ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

502045

Software Engineering

Chapter 06

Lesson 07: Implementation Issues

IMPLEMENTATION ISSUES

- A critical stage of software development process is system implementation where you create an executable version of the software
- Implementation may involve developing programs in high- or low-level programming languages or
- Tailoring and adapting generic, off-the-shelf systems to meet the specific requirements of an organization;

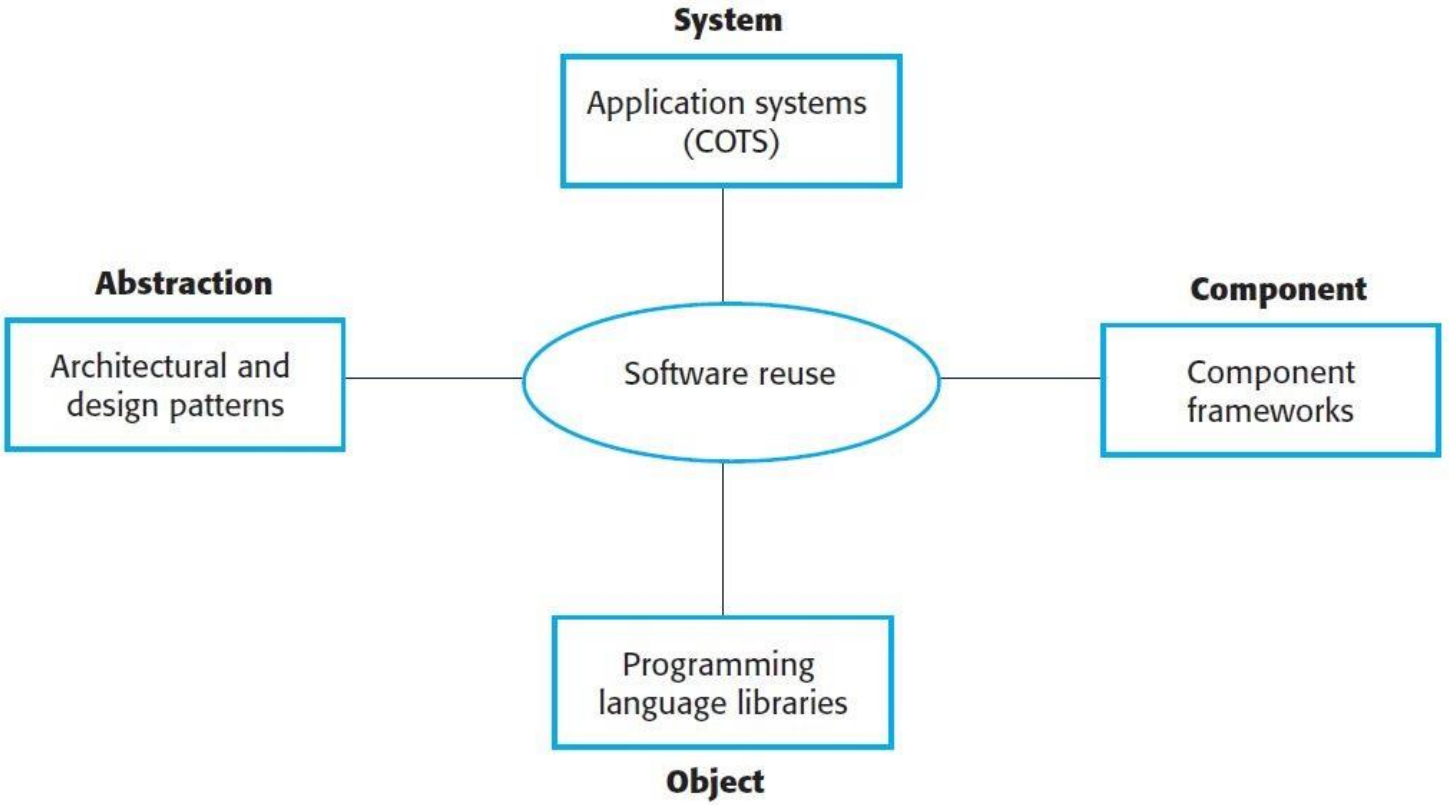
IMPLEMENTATION ISSUES

- *Reuse:* Most modern software is constructed by reusing existing components or systems.
- *Configuration management:* keep track of software components' versions
- *Host-target development:* development computer (the host system) vs execution computer (the target system)

REUSE

- *The abstraction level:* At this level, you don't reuse software directly but rather use knowledge of successful abstractions in the design of your software.
- *The object level:* At this level, you directly reuse objects from a library rather than writing the code yourself.
- *The component level:* Components are collections of objects and object classes that operate together to provide related functions and services
- *The system level:* At this level, you reuse entire application systems.

