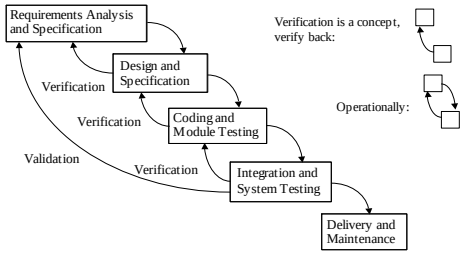
DESIGN & VERIFICATION

Development Cycle within the SLC

- Design Phase
  - transform functions to be performed by system into logical, mathematical & physical processes
  - diagrams, schematics
  - flowcharts
  - screens, menus
  - algorithms

Traditional Life Cycle Using V&V



Objectives of the Design Process

- Develop software design based on and traceable back to the software requirements, and
- Provide complete, consistent, correct, testable and understandable information from which code may be generated

The Design Process produces
- SDD
- STP
- STD
- DBDD (data base design document)
- modified SRS (?)

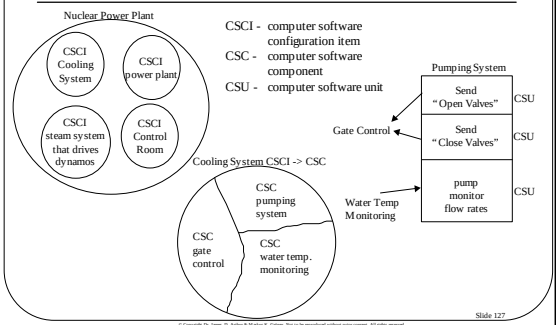
Principal Design Activities

- Allocate Software Requirements (including interface requirements) to design components
- Decompose components to their lowest level of detail necessary for coding the component
- Describe external and internal interfaces for each component
- Define the measurements that will be used to assess whether the design meets its requirements and quality attributes

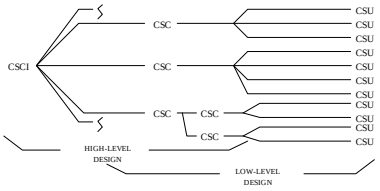
Principal Design Activities
(continued)

- Report on any outstanding problems with the software design (including interfaces) back to software requirements process
- Modify, if necessary, the SRS
- Generate an SDD and DBDD

Level of Design Abstraction
(Nuclear Power Plant Example)



Level of Design Abstraction



High-Level Design Activities

- Architectural design description
  - logical/functional decomposition
  - execution control & data flow
  - design requirements/constraints
  - relationships/interactions among modules
  - memory and processing time per module

High-Level Design Activities
(continued)

- External interface design
  - interaction criteria
  - expected response
  - protocol & conventions
  - error identification, handling & recovery
- Allocation of SW requirements to design
- Database description / Data dictionary
- Human interaction

Detailed Design

- Module description
  - I/O
  - function
  - control & signal flow
  - error handling
  - interfaces among modules

Detailed Design
(continued)

- Information needed to code module
  - detailed design to lowest level
  - functions/operations
  - algorithms
  - data definitions
  - local/global data
  - parameters
  - logic flow
  - error detection & handling details
  - limitations and performance restrictions

Detailed Design
(continued)

- Coding & implementation notes
  - stubs (for incremental development)
  - use of compiler options
- Test plans, design & proposed execution

Objective of Design Verification

To ensure, with a high level of confidence, that the design is ready for coding.

Helps set stage for PDR and CDR
CDR: last stage customer can cost-effectively stop project if deficiencies exist in design

Design Verification Tasks

- Traceability analysis
  - trace of design to requirements
- Design evaluation
  - correctness
  - design quality
- Interface analysis
  - correctness
  - data items across interface

Cautions

- Separate activities
  - verification back to requirements
  - design analysis
  - documents analysis
    - » content completeness
    - » content structure
    - » content correctness (design analysis)

SDD Analysis
(A Detailed Overview)

Using defined standard (Lewis-Fig. 6.3)

(1) Adherence to applicable standards or best practices

(2) How well is design stated?
  - readable
  - understandable

(3) How complete is SDD?

SDD Analysis
(A Detailed Overview)

Using defined standard (Lewis-Fig. 6.3)

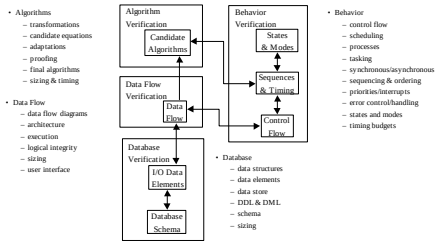
(4) How faithfully and completely are requirements in SRS
  - included, and
  - referenced

Evaluation Source: SRS

- CSCI capability requirements to CSC level
- CSCI internal interfaces
  - data and control flow at CSC level
- CSCI data elements
- External data elements
- Adaptation requirements
- Sizing and timing
- Safety
- Security
- Design constraints
- Quality factors
- Human engineering
- Requirements traceability

SDD Analysis
(A Detailed Overview)

(5) How completely and adequately data flow, behavior, algorithms, databases, and data dictionary are designed.



SDD Analysis
(A Detailed Overview)

(6) How well does STP and STD cover referenced requirements from SRS?

Verifying IDD

- Verify completeness and consistency among interfaces
- Interfaces in IDD and SDD must match
- Items examined during verification
  - interface diagrams
  - purpose of each interface
  - complete description of each element
    - » project-unique ID for each element
    - » brief description
    - » configuration item that is source of data element
    - » configuration item that is user of data element
    - » units of measure
    - » limit/range
    - » accuracy/precision
    - » legality checks
    - » data type/format
  - message descriptions
  - details of communication protocols

Verifying STP

Does STP provide:

- Test Environment
  - identify HW/SW items
  - installation particulars
- CSC Name
  - general test requirements
    - » size and execution time
    - » I/O
    - » test levels (CSC's)
  - test definitions
    - » objectives
    - » special requirements
    - » qualification method
    - » assumptions & constraints
  - test schedule
- Analysis Process

Verifying STD

Does STD stipulate:

- Preparation procedures
  - schedule
  - HW/SW
  - initialization
- Test description
  - initialization
  - inputs
  - expected results
  - criteria for evaluating results
  - test procedures
  - assumptions / constraints

Verifying Data Flow

- Uses DFD and structure charts (structure charts provide additional detail)
- Things to look for
  - completeness & consistency of coverage
  - mixed symbology (bad)
  - each data flow should
    - » have unique name
    - » be in data dictionary
  - each process label is
    - » unique
    - » descriptive
  - hierarchical structure

Verifying Database and Data Dictionary

- Database
  - data flow and data store descriptions as used in DFDs
  - data structure
  - data elements
- Data Dictionary
  - unique name for each data element
  - adequate description
  - origin / destination
  - unit of measure
  - range
  - accuracy
  - data type