

CS 490  
Design Exhibition  
Fall 2010

Sanjeev Setia

<http://www.cs.gmu.edu/~setia/cs490/>

Course Objectives

- ❑ Provide students an opportunity to work on a semester long project
  - Project more substantial than a typical project assigned in a class
  
- ❑ This offering of course intended as a trial run to evaluate feasibility of introducing a required capstone course for CS majors

## Prerequisites

- ❑ Senior status
- ❑ Grade of C or better in CS 310, CS 330 and CS 367
- ❑ CS 421 desirable
- ❑ Adequate background/knowledge of skills needed for project

3

## Project

- ❑ Ideally....
  - Team-oriented (3 - 5 students)
  - Produce software and/or hardware artifacts
  - Follow a software design methodology
  - Result in a demo
- ❑ This semester
  - Will allow smaller teams or individual projects
  - Can do a project where the main focus is research
    - Need a faculty member who agrees to be research supervisor

4

## Schedule

8/30: Course Introduction & Team Formation  
9/13: Project Proposals Due  
9/20: Class Presentations (Project Description & Goals)  
9/27: Design Documents Due  
10/4: Class Presentations (Project Design)  
11/8: Midterm Demos  
12/6: Final Project Reports Due  
12/13: Class Presentations & Demos

9/27 onwards weekly progress reports (submitted electronically)

Bi-weekly class presentation (3-4 slides) going over project progress

5

## Grade

- ❑ Project Deliverables -- 65%
  - Design Document (20%)
  - Final Report, Project generated artifacts (e.g documented code, user manual), Demo (45%)
- ❑ Project Presentations - 15%
  - Proposal, Design, Final
- ❑ Midterm Demo, Weekly Progress Reports, Biweekly presentations (20%)

6

## Project Proposal (due Sept 13)

- ❑ Two-three pages
- ❑ Description of project
  - Overview/Goals
  - Technology
  - Challenges/Issues
  - Team Organization/Division of Labor
  - Milestones/Schedule
  - Midterm Demo
  - Final Demo

7

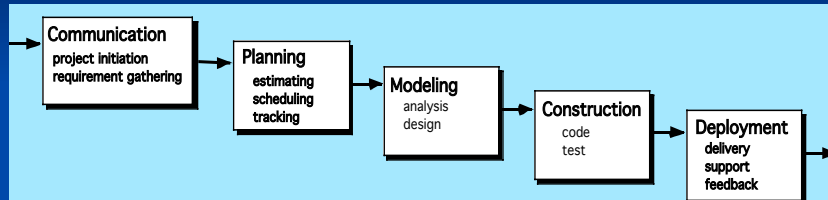
## Review of Software Engineering Process

- ❑ Requirements specification
- ❑ Analysis
- ❑ Design
- ❑ Implementation
- ❑ Testing
- ❑ Maintenance

Following slides borrowed from Prof. Dan Fleck's CS 421 lectures

8

# The Waterfall Model



Use when:

-Requirements are stable and well-understood, very short timeline (maintenance fixes)

9

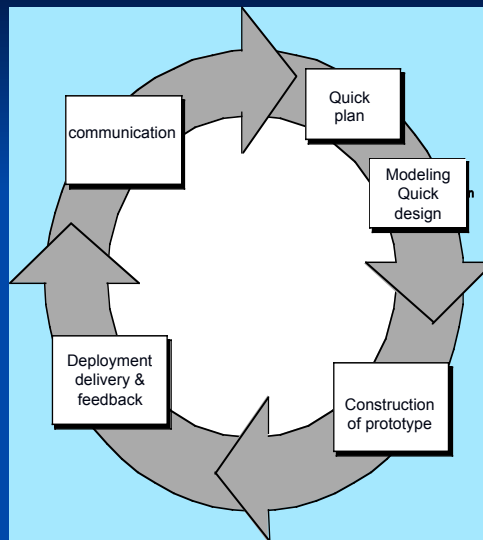
# Evolutionary Models: Prototyping

Used when partial system can be delivered, then evolved into full system

Prototyping is a tool that can be used during any process

Used when customer only has a vague idea of what they want

Plan to either throw-away or evolve into real product -- there will be pressure at the end to evolve into the real product



