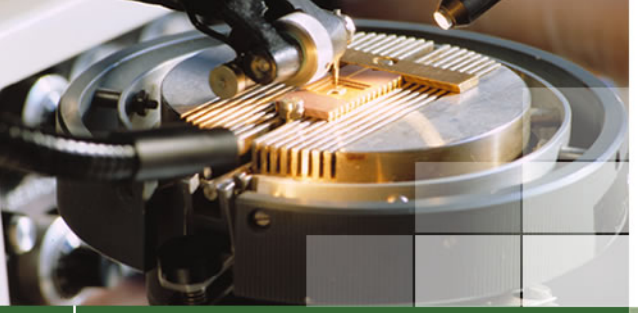


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Planning and Managing Software Projects 2011-12

Class 13

More on requirements

Emanuele Della Valle

http://emanueledellavalle.org

Credits

- This slides are largely based on Prof. John Musser class notes on “Principles of Software Project Management”
- Original slides are available at http://www.projectreference.com/
- Reuse and republish permission was granted

Today

- Class 11 and 12 review
- Capability Maturity Model
- Requirements (most critical activity)
- Other Tips

Risk Management

- Risk Management
  - Types of risk: schedule, cost, requirements
- Risk Identification
- Risk Analysis
  - Risk Exposure (RE = Prob. * Size)
- Risk Prioritization
- Risk Control
- Risk Resolution
  - Avoidance, assumption, transfer, gain knowledge
- Top 10 Risk List

People dimension

- Roles
- Staffing profile, roll-on and roll-off
- Hiring
- Team Models
- Mythical Man-Month
- Optimal team size
- Tools: Responsibility Assignment Matrix and Skill Matrix

People dimension - Team Models

- Business Team
  - flexible
- Chief-Programmer Team
  - Good for creative and tactical execution projects
- “Shunkworks” Team
  - Good for creative projects
- SWAT Team
  - Good for tactical execution projects

People dimension - Mythical Man-Month

- Optimism
- Confuse effort with progress
- Overhead of communication
- Urgency of Client causes Optimistic Estimates
  - E.g., omelet and chef analogy
- The myth of additional manpower

