# 605.601 Foundations of Software Engineering
Fall 2020

## Module 03: Requirements

Dr. Tushar K. Hazra
tkhazra@gmail.com
(443)540-2230

## 605.601 Foundations of Software Engineering
## Course Module 03: Requirements

- Definitions
- Characteristics
- Documentation
- Engineering and Management

## 605.601 Foundations of Software Engineering
## Course Module 03: Requirements

- The hardest single part of building a software system is deciding precisely what to build.
- No other part of the conceptual work is so difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems
- No other part of the work so cripples the resulting system if done wrong.
- No other part is more difficult to rectify later.

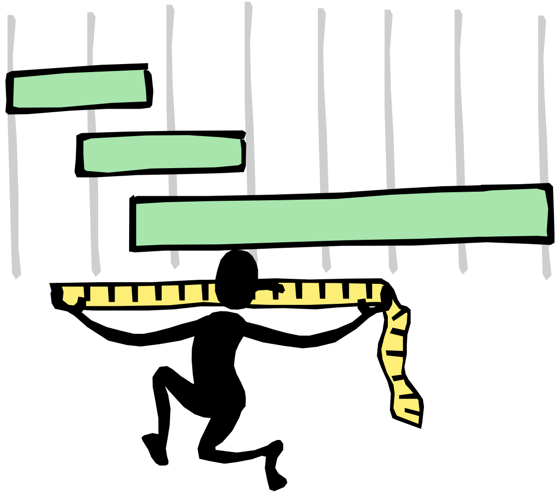
[Brooks, 1986]

## Requirements: Definition

- What is a requirement?
  - A condition or capability needed by a user to solve a problem or achieve an objective
  - A condition or capability that must be fulfilled or possessed by a system to satisfy a contract, standard, or other formally imposed document
  - A specific statement of a function to be provided in a system:
    - A result (e.g., product of the analysis of data)
    - An action (e.g., send a command to the hardware controller)
    - A capacity (e.g., store all ephemeris data for 30 days)
    - A constraint (e.g., process all data in 30 seconds)
    - or any combination of these
- What a requirement is not:
  - The philosophy of the system design is not a requirement
  - The system design approach is not a requirement

# Requirements Cont'd

- Purpose of requirements:
  - Identify to the customer/user what the element will do
  - Provide guidance to the designer/implementor on what to build
  - Provide information to the tester on what to test
- Requirements changes & growth
  - How they come about:
    - Vague requirements subject to interpretation
    - Omissions
    - Human nature
  - What to do about them:
    - The crux of requirements management

