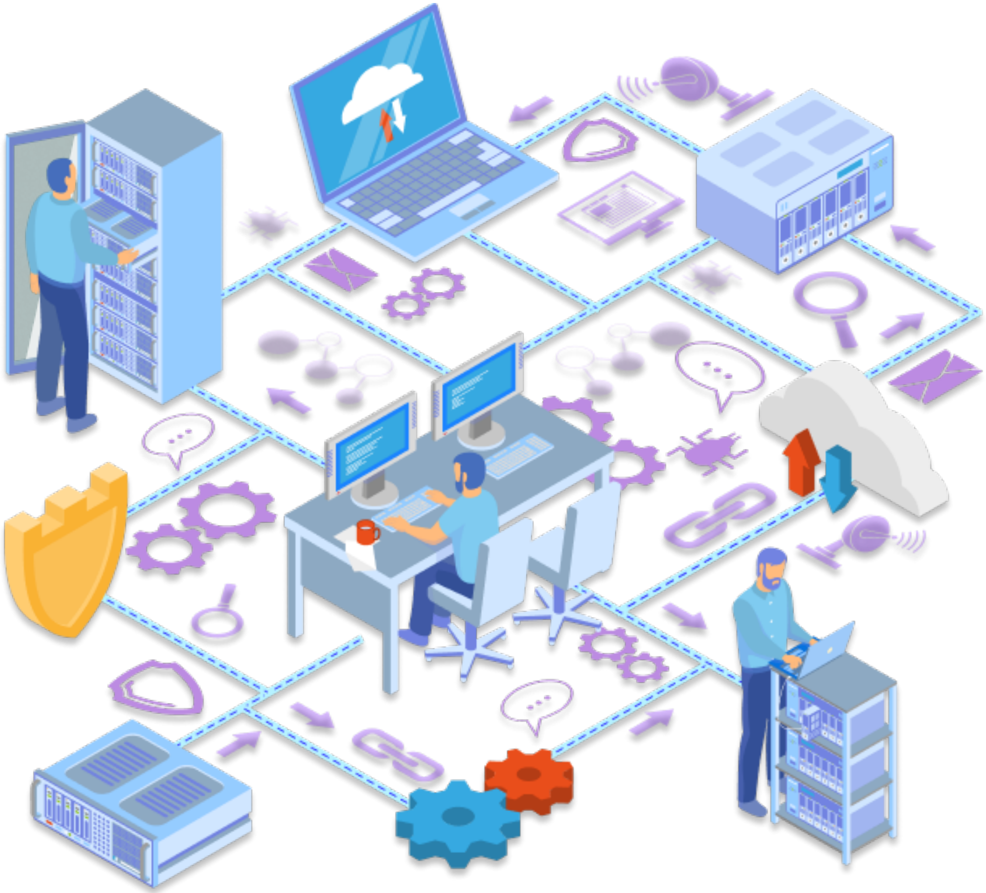
Configuration Management

Software Engineering 2

(3103313-1)

