

Software Engineering Requirements

Erik Fredericks // frederer@gvsu.edu

Adapted from materials provided by Byron DeVries, Jagadeesh Nandigam

Outline

Critical Role of Requirements

Software Requirements – What?

Levels of Requirements

Types of Requirements

- Functional Requirements
- Non-functional Requirements

Requirements Engineering (RE)

- Requirements Development
- Requirements Management
- Common Risks in RE



Critical Role of Requirements

The critical role of requirements to a software project:

*The **hardest single part** of building a software system is **deciding precisely what to build**. No other part of the conceptual work is as 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**.*

- Frederick P. Brooks, Jr.

Software Requirements - What?

A **requirement** is a capability (functionality) or a condition (constraint) to which a system must conform.

Requirements are specifications of **what** should be **implemented**. They are descriptions of how the system should behave, or of system property or attribute. They may be a constraint on the development process of the system.

Levels of requirements

Customer/Business Level:

- Business requirements
- Exists in a vision and scope document

User Level:

- User requirements
- Exists in:
 - Use cases
 - Quality attributes

Implementation Level:

- Functional and non-functional requirements
- Exists in software requirements specification (SRS)

Levels of requirements

Business requirements:

- High-level objectives of the organization or customer who requests the system

User requirements:

- Describe user goals or tasks that the users must be able to perform with the product

Functional requirements:

- Specify the software functionality that the developers must build into the product to enable users to accomplish their tasks to satisfy business requirements

Type	Example
Business	<ul style="list-style-type: none"> • reduce incorrectly processed orders by 50% by the end of next quarter • increase repeat orders from customer by 10% within six months after deployment
User/Stakeholder	<ul style="list-style-type: none"> • add new customer account • view order history • check order status • create new order
Functional/Solution	<ul style="list-style-type: none"> • display customer last name as a link to account history • allow sorting by account opening date
Non-Functional	<ul style="list-style-type: none"> • allow up to 200 concurrent users • require strong passwords of at least 8 characters in length containing a minimum of one non-alphabet character
Implementation/Transition	<ul style="list-style-type: none"> • must run on all Java platforms including 64-bit versions • users must pass an online certification before being allowed to use the system

Types of requirements: **functional**

Also known as **behavioral** requirements

Specific services that an actor can accomplish using the application

- An actor is someone or something (hardware or software) that is external to the system and interacts with the system

Represent main product features/services

Types of requirements: **non-functional**

Qualifies a service or services

- i.e., specifies something about them

Overall qualities or attributes of the resulting system

Can be viewed as restrictions or constraints placed on a service offered by the system

Often more critical than individual functional requirements

Failing to meet a nonfunctional requirement can make the whole system unusable

Major categories – qualities, constraints, external interfaces, and error conditions

Types of requirements

Functional requirements:

- Specifies what the system does.

Non-functional requirements:

- Specifies criteria about the system.

Can anyone think of some examples?

Table of Contents for a SRS Document

1. Introduction

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Project Scope
- 1.5 References

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Features
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 Assumptions and Dependencies

3. System Features

- 3.1 Functional Requirements

4. External Interface Requirements

- 4.1 User Interfaces
- 4.2 Hardware Interfaces
- 4.3 Software Interfaces
- 4.4 Communications Interfaces

5. Nonfunctional Requirements

- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

Related UCAs	Corresponding Safety Constraints
UCA-1.1	SSR1.1- ACC controller should provide acceleration signal when the target vehicle is no longer in the lane.
UCA-1.2	SSR1.2- ACC controller should not increase the speed when the distance to the target vehicle is too close.
UCA-1.3	SSR1.3- ACC controller should not provide the acceleration signal speed before a safe distance is reached.
UCA-1.4	SSR1.4- ACC controller should not increase the speed beyond the value of the speed set by the driver.



