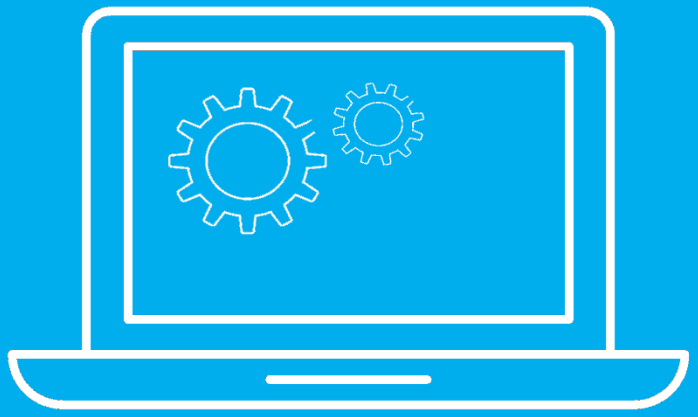


Software Engineering

Requirements

Software Requirements

What is a requirement?

A **requirement** is a singular documented physical or functional need that a particular product must be able to perform.

"The software shall be able to calculate the sum of a column of integers."

A statement that identifies a necessary attribute, capability, characteristic, or quality of a system for it to have *value and utility to a stakeholder*.

Requirement Specification

A requirement's **specification** is a comprehensive technical description of how that requirement will be realized.

The set of specifications will fully describe what the software will do and how it will be expected to perform.

Specification Example

Requirement:

1. The user shall set a password to control account access.

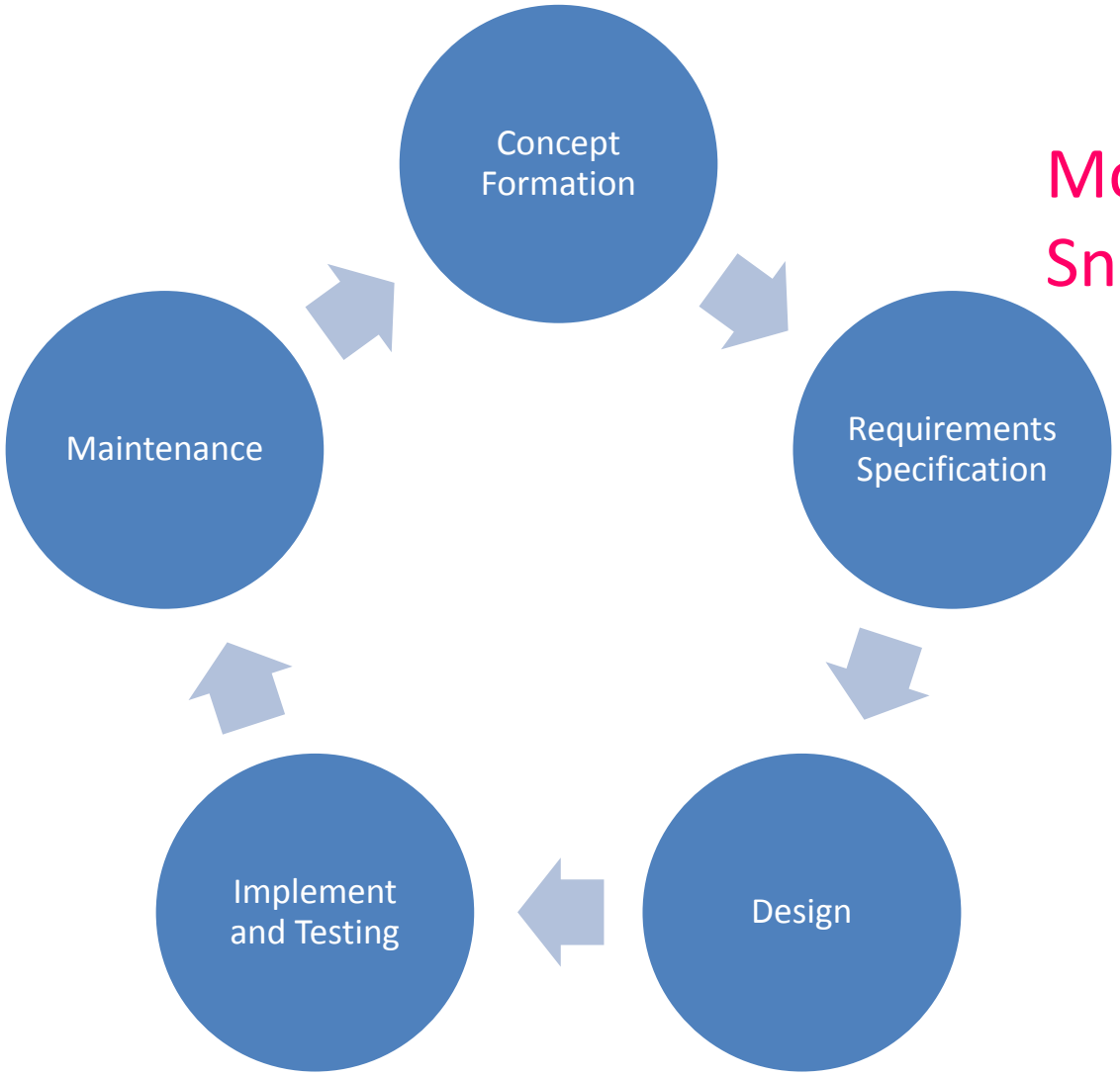
Specification:

1. The password shall be at least eight and no more than sixteen symbols long.
2. The password shall contain at least one lower case and one upper case letter.
3. The password shall be hashed and stored in the password database (Req 3.4).

Specifying Requirements

- Requirements and their specifications are a description of what the system should do.
- They capture **what** the system is doing, not **how** it does it.
  - How = how the code is structured, what algorithms and data structures are used, etc.
  - (that is the design).
- Must be detailed enough to distinguish between the “right” and “wrong” system.

Typical Development Process



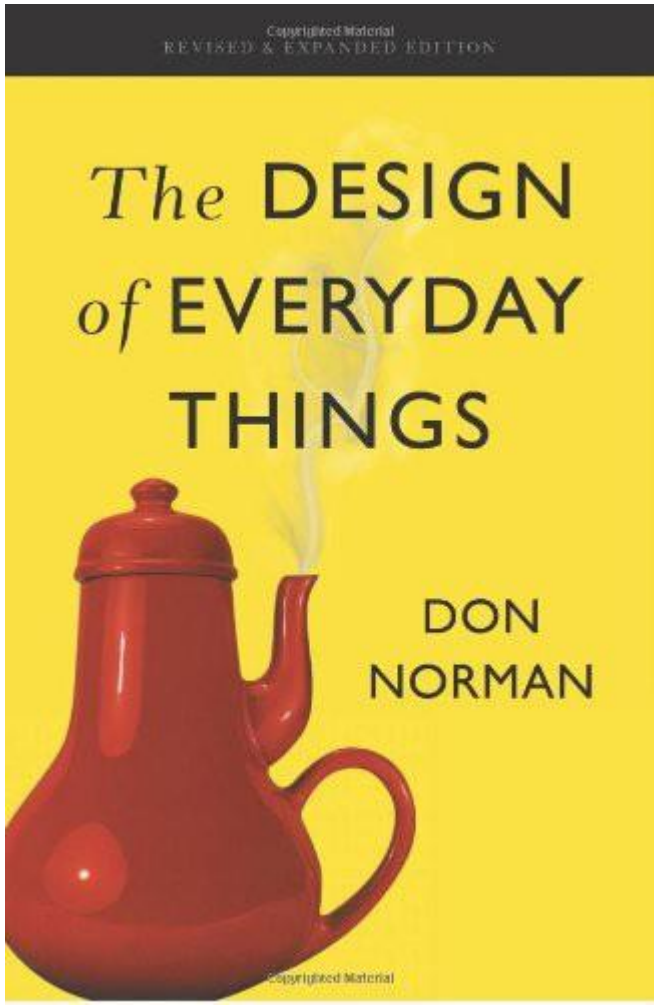
Most Issues Sneak in Here

Importance of Requirements

"The single hardest part of building a software system is deciding precisely what to build… No other part of the work so cripples the resulting system if it is done wrong. No other part is more difficult to rectify later."