Amirkabir University of Technology

Fall 1399-1400

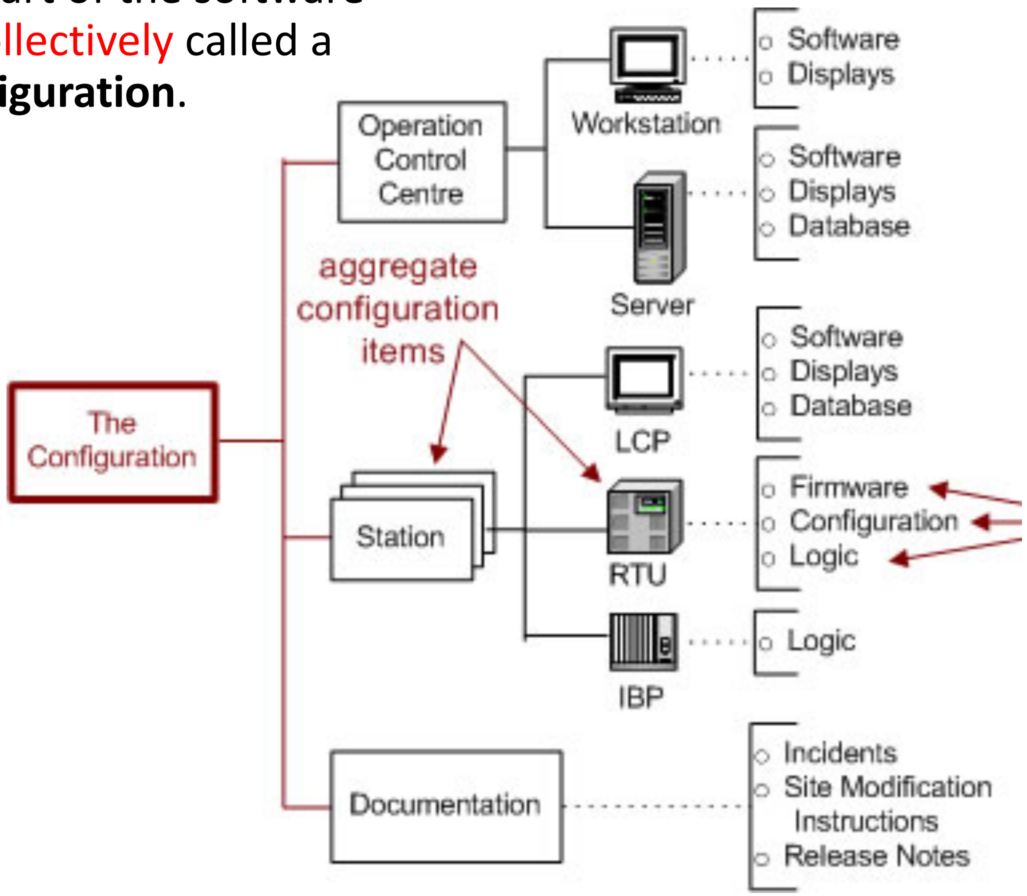


Configuration Management

Configuration management (CM) is concerned with the policies, processes, and tools for managing **changing** software systems.

(Software) Configuration

The items that comprise all information produced as part of the software process are collectively called a **software configuration**.