How the customer explained it



How the Project Leader understood it



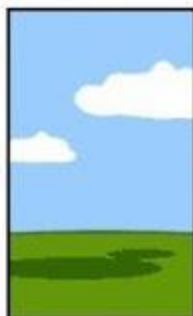
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



How the project was documented



What operations installed



How the customer was billed

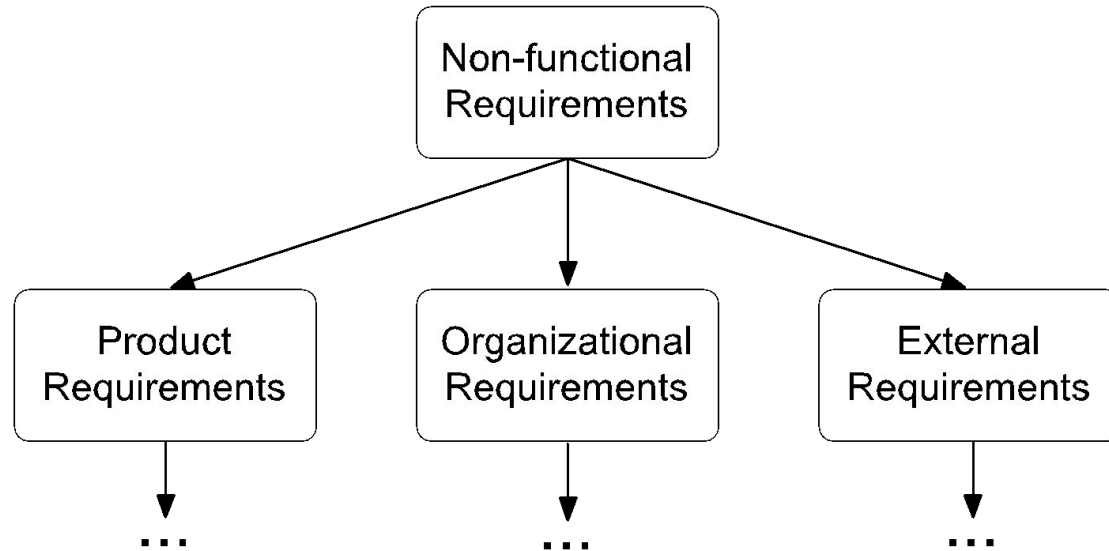


How it was supported

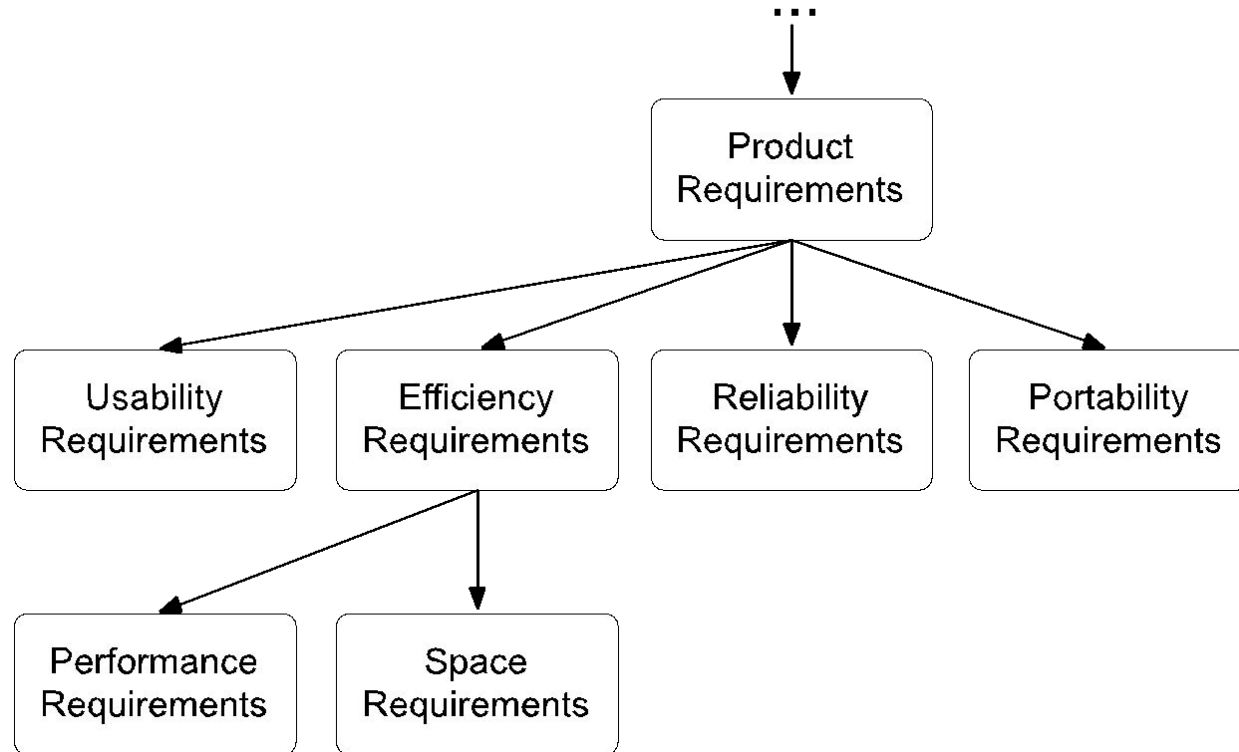


What the customer really needed

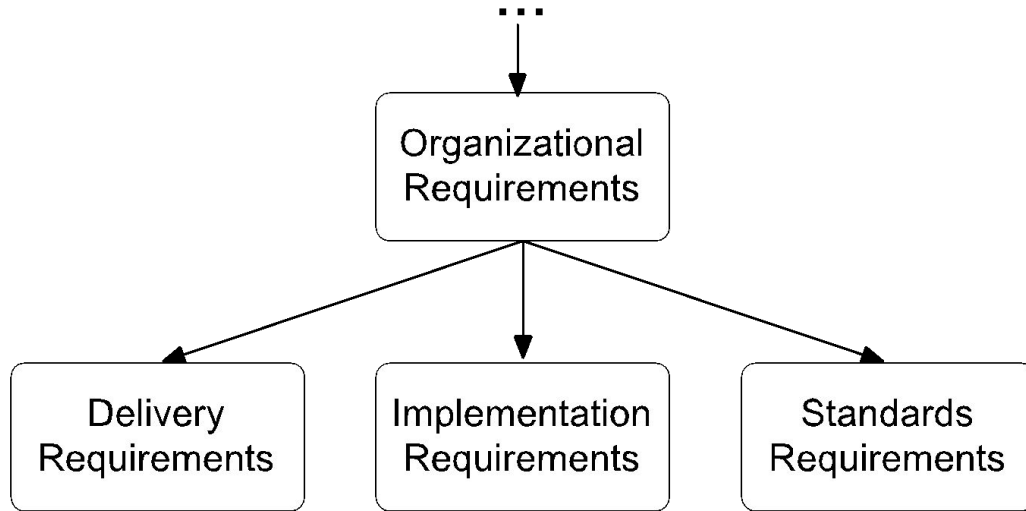
Non-functional requirements



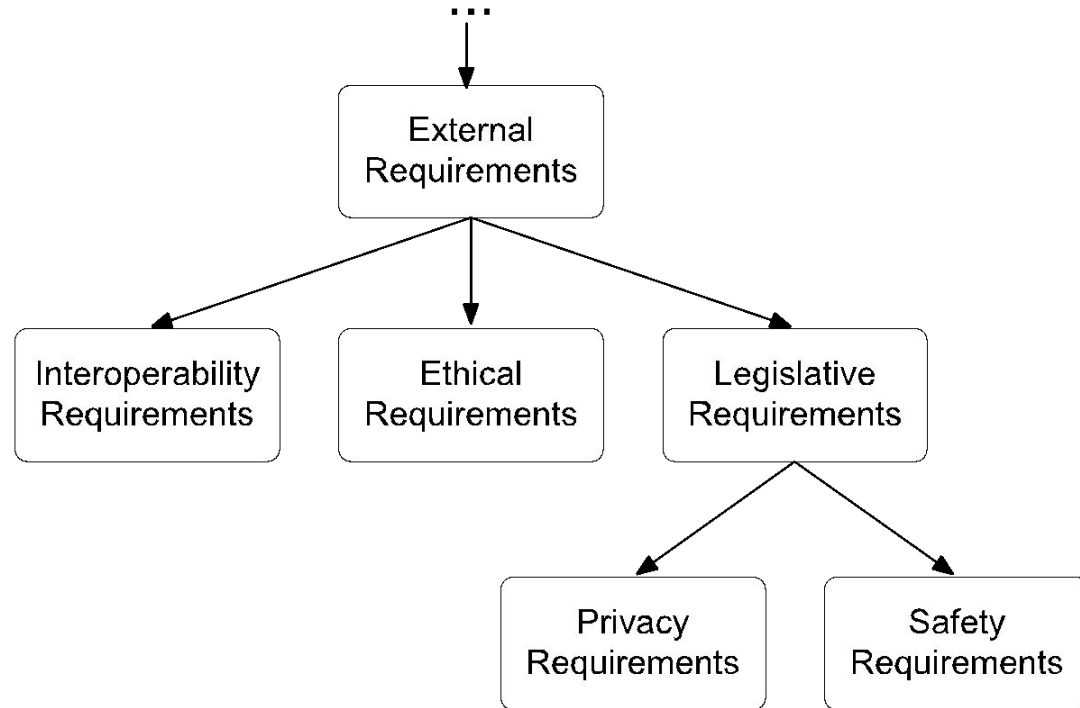
Non-functional requirements



Non-functional requirements



Non-functional requirements



Non-functional requirements

Users or customers state non-functional requirements as general or imprecise goals, for example:

- *The system **shall** be very user-friendly.*
- *The system **shall** recover from a failure quickly.*
- *The system **shall** respond to requests quickly.*

Imprecise non-functional requirements are often **difficult to verify**.

Non-functional requirements must be **specific, quantifiable, and testable**.

