Requirements Analysis
   Determining WHAT the software is to do

Passages
   • Inception
   • Initial development
     – Planning
     – Analysis
     – Requirements Engineering
   • Productive operation
   • Upkeep
   • Retirement

Planning
   • Scope
   • Involve stakeholders
   • Estimate based on knowns
   • Consider Risks
   • Define how to measure quality
   • Determine how to manage change

Purpose of Analysis
   • Definition of what the product must be capable of doing, under what constraints the product will have to work and a plan for developing the product

Analysis Involves
   • Understanding what customer wants
   • Assessing feasibility
   • Negotiating an unambiguous reasonable solution
   • Validating the specification
   • Managing the requirements as they become a system

What are Requirements
   • A statement of a system service or constraint
• IEEE glossary
  – (1) a condition or capability needed by a user to solve a problem or achieve an objective
  – (2) the set of all requirements forms the basis of subsequent development of the system or system component
• Abbott, An integrated approach to software development
  – Any function, constraint, or any other property that must be provided, met, or satisfied to fill the needs of the systems users

**Outputs/Deliverables**
• Requirements document
  – Definition and specification
  – A description of the world into which the intended system must fits and the goals it is to meet

**Brooks: No silver Bullet**
• “The hardest part of building software is deciding what to build…. No other part of the work so cripples the resulting system if done wrong, No other part is more difficult to rectify later”

**Requirements Cover**
• 15% of the total development costs

**Functional Aspects:** What functions the system must be capable of:

*What problem the "user" wants the systems to solve.
*Why the "user" wants the system.
*How the "user" intends to use the system.
*Information the system will have to handle.
**"Data Processing" functions to be performed.

**Non-functional Aspects:** Performance, reliability, security, maintainability ..... 

*other systems & procedures the software will have to interface with.
*Legal constraint ,, privacy issues ... record retention.
*Integrity/security issues & restrictions & access limitations.
*Technical expertise of the "users"
*Expected enhancements
Inverse Conditions:

What the system shall not do:
- constraints on allowable behavior
- software safety

- access limitations
- legal issues
- privacy issues

Design and Implementation Issues:

Boundary conditions on construction and implementation...existing hardware/software usage

- Hardware
- Programming Languages
- Operating System
- Physical constraints---NO AC
- Peripheral devices?
- Expected Load-- simultaneous "Users" / transactions per hour

System Evolution:
- lifecycle model
- Includes how will the system be "installed"

Requirements <-> Design
- Basis and guidelines of design
- Blueprint

Requirements <-> Testing
- Provide a basis
- Acceptance test plan

Requirements <-> Maintenance
- Provide insight through documentation to the impetus for the product, changes that are applicable/not
Steps
• Elicitation
  – Identifying and asking stakeholders their objectives etc
• Analysis and negotiation
  – Categorizing and organizing the requirements
  – Checking for consistency, commonalities, conflicts
  – Identify risks
• Representation/specification
  – Document according to a standard
• Validation
  – Building the right product

Cyclic, iterative, and cooperative process

Requirements Management
• Changes will occur throughout development
• Software Configuration Management
  – Identify, control, and track changes to requirements
  – Unique identifier
  – Traceability tables
    • Used to determine if a change in one requirement will affect different aspects of the system being built

Products of Phase
• User oriented
  – Definition
  – Prototype or user’s manual
  – Acceptance test plan
• Developer oriented
  – Specification
  – Acceptance test plan

Participants
• Developers
• All system stakeholders
  – Direct or indirect contact
Objectives of the Phase
- Achieve agreements on requirements
- Provide basis for design
- Provide a reference point for validation

Metrics for the requirements Phase
- Requirements volatility
  - How frequently the requirements change
    - During requirements phase
    - Rest of the development process
      - Initiated by clients or developers