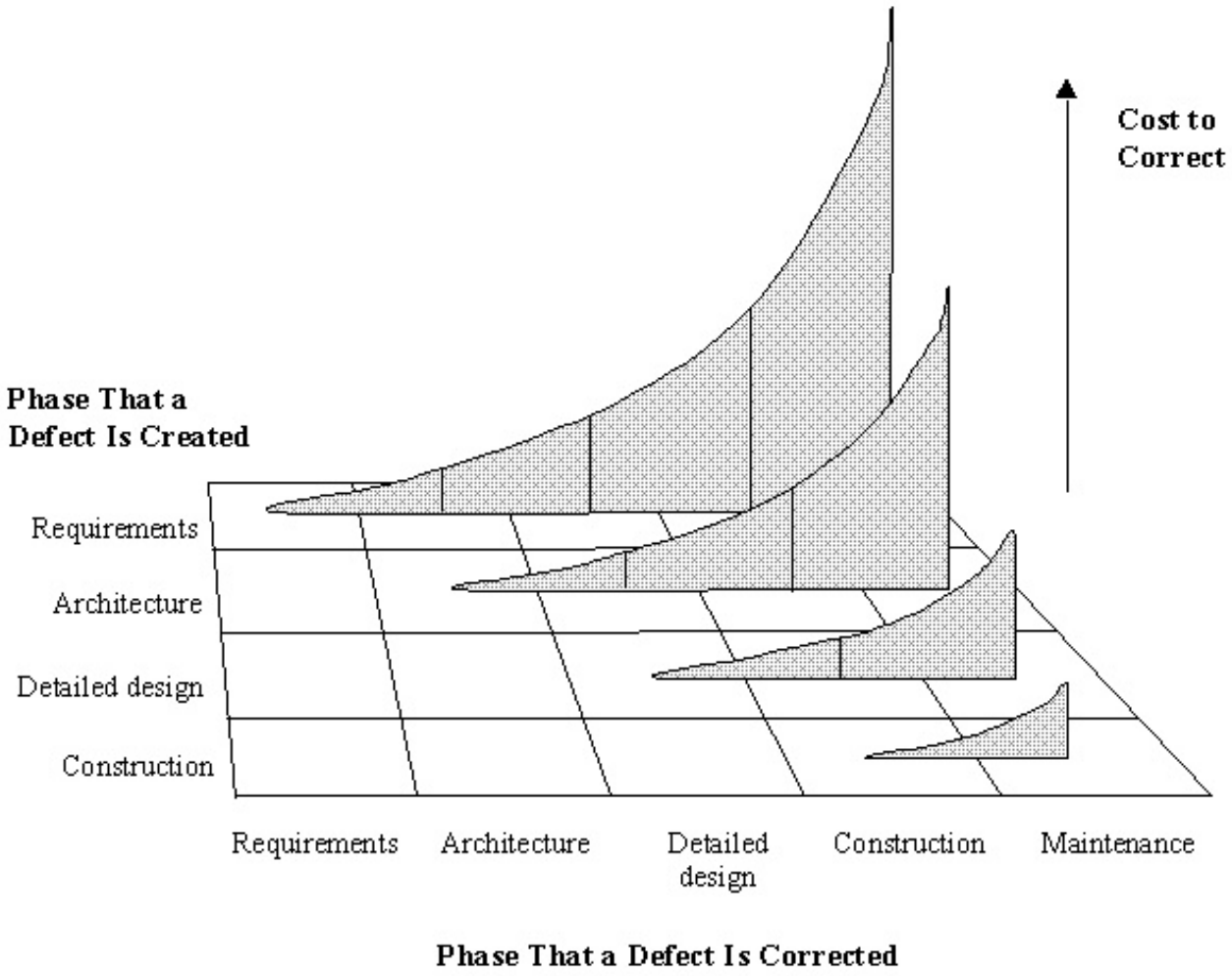
Requirements

- Requirements are capabilities and condition to which the system – more broadly, the project – must conform

Requirements

- Perhaps the most important & difficult phase to control
- Shortchanging it is a ‘classic mistake’
- Can begin with a Project Kickoff Meeting
- Can end with a Software Requirements Review (SRR)
  - For Sponsor and/or customer(s) approval

Why are Requirements so Important?



Characteristics & Issues

- Conflict of interest: developer vs. customer
- Potential tug-of-war:
  - Disagreement on Features & Estimates
  - Especially in fixed-price contracts
- Frequent requirements changes
- Achieving sign-off
- Project planning occurs in parallel

2 Types of Requirements (see lesson 3)

- Functional (behavioral)
  - Features and capabilities
- Non-functional (a.k.a. “technical”) (everything else)
  - Usability
    - Human factors, help, documentation
  - Reliability
    - Failure rates, recoverability, availability
  - Performance
    - Response times, throughput, resource usage
  - Supportability
    - Maintainability, internationalization
  - Operations: systems management, installation
  - Interface: integration with other systems
  - Other: legal, packaging, hardware

The “musts” to remember

- Must be prioritized
  - Must-have
  - Should-have
  - Could-have (Nice-to-have: NTH)
- Must be approved

Used by many people for many purposes

- Management: Yes, that's what I funded
- Users: Yeah, that's what I need
- Developers: Yes, that's what I will build

Input and Outputs (see session 3)

- The “What” phase
- Inputs: SOW, Proposal
- Outputs:
  - Requirements Document (RD)
    - a.k.a.Requirements Specification Document (RSD)
    - Software Requirements Specification (SRS)
  - 1st Project Baseline
  - Software Project Management Plan (SPMP)
  - Requirements Approval & Sign-Off
    - Your most difficult task in this phase

Quotes to Think About

- “There’s no sense being exact about something if you don’t even know what you’re talking about”

  -- John von Neumann

- “When the map and the territory don’t agree, always believe the territory”

  -- taught to all Swedish army recruits

Requirements Gathering Techniques

- Interviews
- Document Analysis
- Brainstorming
- Requirements Workshops
- Prototyping
- Use Cases
- Storyboards
- There are more…

Interview Techniques

- Best practice: use ‘context free’ questions
  - A question that does not suggest a response
  - High-level, early questions to obtain ‘global’ properties of the problem and solution
  - Applicable to any project/product
  - Get the ball rolling

Context-free Questions

- Process, product and meta questions
- Process
  - “Who is the client for project X”?
  - “What is a very successful solution really worth to the client”?
  - “What is the reason for this project”?
- Product
  - “ What problems does this system solve”?
  - “What problems could this system create”?
- Meta-questions
  - “Are these questions relevant”?
  - “Is there anyone else who can give useful answers”?
  - “Is there anything you want to ask me”?

