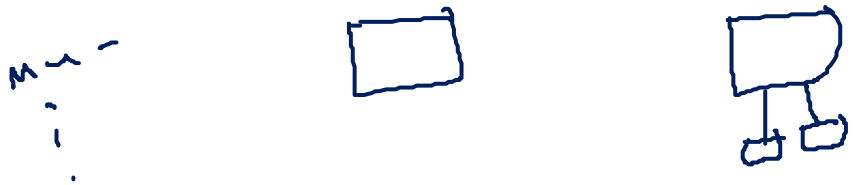
Reuse existing IoT solutions & Reusability Plan

Reuse: Solutions

- In most engineering disciplines, systems are designed by composing existing components that have been used in other systems.
- Reusing of existing components establish development of new technologies in less time and effort.
- We need to adopt a design process that is based on *systematic reuse (a plan)*

Reuse types



Application system reuse

- The whole of an application system may be reused either by incorporating it without change into other systems (Commercial Off-The-Shelf - COTS reuse) or by developing application families.

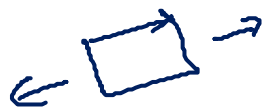
Component reuse

- Components of an application, like, sub-systems or single objects, may be reused.

Object and function reuse

- Software components that implement a single well defined object or function may be reused.

Reuse types and aspects of reuse



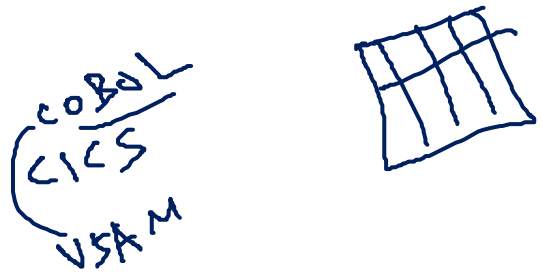
- Application system reuse: Developers of these systems make product available on several platforms.
- Function reuse: domain specific libraries of reusable functions are used.
- Software development with reuse: to design the generic software components for reuse.
- Generator based reuse: domain specific through application generation.
- Software component reuse is **the software engineering practice of creating new software applications from existing components**, rather than designing and building them from scratch.

Problems with COTS

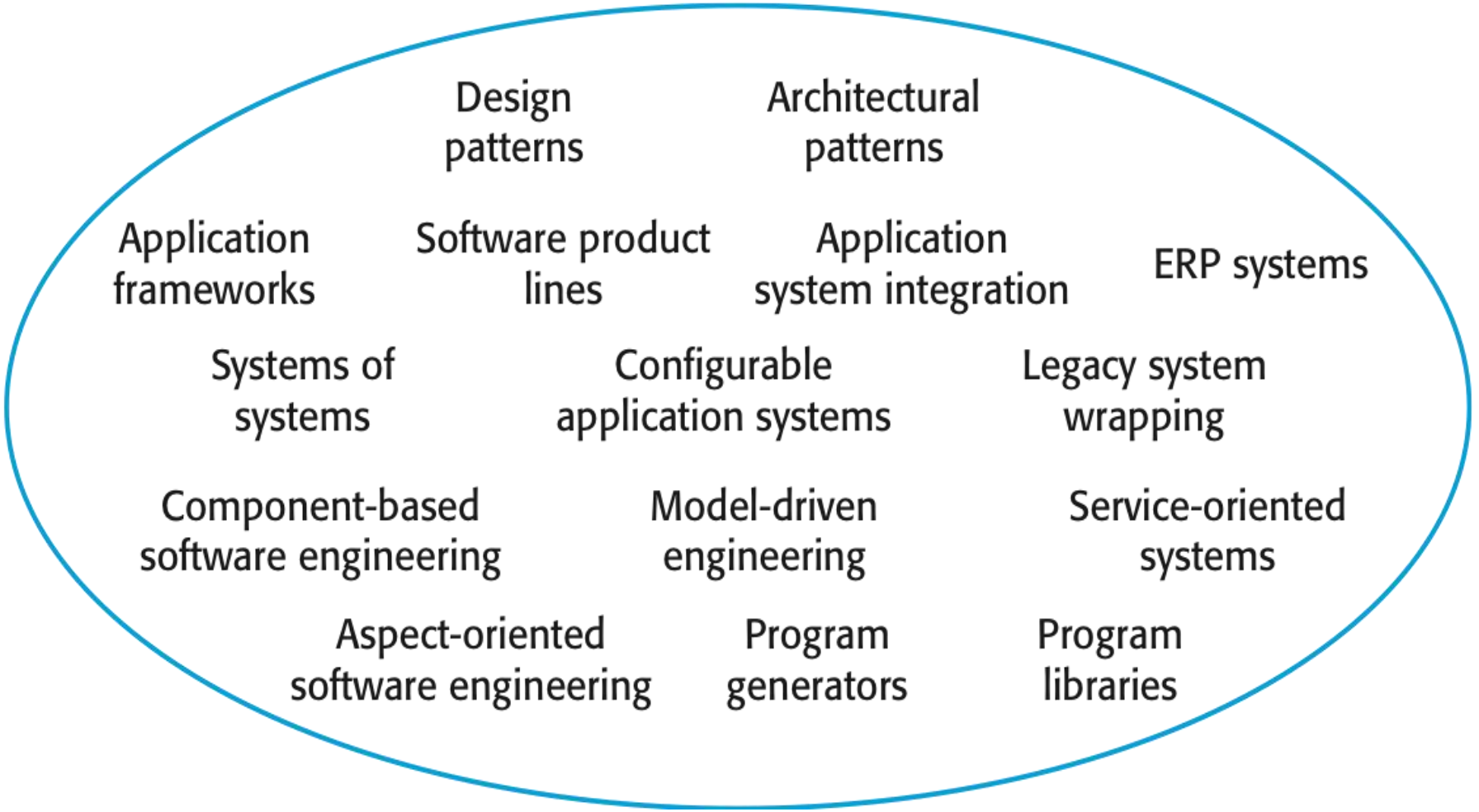
- Lack of developer control over functionality and performance.
- COTS vendors may not offer users any control over the evolutions of its components.
- Vendors may not offer support over the lifetime of a product built with COTS components.