Software Configuration Item (SCI) - a named element of information

SCM Activities

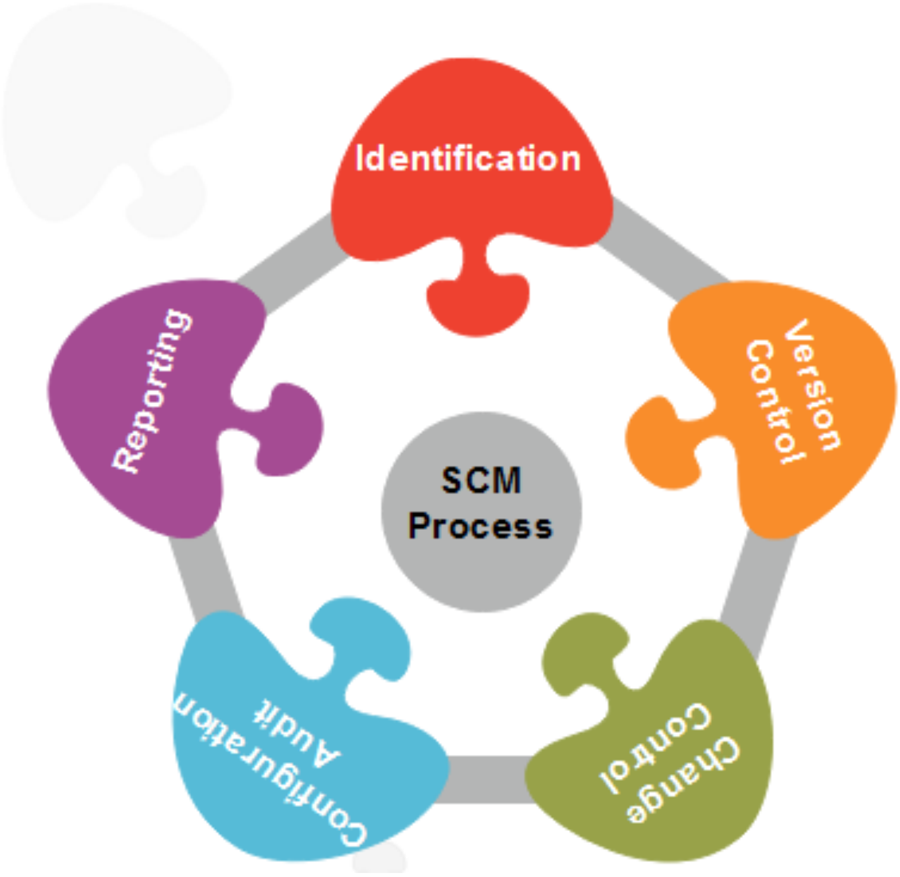
SCM Tasks

- Identification (Configuration Items)

- Version Control
- System Building
- Release Management

- Change Management

- Configuration Audit
- Reporting (Status Accounting)