Document Analysis

- Review of existing documents
  - Business plans
  - Market studies
  - Contracts
  - Requests for proposals (RFP)
  - Statements of work (SOW)
  - Existing guidelines
  - Analyses of existing systems and procedures

Brainstorming

- Idea generation & idea reduction
- Typically via group meetings
- Generation best practices
  - Minimize criticism and debate
    - Editing occurs at end of meeting or later
  - Aim for quantity
  - Mutate or combine ideas
- Reduction best practices
  - Voting with a threshold (N votes/person)
  - Blending ideas
  - Applying criteria
  - Scoring with a weighting formula

Requirements Workshops

- Typically 1-5 days
- Who? Varies. Users & stakeholders
- Pros
  - Good for consensus building
  - Builds participant commitment
  - Can cost less than numerous interviews
  - Provide structure to capture and analysis process
  - Can involve users across organizational boundaries
  - Can help identify priorities and contentious issues
- Joint Application Design (JAD)
  - A type of requirements workshop using a facilitator
  - See http://www.carolla.com/wp-jad.htm

Prototyping

- Quick and rough implementation
- Good communications mechanism
- Can be combined with other techniques such as JAD
- Issues: creating the mis-appearance that development is more complete than it actually is
  - See joelonsoftware's "The Iceberg Secret Revealed"
  - http://www.joelonsoftware.com/articles/fog0000000356.html

The Iceberg Secret

- You know how an iceberg is 90% underwater? Well, most software is like that too
  - 10% of the work goes into a *pretty* UI
  - 90% of the programming work is under the covers
- That's not the secret. The secret is that *People Who Aren't Programmers Do Not Understand This*.
- Corollaries:
  1. Bad interface → bad program
  2. Beautiful interface → program is complete
  3. When demanding nontechnical managers or customers "sign off" on a project, give them several versions of the graphic design to choose from.
  4. When you're showing off, the only thing that matters is the screen shot. Make it 100% beautiful.

[Source: See joelonsoftware's "The Iceberg Secret Revealed" http://www.joelonsoftware.com/articles/fog0000000356.html]

