Requirements Analysis and Specification

- References:
  - G. Kotonya and I. Sommerville, *Requirements Engineering--Processes and Techniques*, John Wiley, 1997.
  - S. Pfleeger and J. Atlee, *Software Engineering--Theory and Practice, Third Edition*, Prentice Hall, 2006, Chapter 4.

Importance of Good Requirements Analysis

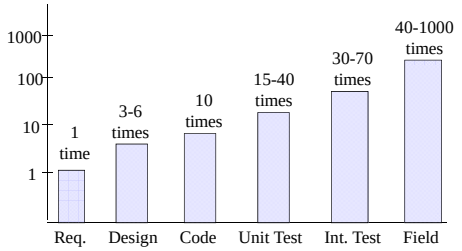
- Standish Group Report, 1995: *The Scope of Software Failures:*
  - Examined 8000 software projects at 350 companies
    - *31% cancelled before completion*
    - *Only 9% of large projects completed at budgeted time and cost*
    - *Only 16% of small projects delivered at budgeted time and cost*

Importance of Requirements Analysis--Continued

- Standish Group Report: Causes of project failure:
  - Incomplete Requirements: 13.1%
  - Lack of user involvement: 12.4%
  - Unrealistic expectations: 9.9%
  - Lack of executive support: 9.3%
  - Changing requirements and specifications: 8.7%
  - Lack of planning: 8.1%
  - System no longer needed: 7.5%

The Case for Careful Requirements Analysis

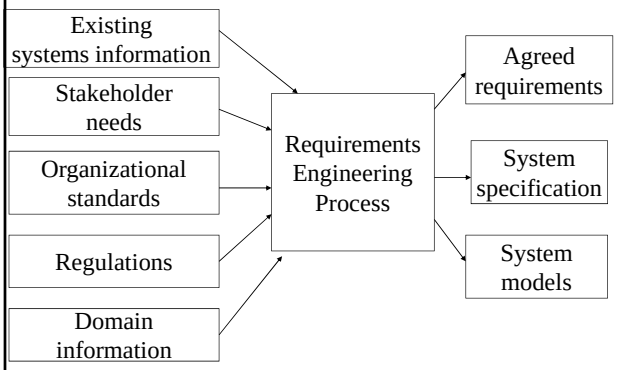
- Boehm and Papaccio, Understanding and controlling software costs, *IEEE Trans. On Software Eng.*, 14(10), October 1988.



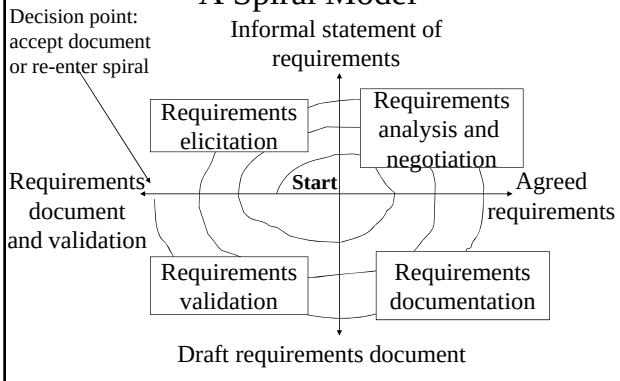
Requirements--What are They?

- A requirement is a description of a system feature, capability, or constraint.
- Requirements generally focus on *what* a system should do, rather than *how* it should do it.
- Classes of Requirements:
  - functional
  - nonfunctional (constraints)
- Requirements Priority
  - essential (“shalls”)
  - highly desirable (“shoulds”)
  - desirable but low priority

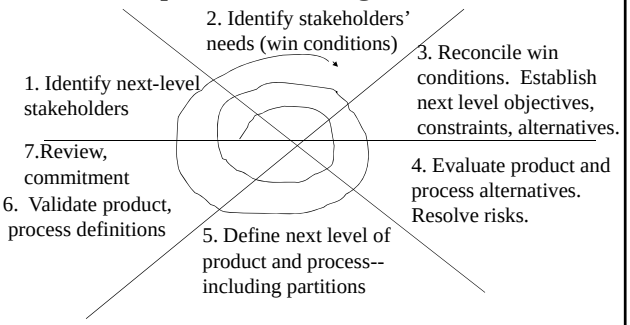
Requirements Engineering



Requirements Engineering Process--A Spiral Model



Boehm’s WinWin Approach to Requirements Negotiation



Stakeholders

- System stakeholders: people or organizations who will be affected by system and therefore have a stake in system requirements
  - end-users
  - managers
  - developers
  - marketers
  - maintainers
  - regulators/certifiers
  - etc.