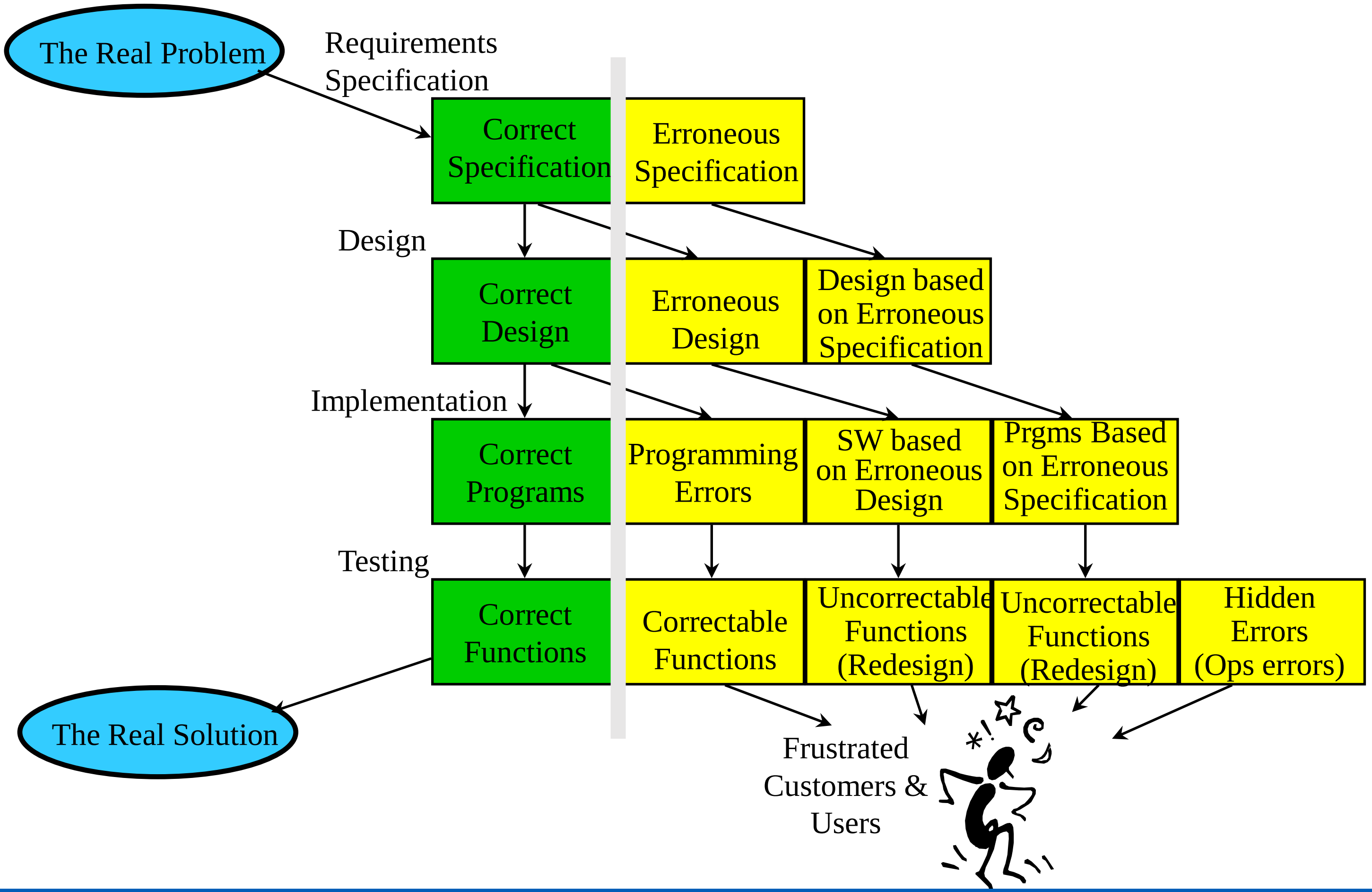


"We have to accept changing requirements as a fact of life, and not condemn them as a product of sloppy thinking"…

…Gerhard Fischer, *IEEE Software, January, 1989*

# Requirements Cont'd

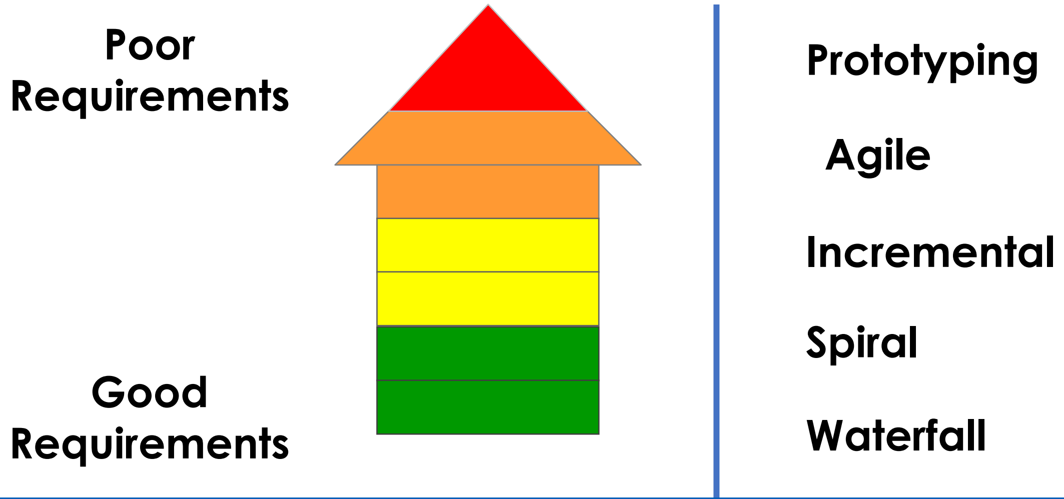
## Consequences of Poor Requirements Process


The Real Problem
Requirements Specification
Correct Specification
Erroneous Specification
Design
Correct Design
Erroneous Design
Design based on Erroneous Specification
Implementation
Correct Programs
Programming Errors
SW based on Erroneous Design
Prgms Based on Erroneous Specification
Testing
Correct Functions
Correctable Functions
Uncorrectable Functions (Redesign)
Uncorrectable Functions (Redesign)
Hidden Errors (Ops errors)
The Real Solution
Frustrated Customers & Users