NONFUNCTIONAL REQUIREMENT EXAMPLES

OPERATION GROUP

Describes the user needs for using the functionality. The user perceives the system as an electronic tool that helps to automate what would otherwise be done manually. From this point of view, the user is concerned with how well the system operates.

ACCESS SECURITY

The extent to which the system is safeguarded against deliberate and intrusive faults from internal and external sources.

Examples

- a. Employees shall be forced to change their password the next time they log in if they have not changed it within the length of time established as "password expiration duration."
- b. Users must change the initially assigned login authentication information (password) immediately after the first successful login. The initial password may never be reused.
- c. The payroll system shall ensure that the employee salary data can be accessed only by authorized users. The payroll system shall distinguish between authorized and non-authorized users.
- d. Employees shall not be allowed to update their own salary information, and any such attempt shall be reported to the security administrator.
- e. Only holders of current security clearance can enter the national headquarters building.
- f. The access permissions for system data may only be changed by the system's data administrator.
- g. Passwords shall never be viewable at the point of entry or at any other time.
- h. Each unsuccessful attempt by a user to access an item of data shall be recorded on an audit trail.
- i. Users shall receive notification of profile changes via preferred communication method of record when profile information is modified.

ACCESSIBILITY

The extent to which the software system can be used by people with the widest range of capabilities to achieve a specified goal in a specified context of use.

Requirements **engineering**

Requirements engineering consists of two activities:

- Requirements development
- Requirements management

Why development *and* management
(and what is management)?