Stakeholders' Needs

- "Informal" requirements
- Capture desired system properties from the perspective of various stakeholder groups
- Provide a basis for formal requirements

Stakeholders' Needs--Continued

- The intent is to capture stakeholders' general expectations
- No need to be rigorous or precise
- Should aim for completeness—can always scale back expectations later

Why Bother With Stakeholders' Needs Analysis

- Makes sure that needs of all important constituencies have been considered
- Provides a validation check for necessity and completeness of system requirements
  - can map from stakeholders' needs to requirements to check completeness
  - cam map from requirements to stakeholders' needs to check necessity

Stakeholders' Needs Analysis

An example

Stakeholder Needs Example

Own An On-Line Auction House

Your customers will marvel at the

- √ Custom auction categories
- √ Custom news groups and chat rooms
- √ Easy registration
- √ Credit card security
- √ Easy sale submission
- √ Easy bidding
- √ A history of their personal transactions

Auction Administrator

- Enforce fair payment policies
- Collect auction fees
- Track customer profiles and transactions
- Manage auctions, chat rooms and new groups

Auction Administrator

- Check for credit card fraud
- Document your revenue stream
- Add links and advertisements to key web pages
- Profile the popularity of every clickable item on your site
- Recovery software for server failures and power loss

Stakeholders in Both Net Franchise Corp and Auction Owner

- Management leadership
- Sales
- Engineering
- Information systems
- Support services
- Financial
- Legal

Additional

- Customers
  - Sellers of Merchandise
  - Buyers of Merchandise
- Federal, State, and Local Governments

Net Franchise Corp Needs

- ***Management Leadership***
  - Minimize capital investment by manufacturing no hardware
  - Outsource software development to keep workforce costs low
  - Highly responsive customer support staff

Net Franchise Corp Needs

- ***Sales***
  - Demonstrate Net Auctioneer to Customer over the web on a lap top computer
  - Tailor auction type and chat rooms as demonstrations to customer
  - Customer success stories

Net Franchise Corp Needs

- ***Engineering***
  - Well defined extensible software architecture with interfaces that minimize the cost of future changes
  - Both Unix and Windows server compatible
  - Use commercially available and widely supported application frameworks
  - Independence from hardware

Net Franchise Corp Needs

- ***Engineering***
  - SQL database for long term, large data stores
  - Allow for growth in number of customers, number of servers, sizes for databases, and performance
  - Use third party software components when feasible
  - The best development, testing, and component integration tools available

Net Franchise Corp Needs

- ***Information Systems***
  - Access to franchise sites all over the world
  - Franchise data collection, tracking and reporting
  - To know the trends in customers, revenue, and types of auctions for each franchise
  - Produce accounting and billing records for every franchise

Net Franchise Corp Needs

- ***Information Systems***
  - Provide trend and summary reports for all franchises
  - Timely billing for franchise licenses, custom software, training and services
  - Problem reporting system and help desk support

Net Franchise Corp Needs

- ***Support Services***
  - Easy to use interfaces for customer administration of their franchise
  - Problem reporting system with help desk 24 hours a day

Net Franchise Corp Needs

- ***Financial Accounting***
  - Number of franchises active, contracts, data on franchise owners
  - Current and projected revenue from each franchise
  - For each auction, the number of auctions and each gross auction revenue
  - Service and custom software contracts
  - Installation costs for each franchise

Net Franchise Corp Needs

- ***Legal Services***
  - Access contracts and ownership information for each franchise
  - Notification of legal problems
  - History of previous agreements

Net Auctioneer Customers

- ***Management Leadership***
  - 100 times increase in auction revenue
  - High reliability with quick fixes when problems occur
  - Move from manual auctions to on-line auctions
  - Increase auction customer base
  - Minimize installation and service costs
  - Avoid legal entanglements

Net Auctioneer Customers

- ***Auction Administration***
  - Interface with existing databases of buyers and sellers
  - Rapid definition of new auction categories and policies for those auctions
  - Customize the look of the web-pages
  - Change and upgrade the customer interface

Net Auctioneer Customers

- ***Auction Administration***
  - Provide historical search data for the customers
  - Reports on the success of auctions and analysis of what makes them successful
  - Policing of auctions for underhanded sellers
  - Ability to hold money for buyers until transaction is complete
  - Financial reports