# Requirements Cont'd

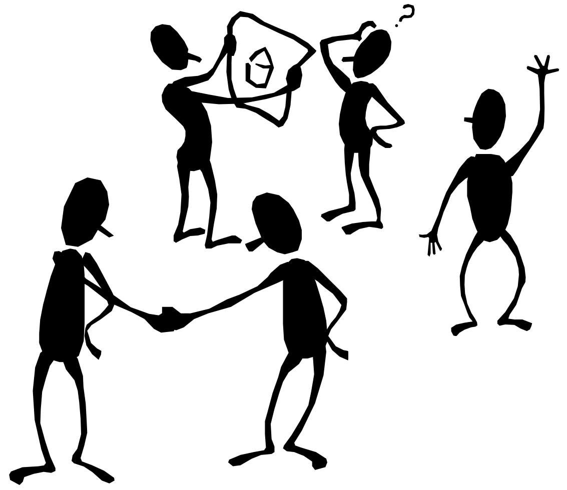
- Successful estimation is directly related to requirements definition and requirements management
  - Allocated requirements are the basis for software size estimates
  - Software size estimates are the basis for effort and schedule estimates
  - Effort and schedule estimates drive plans, products, and activities
  - Changes to requirements impact original estimates and must be controlled and negotiated - size, effort, and schedule must be re-estimated

– Specificity of requirements is an important criteria for selection of the development process ( and indirectly development cost)



## Requirements Cont'd

- Integrated Product Development (IPD) - Is this the answer?
  - Central concept: the product belongs to all
    - customer
    - user
    - program manager
    - systems engineers
    - hardware engineers
    - software engineers
    - support groups



– If all the stakeholders really believe the product is theirs, then they will not let requirements problems ruin the product or program

## Requirements Cont'd

- What makes requirements specific or vague?
    - A requirement is vague when it contains words such as: often, frequently, occasionally, sometimes, as needed, as appropriate, big, small, large, little
    - A requirement is vague when it does not specify what data a given operation is to be performed on, or how often, or to what precision
    - A requirement is specific when it specifies the type of data to be processed, the algorithms to be used, and time constraints involved
- How to make requirements specific
    - Produce a sufficient number of derived requirements to be specific
    - but….this will impose limitations on the capability of the system

## Definitions: Requirements, Specification, and Design

### Requirement

- An expression of desired behavior
- Description of real-world phenomena without referencing the proposed system
- Not *how* the system will be implemented (unless mandated by the customer)

### Specification

- The identification of what *requirements* will be fulfilled by software

### Design

- A plan for how specific behavior will be implemented

# Requirements

## Functional Requirement

- A functional requirement describes the required behavior in terms of required activities and the state of each entity before  and after an activity

## Requirements

### Non-Functional Requirement

- A non-functional requirement is a  characteristic that the system must possess
  - May seem like "motherhood" characteristics that all systems should possess
- Example
  - Fast response time
  - High reliability
  - Ease of use
  - Low maintenance cost
- Think of quality requirements as design criteria to be optimized

## Requirements: Constraints

Design

- A design constraint is a design decision that restricts the set of solutions to the problem

Process

- A process constraint is a restriction of the techniques or resources that can be used

Fit Criteria

- Fit criteria quantify the extent to which each requirement must be met
  - Provide objective standards to judge if a proposed solution satisfies the requirement

## Requirements: Characteristics I

### Consistency

- Two requirements are inconsistent if it is impossible to satisfy  both simultaneously

### Ambiguity

- Requirements are ambiguous if they allow multiple different, but valid, interpretations

### Testable

- Requirements are testable if they allow an acceptance test that clearly demonstrates whether the product satisfies them

## Requirements: Characteristics II

### Traceability

- Requirements should be traceable with every requirement organized an labeled for reference with corresponding entries in the *requirements definition* and *requirements specification*

### Completeness

- Requirements are complete if they specify the required behavior and output for all possible inputs in all possible states of the system under all possible constraints
  - Requirements are externally complete if all states, state changes, inputs, products, and constraints are described
  - Requirements are internally complete if there are no undefined terms

## Requirements: Documentation

A record of the requirements expressed in the customer's terms

- What should it include?
  1. General purpose and scope—relevant benefits, objectives, and goals
  2. Background and rationale for a new system
     - Must explain why the existing approach is unsatisfactory
  3. Essential characteristics of an acceptable system
  4. Description of solution proposed by customer (if applicable)
     - Note: The purpose of requirements documentation is the *problem*, not the *solution*!
  5. Assumptions about the environment

## Requirements: Specification

A record of the requirements expressed from the perspective of developers

Content

- System interface—sources of input, destination of outputs,  protocols, etc.
- Requirement functionality in terms of interfaces' inputs and outputs
- Fit criteria for non-functional or *quality requirements*

## Requirements Gathering

Development of Problem Statement

Problem Statement establishes. . .

