COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

CHAPTER 13

SOFTWARE CONFIGURATION MANAGEMENT

PROF DR KAMRUDDIN NUR
PROFESSOR, CS, AIUB

BASELINE

- A *baseline* is a software configuration management concept that helps you to control change without seriously impeding justifiable change.
- The IEEE (IEEE Std. No. 610.12-1990) defines a baseline as:

  A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

- For example, the elements of a design model have been documented and reviewed. Errors are found and corrected. Once all parts of the model have been reviewed, corrected, and then approved, the design model becomes a baseline.

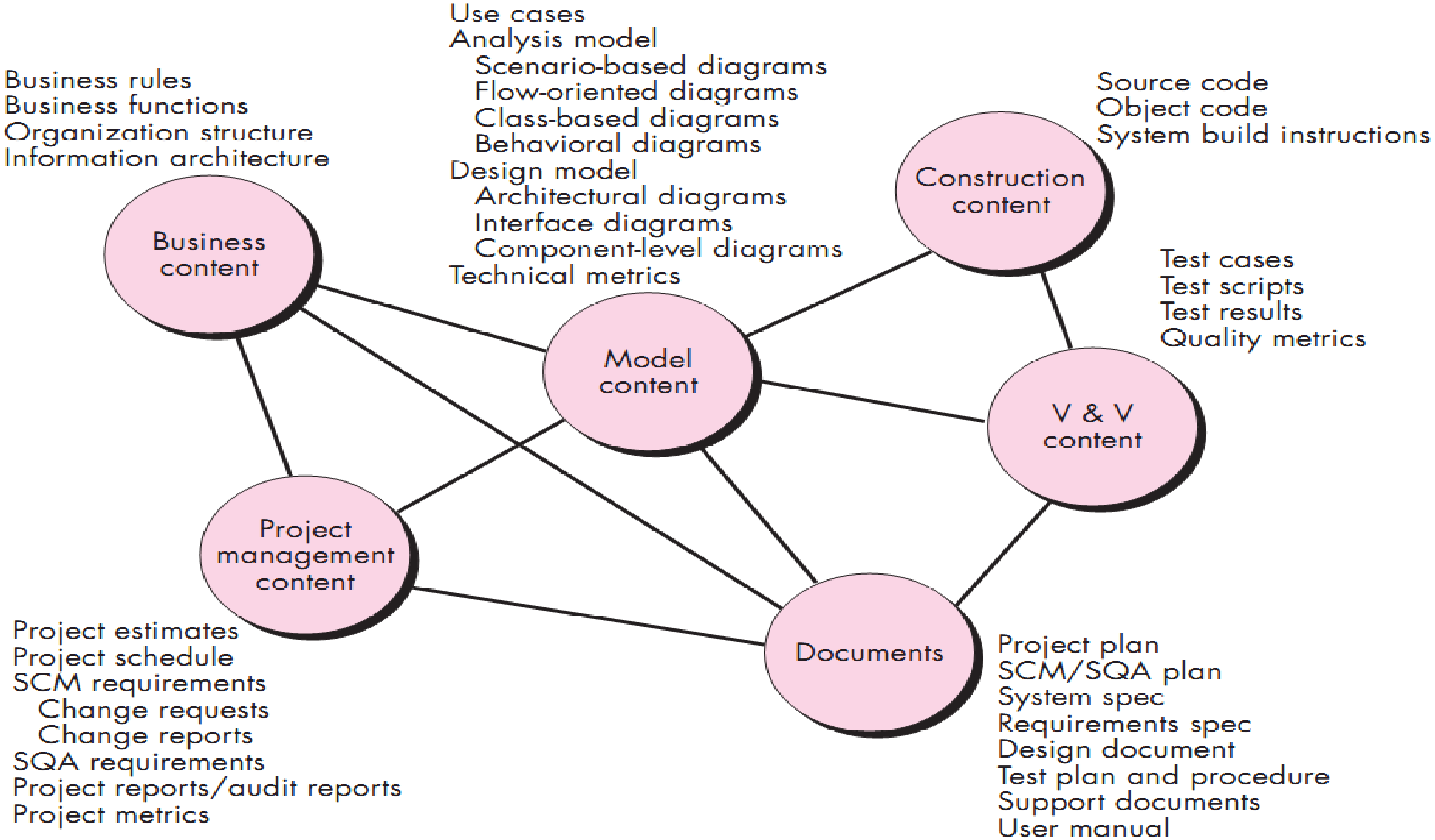
SCM REPOSITORY

- The items that comprise all information produced as part of the software development process are collectively called a *software configuration*.
- A database that acts as the center for both accumulation and storage of software engineering information
- The SCM repository is the set of mechanisms and data structures that allow a software team to manage change in an effective manner

SOFTWARE CONFIGURATION MANAGEMENT

- In software engineering, software configuration management (SCM) is the task of tracking and controlling changes in the software.
- SCM practices include revision control and the establishment of baselines.
- In configuration management, a **baseline** is an agreed description of the attributes of a product, at a point in time, which serves as a basis for defining change.