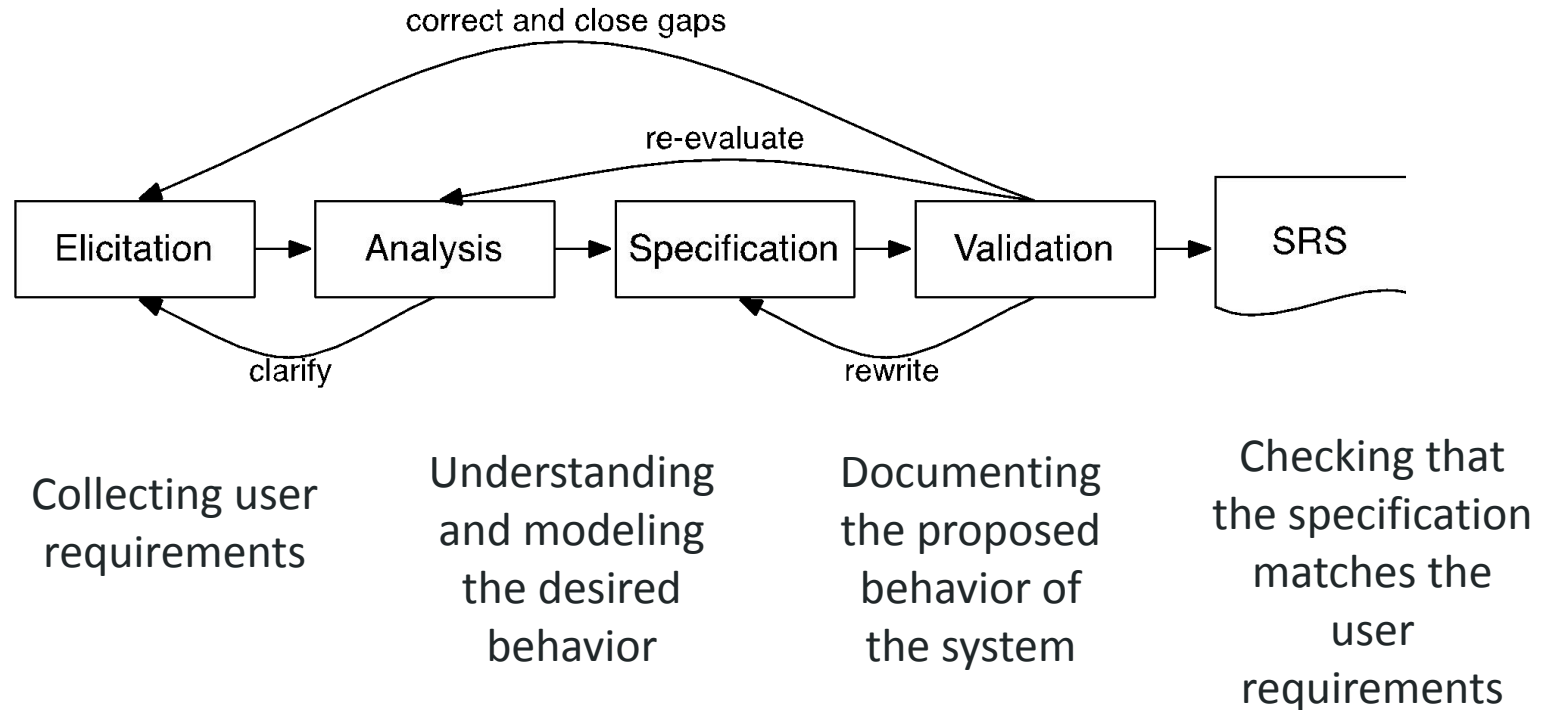
STOP

A REVIEW (of sorts)

What is the difference between a non-functional and a functional requirement?

Why do we need them in the first place?