SCM Activities

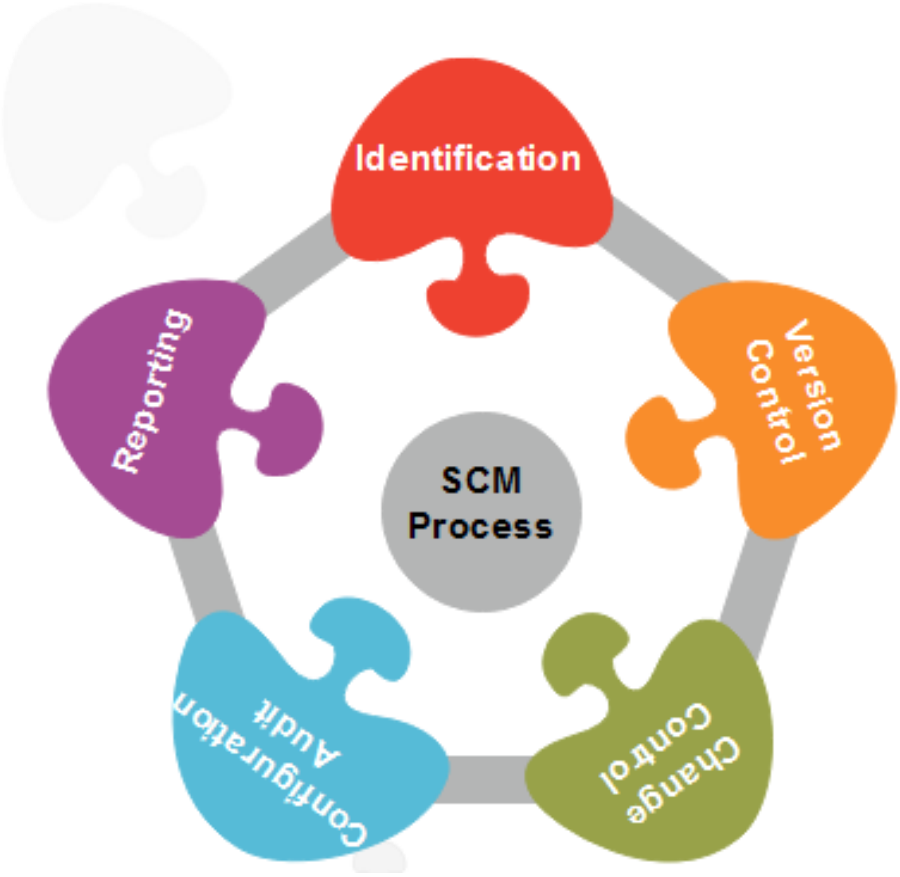
SCM Tasks

- **Identification (Configuration Items)**

- Version Control
- System Building
- Release Management

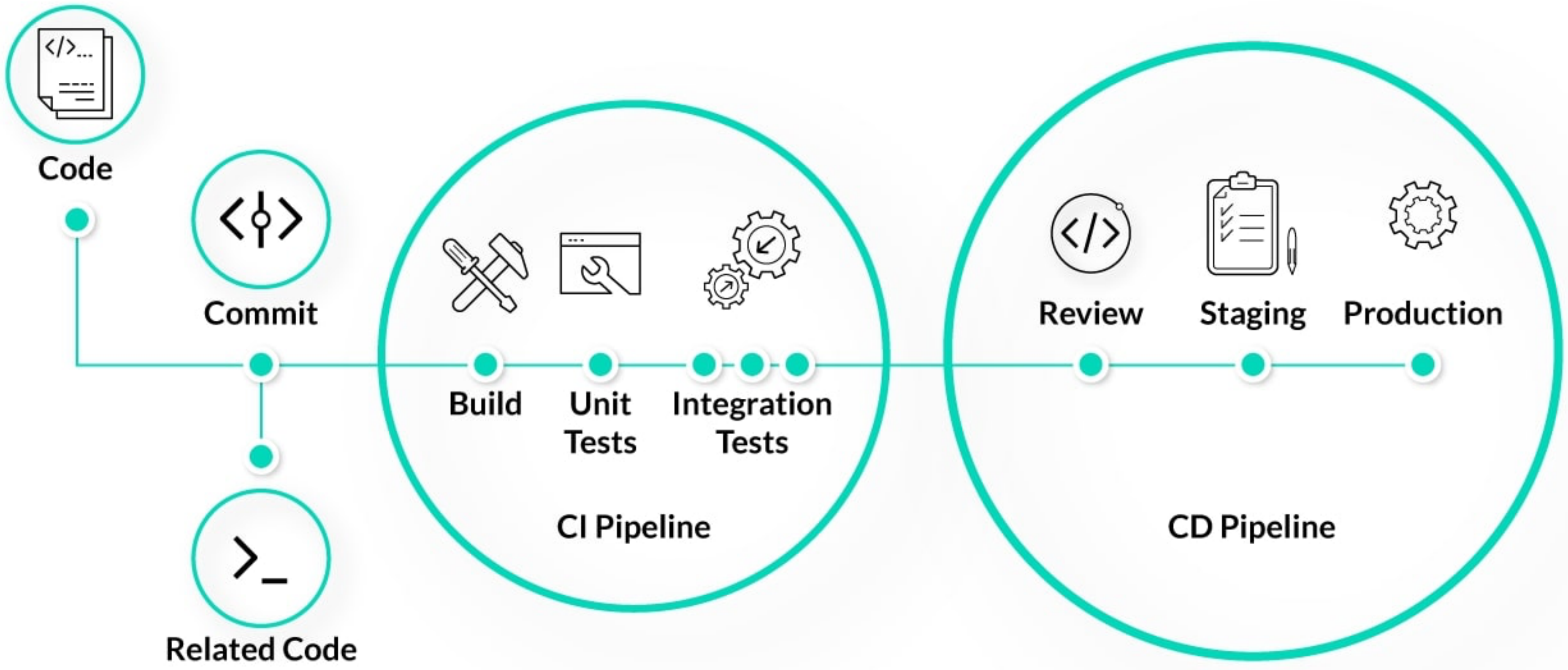
- Change Management

- Configuration Audit
- Reporting (Status Accounting)

