

SOFTWARE CONSTRUCTION

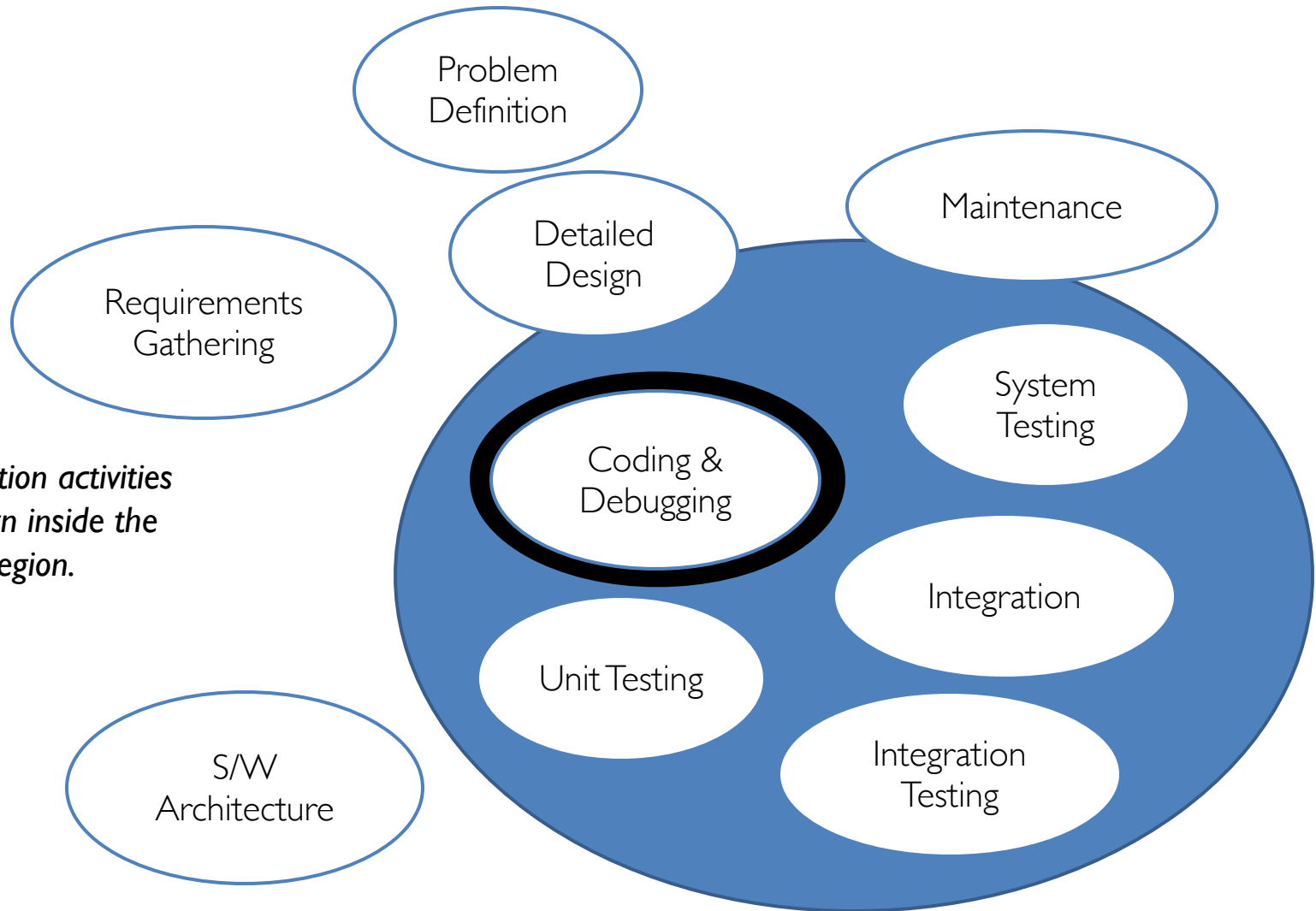
Chapter 7

Software Construction

■ What is Software Construction?

- -> Create an executable version of the software
- Software Construction can be defined as detailed creation of **working, meaningful software** through a combination of **coding, verification, unit testing, integration testing, and debugging.**
- Involves:
 - - developing programs in high- or low-level programming languages
 - - Tailoring and adapting generic, Off-the-Shelf systems to meet the specific requirements of an organization

Software Construction Activities



Construction activities are shown inside the shaded region.

Software Construction Fundamentals

- The fundamentals of software construction include:
 - Minimizing complexity
 - Anticipating change
 - Constructing for verification
 - Standards in construction

Minimizing Complexity

- As functionality increases, software complexity increases.
- We always aim to reduce complexity throughout the lifecycle.
- Minimizing complexity is one the of **strongest drivers** in software construction.
- In software construction, reduced complexity is achieved through emphasizing code creation that is simple and readable.

Anticipating Change

- Most software will change over time, and the anticipation of change drives many aspects of software construction;
- Changes in the environments in which software operates also affect software in diverse ways.
- Anticipating change means software engineers build extensible software, so they can enhance a software product without disrupting the underlying structure.
- Anticipating change is supported by many specific coding techniques.

Anticipating Change

- Anticipation of change affect how software is constructed in different ways:
 1. Use of control structures
 2. Handling of errors
 3. Source code organization
 4. Code documentation
 5. Coding standards

Standards in Construction

- Restricting the use of complex or hard-to-understand language structures, among others.
- Applying external or internal development standards during construction helps achieve a project's objectives for efficiency, quality, and cost.

Reuse

Most modern software is constructed by reusing existing components or systems. When you are developing software, you should make as much use as possible of existing code.

▷ Develop new systems more quickly, with fewer development risks and lower costs.

▷ Reuse improves Software reliability.

Reuse

Systematic reuse can enable significant software productivity, quality, and cost improvements.

Reuse has two closely related facets:

Construction for reuse: Create reusable software assets.

Construction with reuse: Reuse software assets in the construction of a new solution.

Open source software

- Open source software is released under a [license](#) in which the [copyright](#) holder grants users the right to use, change, and [distribute the software](#) and its [source code](#) to anyone and for any purpose.
- Its roots are in the Free Software Foundation (www.fsf.org), which advocates that source code should not be proprietary but rather should always be available for users to examine and modify as they wish.