

# Requirements Analysis

---

## Outline

Analysis of “Analysis”

Content of RS Document

Specificity & Flexibility

Requirements Collection

Presentation

Checklists - Process

Checklists - Planning and Management

Checklist - Loose Ends

# Requirements Analysis: Product Specifications

## Why emphasize?

- incorrect specification is major reason that systems projects fail
- late discovery of problems is costly

## What?

- features of a good (or bad) requirements analysis
- requirements analysis outlines

## How?

- static (descriptive) specifications
  - system architecture
  - Entity-Relationship diagrams
- dynamic (procedural) specifications
  - Data Flow diagrams
  - decision tables
- supra-operational specifications
  - user interface/human factors
  - performance
  - portability, modifiability, *etc.*

# Requirements Analysis: Process Specifications

## Why?

- software is process-intensive
- evaluation of a finished software product is
  - difficult to do
  - too late

## How?

- "management" is not a dirty word

## What?

- plans  
whole life-cycle is "document driven"
- procedures & standards
- quality assurance specifications  
monitoring the process  
verification and validation
- tools  
Computer Aided Software Engineering (CASE)

## Requirements Specification Qualities

- clear, unambiguous, & understandable
- consistent
- complete
- cost-effective
  
- correct
  - to the extent possible
  
- ★ “A good specification document should be precise enough [to define] exactly what output is generated for any given input.”
  - A. von Mayrhauser, *Software Engineering*

## Functionality *versus* Qualitative Requirements

### Functionality Requirements (FR)

- as in “what functions does the system perform?”
- more tied to specific objectives
  - ◇ operational
  - ◇ measurable or observable
  - ☆ information
- also called “Functional Requirements”

### Qualitative Requirements (QR)

- ◇ “user friendly” is an example
- also called “Non-functional Requirements”

### FR & QR tricky to distinguish

- depends upon perspective
  - ◇ manager: “easy to use” is qualitative
  - ◇ clerk: “menu choice for commands” is functionality
- depends upon degree of specificity
  - ◇ “portability” - qualitative
  - ◇ “portable to Unix” - functionality

# Major Components

## Goals & Objectives

### Environment

- staff
- facilities
- current systems
- other [information] systems  
computer & paper  
current & proposed
- impact on operations

### Functionality Requirements

- information  
models & concepts flow
- operations  
examples & scenarios

### Qualitative Requirements

## Checklist/Suggested Outline for RS

0. Executive Summary
1. Introduction
2. Background of Client
3. Goals of the Project
4. Environment
  - 4.1. Users and Roles
  - 4.2. Hardware and Software Platforms
  - 4.3. Interaction with Other Information Systems
  - 4.4. Impact on Operations
5. Current System
  - 5.1. Current Entity-Relationship Model
  - 5.2. Current Data Flow Diagrams
  - 5.3. Other Supplementary Material
6. Proposed System
  - 6.1. Overview
  - 6.2. Proposed Entity-Relationship Model
  - 6.3. Data Flow Diagrams
  - 6.4. Interaction of ER and DF Models
  - 6.5. Functionality Requirements  
a long list of required functionality
  - 6.6. Business Rules/Constraints
  - 6.7. Maintenance Requirements
  - 6.8. Qualitative Requirements
  - 6.9. User Interface
  - 6.10. Implementation/Installation
  - 6.11. Other Considerations
  - 6.12. Limitations
7. Appendices
8. Client Comments

## Requirements Analysis: Specification Principles

A good specification should:

- separate functionality from implementation
- use process-oriented system specification language
- use static-structure specification language
- encompass the system of which the software is a component
- encompass the environment in which the system operates
- use cognitive models
  - reflect real-world
  - communicate with user
- provide operational specificity
  - using scenarios
  - through tests & measures
- provide ability to change and grow
- have localized, loosely-coupled structure
  - modularity

– after Balzer and Goldman (1979)

## Information Requirements

Entity-Relationship (ER) models central to  
*information RS*

Hence several lectures on ER modeling

ER model leads to functionality requirement:

- “store, update, query, manage information as specified in the in ER model”
- typically update has most complexity

Note that “apply” requires additional functionality requirements

# Requirement Specificity

## Models

- conceptual
- ⇒ preliminary design

## Lists

- operations / characteristics
- ⇒ detailed design

## Standards / metrics

- ⇒ guide subsequent steps
- ⇒ verification

## Tools & methodologies increase specificity

- conceptual
- notational

## Requirement Flexibility

Never say 'never'.