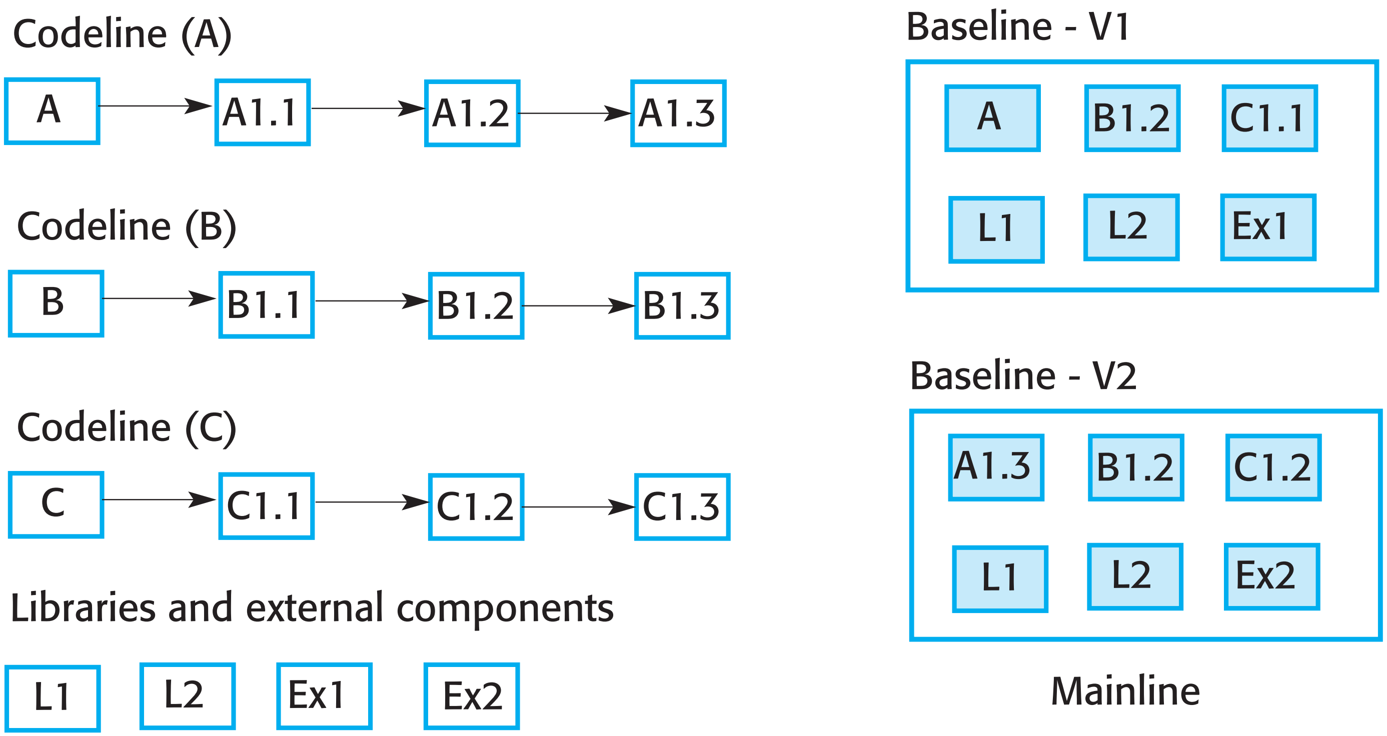


Version Control, System Building, & Release Management

- Repository
- Version Management
- Build Facility
- Issue Tracking



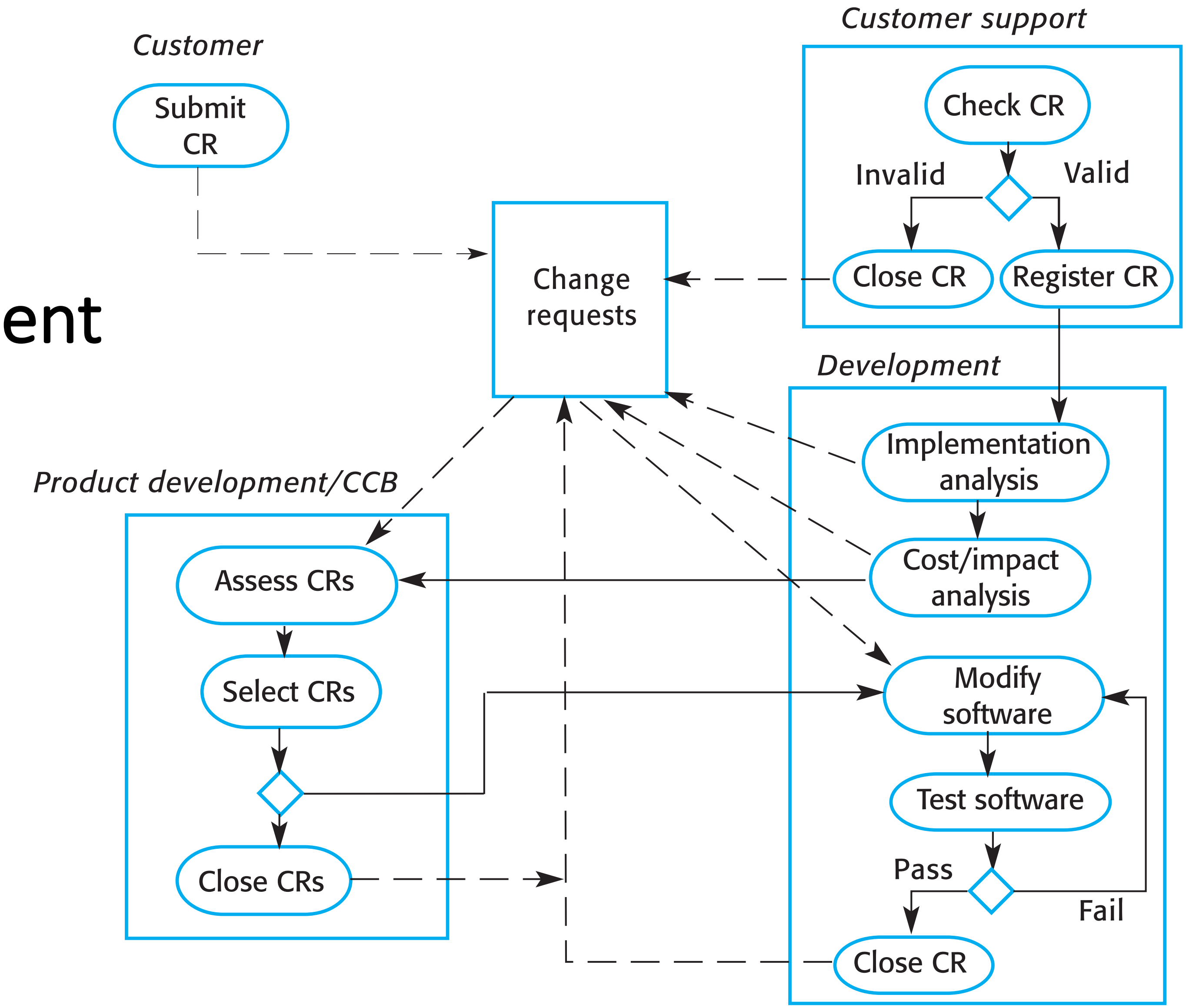
Codelines and Baselines



Version Control Systems

- File versioning, Check-in / Check-out, Branching, Merging, Multiple users working on a project, …
- Centerlised and Distributed
- CVS, SVN, Git, …