Requirements development

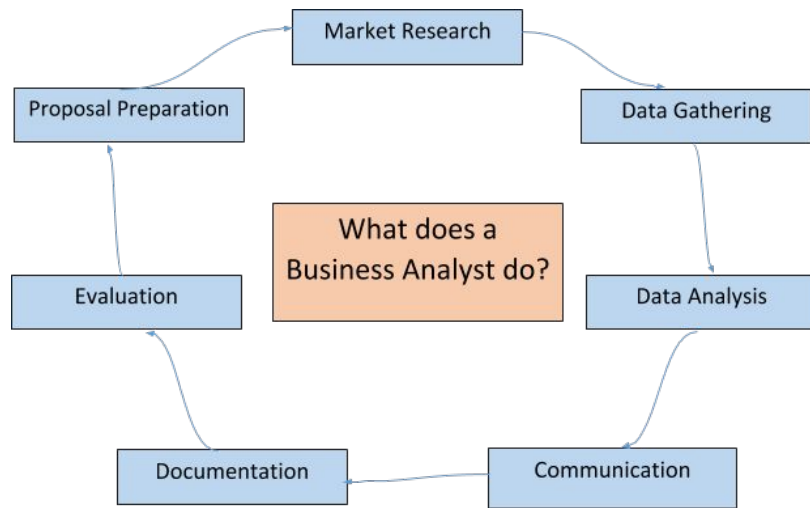


Requirements development: Analyst

Individual who has the primary responsibility to gather, analyze, document, and validate the needs of the project stakeholders.

Project role, not necessarily a job title

Bridges communication between customer and development stakeholders.



Requirements development: Analyst

Skills:

- Listening skills
- Interviewing and questioning skills
- Analytical skills
- Facilitation skills
- Observational skills
- Writing skills
- Organizational and interpersonal skills
- Modeling skills

Requirements Elicitation

Most communication-intensive aspect of software development

Elicitation strategies

- Collaborative group workshops
- Interviewing
 - Close-ended vs. open-ended
 - Structured vs. unstructured
 - Surveys and user questionnaires
 - Observing users at work – Ethnography
 - Scenario analysis of user tasks
 - Essence of use-case approach/modeling
 - Problem reports and enhancements requests

Requirements Elicitation

Classifying Requirements:

- Functional Requirements
- Business Rules
- Quality Attributes
- External Interface Requirements
- Constraints
- Data Definitions
- Solution Ideas (May impose design constraints)

Requirements Elicitation: Identifying Users

User involvement is key to delivering excellent software

Identify product user classes/groups

User groups based on:

- Frequency with which they use the product
- Their application domain experience
- Their computer system expertise
- **Features they use / Perspective of application or data**
- Tasks they perform
- Their access privilege or security levels



Requirements Elicitation: Conflicts

Different stakeholders have different sets of **potentially conflicting** requirements.

Resolving conflict can be done by prioritization:

- **Essential:** Absolutely must be met
- **Desirable:** Highly desired but not necessary
- **Optional:** Possible but could be eliminated

What are examples of each of these (next slide)?

What are examples of conflicting requirements?

Requirements Elicitation: Product Champions

A **representative of a user class/group** to provide the voice of the customer.

Product champion serves as the **primary interface** between the members of a single user class and the requirements analyst.

Important to agree on who the requirements **decision makers** are for the project.

Given the diversity of viewpoints, what issues could arise due to the use of product champions?

What issues could a product champion
bring?

Requirements specification

Requirements are the result of requirements elicitation and analysis, and stored in the Software Requirements Specification (SRS).

The SRS includes:

- System features
- External interface requirements
- Functional requirements
- Non-functional requirements
- Classifications of users
- ...

Requirements validation

Validates that the requirements actually define the system the customer wants and intends.

Validation should include checking that each requirement / set of requirements is:

- Correct
- Consistent
- Unambiguous
- Complete
- Feasible
- Relevant
- Testable
- Traceable

Requirements validation

Validation techniques can include:

- Requirement Reviews, including
 - Informal Reviews
 - Formal Reviews
- Prototyping
- Designing black-box (functional) test cases
- Defining user acceptance criteria/tests

Validation checks that you are building: the right thing

Verification checks that you are building: the thing right

Requirements management

Managing requirements is similar to managing code. It requires:

- Change Control
- Version Control
- Requirements Status Tracking
- Requirements Tracing

Requirements management: Change control

Change control is necessary to manage scope creep.

