

Requirements Capture

- Determine what user/client wants
- Agree on a contract
- What problem is the program supposed to solve?
- Difficulties:
 - Users do not know what they want
 - Users may ask for impossible things
 - Requirements are usually vague, incomplete, contradictory
- Communication between user and developer

Analysis

- Understand the problem domain
 - Business modeling
 - Workflow modeling
- Look for unknown terms and concepts
 - Technical concepts of the user domain
 - Background knowledge needed to understand the problem
- Understand the tasks that users want to perform
- Identify conceptual classes and their relationships
 - Just enough modeling to express the tasks of the program

Design

- Take the analysis model and refine it into something that can be implemented
- Consider concerns of the programming language
 - Multiple inheritance?
 - Interfaces?
 - Built-in data types?
- Consider reuse
 - Standard libraries
 - Commercial software
- Other technical concerns
 - GUI
 - Database
 - Performance
- Add enough detail so that implementation will be easy

Implementation

- Working code!

Testing

- Find all kinds of faults and remove them
- Alpha-testing: in-house
- Beta-testing: by users
- Unit test: by developer
- System test: by opposing tester
- Faults introduced in:
 - Requirements
 - Analysis
 - Design
 - Implementation
- Different testing techniques find different faults

Maintenance

- All development after release
- Upgrades (“perfective”)
 - New features
- Fixes (“corrective”)
 - Problems that should have been solved before
- Reasons:
 - Users change their minds
 - Environment changes
 - Better technologies become available

