

# **CSC 309 Lecture Notes Week 1**

**Introduction to the Course**  
**Introduction to the Design Process**

# **I. First-day handouts**

## **A. Syllabus**

## **B. Milestone 1, including**

- 1. SCO format**
- 2. work breakdown format**
- 3. meeting minutes format**

# Syllabus Review

## Page 1:

- Instructor
- Course Objectives
- Class Materials
- Activities

# Syllabus Review, Cont'd

## Page 2:

- Project Milestones
- Evaluations

# Syllabus Review, Cont'd

## Page 3:

- Bi-Weekly Activity Reports
- How to Submit Project Work
- Team Work
- Computer Work

# Syllabus Review, Cont'd

## Page 4:

- Lecture, Lab, Milestone & Exam Schedule

# Milestone 1

# Milestone 1

1. Form teams.



# Milestone 1

1. Form teams.
2. Select project.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.
4. Write initial SCOs.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.
4. Write initial SCOs.
5. Define initial levels of completion.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.
4. Write initial SCOs.
5. Define initial levels of completion.
6. Determine initial work breakdown.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.
4. Write initial SCOs.
5. Define initial levels of completion.
6. Determine initial work breakdown.
7. Install admin templates.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.
4. Write initial SCOs.
5. Define initial levels of completion.
6. Determine initial work breakdown.
7. Install admin templates.
8. Create project repository, release.

# Milestone 1

1. Form teams.
2. Select project.
3. Read specs.
4. Write initial SCOs.
5. Define initial levels of completion.
6. Determine initial work breakdown.
7. Install admin templates.
8. Create project repository, release.
9. Consider implementation platform.



## **M1, Task 1 -- Form Project Team**

- In lab today.
- You may change teams & projects.
- Admin tasks:
  - a. Exchange contact info.
  - b. Determine meeting times.
  - c. Elect officers

## **M1, Task 2 -- Select a Project**

- EClass, Grader, Scheduler, TestTool
- From last quarter, and/or before.
- We'll discuss in lab today and Wed.

## **M1, Task 3 -- Read the specs.**

- Goto 308 specs link.
- Look for
  - a. important features missing
  - b. features not spec'd clearly
  - c. features that could be "better"
  - d. how to merge features, if appro
  - e. models inconsistent with scenarios

## **M1, Task 4 -- Initial SCO's**

- Choose a base spec.
- Add features as appropriate from other specs

## **M1, Task 5 -- Levels of completion:**

- LEVEL 1: fully design and fully implement
- LEVEL 2: fully design but no implementation
- LEVEL 3: provide design hooks

## **M1, Task 6 -- Initial Work Breakdown**

- For high-level design phase
- Will refine as quarter progresses

## M1, Task 7 -- Fill in Templates

- In handouts dir:
  - o `scos-template.html`
  - o `work-breakdown-template.html`
  - o `meeting-minutes-template.html`
- Commit to repository when ready

## **M1, Task 8 -- Repository**

- Librarian creates repository.
- Template filler inners commit.
- Librarian releases by 7PM Monday 9 January.



## **M1, Task 9 -- Imple'n Platform?**

- Consider what you'd like to use.
- Instructor component must be desktop app
- Students can use web app
- Java recommended, Python an alternative

