



UML Diagram Types

Dynamic Models	Structural Models
■ activity diagrams	■ <i>class diagrams</i>
■ <i>statechart diagrams</i>	■ object diagrams
■ interaction diagrams	■ packages
– sequence diagrams	Architectural Models
– collaboration diagrams	■ component diagrams
■ <i>use case diagrams</i>	■ deployment diagrams



Software Engineering Process Models

Models to be discussed

- Sequential/Waterfall Model
- Prototype Model
- Incremental Model
- Spiral Model
- Embedded Controller Engineering Process



Generic Parts of Each Model

- System Engineering: establishing requirements for how system being designed fits into larger system
- Requirements Analysis: focus on specific software implementation and what the software system must do
- Design: focus on how the software system will be implemented
 - data structure
 - software architecture
 - interface representations
 - algorithmic detail
- Code: writing software
- Test: testing software against some criteria (requirements, design, etc)

Sequential/Waterfall Model

Stepped process of:

- analysis
- design
- code
- test

■ Cons

- projects rarely follow described path
- often difficult to state requirements upfront
- working version not available to the end

Prototype Model

Build, then throw away

Quick design to show capability

■ Cons

- customer want apparent working system
- implementation compromises to obtain quick working system

Incremental Model

■ Pipelined waterfall model

■ Move resources at completion of increment

■ Primary core path, followed by add'l functionality paths

■ Pros

- specialization and redeployment of implementation teams
- gradual increase of personnel
- increments planned to manage technical risks



Spiral Model

- Evolutionary combination of prototyping and waterfall models
- Communication key to success
- Pros
 - realistic approach to large-scale systems
 - systematic stepwise refinement of software



Embedded Controller Engineering Process

- Entire process employed by GM Powertrain Electronics Integration and Software
- SWE model is portion of overall controller system implementation