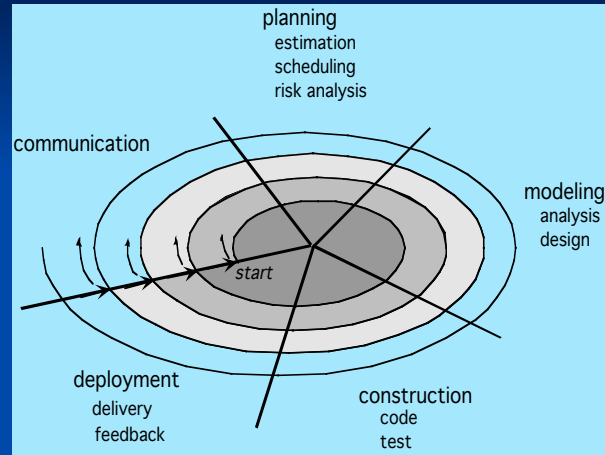
10

## Evolutionary Models: The Spiral

Complete highest risk items first

Used to mitigate risk on risk-intensive projects

Every spiral revises cost/ budget/schedule/ etc...



11

## Typical Prescriptive Requirements Engineering Process

- **Inception**—ask a set of questions that establish ...
  - basic understanding of the problem
  - the people who want a solution
  - the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and collaboration between the customer and the developer
- **Elicitation**—elicit requirements from all stakeholders
  - to identify the problem
  - propose elements of the solution
  - negotiate different approaches, and
  - specify a preliminary set of solution requirements
- **Elaboration**—create an analysis model that identifies data, functional and behavioral requirements
- **Negotiation**—agree on a set of requirements among all stakeholders (realistic, non-conflicting, inline with budget)

12

## Requirements Engineering-II

- **Specification**—can be any one (or more) of the following:
  - A written document
  - A set of models
  - A formal mathematical
  - A collection of user scenarios (use-cases)
  - A prototype
- **Validation**—a review mechanism that looks for
  - errors in content or interpretation
  - areas where clarification may be required
  - missing information
  - inconsistencies (a major problem when large products or systems are engineered)
  - conflicting or unrealistic (unachievable) requirements.
- **Requirements management**

13

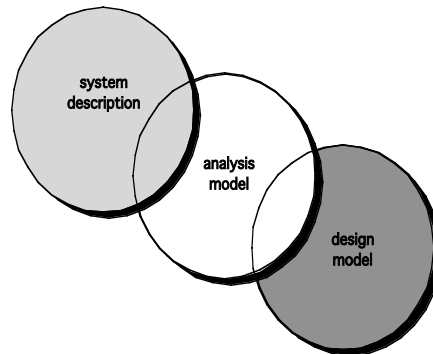
## Typical Prescriptive System Requirements Specification (SRS) Document Structure

- Purpose
- Overall Description
- System Features (Functional Requirements/Use Cases)
- External Interface Requirements
  - User interface requirements or standards
  - Hardware interfaces to other systems
  - Software interfaces to other systems
  - Communication Interfaces
- Non-functional Requirements
- Appendices - analysis model diagrams
- See SRS Template on class web page

(provided by <http://www.processimpact.com/> ... but essentially from IEEE).