## **II. Milestone 2 Discussion**

**A.** See the writeup.

**B.** We'll go over key points in class.

## Milestone 2, cont'd

### C. Objectives:

1. Package design
2. Model/View class design
3. Initial implementation of  
model/view communication

## Milestone 2, cont'd

### D. Deliverables

1. Package structure
2. `overview.html`
3. `package.htmls`
4. Compilable model and view classes

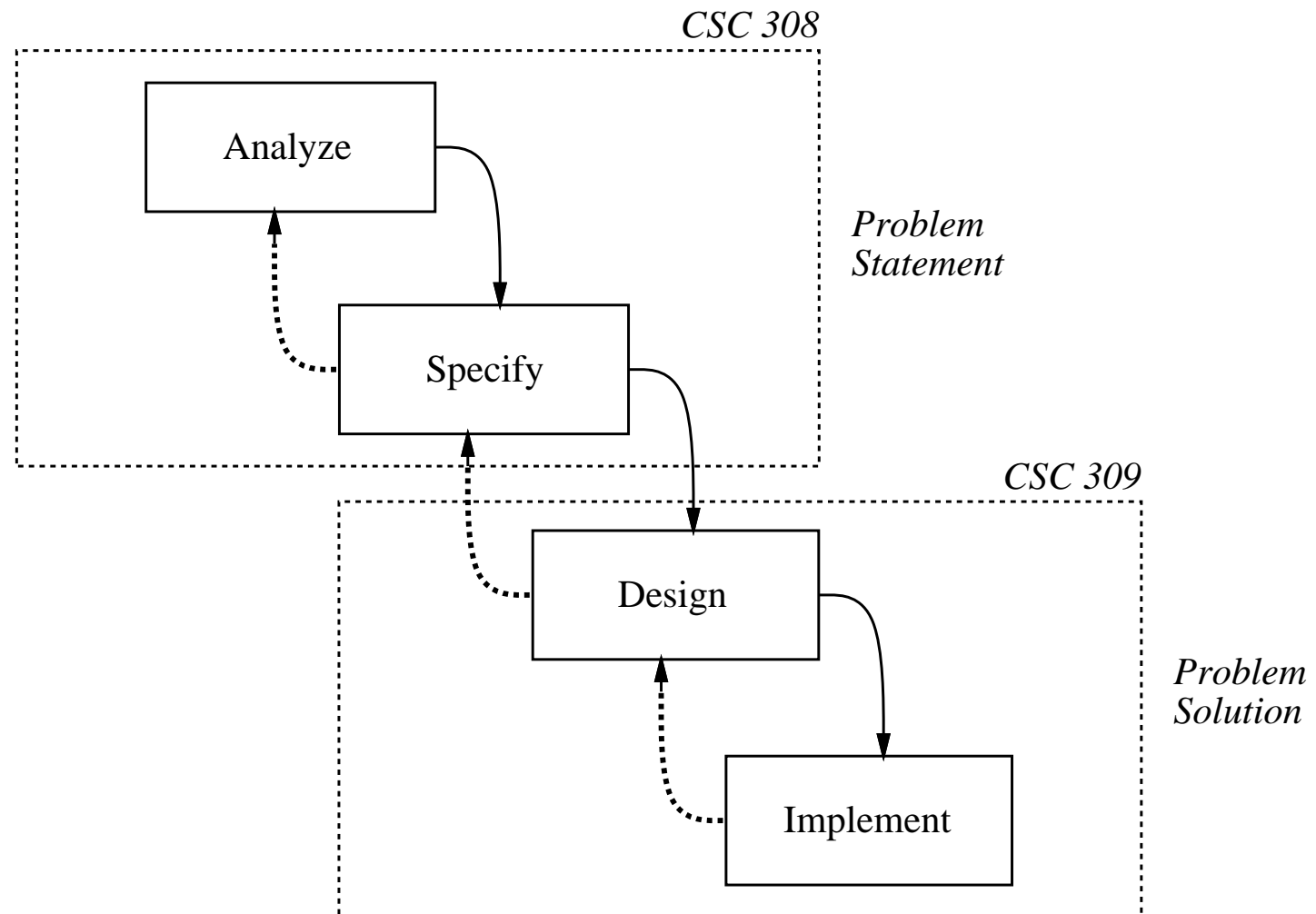
## Milestone 2, cont'd

5. Menubar or equivalent top-level UI
6. At least two model/view class integrations
7. Javadoc (pydoc) commentary for all

## Milestone 2, cont'd

8. `admin/m2-duties.html`
9. `admin/work-breakdown.html`
10. `HOW-TO-RUN.html`
11. Generated javadoc, or equiv
12. A runnable `.jar` file, or equiv

### III. Review of software systems life cycle.



## Life Cycle, cont'd

- A. *Requirements Analysis* determines what end users want and need.
- B. *Specification* formally defines user requirements.
- C. *Design* defines and organizes operational parts.
- D. *Implementation* defines operational details.



## **IV. Review of requirements analysis and specification phases**

- A.** *What the system does as opposed to how it works.*
  
- B.** The domain of CSC 308.