- Fred Brooks

In other words:
"The spec was vague"



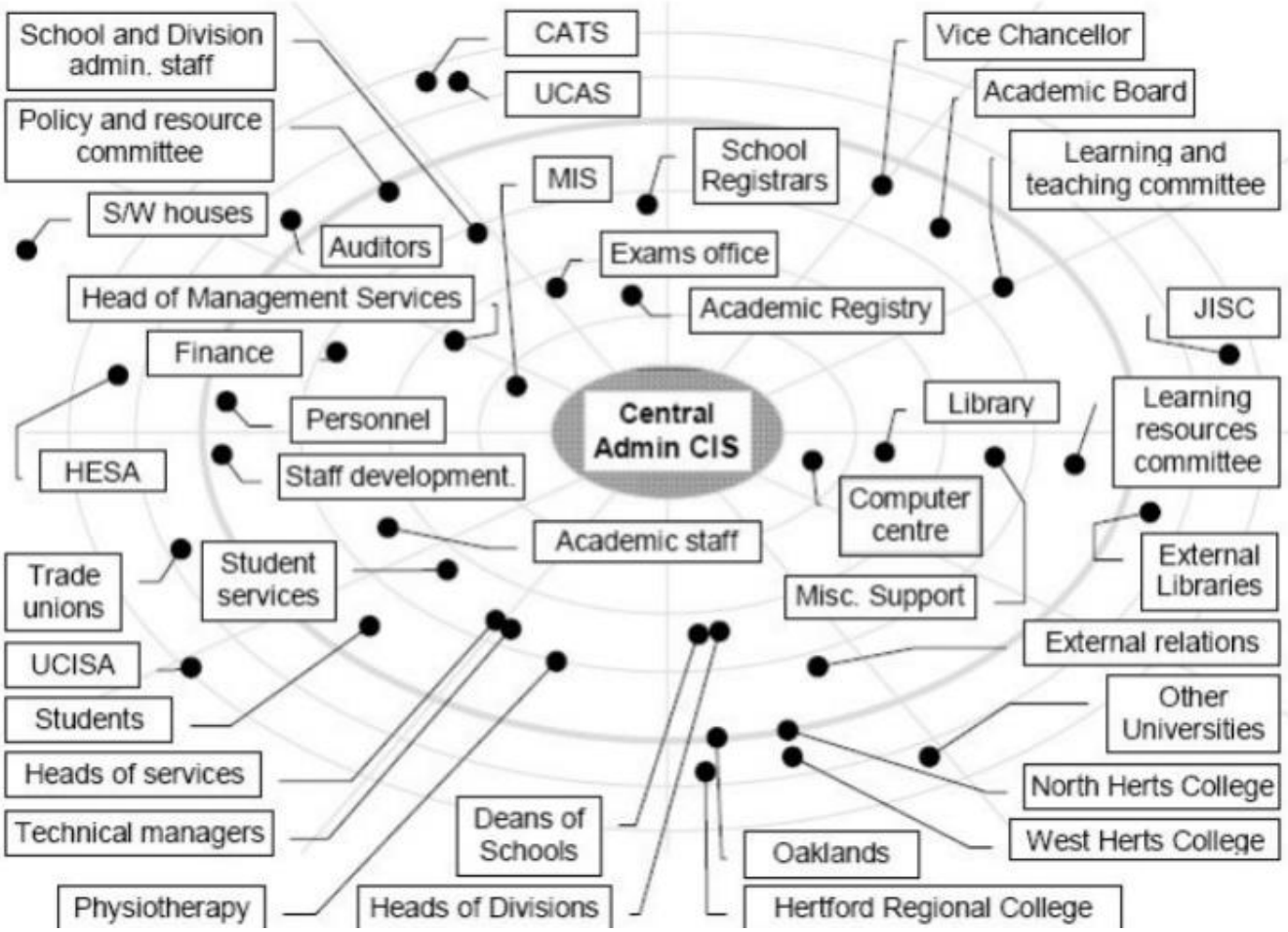
The Stakeholders

The specification process must involve all stakeholders:

- Clients
- Engineers
- Regulatory Agencies
- Users

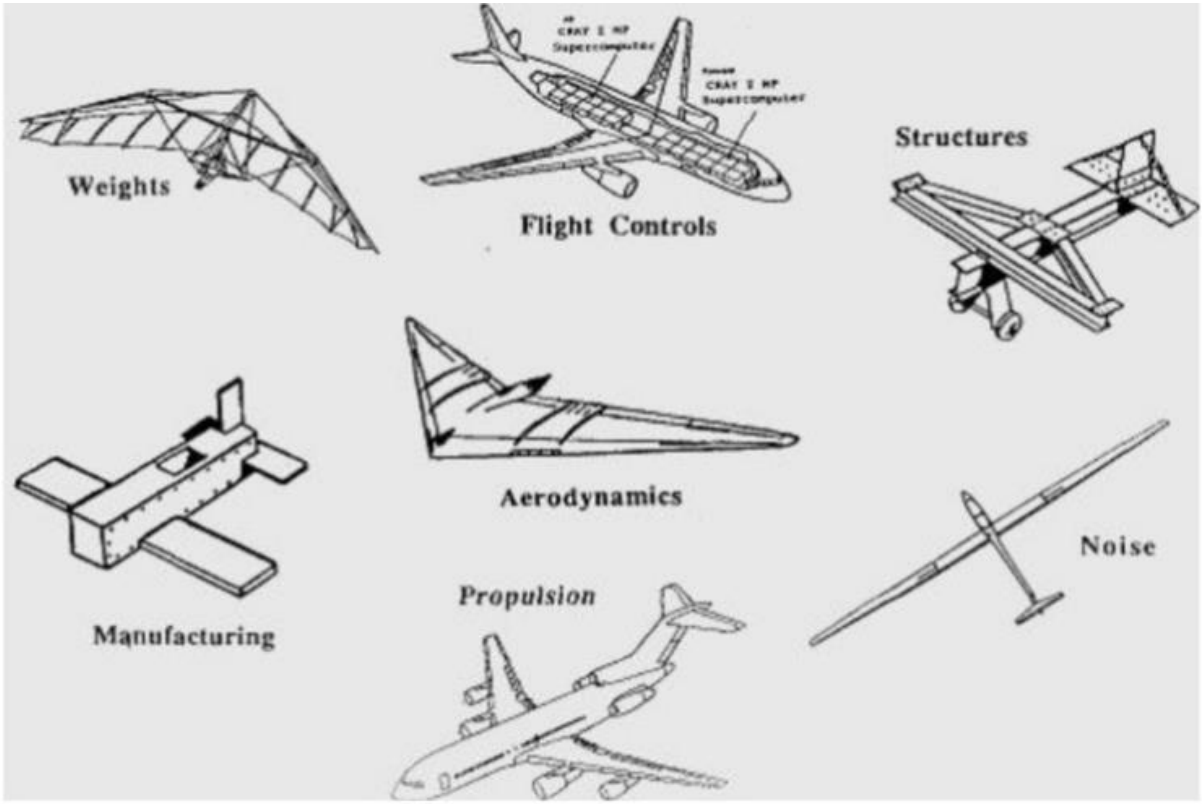
Do all of these people want the same thing?

