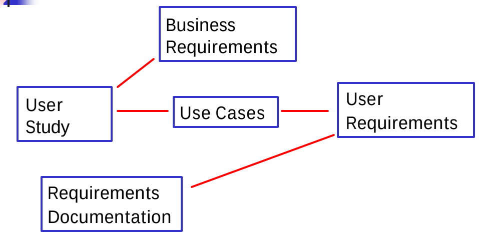
CSE 403, Software Engineering Lecture 4

Documenting and Using Requirements

Today

- Representation of requirements
- Use of requirements
- Key parts of the discussion will (probably) be about process, not artifacts
- Non-functional requirements

User requirements



Who are requirements for?

- Customer
  - To know what they are getting
- Management
  - To know what they are sponsoring
- Marketing
  - To know what they are selling
- Test
  - To know what they are testing
- Dev
  - To know what they are building

Managing the requirements process

- Recognizing requirements documents
- Process for updating requirements
- Tracking process for requirements changes
- Arbitration process for resolving ambiguity and inconsistency

Documenting requirements

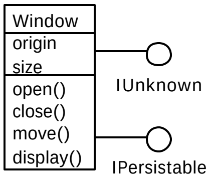
- Multiple representations are probably needed
- Contradictory goals
  - Conciseness vs. completeness
  - Formality vs. comprehensibility

Representations for requirements

- Requirement lists
- Diagrams
  - A picture is worth a thousand words
  - Entity Data Diagrams
  - Data Flow Diagrams
  - State Charts
  - Activity Diagrams

Unified Modeling Language

- UML (1997)
- Object diagrams
- Behavioral Notations
- Diagram notations
  - Use case diagrams
  - Interaction diagrams
  - Activity diagrams
  - Statechart diagrams



Window
origin
size
open()
close()
move()
display()
IUnknown
IPersistable

Approaches to UML

- "Whiteboard UML"
  - Use notation for expository tool
  - Flexible use
  - Partial tool support
- "Formal UML"
  - Substantial tool support
  - Restrictive
  - Assigning semantics very challenging
    - 37 different semantics for Statecharts

UI Mockup

- Advantages
- Disadvantages

Write the user's manual first

- Advantages
- Disadvantages

Redundancy: Good or bad?

- Multiple forms of documentation
  - Used for different audiences or views
  - But risk of inconsistency
- Functional spec and user spec
  - Should describe the same thing!
  - But what if they differ
    - Isn't that Test's job?
- Development principle
  - Avoid computing the same thing in multiple places

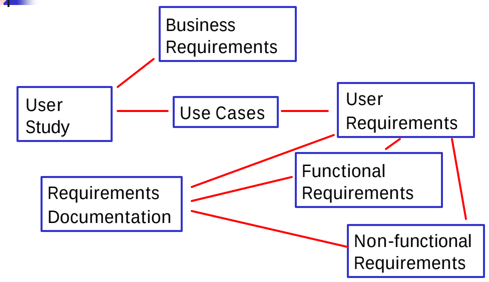
Brooks on flow charts

- The flow chart is the most thoroughly oversold piece of program documentation
- I have never seen an experienced programmer who routinely made detailed flow charts before beginning to write programs. Where organization standards require flow charts, these are invariably done after the fact.

Flow charts

- "The emperor has no clothes"
- Formal processes
  - Many processes are onerous and unpleasant to follow – but enhance overall product quality
  - Some are without value, and should be dropped
  - Process is not the end in itself

User requirements



Non-functional requirements

- Requirements beyond user interaction with the system
- Kulak and Guiney
  - Availability, cost of ownership, maintainability, data integrity, extensibility, functionality (?), installability, reuse, operability, performance, portability, quality, robustness, scalability

Non-functionality requirements

- Wiegers
  - Performance requirements
  - Safety requirements
  - Security requirements
  - Software quality attributes

Safety requirements

- Safety critical applications
  - Where bugs can kill
- Famous cases
  - Therac-25 radiation therapy machine
  - US Air traffic control which failed in UK
    - Reflected map on Greenwich Median
  - US Aviation software failed in Israel
    - Encountered negative altitudes over Dead Sea

Safety critical systems

- Very high cost of failure
- Software component of a large system
  - e.g. nuclear reactor
- Characteristics of software lead to failures
- Safety requirements
  - Low probability of failure (risk analysis)
  - Understood failure modes

Security requirements

- Applications are run in a hostile world
- Application compromise vs. system compromise
- Example requirements
  - Only authenticated users can change data
  - Application can change security permissions or execute programs
  - Malicious user cannot crash system with bad data
- Threat analysis

Security requirements for multiplayer games

- Cheating ruins game play (and consequently market)
- Threats
  - Players introducing counterfeit weapons
  - Sending packet of death across network
  - Using profiling tools to detect areas of activity in dungeons