## Review of requirements, cont'd

- C. In 309, we'll work on EClass, Grader, TestTool
  1. from last quarter
  2. and/or from previous quarters
  3. we'll discuss in week 1 lab

## Review of requirements, cont'd

**D.** 308 specs located at:

```
http://www.csc.calpoly.edu/  
~gfisher/classes/309/specs
```

## V. Major goals of the design process

### A. Adhere to the specification

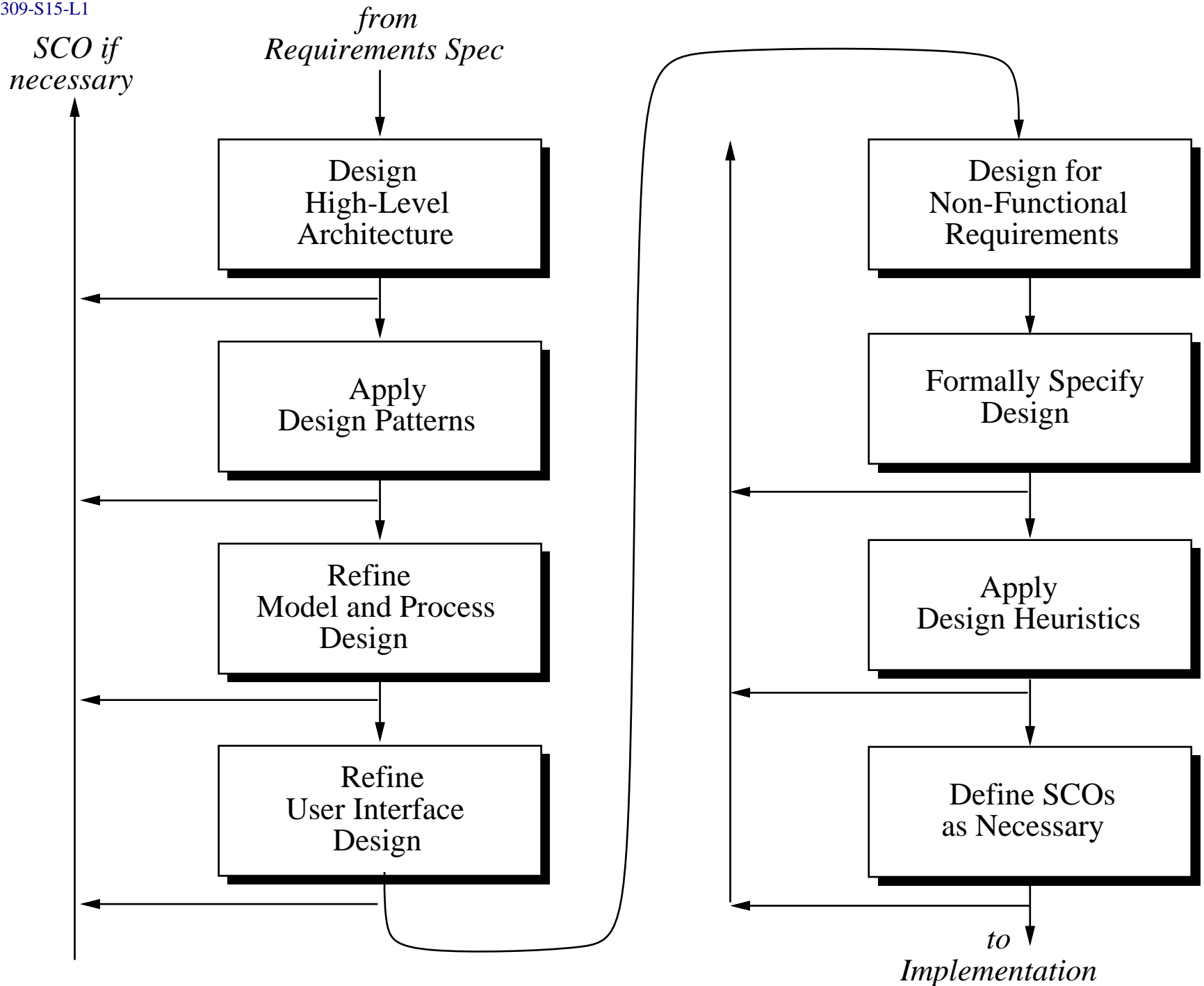
1. Any deviation in a SCO
2. The spec + SCOs form binding *contract*
3. No changes without consulting customer