Net Auctioneer Customers

- ***Auction Administration***
  - Creation and management of news or chat groups
  - Profile customers
  - Plug -in advertising selection, profiling, and display components
  - Switch database support systems
  - Interface with existing customer and accounts databases
  - Increase the hardware performance for communication, dial-in, and the servers

Net Auctioneer Customers

- ***Sales***
  - On-line tutorials for selling and buying
  - Easy to use interfaces for registration, selling, and purchasing
  - Reliability without misrepresentation

Net Auctioneer Customers

- ***Engineering***
  - Integration with existing databases
  - Use of standard platforms and installation tools
  - Identification of problems that can be fixed locally
  - Help-line from vendor
  - Ability to design web site, add links, add advertising, etc
  - Integrate third party web applications
  - Bring new versions on line seamlessly

Net Auctioneer Customers

- ***Information Systems***
  - Track financial data with audit trails
  - Access history and profiles of customers
  - Produce reports for taxes
  - Project growth of revenues

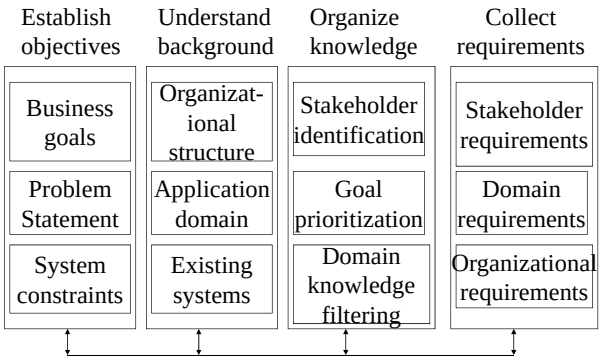
Net Auctioneer Customers

- ***Financial Accounting***
  - Audit auction records
  - Track transactions, fees, and margin from sales
  - Predict net and gross revenues
  - Automatic transfer of financial data to/from Net Franchise Corp

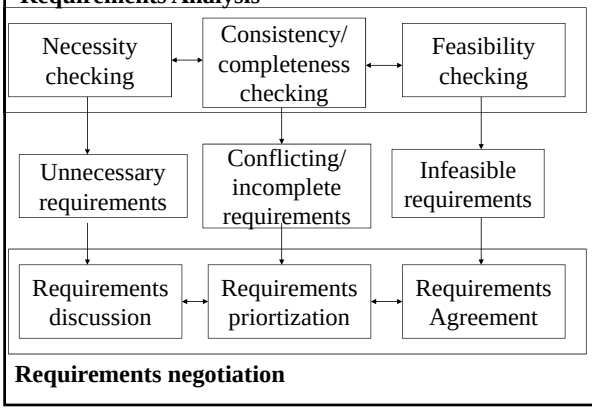
Net Auctioneer Customers

- ***Legal Services***
  - Secure records of sales and purchases
  - Conformity to all federal and state regulations

Eliciting Requirements



Requirements Analysis and Negotiation



Requirements Analysis and Negotiation--Continued

- Analysis Checklist Items From: Kotonya & Sommerville, *Requirements Engineering*, Jon Wiley, 1998
  - premature design
  - combined requirements
  - unnecessary requirements
  - use of non-standard hardware
  - conformance with business goals
  - ambiguity
  - realism
  - testability

Analyzing Requirements--Continued

- Requirements Checklist Items From: Pfleeger, *Software Engineering Theory and Practice*, Prentice hall, 1998.
  - Correct?
  - Consistent?
  - Complete?
  - Realistic?
  - Needed?
  - Verifiable?
  - Traceable?

Requirements Analysis Example:

- Consider the following requirements:
  - The system shall provide real-time response to queries.
  - Accuracy shall be sufficient to support mission planning
  - The system shall maintain records of all library materials, including books, newspapers and magazines, and CDs. The system shall allow users to search for an item by author, title, or ISBN.
- What is wrong with each of these?

Documenting Requirements

- Requirements Documents go by various names:
  - requirements document
  - specification document
  - system requirements specification (SRS)
- Sometimes there will be two documents
  - requirements definition--customer-oriented
  - requirements specification--developer oriented

Requirements Documentation--Continued

- Elements of a typical requirements document
  - Preface/Introduction
    - intended purpose and audience
    - product background
    - problem statement and rationale
    - context
  - Glossary
    - definition of technical terms
    - abbreviations and acronyms