- a basic understanding of the problem
- the people who want a solution,
- the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and  collaboration between the customer and the developer

## Requirements Gathering

Development of Problem Statement: Tasks involved

- Identify stakeholders
  - .., “Who else do you think I should talk to?”
- Recognize multiple points of view
- Work toward collaboration
- The first questions
  - .., Who is behind the request for this work?
  - .., Who will use the solution?
  - .., What will be the economic benefit of a successful solution?
  - .., Is there another source for the solution that you need?

## Requirements Gathering

Development of Problem Statement

Problem Statement establishes. . .

- a basic understanding of the problem
- the people who want a solution,
- the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and  collaboration between the customer and the developer

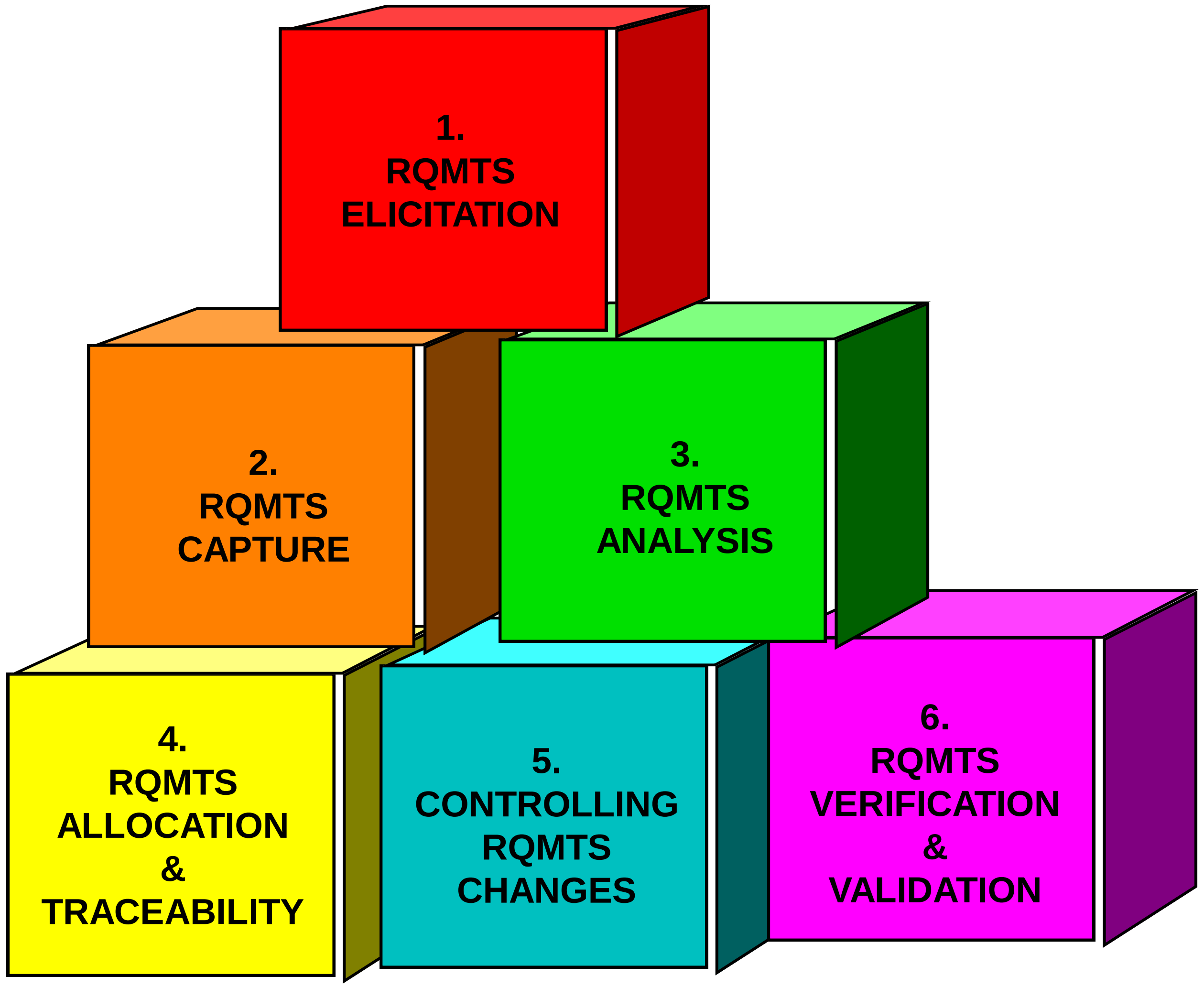
## Requirements Gathering

Development of Problem Statement

Problem Statement covers. . . 5 ‘W’s and 1 ‘H’

- Who – uses the software
- What – users need the software to do
- When – is the software needed
- Where – will the software be used
- Why – users need the software
- How – users want to use the software

## Requirements Engineering & Management



## Requirements Engineering & Management Cont'd

- Requirements Elicitation
  - Extracts or derives requirements from original sources
    - Stakeholders
    - Textual documents
    - Other formats (e.g., tables, charts, graphics)
  - Provides a set of requirements that reflects customer needs
  - Differentiates between requirements and their supporting information
    - Defines the scope of the problem