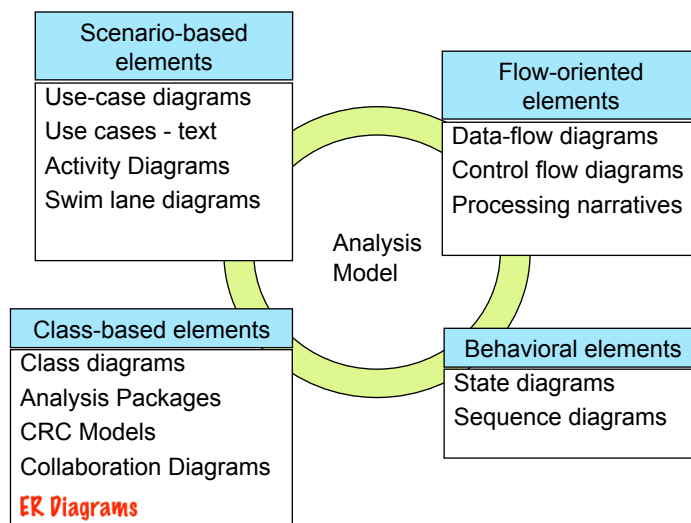
14

# Analysis Phase: What is it?



- Three objectives:
- To describe what the customer requires
  - To establish a basis for the creation of a software design
  - To define a set of requirements that can be validated once the software is built

# Elements of the Analysis Model





## Elements of the Analysis Model



### Scenario-based elements

High level idea of the system from user's or a functional perspective

### Flow-oriented elements

How information flows throughout the system (data and control flow)

### Behavioral elements

How the system responds to external stimuli

### Class-based elements

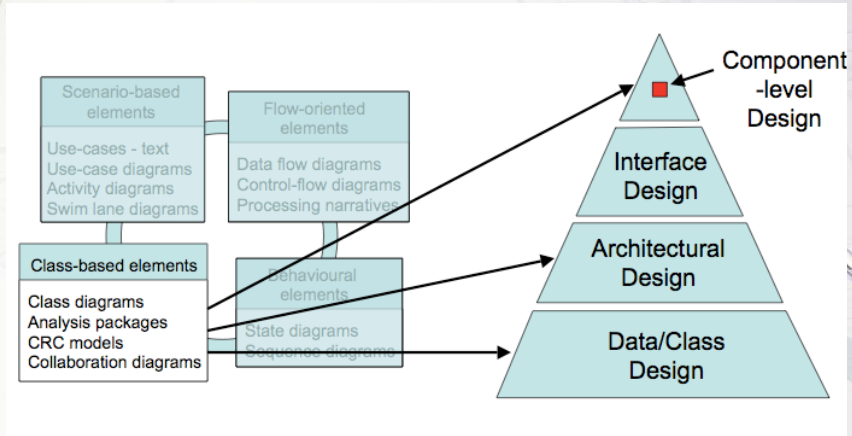
Static view of the system and how the different parts are related. Tries to show standard ideas of object oriented development

17

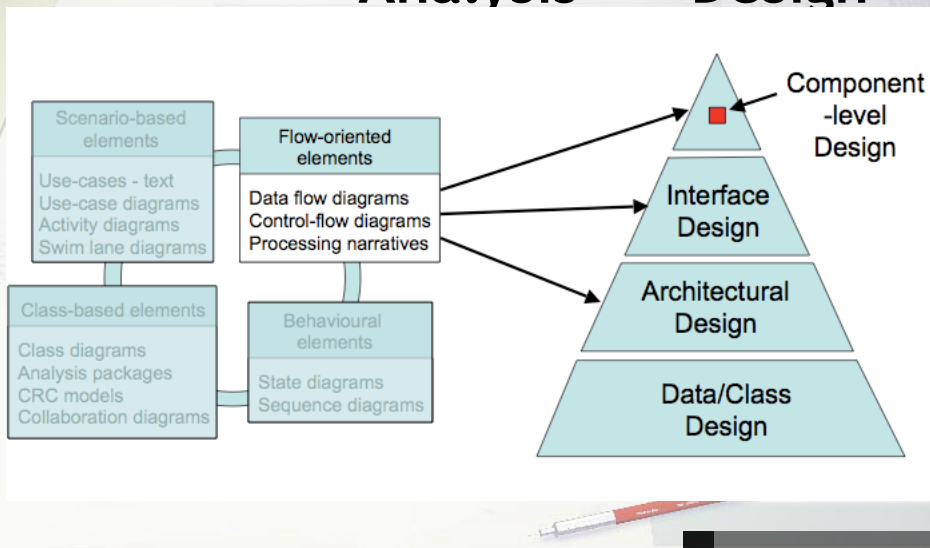
## What is the design phase?