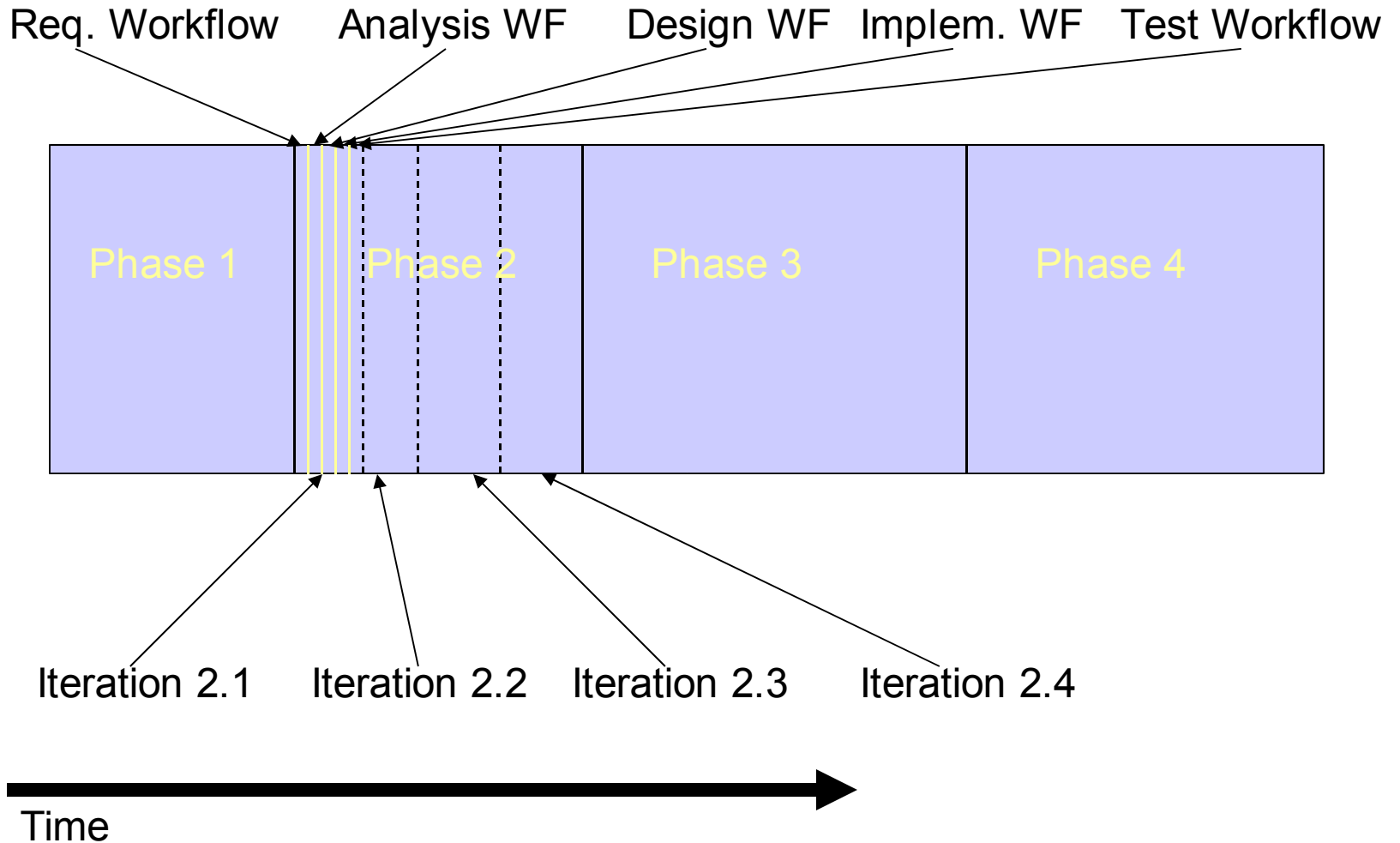
Some Process Models

- Waterfall
 - sequential with little feedback
- Incremental
 - waterfall in a loop
- Synchronize and Stabilize
 - concurrent
- Spiral
 - risk management and verification
- Unified Software Development Process
 - complex, object-oriented, adaptable
- Extreme Programming
 - light-weight, flexible, small
- Open Source
 - widely distributed development

3.2 Unified Process

- Unified Software Development Process (USDP)
- A process to work with the UML
- Three principles:
 - use-case driven
 - architecture-centric
 - incremental and iterative
- Phases and workflows
 - phases structure whole development cycle:
Inception, Elaboration, Construction, Transition
 - core workflows structure tasks
Requirements, Analysis, Design, Implementation, Test
- Several iterations make up a phase
- Complex, configurable

USDP Timeline



USDP Workflows

- Requirements, Analysis, Design, Implementation, Test
- Each iteration goes through all workflows
- Each workflow can happen in all phases
- A USDP workflow is a set of related tasks
 - each kind of task is done by a worker (role)
 - each task produces an artifact
- The following slides list the workers for each workflow, and the kinds of tasks for each worker

Requirements Capture Workflow

- Workers:
 - System Analyst
 - Find Actors and Use Cases
 - Structure Use Case Model
 - Use-Case Specifier
 - Detail a Use Case
 - User-Interface Designer
 - Prototype User Interface
 - Architect
 - Prioritize Use Cases
- Artifacts:
 - Domain model
 - Use-case model
 - User interface prototype
 - Supplementary specification

Analysis Workflow

- Workers:
 - Architect
 - Architectural Analysis
 - Use-Case Engineer
 - Analyze a Use Case
 - Component Engineer
 - Analyze a Class
 - Analyze a Package
- Artifacts:
 - Analysis model: classes, packages
 - Use-case realizations
 - Architecture description

Design Workflow

- Workers:
 - Architect
 - Architectural Design
 - Use-Case Engineer
 - Design a Use Case
 - Component Engineer
 - Design a Class
 - Design a Subsystem
- Artifacts:
 - Design model: subsystems, classes, operations, attributes, relations
 - Use-case realizations
 - Architecture
 - Deployment model

Implementation Workflow

- Workers:
 - Architect
 - Architectural Implementation
 - System Integrator
 - Integrate System
 - Component Engineer
 - Implement a Class
 - Implement a Subsystem
 - Perform Unit Test
- Artifacts:
 - Implementation of classes and interfaces (unit tested)
 - Updated architecture

Test Workflow

- Workers:
 - Test Engineer
 - Plan Test
 - Design Test
 - Evaluate Test
 - Component Engineer
 - Implement Test
 - Integration Tester
 - Perform Integration Test
 - System Tester
 - Perform System Test
- Artifacts:
 - Test cases
 - Test procedures
 - Test components (for automating tests)

USDP Phases

- Phases structure the process by time
 - Workflows structure the process by type of task
- Each phase consists of several iterations
- Inception, Elaboration, Construction, Transition

- Don't confuse phases, iterations, workflows!

USDP: Phases and Workflows

	Inception	Elaboration	Construction	Transition
Requirements				
Analysis				
Design				
Implementation				
Test				

Filled in: workflow is used a lot in this phase

Inception Phase

- Why do we do this project?
- How long?
- How expensive?
- Can it succeed?
- Deliverables:
 - feature list
 - draft of domain model
 - drafts of use-case model and analysis model
 - possibly: draft of design model, prototype
 - project plan

Elaboration Phase

- Identify most of the use cases
- Design architecture
- Trade-offs between requirements and architecture
- Develop first increments
- Make sure you have necessary resources
- Deliverables:
 - Complete domain model
 - Almost complete use case model
 - Almost complete analysis model
 - Draft of design model
 - First version of implementation, especially user interface
 - Plan of Construction and Transition

Construction Phase

- Create a beta release
- Adds in all the details
- Emphasis shifts from understanding to constructing
- Use cases are prioritized by business need
- Refactoring
- Lots of iterations: a few use cases in each
- Deliverables:
 - Executable, alpha-tested program
 - All models (use-case, analysis, design)
 - Draft of user manual
 - Plan for transition

Transition Phase

- Acceptance testing and beta-testing
- Fixing problems
- Installation
- Teaching users
- Things that cannot be done iteratively
- Deliverables:
 - Final release of the executable program
 - Installation software
 - Updated models and documentation
 - Setup of user support organization