## Goals of design, cont'd

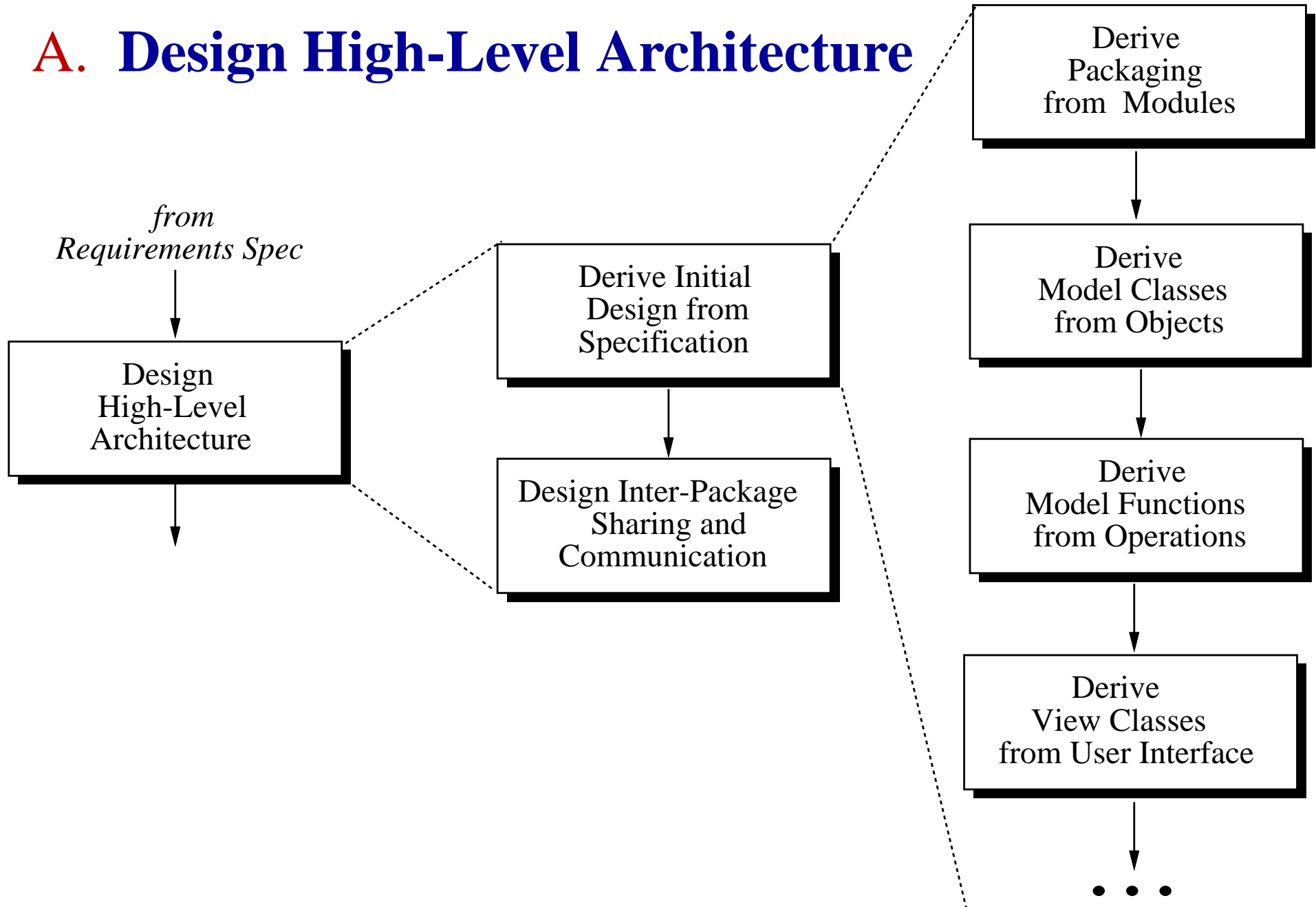
### B. Achieve design quality goals:

1. *Traceability*
2. *Modularity*
3. *Portability*
4. *Maintainability*
5. *Reusability*