- **Analysis phase** describes **what** the system should do
- Analysis has provided a collection of classes and descriptions of the scenarios that the objects will be involved in. These functions are clustered in groups with related behavior.
- The **design phase** is to work out **how** the system should do these things. This is the goal of the design phase.

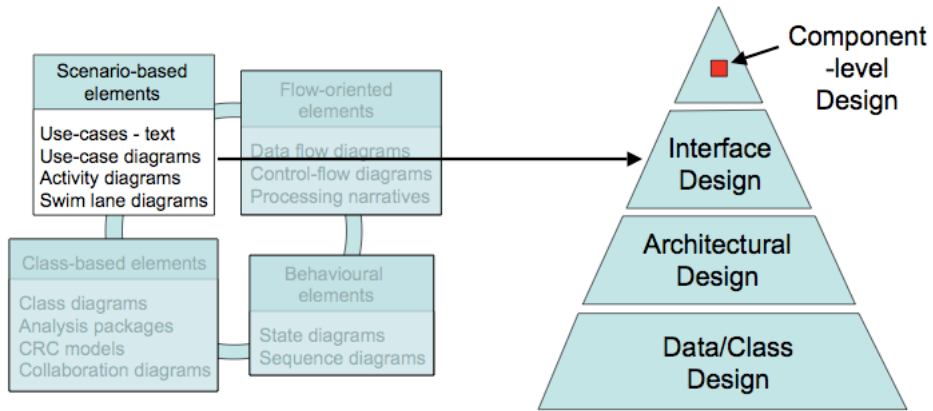
# Analysis --> Design



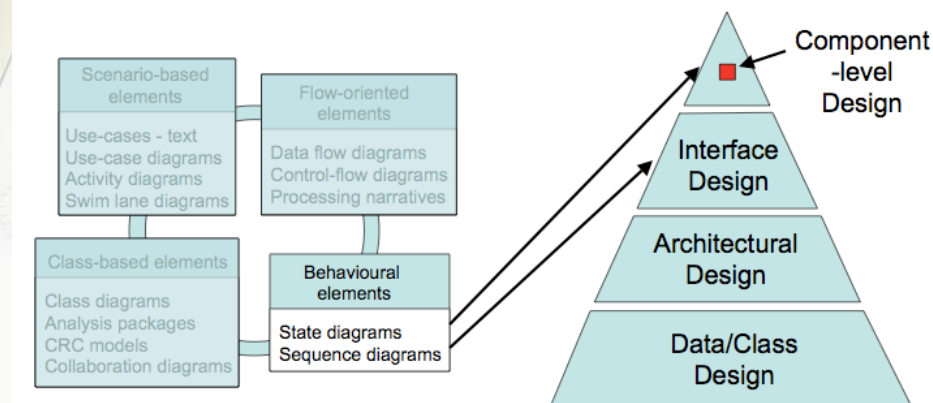
# Analysis --> Design



# Analysis --> Design



# Analysis --> Design





# The Design Spec

## Architecture Design -

- Layers of the software (e.g. model, view, controller (MVC))
- Categories of classes (e.g. UI, Business logic, interfaces)

## Component design -

- Description of classes/methods/algorithms
- State machines for classes
- (Think: individual classes)

## UI design

- sample screens
- UI guidelines/standards we're using
- detailed description of how UI components work

## Data design -

- database design
- data structures we're using.



# The Design Spec

But really, how do I create a design spec?

Find examples and use what you think is helpful from them!

<http://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs2/design.pdf>

<http://www.cmcrossroads.com/bradapp/docs/sdd.html>

## This class...

- ❑ Compressed Design Phase
  - Design document due 9/27
  - Does not need to be as detailed as the design template
  - You can keep refining the design
- ❑ Need some working code for a midterm demo 11/8
- ❑ Final Demo 12/13

25

## Other Issues

- ❑ Use SVN for version control
- ❑ IT&E Labs can provide machines/software for projects, e.g. application servers, web servers

26