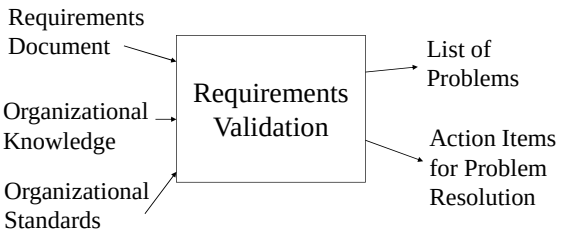
Requirements Documentation--Continued

- Elements of a Requirements Document--Continued
  - Functional requirements—May be organized by Stakeholder
  - Nonfunctional requirements– Reliability, performance, etc
  - System Architecture (we will do this in a separate document)
  - Hardware Specification (We will leave this to the architecture document)
  - Mapping to/from Stakeholder Needs
  - Appendices
  - Index

Requirements Validation

- Validation is the process of establishing that the requirements specification is is accurate, consistent, and complete with respect to the stakeholders' needs.
- Note that it is not possible to formally validate a requirements specification with respect to the stakeholders' expectations
- The most common form of requirements validation is a Requirements Review.
  - Formal meeting similar to a walkthrough
  - review team made up of various stakeholders

Requirements Validation--Inputs and Outputs



Informal Requirements Validation

- General problems to look for
  - incompleteness
  - ambiguity
  - violation of standards
  - redundancy
  - inconsistency (conflict)
  - traceability
    - is this requirement traceable to one or more stakeholder needs
  - testability
    - in most cases it is a good idea to define tests as part of the requirements generation process

Requirements Validation--Continued

- Some types of requirements are difficult to validate
  - General system requirements
  - Exclusive requirements
    - exclude some specific type of behavior
    - e.g. system failure shall never corrupt the database
  - Nonfunctional requirements
    - e.g. reliability
- These should be identified up front, and validation criteria should be negotiated with stakeholders

Requirements--Common Problems

- Failure to completely capture functional requirements
  - transfer of control
  - ownership
  - window resizing
  - tracking of mouse movement
- Failure to define terms

Requirements--Common Problems (continued)

- Vagueness
  - The system shall be reliable.
  - The code shall be easily readable.
  - The system shall support remote access
  - The system shall allow the user to store and open files.

Requirements--Common Problems (Continued)

- Overly constraining requirements
  - The system shall be platform-independent
  - The system shall update the display within one second.
- Failure to prioritize Requirements

Requirements Engineering--Feature Set Control

- Feature creep is one of the most common causes of schedule and cost overrun and/or project failure.
- Feature creep is insidious and hard to prevent.
  - All stakeholders, including development team, may contribute to the problem.
  - Control requires discipline and effective management.

Feature Set Control

- Early project control
  - define feature set consistent with project budget and schedule
- Mid-project control
  - avoid feature creep
- Late project control
  - trim or delay lower priority features to meet schedule or budget.

Feature Set Control--Early Project

- Minimal Specification
- Requirements Scrubbing
- Versioned Development

Feature Set Control--Early Project

- Minimal Specification
  - Find the proper level of detail that captures the essential requirements without overspecifying the system.
  - examples of overspecification:
    - specifying details or characteristics that are unimportant to the user/customer.
    - Specifying features or technologies that are known to be subject to change.
    - Overly constraining design.
  - It is essential that the goals and objectives of the project are made crystal-clear.

Minimal Specification--Some Suggestions

- Start with a vision statement that describes both what the product **is** and **is not**.
- Consider using a User Manual or On-line Help system as the spec.
- Consider using GUI prototypes or storyboards in lieu of (or in addition to) prose specification.
- Make sure developers clearly understand the goals and objectives of the project.

Minimal Specification--McConnell's Keys to Success

- Use only when requirements are flexible.
  - Note that keeping the initial requirements flexible may itself be a key to success in some projects.
- Don't use as an excuse for not doing careful requirements analysis
- r.e. specification items: "When in doubt, leave it out".
- Make sure that you DO capture the essential requirements.
- Use with flexible development approach.

Feature Set Control--Early Project

- Requirements Scrubbing:
  - Eliminate all requirements that are not absolutely essential.
  - Simplify overly complicated requirements.
  - Substitute cheaper options where possible.
- Be careful. Remember that reinstating requirements at a later development stage will cost much more.

Feature Set Control--Early Project

- Versioned Development
  - Defer some requirements to later versions.
  - Use evolutionary or staged development process.
- Success of this approach requires a strong architectural perspective that accommodates the addition of deferred features.