REUSE


System
Application systems (COTS)
Abstraction
Architectural and design patterns
Software reuse
Component
Component frameworks
Programming language libraries
Object

REUSE COSTS

- The costs of the time spent looking for software to reuse
- The costs of buying the reusable software
- The costs of adapting and configuring the reusable software components or systems
- The costs of integrating reusable software elements with each other

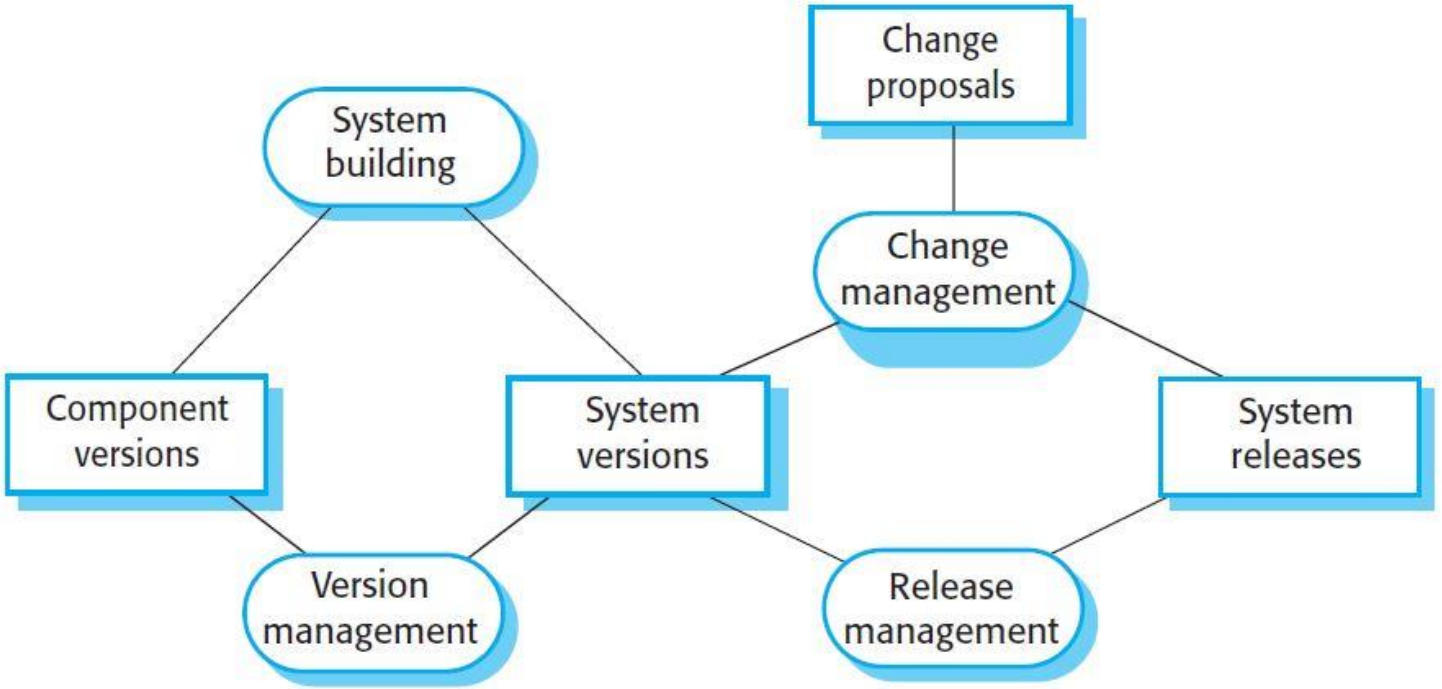
CONFIGURATION MANAGEMENT

- Configuration management is the general process of managing a changing software system
- The aim of configuration management is to support developers:
  - Access the project code and documents in a controlled way
  - Find out what changes have been made
  - Compile and link components to create a system