There Can Be Many Stakeholders

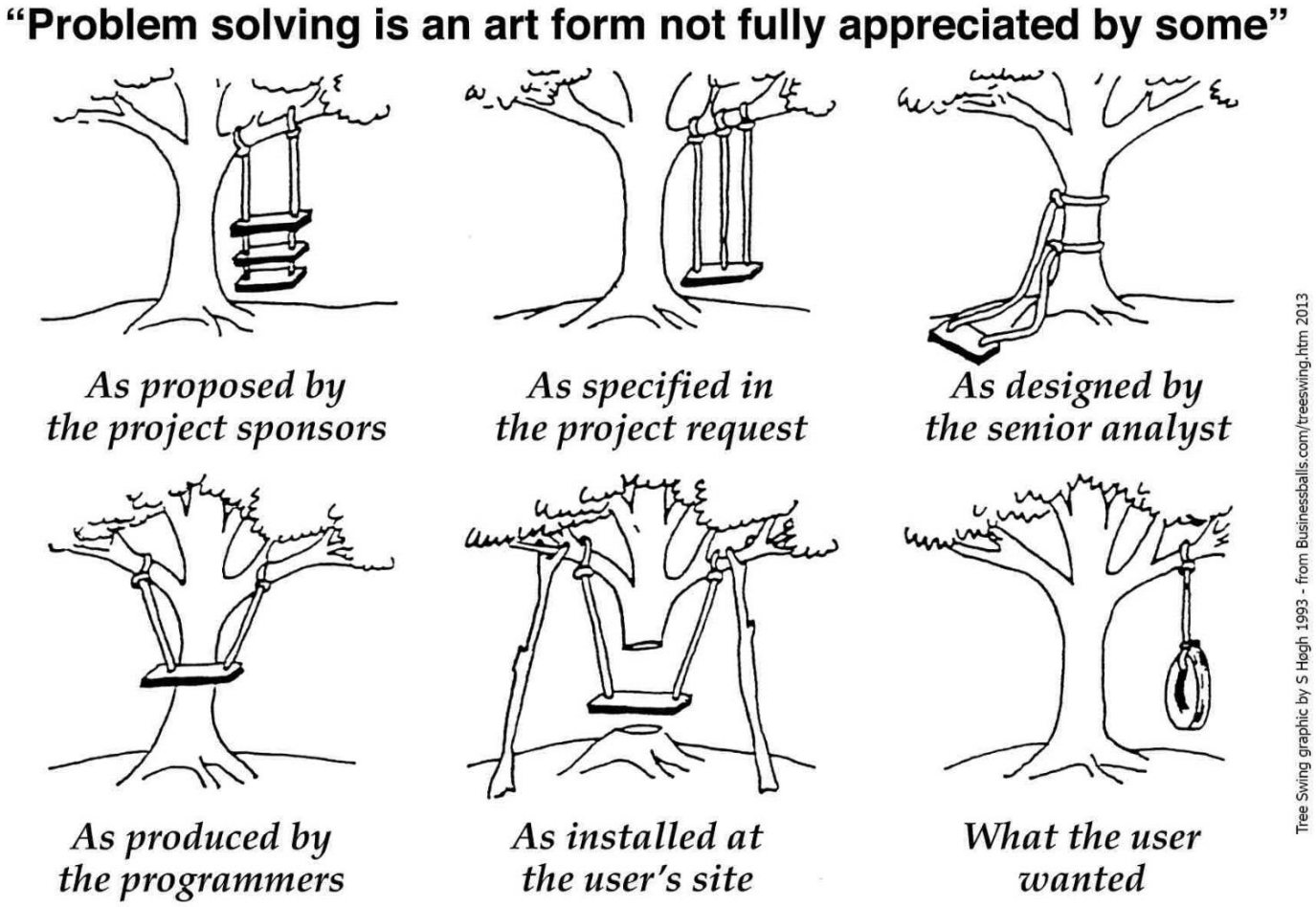


The Stakeholders

One requirement can have many meanings depending on the stakeholder's perspective.



Importance of Requirements



Importance of Requirements

The Engineering Argument: Engineering is about developing solutions to problems. A good solution can only be developed if *engineers understand the problem*.

The Economic Argument: Errors cost more to correct the longer they go undetected.

The Empirical Argument: Failure to understand and manage requirements is the biggest single cause of cost and schedule over-runs

Types of Requirements

User Requirements

- Properties that the user wants to see out of the final system.
- Usually high-level and lacking in technical detail.
- Gathered from the user, and form the basis for technical requirements.