Requirements must be *general* and *flexible* because:

- user may change
- environment may change
- goals may change
- goals/specifications may be wrong
- a specific system may “lock in” bad habits

## Balance

Does specificity conflict with flexibility?

No!

- the requirements must be specific in their description of the proposed system
- the system described must be flexible
- the requirements should be generalized within well-conceived parameters
- Essential *versus* Accidental

## Characteristics of Good Information Collection

- quantity
- reliability
- scope
- perspective
- organization
- refinement

## Discovering Requirements

### Carry-forward

feasibility study

“domain knowledge”

### Observation & Interviews

⇒ operations

★ ask the question

*What makes a difference in your job?*

### Documents, etc.

⇒ Current system, policies & procedures, ...

### Look for

- data
  - formal
  - informal
- uses of data
  - formal & informal
  - manual & computerized
  - decision analysis

Remember: put yourself in the clients' perspective

technical/official

human/informal

## Representation Principles

- representation form and content should be relevant to the problem
- information contained within the specification should be well-structured
  - ⇒ client
  - ⇒ later project phases
- notational forms should be restricted in number and consistent in use
  - ⇒ client
- representations should be revisable
  - ⇒ CASE tools

## The Seven Deadly Sins of Specification

1. Noise
  - ◇ irrelevant information
  - ◇ confusing presentation
2. Silence
  - ◇ omissions
3. Over-specification
  - ◇ premature implementation decisions
4. Contradiction
  - ◇ inconsistency
5. Ambiguity
6. Forward reference
7. Wishful thinking

– B. Meyer, 1985

## Selecting an Alternative - Questions

Following questions apply, with variations, to specification, design, planning, *etc.*

### *Process*

1. Are all alternatives fully reviewed and evaluated?
2. Are the alternatives ranked in terms of their ability to meet the system needs?
3. Is there a technical-management team with authority to select the most appropriate alternative?

### *Content*

4. Does one alternative clearly outrank the others?
5. Which alternatives are better to implement in terms of time, cost, technical risk, *etc.*?
6. Which alternatives have clear conceptual foundations?
7. Which alternatives support the best user interaction?
8. Which alternatives do the clients support?
9. Which alternatives are likely to last the longest?

## Selecting a Design Alternative - Deliverables

1. A detailed comparison of alternatives
2. A ranking of alternatives
3. A specific recommendation as to the alternative that is best to pursue
4. A report to the clients on the alternative selected
5. A summary of reasons for rejecting other alternatives

## Analysis Process - Questions

1. Is the overall system flow modeled?
2. Are all data elements, flows, and expected processing steps defined?
3. Are associated clerical procedures outlined?
4. Are the content and uses of inputs and outputs defined in a general way?
5. Are procedural and organizational changes that the new system will generate defined and evaluated?
6. Has the volume of data and transactions been estimated?
7. Are the facilities requirements for the new system described?
8. Are the security and accuracy requirements of the data being considered?
9. Is there a tentative data conversion plan?
10. Is there a list of expected system modules?
11. Is a preliminary system implementation plan available?
12. Are testing procedures for the new approach thoroughly defined?

## Plans for the Next Phase - Questions

1. Are there work tasks and resource estimates for the general design work?
2. Is there a resource loading plan that shows requirements by work task?
3. Are client support tasks identified and planned? Are the clients aware of them?
4. Are target dates set to obtain authorization to proceed with the next phase?
5. What is the expected completion date of the proposed work?

## Plans for the Next Phase - Deliverables

1. The work plan and the resource estimates
2. The user support plan
3. A narrative on the approach to managing the next phase
4. A preliminary testing plan
5. A tentative implementation plan

## Management Presentation and Review - Questions

1. Are all levels of management in the technical and client areas briefed on the results and recommendations?
2. Are the presentations clearly and logically formulated?
3. Are management's concerns and questions documented and answered?
4. Has the proposed alternative survived management's scrutiny?
5. Does the team have any doubts about the project approach?
6. Have minority opinions and negative comments been properly addressed?

## Management Presentation and Review - Deliverables

1. Presentation critiques and internal reviews
2. Presentation reports and visual aids
3. Authorization to proceed

## Possible Additional Issues

1. Hardware and software driven issues  
e.g., is the design of the system driven by hardware characteristics or software requirements?
2. Algorithm design and analysis considerations
3. Mismatch of tools and needs
4. Performance, accuracy, and reliability
5. The interrelation of all system elements and the communication among various development groups e.g., developers and systems operations
6. The impact of a technology that may change before the project development is complete