CONFIGURATION MANAGEMENT

- *Version management*: keep track of the different versions
- *System integration:* define what versions of components are used to create system
- *Problem tracking:* report bugs and other problems
- *Release management*: plan the functionality of new releases and organizing the software for distribution

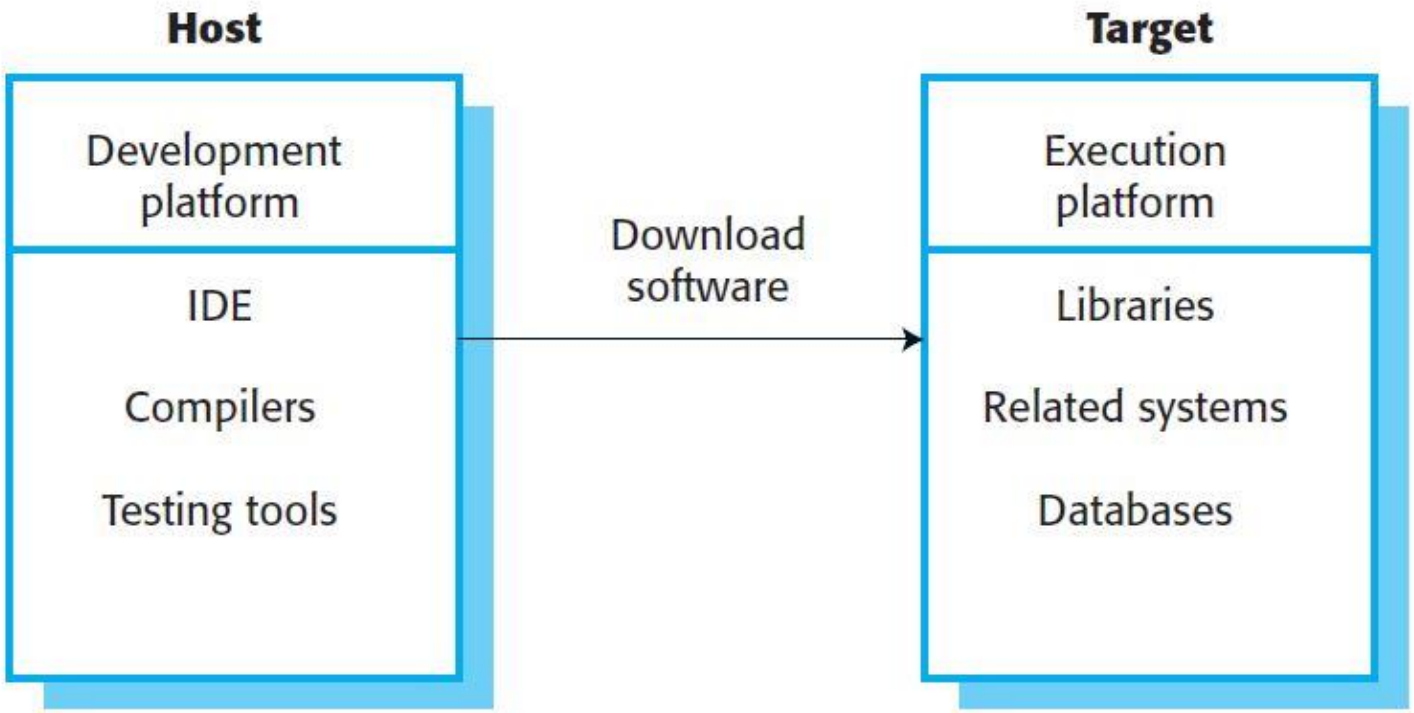
CONFIGURATION MANAGEMENT



HOST-TARGET DEVELOPMENT

- Software is developed on one computer (the host) but runs on a separate machine (the target)
- It is more than just hardware
- It includes operating systems plus other supporting software such as a database management system

HOST-TARGET DEVELOPMENT



SOFTWARE DEVELOPMENT PLATFORM

- An integrated compiler and syntax-directed editing system
- A language debugging system
- Graphical editing tools, such as tools to edit UML models
- Testing tools, such as Junit
- Tools to support refactoring and program visualization
- Configuration management tools to manage source code versions and to integrate and build systems.

TARGET PLATFORM

- The hardware and software requirements of a component
- The availability requirements of the system
- Component communications

Q&A