Change Management



SCM Activities

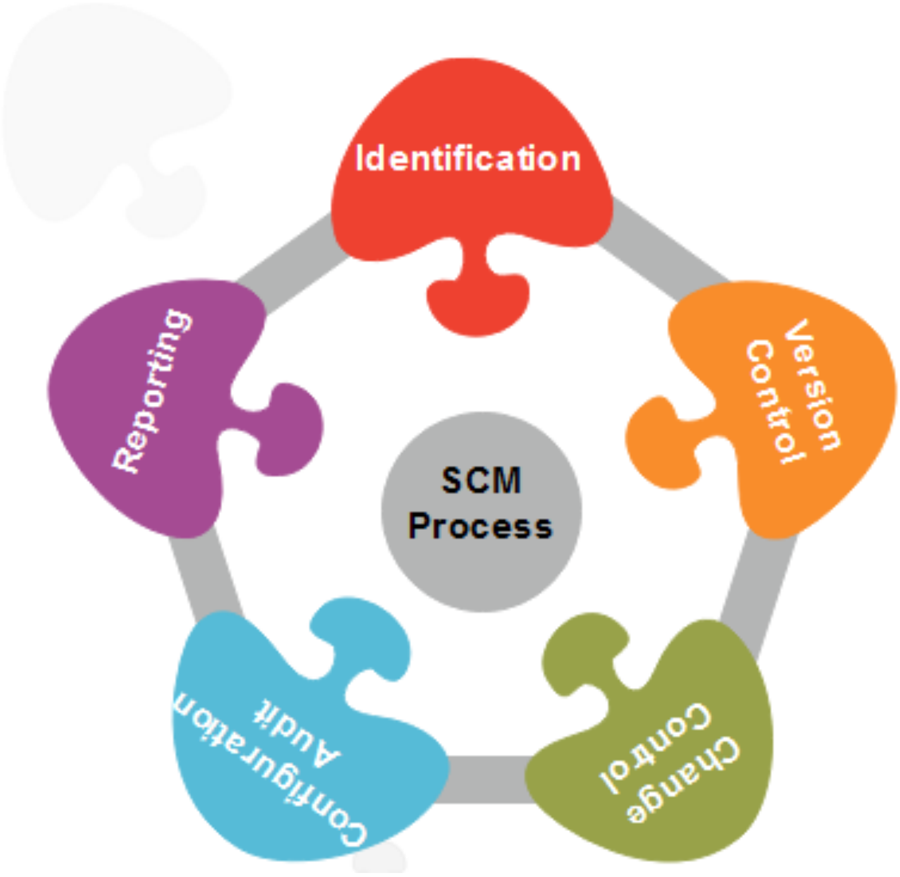
SCM Tasks

- Identification (Configuration Items)

- Version Control
- System Building
- Release Management

- Change Management

- **Configuration Audit**
- **Reporting (Status Accounting)**



Tool Support

- Large volume of information to be managed and the relationships between configuration items, requires tool support.

- Enforcement
- Cooperating Enablement
- Version Control Friendly
- Enable Change Control Processes
- Abstraction

CHEF™
ANSIBLE
puppet labs®
SALTSTACK

break

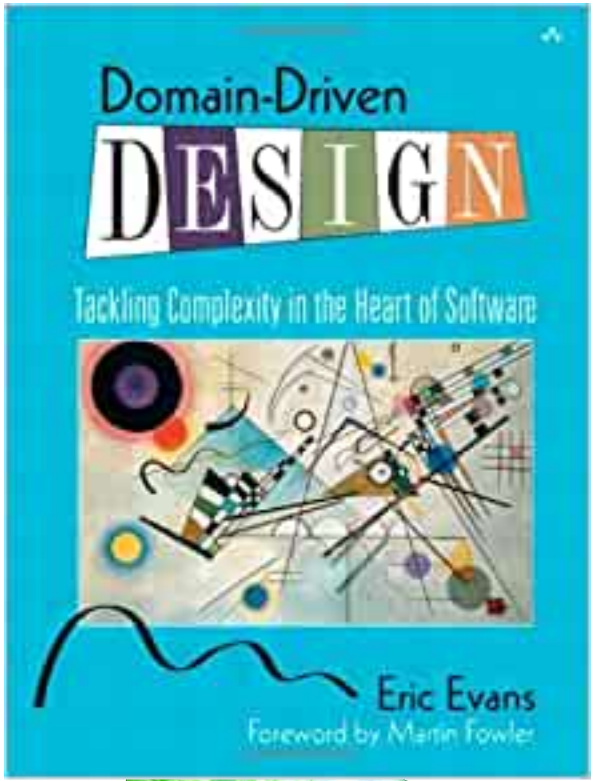
Domain-Driven Design: Tackling Complexity in the Heart of Software

- how you can make the design of your software match your mental model of the problem domain you are addressing.
- about how you think of your domain, the language you use to talk about it, and how you organize your software to reflect your improving understanding of it.

“I see this book as essential reading for software developers—it is a future classic.”

—**Ralph Johnson,**

author of *Design Patterns*


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Vaughn Vernon,
Addison-Wesley
Professional, 2016