    - Prevents over or under design
    - Prevents unnecessary design constraints
  - Majority of system requirements are documented in English text via requirements specification with accompanying concept of operation (spec is easy to structure, test, trace; conop is easy to read and understand)

# Requirements Engineering & Management Cont'd

- Requirements Elicitation Checklist

**Source Documents**

- 🗁 Customer needs statements
- 🗁 Market studies
- 🗁 Lessons learned
- 🗁 Regulations
- 🗁 White papers
- 🗁 Standards
- 🗁 Early system conops
- 🗁 Product visions
- 🗁 Top level rqmts documents

**Vendors/subcontractors/partners**

- 🗁 Marketing discussions
- 🗁 Product briefings
- 🗁 Teaming discussions

**Related Systems**

- 🗁 Similar systems
- 🗁 Predecessor systems
- 🗁 Prototypes
- 🗁 Existing subsystems
- 🗁 Interfacing systems

**Relevant Experts**

- 🗁 System subject matter
- 🗁 Research & development
- 🗁 Supporting disciplines
- 🗁 Universities

**Stakeholders**

- 🗁 Interview
- 🗁 Focus groups
- 🗁 Surveys
- 🗁 Brainstorming

**Future Changes**

- 🗁 Technology
- 🗁 Procedures
- 🗁 Environment
- 🗁 Processes

## Requirements Engineering & Management Cont'd

- Requirements Capture - Objectives
  - Documents customer needs
    - Provides a vehicle for communication between customers & system developers
    - Starting point for all activities in the development process
  - Formally represents requirements
    - English language is imprecise
    - Minimize conflicts between various domain experts
    - Distinguish between requirements & constraints
  - Provides a well structured representation
    - Enhance requirements analysis
    - Enable requirements traceability
    - Enhance testing