User Requirement Example

“The user shall be able to book a concert ticket through the system.”

Hardware Requirements

- Define the properties expected from and constraints on the physical hardware in a system.
- Expectations on storage space, CPU, network latency, sensor inaccuracy bounds, etc.

Hardware Requirement Example

from a pacemaker

The Device Control Monitor shall either:

- use an inductive telemetry wand to communicate with the pulse generator, maintaining consistent communication over the range of 0 cm to 5 cm between the wand and the pulse generator; or,
- use some other medium, such as RF or ultrasound, that is safe and legal to use, for maintaining consistent telemetry with an implanted medical device.

Software Requirements

Software Requirements

- Properties of and constraints on the software being executed on the system.
- Typically the Bulk of requirements

Software Requirements Example

from a pacemaker..

If dynamic, the AV delay shall be determined individually for each new cardiac cycle based on the duration of previous cardiac cycles.

- The previous cardiac cycle length is multiplied by a factor stored in device memory to create the dynamic AV delay.
- The AV delay shall vary between a programmable maximum paced AV delay and a programmable minimum paced AV delay

Software Requirements

- Define how software and hardware must work together to perform functionality.
- Often related to interfaces, input, or output - how hardware can influence the software.

Types of Requirements

Functional Requirements:

- Related to the functionality of the system.
- Describe system services or actions.
- Can be used to distinguish correct output from incorrect.

Non-Functional Requirements:

- Constraints on the system or the development process.
- Related to how functionality is performed, not whether the end result is correct.

Non-Functional Requirements

- **System properties and constraints**
  - Reliability, response time, storage requirements
  - Constraints on I/O device capability, system representations, etc.
- **Process requirements**
  - Team organization, tool support, programming language, or development method.
- May be more critical than functional requirements.
  - If not met, the system is useless.

Non-Functional Requirement Types

Product Requirements: Delivered product must behave in a certain manner: speed, reliability, disk space

Organizational Requirements: Consequence of organizational policies and procedures: process, implementation requirements

External Requirements: Arise from factors external to the product and development process: legislative requirements, communication protocols, interoperability standards.

Non-Functional Requirements

Product Requirements

Withdrawals must be available with no more than 30 minutes of maintenance time per day.

Organizational Requirements

The system development process shall follow the SCRUM model.

External Requirements

The system shall keep a log of transactions in the Microsoft Excel file format.

Non-Functional Requirements

- Accessibility
- Availability
- Capacity
- Compliance
- Cost
- Documentation
- Fault Tolerance
- Privacy
- Reliability
- Reusability
- Security
- Usability
- …

Writing Requirements

Writing Requirements

Normal Method:

Natural language statements, supplemented by diagrams and tables (like the examples so far).

This is universally understandable, but problems can arise.

Issues with Natural Language

Lack of clarity: It is hard to write unambiguously.

Missing details: It is easy to forget to include details when using natural language.

Requirements amalgamation: Several different requirements may be expressed together.

Level of Detail

The level of detail of requirements may range from a high-level abstract statement to a detailed mathematical functional specification.

Requirements serve dual functions:

- May be the basis for a bid for a contract
  - Therefore, open to interpretation.
- May be the basis for the contract itself
  - Therefore, must be defined in detail.

Capturing Good Requirements

Three common problems:

Poorly written individual requirements

- Ambiguous, contradictory, unclear.

Untestable requirements

- Cannot verify that the final system meets the requirements, as written.

Poorly structured requirements documents

- Unable to find necessary information.

Requirements Sample (NASA)

While acting as the bus controller, the C&C MDM CSCI shall set the e, c, w indicators identified in Table 3.2.16-2 for the corresponding RT to "failed" and set the failure status to failed for all RT's on the bus upon detection of transaction errors of selected messages to RTs whose 1553 FDIR is not inhibited in two consecutive processing frames within 100 milliseconds of detection of the second transaction error if: a backup BS is available, the BC has been switched in the last 20 seconds, the SPD card reset capability is inhibited, or the SPD card has been reset in the last 10 major (10 second) frames, and either

1. the transaction errors are from multiple RT's, the current channel has been reset within the last major frame, or
2. the transaction errors are from multiple RT's, the bus channel's reset capability is inhibited, and the current channel has not been reset within the last major frame.