CONTENT OF SCM REPOSITORY



Business rules
Business functions
Organization structure
Information architecture

Use cases
Analysis model
  Scenario-based diagrams
  Flow-oriented diagrams
  Class-based diagrams
  Behavioral diagrams
Design model
  Architectural diagrams
  Interface diagrams
  Component-level diagrams
Technical metrics

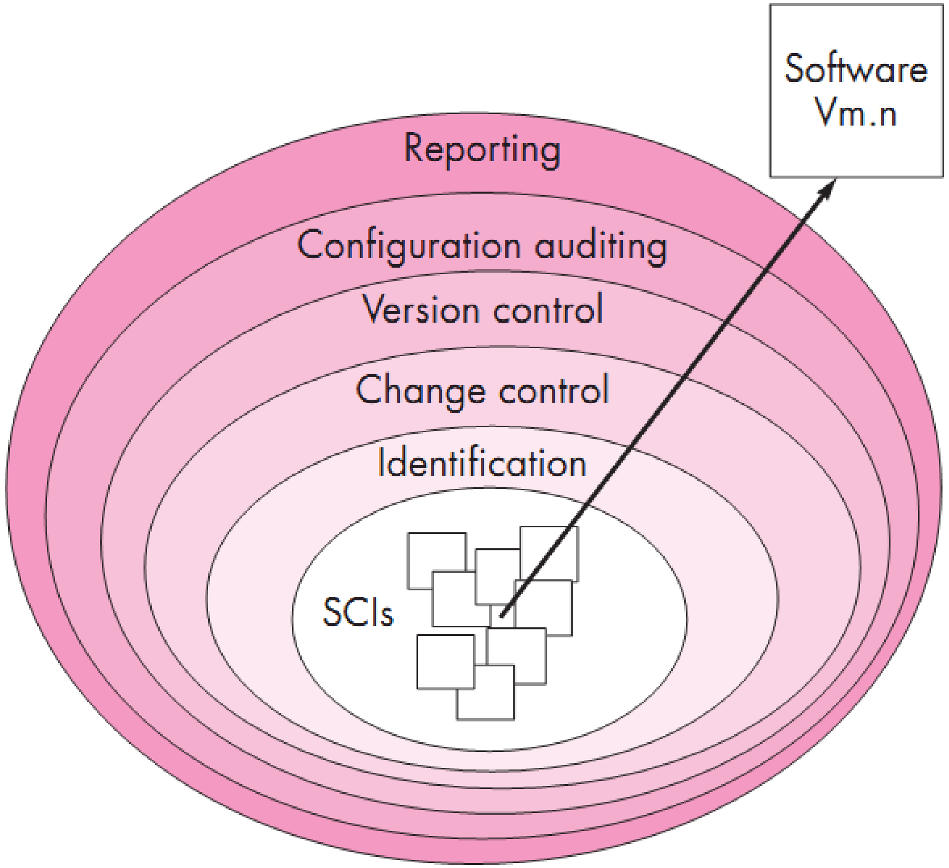
Source code
Object code
System build instructions

Test cases
Test scripts
Test results
Quality metrics

Project estimates
Project schedule
SCM requirements
  Change requests
  Change reports
SQA requirements
Project reports/audit reports
Project metrics

Project plan
SCM/SQA plan
System spec
Requirements spec
Design document
Test plan and procedure
Support documents
User manual

SCM LAYERS



- S/w configuration items (SCIs) flow outward through these layers throughout their lifetime
- When a new SCI is created, it must be identified, however, if no changes are requested for the SCI, the changer control layer does not apply.
- Each SCI created is assigned to a specific version of s/w
- A version control captures all changes to all files in the configuration along with the reason for changes and details of who made the changes and when
- The record of all these SCI's (i.e. name, creation date, version, etc.) is maintained for configuration auditing purpose.

CHANGE CONTROL PROCESS

1. Need for change is recognized
2. Change request from user
3. Developers Evaluates
4. Change report is generated
5. Change control authority decides
6. If (Request is queued for action)….else (change request is denied) → user is informed
7. Assign individuals to configuration objects
8. “Check out” configuration objects (items)
9. Make the change (in design)

CHANGE CONTROL PROCESS

12. Establish a baseline for testing
13. Perform quality assurance and testing activities (regression testing)
14. Promote changes for inclusion in next release (revision)
15. Rebuild appropriate version of software
16. Review the change to all configuration
17. Include changes in new version
18. Distribute the new version

REFERENCES

- R.S. Pressman & Associates, Inc. (2010). *Software Engineering:A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992).An analysis of defect densities found during software inspections.*Journal of Systems and Software*, *17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K.,Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation.*IBM Systems Journal*, *33*(1), 182-214.