## Requirements Engineering & Management Cont'd

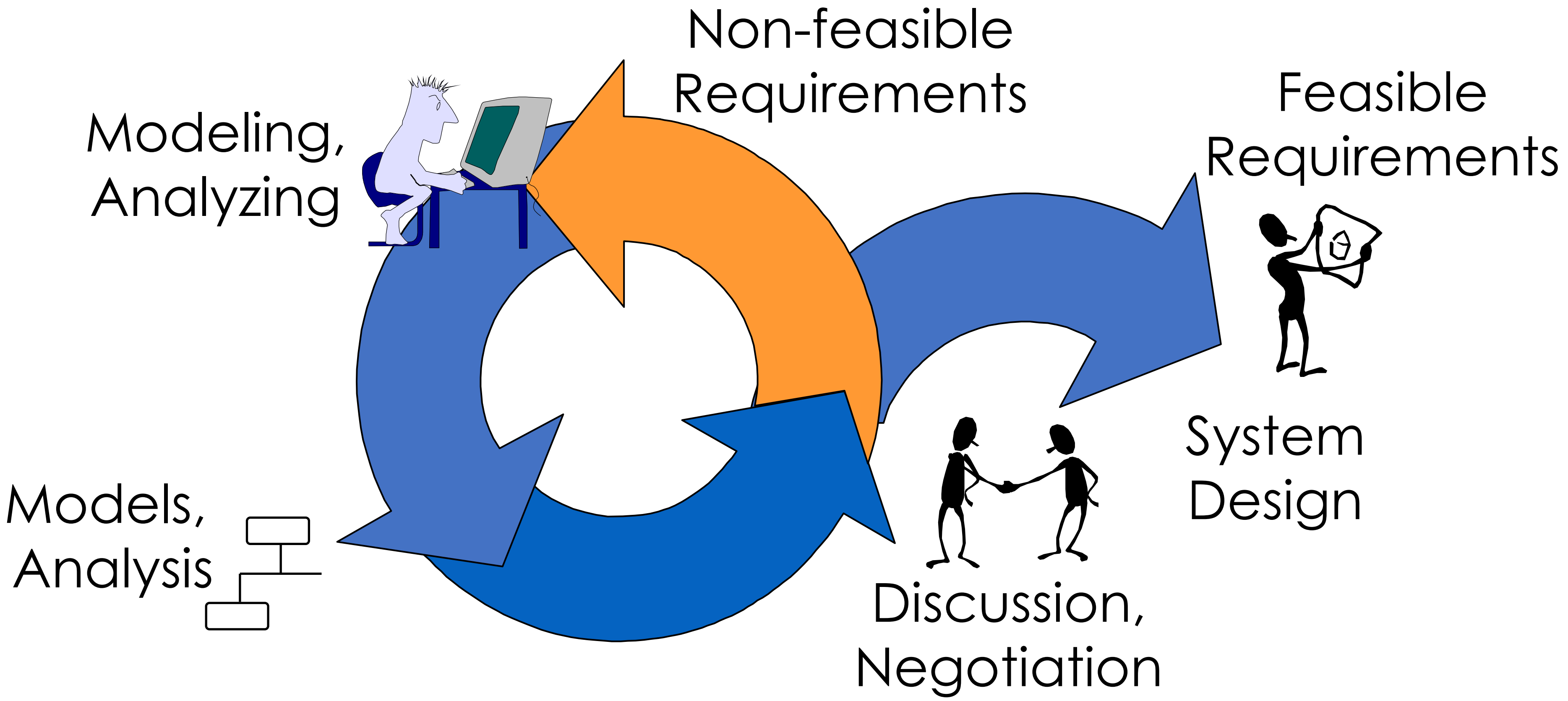
- Requirements Capture - Structure
  - A robust organization of requirements
    - Useful for all activities in the development process
    - Easy to access
    - Automatable (e.g., RTM, DOORS)
  - Include all requirements elements
    - Functional (display the cartesian coordinates)
    - Performance ( respond within 30 seconds)
    - Capacity (store up to 30 days of historical data)
    - Implementation (interfaces, resource constraints)
    - Verification & validation (inspection, analysis, etc.)
    - Tests (integration, system, DT&E, etc.)
    - Priority

## Requirements Engineering & Management Cont'd

- Requirements Analysis - Objectives
  - Provides better understanding of requirements
    - Understand customer's needs and constraints
    - Understand system interfaces
    - Establish feasible design constraints
    - Minimizes redesign and re-implementation
    - Use of models converts English text to graphical or tabular representation
  - Supports requirements allocation
    - Provides initial guidance for allocation
    - Rapidly focuses the design
  - Prioritize requirements
    - Identifies required critical factors
    - Assists with build planning & prototyping objectives

# Requirements Engineering & Management Cont'd

- Requirements analysis & modeling (producing a representation or simulation of the logical system)



# Requirements Engineering & Management Cont'd

- Requirements analysis - Modeling Methods

| Tools/Modeling Methods | |
|---|---|
| Entity relationship diagramming | State transition diagramming (STD) |
| Object diagramming | State transition matrices (STM) |
| Functional flow diagramming | Decision table & trees |
| Data flow diagramming | Petri-nets |
| N-square diagramming | Specification & description language |
| Decomposition diagramming | State charts |
| Schematic block diagramming | Process-oriented, Applicable, and Interpretable Specification Language (PAISLey) |