# VI. 309 design process

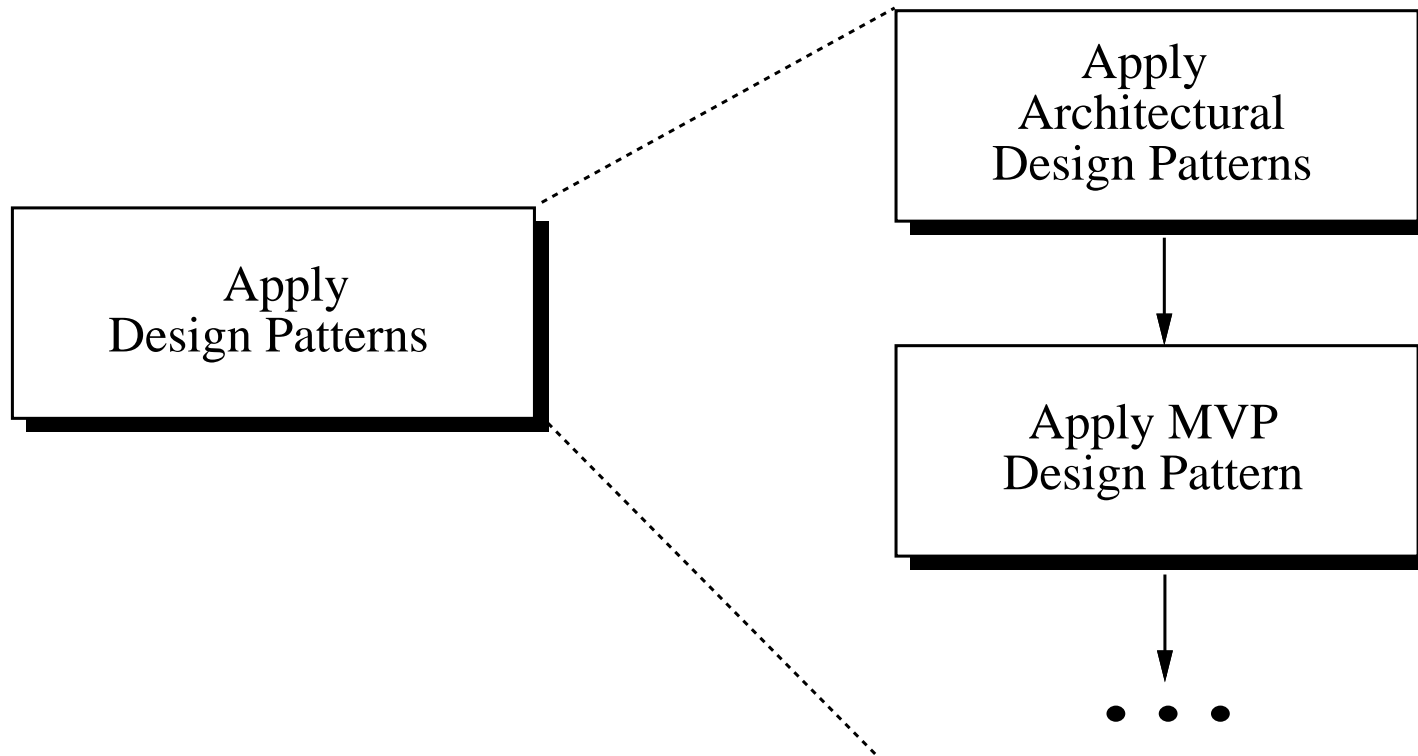


## A. Design High-Level Architecture

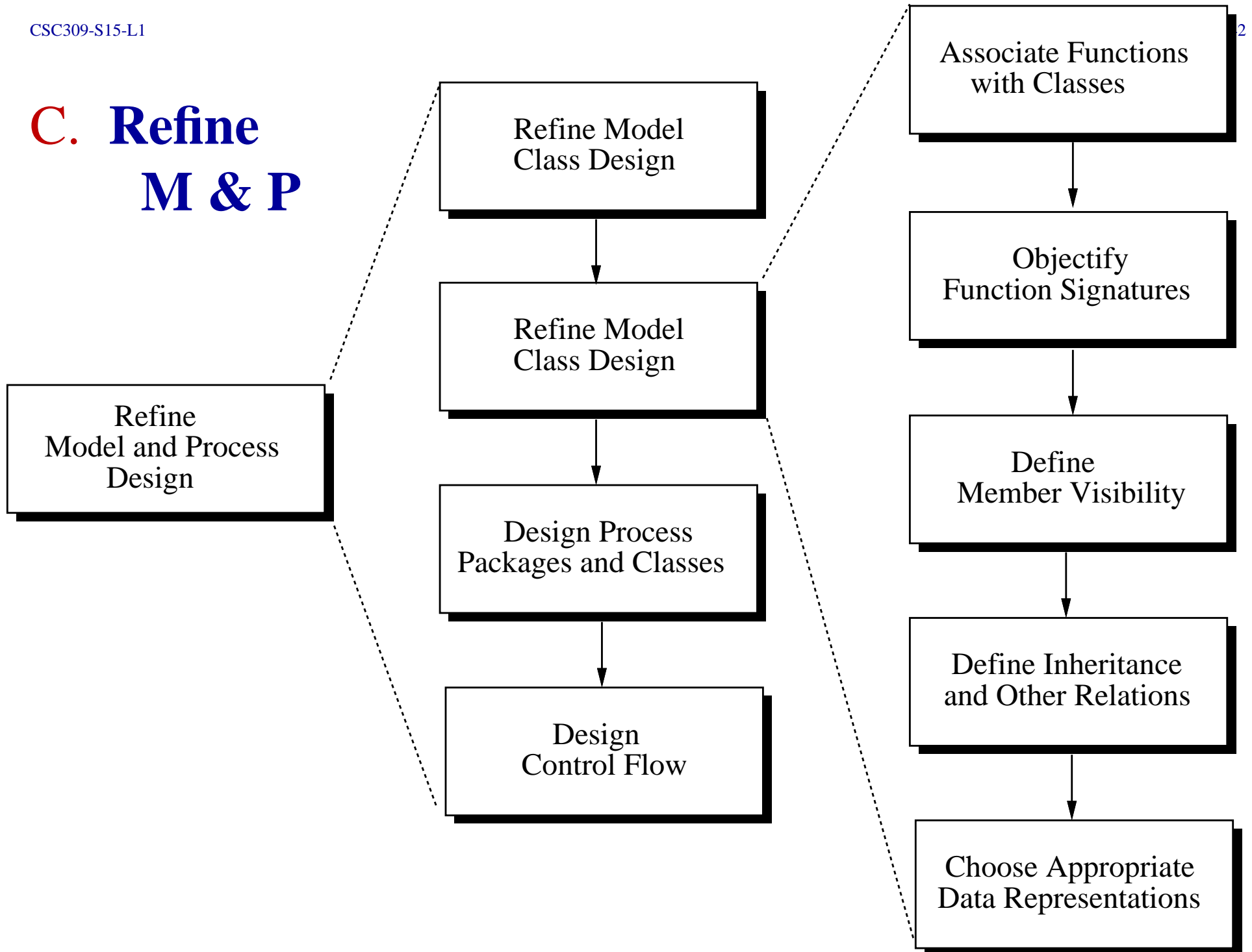




## B. Apply Design Patterns



# C. Refine M & P



## Design process, cont'd

### D. Refine UI Design

1. The fourth step.
2. Relies heavily on libraries.
3. Commonly-used interface elements and layouts.
4. Model classes must be refined.
5. Particularly useful is "Observer/Observable".

## Design process, cont'd

### **E. Design for Non-Functional Requirements**

1. Any non-functionals not yet incorporated.
2. Ensure system-related non-functionals are fully addressed.

## Design process, cont'd

### F. Formally Specify Design

1. As detailed program design established.
2. Precise def of function signatures and pre/post.
3. Derived from pre/posts defined in ops.

## Design process, cont'd

### G. Apply Design Heuristics

1. Applied throughout the process.
2. Minimizing coupling.
3. Maximizing cohesion.
4. Other heuristics, such as controlling size.

## Design process, cont'd

### H. Define SCO and Iterate Back

1. Aspects of requirements spec may need to be modified or enhanced.
2. Designer defines a *specification change order*.
3. In keeping with our "traditional" process.

## **VII. Comments on the 309 Design Process**

**A.** Employs a number of design methodologies:

**1.** UML

**2.** "Classical" structured design techniques.

**3.** MVP (Model-View-Process),  
(aka, MVC -- Model-View-Controller),



## Design process comments, cont'd

- B.** Works for systems with substantial HCIs.
- C.** Also for types of system, with adjustments.
- D.** Types of system without significant HCI:
  1. Realtime systems
  2. Utility systems
  3. Embedded systems

## **VIII. Languages of specification and design.**

- A.** Sometimes problems in translation from spec language into imple'n language.
- B.** Spec lang may differ from prog lang.
- C.** Not the case in 309 this year.