Typical change control processes include:

- Making change requests
- Impact analysis
- Change control board (that approves or rejects)
- Updating requirements documents
- Measuring requirement volatility

Requirements management: Version control

Version control ensures:

- Identification scheme for requirements
- Access to current set of requirements
- Only designated individuals can update requirements

Possible states for requirements status tracking within version control can include:

- Proposed
- Approved
- Implemented
- Verified
- Deleted
- Rejected

Requirements management: Tracing

Tracing is necessary to document the dependencies between individual requirements and other system elements, including:

- Business Rules
- **Use cases**
- Design Components
- Source Code Modules
- **Test Files**
- Help Files
- ...

Requirements Traceability Matrix:

Represents the links between requirements and other system elements, especially **higher and lower level requirements and tests.**



Hub

REQUIREMENT TRACEABILITY MATRIX

Select Test Plan: Stryka 12/02/2017



Advance Search

Hide Stats

Show Requirements



Hide



86

Total Requirements

Approved Pending Completed Verified
Deployed Submitted



1403

Total Linked Test Cases



201

Found Defects

REQ ID	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031	0032	0033	0034	0035	0036	
Test Case ID	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031	0032	0033	0034	0035	0036	
TC-000001	Pass	Pass	Pass			Pass	Pass	Pass																											Fail	Pass	
TC-000002					Pass	Pass	Pass	Pass	2	Pass	Pass	Pass	Pass	Pass	Pass	Pass		Pass	Pass																Fail		
TC-000003							Pass	Fail	1			Blocked	Pass	Pass							Pass	1	5												Pass	Fail	
TC-000004											Pass	Pass	Fail							Pass	Fail	Pass													Pass		
TC-000005													Pass	Pass	Pass			Pass	1	Pass		Pass	Fail	Pass			Pass	Pass	Pass						Pass	Pass	
TC-000006					Blocked	Fail	12				Pass	2	Pass				Pass	Pass	Pass	Pass	Blocked	Pass				Fail	Pass	Pass									
TC-000007										Pass	Pass	1									Pass	Pass					Pass	Pass	Pass								
TC-000008								Pass										Pass	1	Pass						Pass	Fail	1								Pass	
TC-000009							Pass	Pass	Pass														Pass			Pass		Pass	Pass	Blocked			Pass	Pass	Pass	Pass	
TC-000010										Pass								Pass			Pass	Pass	Pass				Pass									Pass	
TC-000011	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	12	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Fail	Pass	Pass	Pass	Pass	Pass	Pass	Fail	3	Fail	8	7	4	7	Pass	Pass	Pass	Pass	Pass	
TC-000012		Pass	Pass	Pass								Pass	Pass	Fail							Pass	Pass	Pass						Fail								
TC-000013							Pass									Pass		Pass	Pass	1							Fail	3	Fail								
TC-000014					Pass	Pass	Fail							Pass	Pass	Pass					Pass									Fail							
TC-000015							Pass								Pass	Pass			Pass	Blocked					Pass			Fail	Fail								

Risks in Requirements Engineering

Common risks:

- Insufficient Customer/User Involvement
- Overlooked User Classes
- Vague and Ambiguous Requirements
- Unprioritized Requirements
- Building Functionality No One Uses
- Analysis Paralysis
- Scope Creep
- Inadequate Change Process
- Insufficient Change Impact Analysis
- Inadequate Version Control



In-Class Assignment

I have a pressing need for **four requirements (2 functional, 2 non-functional)** for the following project use cases:

*Project manager: I need you to develop a web application that acts as both a customer and employee front-end to our back-end database system. Naturally you're aware that we are the largest retailer of home improvement products and clearly have a logo color different than orange or blue *cough*. We need customers to be able to search for products, see what is available in stores near their area, and to be able to purchase products. Our employees need to be able to login, update inventory and store information, and also log their hours into a hidden area available only to them. Since it is (currently) 2021, we should be able to handle any device type or screen size that the user has without any loss of functionality.*

(Same as before ... small teams, put everybody's name on it, submit to Blackboard)