## Requirements Engineering & Management Cont’d

- Requirements allocation & traceability
  - Objective is to ensure that the products meet their requirements at every instance of the development process
  - Enables the development organization to effectively isolate the impact of requirements, design, and implementation changes
  - Each requirement must be allocated to at least one design element
  - Each design element must be traced back to at least one requirement (either an original or a derived requirement)
  - Rationale for requirement allocation must be documented
  - Any change of requirement allocation must be managed
    - Justify
    - Analyze
    - Authorize
    - Trace to design

# Requirements Engineering & Management Cont’d

- Requirements allocation & traceability cont’d
  - Traceability rules
    - Linkages must be established to maintain requirement traceability
      - Primary requirements to derived requirements
      - Requirements to design elements
      - Requirements to implementation
      - Requirements to test
      - Design elements to implementation
      - Design elements to test
    - Linkages and objects should have associated attributes
      - e.g., active or retired, build allocation, priority, etc.
      - To better understand the relationships among objects
      - To support requirements management
    - Automation through COTS tools can support
      - Generation and maintenance of linkages
      - Creation and maintenance of attributes
      - Forward and backward traceability

# Requirements Engineering & Management Cont'd

- Manage requirements elicitation
  - Develop guidelines for requirements elicitation
    - Who is authorized to elicit requirements?
    - How are requirements elicited?
    - When is requirements elicitation conducted (when does it stop)?
  - Identify standard mechanism
    - What questions should be asked
    - What tools should be used
- Manage requirements capture
  - Define policy for baseline requirement
  - Define policy for modifying requirement
  - Identify responsible personnel
  - Assign unique requirement identifications
  - Explicitly identify derived requirements & maintain traceability

## Requirements Engineering & Management Cont'd

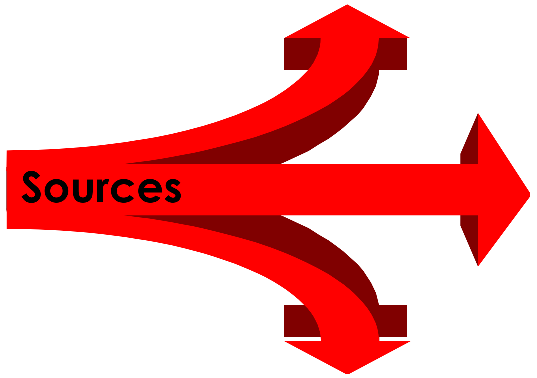
- Manage requirements analysis
  - Identify the appropriate type of analysis
  - Select specific analysis methods & tools
  - Identify sources of information for analysis
  - Control the transition of analysis results into design
  - Maintain traceability between requirements and analysis results
- Importance of baselining requirements
  - Provides a controlled set of requirements for design
  - Reflects common understanding between developer/customer as to what the system will do at a particular instance in the development process
  - Changes can be made with proper procedure
    - Approval from customer
    - Documented changes and rationale for change
    - Traceability

## Requirements Engineering & Management Cont'd

- Controlling Requirements Changes - Sources of Changes

User/customer learning process

- Awareness of possibilities
- Understanding implementation issues
- Training and education
- Familiarity with technology
- Growth in user community
- New & perceived problems & threats



