Chapter 3
Part 1

The Workflows and Activities
Specification Document
Artifacts

1

3.3 The Requirements Workflow

- The aim of the requirements workflow
  - To determine the client's needs

2

Overview of the Requirements Workflow

- First, gain an understanding of the *application domain* (or *domain*, for short)
  - That is, the specific business environment in which the software product is to operate
- Second, build a business model
  - Use UML to describe the client's business processes
  - If at any time the client does not feel that the cost is justified, development terminates immediately

3

Overview of the Requirements Workflow

- It is vital to determine the client's constraints
  - Deadline
    - Nowadays, software products are often mission critical
  - Parallel running
  - Portability
  - Reliability
  - Rapid response time
  - Cost
    - The client will rarely inform the developer how much money is available
    - A bidding procedure is used instead

4

Overview of the Requirements Workflow

- The aim of this *concept exploration* is to determine
  - What the client needs
  - *Not* what the client wants

5

3.4 The Analysis Workflow

- The aim of the analysis workflow
  - To analyze and refine the requirements
- Why not do this during the requirements workflow?
  - The requirements artifacts must be totally comprehensible by the client
- The artifacts of the requirements workflow must therefore be expressed in a natural (human) language
  - All natural languages are **imprecise**

6

The Analysis Workflow

- Example from a manufacturing information system:
  - "A part record and a plant record are read from the database. If it contains the letter A directly followed by the letter Q, then calculate the cost of transporting that part to that plant"
- To what does it refer?
  - The part record?
  - The plant record?
  - Both??
  - Could it mean the database??

7

The Analysis Workflow

- Two separate workflows are needed
  - The requirements artifacts must be expressed in the language of the client
  - The analysis artifacts must be precise, and complete enough for the designers

8

The Specification Document

- Specification document ("specifications")
  - It constitutes a contract
  - It must not have imprecise phrases like "optimal," or "98% complete"
- Having complete and correct specifications is essential for
  - Testing and
  - Maintenance

9

What we have seen so far

- Project Name
- Vision
- Boundaries
- Business Case
- Risks
- Supplemental Requirements (Non-Functional)
- Specific Requirements
- –Supporting use-case
- –Supporting user-story
- Glossary

10

The Specification Document

- The specification document must not have
  - Contradictions
  - Omissions
  - Incompleteness
- But it will

11

Software Project Management Plan

- Once the client has signed off the specifications, detailed planning and estimating begins
- We draw up the software project management plan, including
  - Cost estimate
  - Duration estimate
  - Deliverables
  - Milestones
  - Budget
- This is the earliest possible time for the SPMP

12

3.5 The Design Workflow

- The aim of the design workflow is to refine the analysis workflow until the material is in a form that can be implemented by the programmers
  - Many nonfunctional requirements need to be finalized at this time, including
    - Choice of programming language
    - Reuse issues
    - Portability issues

13

Classical Design

- Architectural design
  - Decompose the product into modules
- Detailed design
  - Design each module:
    - Data structures
    - Algorithms

14

Object-Oriented Design

- Classes are extracted during the object-oriented analysis workflow and
  - Designed during the design workflow
- Accordingly
  - Classical architectural design corresponds to part of the object-oriented analysis workflow
  - Classical detailed design corresponds to part of the object-oriented design workflow

15

The Design Workflow

- Retain design decisions
  - For when a dead-end is reached
  - To prevent the maintenance team reinventing the wheel

16

3.6 The Implementation Workflow

- The aim of the implementation workflow is to implement the target software product in the selected implementation language
  - A large software product is partitioned into subsystems
  - The subsystems consist of *components* or *code artifacts*

17

3.7 The Test Workflow

- The test workflow is the responsibility of
  - *Every* developer and maintainer, and
  - The quality assurance group
- Traceability of artifacts is an important requirement for successful testing

18

3.7.1 Requirements Artifacts

- Every item in the analysis artifacts must be traceable to an item in the requirements artifacts
  - Similarly for the design and implementation artifacts

19

3.7.2 Analysis Artifacts

- The analysis artifacts should be checked by means of a review
  - Representatives of the client and analysis team must be present
- The SPMP must be similarly checked
  - Pay special attention to the cost and duration estimates

20

3.7.3 Design Artifacts

- Design reviews are essential
  - A client representative is not usually present

21

3.7.4 Implementation Artifacts

- Each component is tested as soon as it has been implemented
  - *Unit testing*
- At the end of each iteration, the completed components are combined and tested
  - *Integration testing*
- When the product appears to be complete, it is tested as a whole
  - *Product testing*
- Once the completed product has been installed on the client's computer, the client tests it
  - *Acceptance testing*

22

Implementation Artifacts

- COTS software is released for testing by prospective clients
  - Alpha release
  - Beta release
- There are advantages and disadvantages to being an alpha or beta release site

23

3.8 Postdelivery Maintenance

- Postdelivery maintenance is an essential component of software development
  - More money is spent on postdelivery maintenance than on all other activities combined
- Problems can be caused by
  - Lack of documentation of all kinds

24

Postdelivery Maintenance

- Two types of testing are needed
  - Testing the changes made during postdelivery maintenance
  - Regression testing
- All previous test cases (and their expected outcomes) need to be retained

25

3.9 Retirement

- Software can be unmaintainable because
  - A drastic change in design has occurred
  - The product must be implemented on a totally new hardware/operating system
  - Documentation is missing or inaccurate
  - Hardware is to be changed — it may be cheaper to rewrite the software from scratch than to modify it
- These are instances of maintenance (rewriting of existing software)

26

Retirement

- True retirement is a rare event
- It occurs when the client organization no longer needs the functionality provided by the product

27

Technical Context Vs. Business Context

- Workflow
  - Technical context of a step
- Phase Increments and Episodes
  - Business context of a step

28

The End of Part 1

29