Feature Set Control--Mid-Project

- Controlling feature creep
  - Having "lean-and-mean" requirements doesn't protect against feature creep.
  - All stakeholders are potential contributors to this problem.
- Some common sources of feature creep:
  - "Killer-app" syndrome
  - Vague, overly-ambitious goals
  - Gold-plating of unimportant details

Controlling Feature Creep

- In most projects, some volatility of requirements is inevitable.
  - Customers don't know what they want
  - Markets change rapidly.
  - Technology changes rapidly
  - Developers may need latitude-c.f. earlier discussion of Minimal Specification
  - Errors or incompleteness may be found in the specification.

Classes of Requirements Volatility

From: Kotonya and Sommerville, *Requirements Engineering*, John Wiley, 1997

- Mutable Requirements
  - subject to changes in environment
  - e.g. tax software subject to changes in tax laws.
- Emergent Requirements
  - Cannot be completely defined at specification time.
- Consequential Requirements
  - based upon usage assumptions that may be wrong.
- Compatibility Requirements
  - depend on other systems that may change.

Controlling Requirements Changes

- Plan for change--have a Change-Management Plan
- Some goals of a Change-Management Plan:
  - Critically assess proposed changes w.r.t. impact on schedule, budget, and product.
  - Involve all affected stakeholders in the decision-making process.
  - Provide an audit trail of requirements change decisions
- A *Change-Control Board* may be effective.

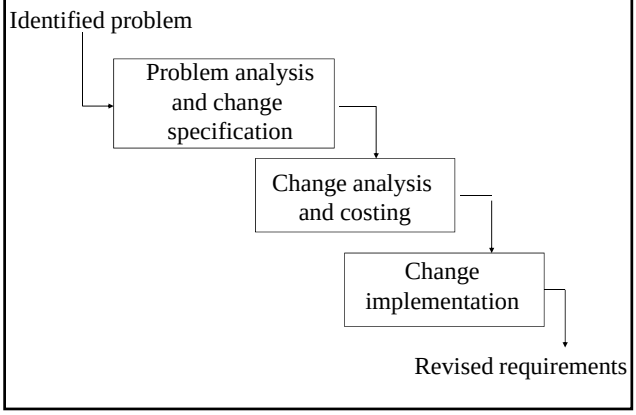
Feature Set Control--Late Project

- Eliminating low-priority features is often a good way of managing schedule and budget risks.
- This requires good up-front requirements priortization
- Good approach:
  - Start with scrubbed requirements + prioritized list of additional features
  - Add new features as time an budget permit.

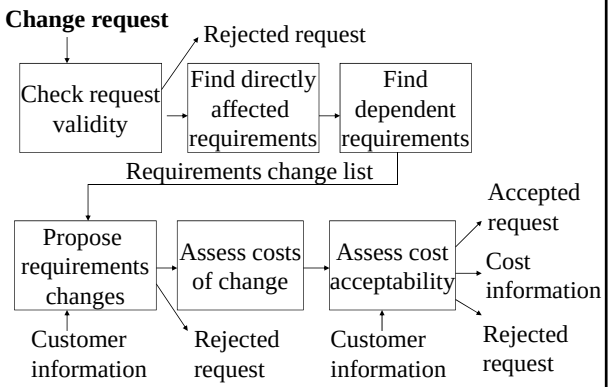
Requirements Management

- Principal elements:
  - managing changes to agreed-upon requirements
  - managing interdependencies among requirements
  - managing dependencies between requirements document and other documents produced during the development process (traceability)

Managing Changes to Requirements



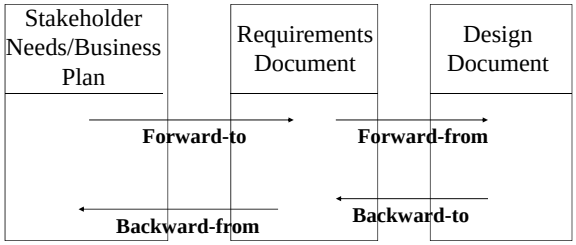
Change Analysis and Costing



Tracing Requirements

- Four types of traceability
  - Backward-from
    - links requirements to stakeholder needs
  - Forward-from
    - links requirements to design and implementation components
  - Backward-to
    - links design and implementation components back to requirements.
  - Forward-to
    - links stakeholder needs to relevant requirements

Backward and Forward Traceability



Other Types of Traceability

- Requirements-to-requirements
  - links dependent requirements
  - two way: dependent and is-dependent-on
- Requirements-to/from-architecture
- Requirements-to/from interfaces

Keeping Track of Traceability Relationships

- Traceability tables
  - record relationships in matrix form
  - generally practical only for simple systems
- Case-tools
  - automated change request process
  - requirements database with change management process.