Withdrawal Requirement

2.6: Withdrawal

If the card is accepted, the user has entered the correct PIN, and if there are sufficient funds in the account, the amount of cash shall be dispensed. If the card is invalid (in which case it should be ejected), the PIN does not match the one required for the card (in which case a tone shall sound and the user given the option to try again - the tries shall be limited to 3), or the balance is insufficient (in which case a tone shall sound and the user shall have the opportunity to enter a new amount) cash shall not be dispensed.

Withdrawal Requirement (Take 2)

2.6: The system shall support cash withdrawals by the user.

High-level requirement, defining a function of the system.

2.6.1: A withdrawal shall be allowed if, and only if:

- The card can be validated (Req 2.7).
- The PIN is valid for the card (Req 2.8).
- The funds in the card account exceed the funds requested in the withdrawal request.

Conditions necessary to perform the action.

2.6.2: If a withdrawal is allowed (Req 2.6.1), the exact amount requested shall be dispensed.

Action to be performed if preconditions are met.

Using Templates to Present Information

- Templates define a standard requirement structure.
- You should establish templates for the requirement descriptions.
  - Ensure readers are familiar with the document.
  - Acts as a checklist so that no sections are forgotten.
  - Makes it easy to find the needed information.

Suggested Template Items

- **Number:** A unique identifier.
- **Use Case:** Which use-case is this requirement linked to.
- **Introduction/Definition:** What is this requirement about?
- **Rationale:** Why does the requirement exist?
- **Source:** Who came up with the requirement?
- **Author:** Who wrote it down?
- **Inputs:** What are the necessary inputs to use this function?
- **Required Function:** What is the requirement?
- **Outputs:** What is output as a result of this function?
- **Related Requirements:** List of relevant requirements?
- **Conflicts:** Are there requirements in conflict with this one?
- **Support Material:** Docs, figures, tables, etc.
- **Test Cases:** How do we test this requirement?
- **Date:** When was this requirement modified last?
- **Priority:** How important is this requirement?

Withdrawal, Using Template (Still Bad)

Introduction: The most common action performed at an ATM is the withdrawal of funds. The ATM must support this functionality.

Rationale: Survey ABC-345 indicates that withdrawals are highly desirable. The success of the product hinges on successful withdrawal of funds.

Inputs: Card number, PIN, requested amount

Description: If the card is accepted, the user has entered the correct PIN, and there are sufficient funds in the account, the amount of cash shall be dispensed. If the card is invalid (in which case it should be ejected), the PIN does not match the one required for the card (in which case a tone shall sound and the user given the option to try again—the tries shall be limited to 3), or the balance is insufficient (in which case a tone shall sound and the user shall have the opportunity to enter a new amount) cash shall not be dispensed.

Outputs: Customer Receipt, Requested Amount of Money, Alarm

Persistent Changes: The Requested Amount will be deducted from the Customer Account.

Withdrawal, Using Template (Better)

Introduction: The most common action performed at an ATM is the withdrawal of funds. The ATM must support this functionality.

Rationale: Survey ABC-345 indicates that withdrawals are highly desirable. The success of the product hinges on successful withdrawal of funds.

Inputs: Card number, PIN, requested amount

Description: The user shall be able to withdraw funds from the ATM

- A withdrawal shall be allowed if and only if:
  - The Card Number can be validated (Req 35)
  - The PIN is valid for the card (Req 45)
  - The funds in the card account exceeds the Requested Amount requested in the withdrawal
- If a withdrawal is allowed, the Requested Amount of money shall be dispensed.

Outputs: Customer Receipt, Requested Amount of Money

Persistent Changes: The Requested Amount will be deducted from the Customer Account.

Related Requirements: 35, 45