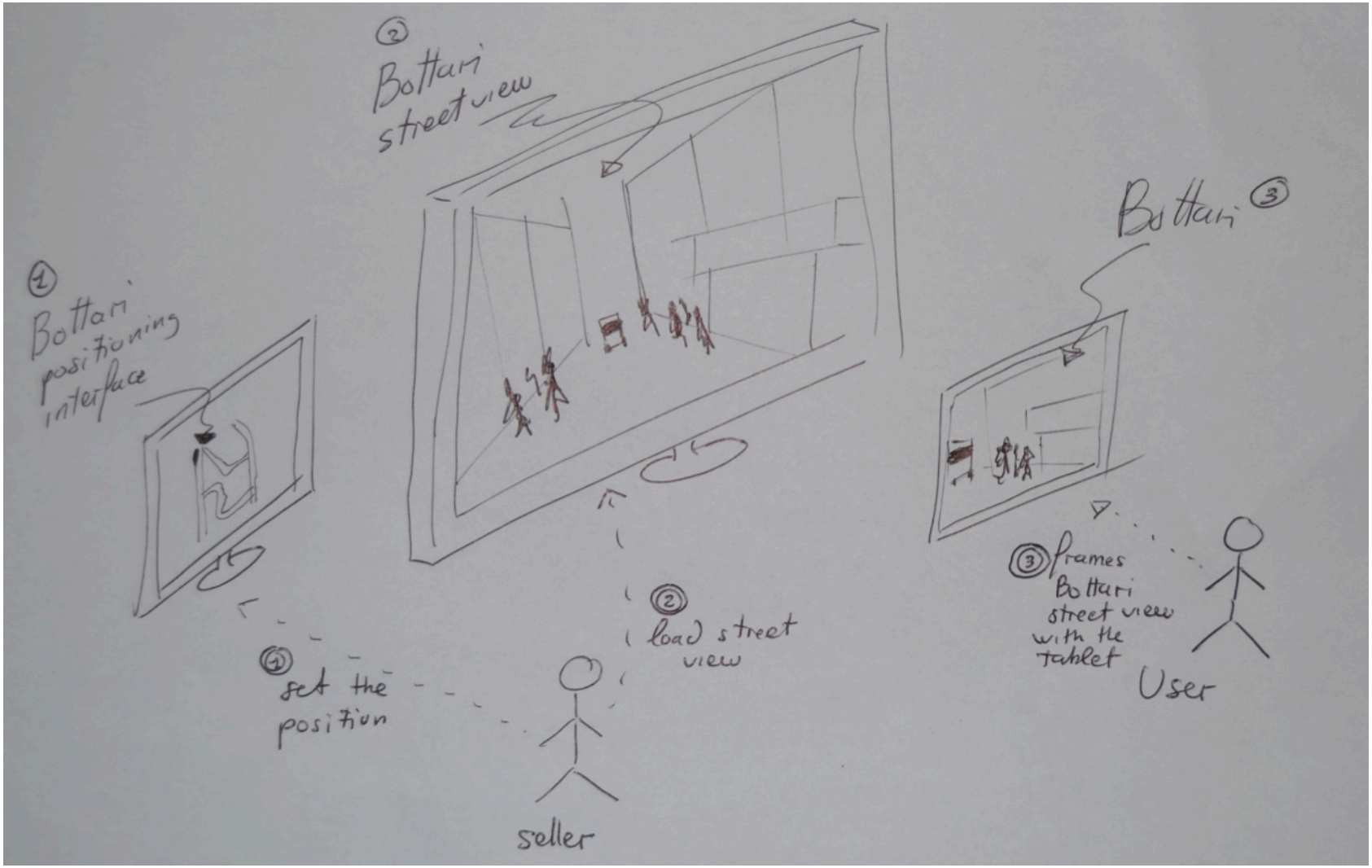
Use Cases

- Picture plus description
- Often misunderstood: the text is more important
- Text structure
  - Use case name and number (ID)
  - Goal
  - Pre-conditions
  - Primary & secondary actors
  - Trigger
  - Description
  - Business rules
  - Open issues
- See also http://alistair.cockburn.us/Use+cases

Storyboards

- Set of drawings depicting user activities
- Paper prototyping
- Drawing screens or processes

Storyboards: e.g., the one of BOTTARI [1]



See also http://streamreasoning.org/demos/bottari

An example of requirements I'm proud of

- A deliverable of my Planet Data project documenting Urban Games Requirements
  - http://wiki.planet-data.eu/uploads/6/62/D9.1.pdf
- See
  - The process in Figure 3.1 at page 20
  - The series of brainstormings in Figure 3.2, 3.3, and 3.4 at page 25, 26 and 27
  - The Use Case diagrams in Figure 4.2 at page 30
  - The Activity Diagrams in Figure 4.3, 4.4, 4.5, 4.6, 4.7 at pages 32-36
  - The Storyboards/GUI mockups in Figure 4.8,4.9,4.10,4.11,4.12,4.13 and 4.14 at pages 37-41

Who?

- Customers and users (note: often not the same)
  - Customers can be users, but rarely opposite
  - Sometimes user constituencies need to be ‘found’
- Subject matter experts
- Other stakeholders
  - Marketing, sales
  - Product managers

Other Tips

- Meetings
  - Treat them like a tool: design them
  - Boy scout motto: “Be Prepared”
  - As small as possible – but no smaller
  - Make it safe not to attend
    - Publish an agenda before
    - Publish a summary after
  - Generate a ‘related issues’ list
  - Can formal/informal, scheduled/ad-hoc

Other Tips

- Manage expectations
  - Don't forget to ask people theirs
  - Listen
  - Make explicit otherwise implicit decisions
    - PDA: Possibility, Deferred, Absolutely impossible
- Technical reviews
  - Requirements can be wrong by: inadequate or inaccurate
    - Quantity & quality
  - Reviews help with the latter

Tools

- Borland Caliber Analyst
  - Collaborative Software Requirements Engineering
  - http://www.borland.com/us/products/caliber/index.html
- IBM Telelogic DOORS
  - Requirements Management for Complex Systems and Software Development
  - http://www-01.ibm.com/software/awdtools/doors/
- Both offer
  - Databases of requirements
  - Displayable in various formats
  - Requirements control metrics:
    - Requirements Stability Index
      - See http://sunset.usc.edu/classes/cs577b_2001/metricsguide/metrics.html#p36
    - To help manage feature creep and ‘vibration’

Optional Readings

- McConnell: 14 "Feature-Set Control”

Questions?