Requirements for reuse

- Able to find the appropriate reusable components from the components database.
- Components users should have the knowledge of components and how to reuse, also the potential cost of reuse.



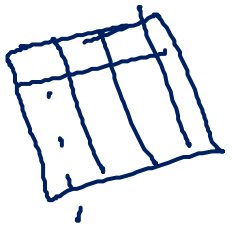
The reuse landscape



Reuse – Approaches

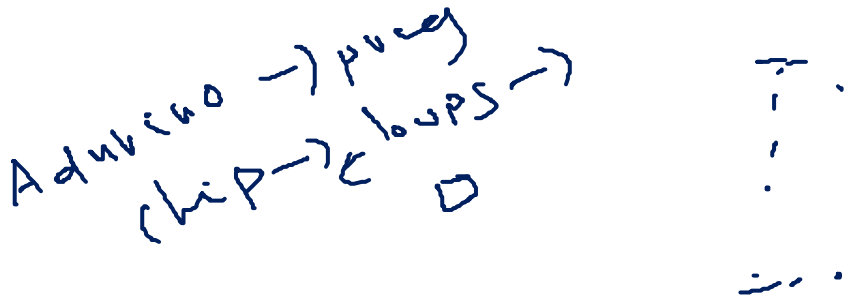
Design patterns	Generic abstractions that occur across applications are represented as design patterns that show abstract and concrete objects and interactions.
Component-based development	Systems are developed by integrating components (collections of objects) that conform to component-model standards.
Application frameworks	Collections of abstract and concrete classes that can be adapted and extended to create application systems.
Legacy system wrapping	Legacy systems that can be 'wrapped' by defining a set of interfaces and providing access to these legacy systems through these interfaces.
Service-oriented systems	Systems are developed by linking shared services that may be externally provided.

Reuse – Approaches



Application product lines	An application type is generalised around a common architecture so that it can be adapted in different ways for different customers.
COTS integration	Systems are developed by integrating existing application systems.
Configurable vertical applications	A generic system is designed so that it can be configured to the needs of specific system customers.
Program libraries	Class and function libraries implementing commonly-used abstractions are available for reuse.
Program generators	A generator system embeds knowledge of a particular types of application and can generate systems or system fragments in that domain.
Aspect-oriented software development	Shared components are woven into an application at different places when the program is compiled.

Reusability-Benefits

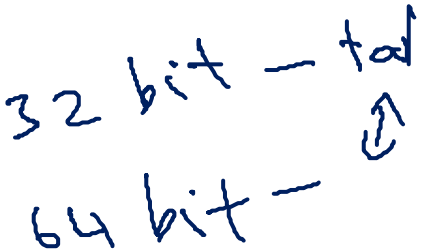


- IoT systems are developed from existing IoT components rather than from scratch. The reusability of IoT components decrease bugs and defects in the systems, to shorten the development time and cost, and to increase productivity.
- Developers find it easier to adapt an existing system.
- Sensor or machine to enable remote access and data gathering.
- There is scope to reuse design patterns, architectures, and standards for maximum efficiency.
- Examples of common functions in all IoT systems include communications management, device management, and security.

Reuse: Benefits

- ✓Increased dependability
- ✓Reduced process risk
- ✓Effective use of specialists
- ✓Standards compliance
- ✓Accelerated development

Reuse problems



- Increased maintenance cost
- Lack of tool support
- Create and maintain a component library
- Identify and adapt reusable components