OO Development Process

UML and Process

- UML standardizes notation, not process
  - Increase likelihood of widespread acceptance
- There is significant variability in what constitutes an appropriate process depending on particular problem, staff, tools, etc.

Recommended Process Models

- Booch
- OOSE
- OMT
- Fusion
- Martin-Odell
- Responsibility Driven Design

Guiding Principles

- Process should be iterative and incremental.
  - Flaws in the waterfall model include:
    - complexity overload
    - delayed feedback
    - specifications frozen early (while business environment may change)
- Process should be use case driven.
- There should be early emphasis on defining the architecture of the system.

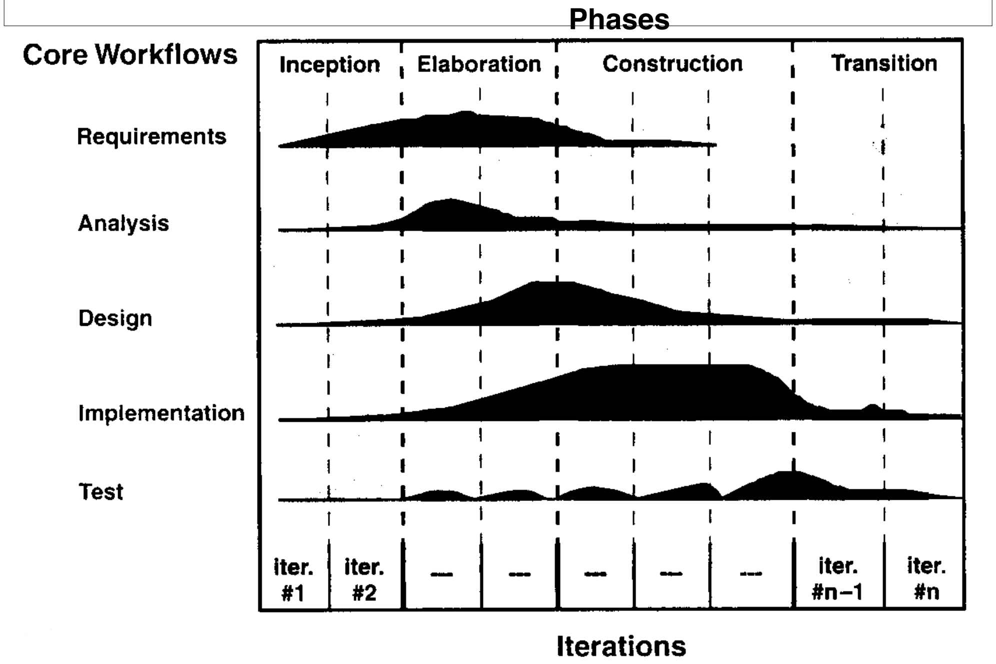
Iterative and Incremental

- Successive enlargement and refinement of a system through multiple cycles of analysis, design, implementation, and testing
- Each cycle involves a relatively small set of requirements.

Iterative and Incremental (2)

- Iterative development is a planned process, not haphazard.
- Each iteration may add new functionality or may refine and tune existing functionality.
- Incremental development refers to adding functionality to a system over several release cycles.
- Each release cycle includes multiple iterative development cycles.

Release Cycle Phases



Benefits of Iterative/Incremental

- Complexity is never overwhelming. No “analysis paralysis” or “design paralysis”.
- Early feedback can inform and enhance the analysis of subsequent cycles.
- Development team can apply their growing skills to subsequent cycles.
- Easier to modify requirements.

Disadvantages

- Early demonstrations may unfairly raise expectations of client and management:

“user interface working” = “finished system”

Use Cases Drive

- Requirements organization
- Estimates
- Schedules
- Concepts and software classes
- Test cases

Major Development Phases

- Plan and Elaborate
  - Project planning, requirement definition, prototype building, etc.
- Build
  - System construction
- Deploy
  - Put the system to use

Plan and Elaborate Activities

1. Define Draft Plan
   - Schedule, resources, budget, motivation, etc.
2. Conceptual (Problem Domain) Modeling
3. Requirement/Use Case Definitions
4. Develop Glossary
5. Define Draft System Architecture
6. Implement Prototype
7. Refine Plan

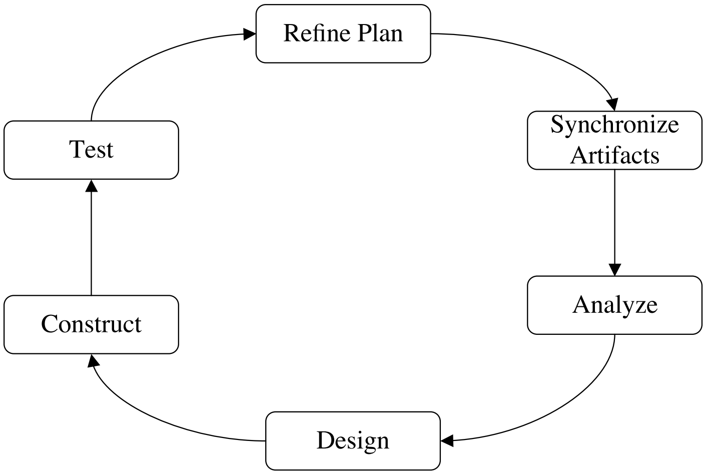
Kinds of Use Cases

- *Essential* use cases describe interactions in terms of the essential activities and motivation.
- *Real* use cases include information about the design and implementation such as user interface details.
- *High level* use cases are very brief descriptions of (essential) use cases.

Analysis During Planning Phase

- Trade off between too much complexity and benefits:
  - Comprehension, risk management, scope, estimating, etc.
- Create all the high-level use cases, but only expand the most critical and important use cases during the plan and elaborate phase.
- Defer the rest to the analysis step of the build cycle in which they are tackled.

Build Cycle Steps



Time-Boxing Build Cycles

- Time for each cycle is rigidly fixed
- Time box should be at least two weeks
  - Less is not enough to complete tasks
- Time box should be at most two months
  - More results in too much complexity and delay of feedback
- Requirements must be chosen carefully (and by the development team).

Use Case Driven Build Cycles

- Iterative development cycles are organized by use case requirements
- A single cycle might implement:
  - One or more complete use cases
  - A basic use case without extensions
  - One or more extensions to a previously implemented use case
- Use cases are prioritized

Build Emphasis

- Build cycles are often included in the Plan and Elaborate phase (Phase I), with an emphasis on exploration (1 to 3 months).
  - requirements, estimates, schedule
- Early Phase II (build phase) build cycles emphasize infrastructure.
  - architecture, frameworks, layers, subsystems
- Later Phase II build cycles emphasize the remaining system functions.

Analysis During Build Cycle

1. Define/Refine Essential Use Cases
2. Refine Use Case Diagrams
3. Refine Conceptual Model
4. Refine Glossary
5. Define System Sequence Diagrams
6. Define System Operation Contracts
7. Define System State Diagrams

Design

1. Define Real Use Cases
2. Define User Interface, Reports
3. Refine System Architecture
4. Define/Refine Design Class Diagrams
5. Define Interaction Diagrams
6. Define Database Schema

Key Points

- Development process is iterative, incremental and use case driven.
- Work is organized in time-boxed development cycles.
- Parallel team development may be organized according to architecture.