Environmental Changes

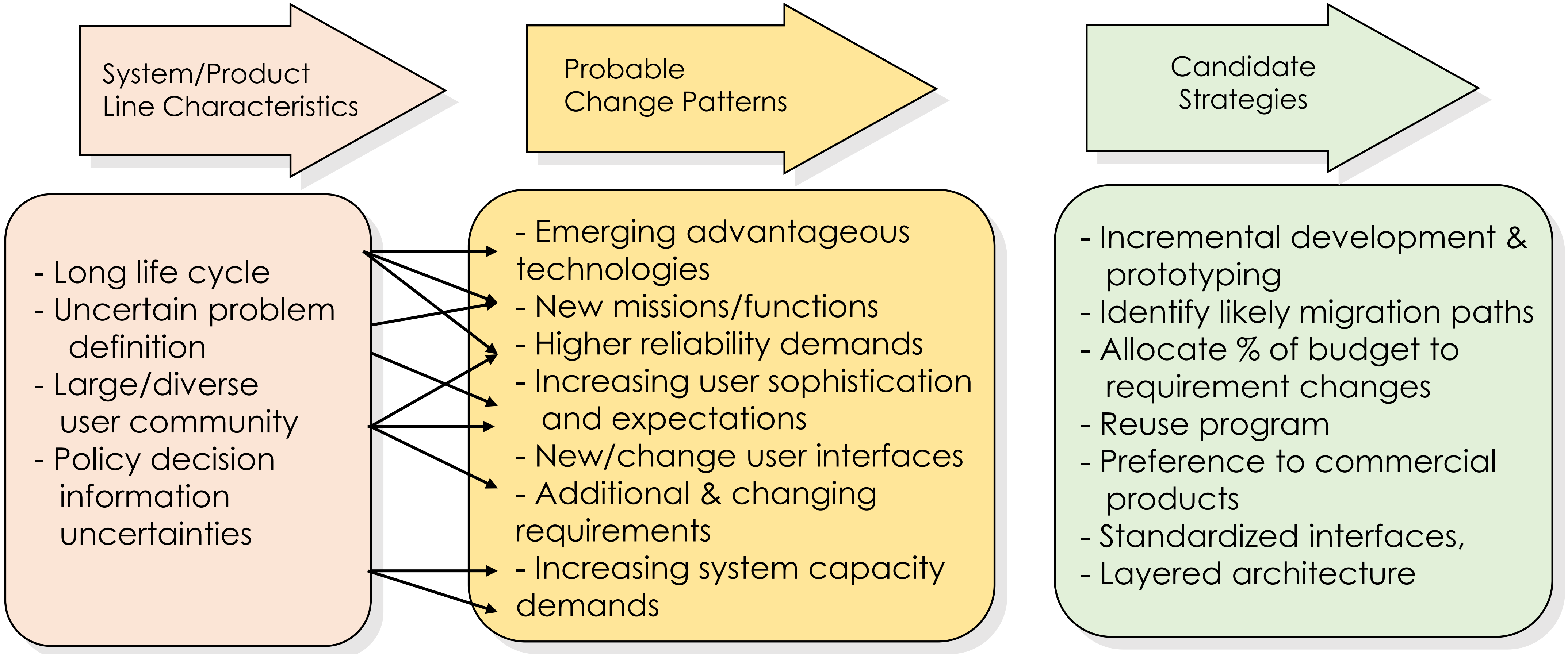
- Interfaces & interfacing systems
- System scope & responsibilities
- Organizational policies/procedures
- Regulation
- Systems throughput meeds
- Competitive environment
- Funding

Technologies Evolution

- Material & fabrication
- Sensors & actuators
- Computer platforms
- Storage capacity & throughput
- Computer-human interfaces
- Communications performance
- Algorithms & applications
- Tools
- Network topology, complexity, and management
- Standards maturity

# Requirements Engineering & Management Cont'd

- Controlling Requirements Changes - Dealing with change & uncertainties



System/Product Line Characteristics

- Long life cycle
- Uncertain problem definition
- Large/diverse user community
- Policy decision information uncertainties

Probable Change Patterns

- Emerging advantageous technologies
- New missions/functions
- Higher reliability demands
- Increasing user sophistication and expectations
- New/change user interfaces
- Additional & changing requirements
- Increasing system capacity demands

Candidate Strategies

- Incremental development & prototyping
- Identify likely migration paths
- Allocate % of budget to requirement changes
- Reuse program
- Preference to commercial products
- Standardized interfaces,
- Layered architecture

## Requirements Engineering & Management Cont'd

- Controlling requirements changes
  - Evaluate the impact
  - Do whatever has little or no impact
  - Let the customer decide: impact the cost and/or the schedule of the current build or hold the change for the next (or later) build
  - Plan increments for the least amount of time possible (so that waiting for the next build is not intolerable)
  - If the decision is to increase cost/schedule, then ask for formal notification through contracting office
  - Make sure that:
    - Plans, documentation, activities, and products are revised to agree with requirements changes
    - Records and traceability of such changes are maintained by configuration management
    - Re-negotiate commitments with groups affected by requirements changes

# Requirements Engineering & Management Cont'd

- Requirements verification & validation
  - Validation: *Am I building the right thing?*
    - The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. Requirements validation is done against either the real world user needs or system level requirements.
  - Verification: *Did I build what I said I was going to build?*
    - The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Products are typically verified against lower level requirements, design, or test procedures.
  - Requirements should be verified & validated against stakeholder needs obtained through
    - Review & walkthrough
    - Demonstrations
    - Scenario reviews
    - Prototyping
    - Formal reviews and inspections
    - Sign-off

## Requirements Engineering & Management Cont'd

- How would you like to communicate with the user of your software once you collected the requirements?
- How would you present the requirement(s) to the user and validates?
  - A visual Model
  - A Diagram
  - A Model