Software Reuse

Software Reuse

- In most engineering disciplines, systems are designed by composing existing components that have been used in other systems.
- Software engineering has been more focused on original development but it is now recognised that to achieve better software, more quickly and at lower cost, we need a design process that is based on systematic software reuse.

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 software reuse

- **Reduced Development Time :** By reusing existing software components, fewer programs are written, and thus less time is spent in programming.

- **Improved Quality:** Because software components are often repeatedly reused, the defect fixes from each reuse accumulate, resulting in higher quality of the developed software systems.

- **Easy Maintenance :** Reusable components contribute to easy maintenance not only because they have fewer defects, but also because they facilitate communication among software developers by providing a set of common vocabulary, especially for the indirect communication between system builders and system maintainers.

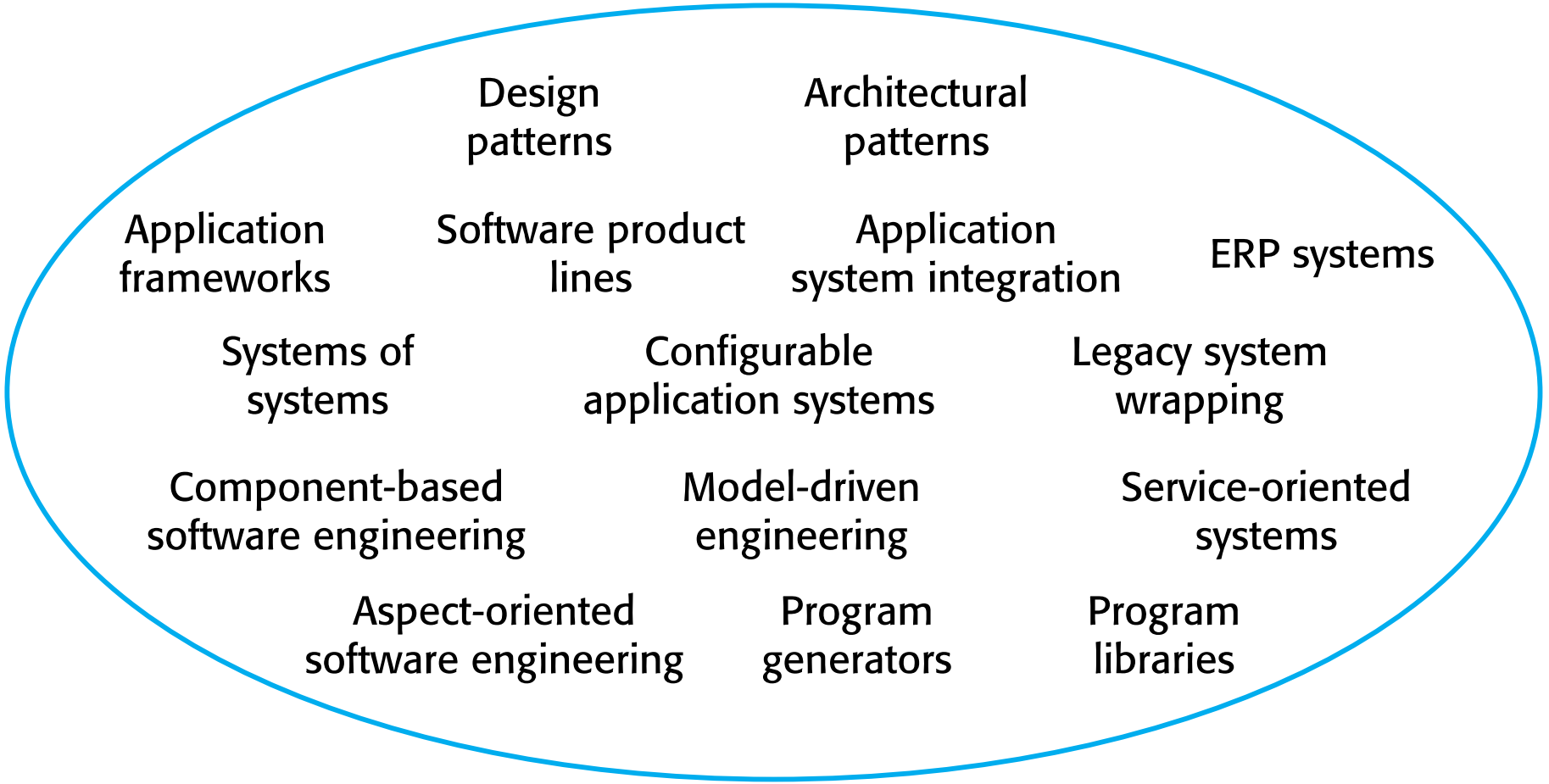
- **Improved Evolvability:** To cope with constantly changing requirements and implementation platforms, software systems must be able to evolve. Reusing software components improves the evolvability of software systems because it can limit the needed change to components instead of identifying and changing all occurrences distributed all over the system

- **Increased Problem Framing Ability:** The representation of a problem is animportant determinant of the range of solutions that will be considered, as well as an important source of problem-solving difficulty.

The reuse landscape

- Although reuse is often simply thought of as the reuse of system components, there are many different approaches to reuse that may be used.
- Reuse is possible at a range of levels from simple functions to complete application systems.
- The reuse landscape covers the range of possible reuse techniques.

The reuse landscape



Application Framework

- Framework Definition:

"..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."

Application frameworks

- Frameworks are moderately large entities that can be reused. They are 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.

Benefits of Framework

1.Reusability: Programmers reuse framework to solve a problem of a particular class without developing independently.

2. Standardization: Program development costs can be reduced through adherence to organizational standards, including the use of commercial components.

3. Interoperability Applications using a framework are more likely to be compatible with each other.

4. Scalability: With the use of architectural infrastructure, systems will operate with minimum hardware resources improving scalability.

5. Portability: It promotes vendor independence for both hardware and software with increase portability through the use of open system concept.