Software Quality Engineering

Week 2

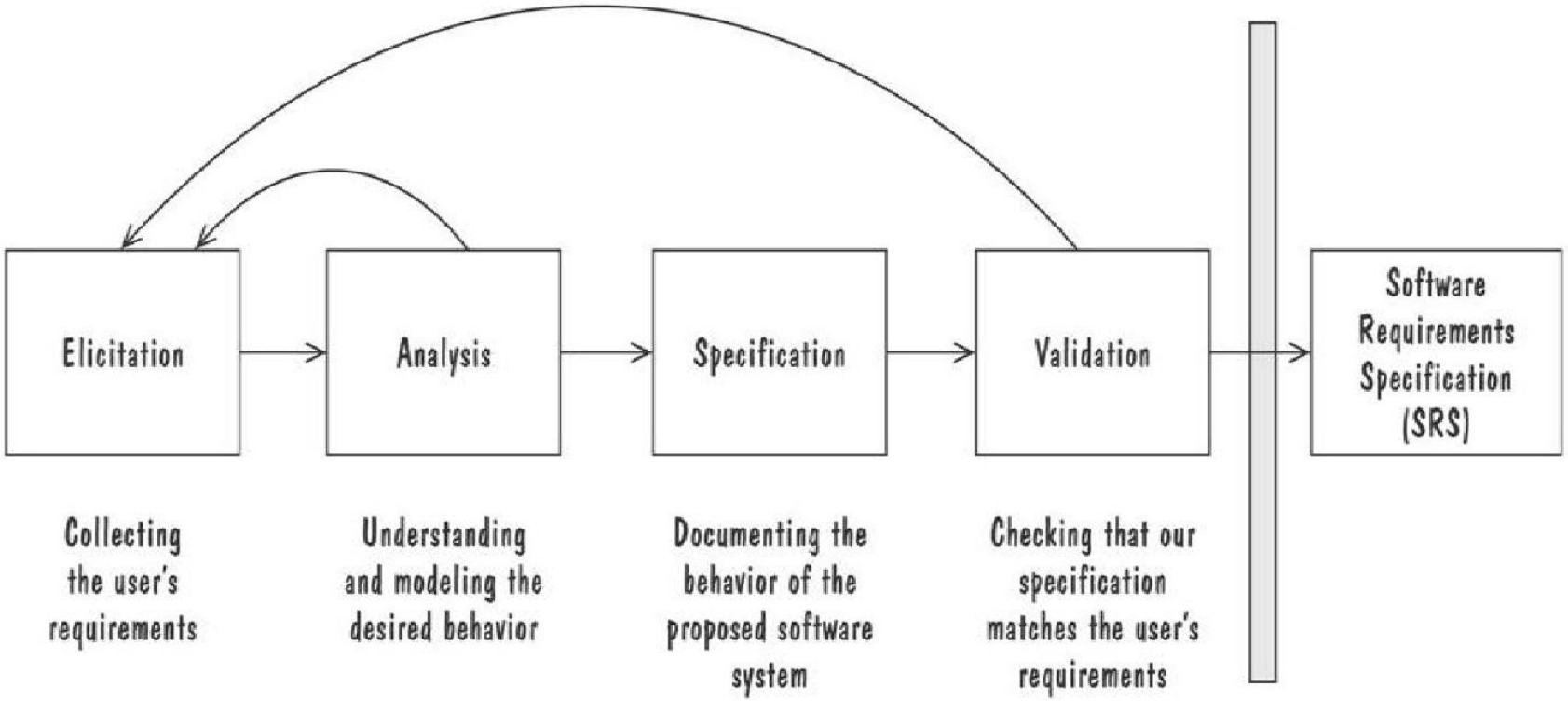
Engr. Muhammad Umer Haroon

FOLLOW THE RULES

- Use of Mobile (no)
- Late coming (no)
- Short Attendance (no)
- Plagiarism (no)
- Cheat (no)
- Disrespect (no)

The Requirements Process
(Process for Capturing Requirements)

- Performed by the req. analyst or system analyst
- The final outcome is a Software Requirements Specification (SRS) document



Functional Requirements	Non Functional Requirements
A functional requirement defines a system or its component.	A non-functional requirement defines the quality attribute of a software system.
It specifies "What should the software system do?"	It places constraints on "How should the software system fulfil the functional requirements?"
Functional requirement is specified by User.	Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers.
It is mandatory.	It is not mandatory.
It is captured in use case.	It is captured as a quality attribute.

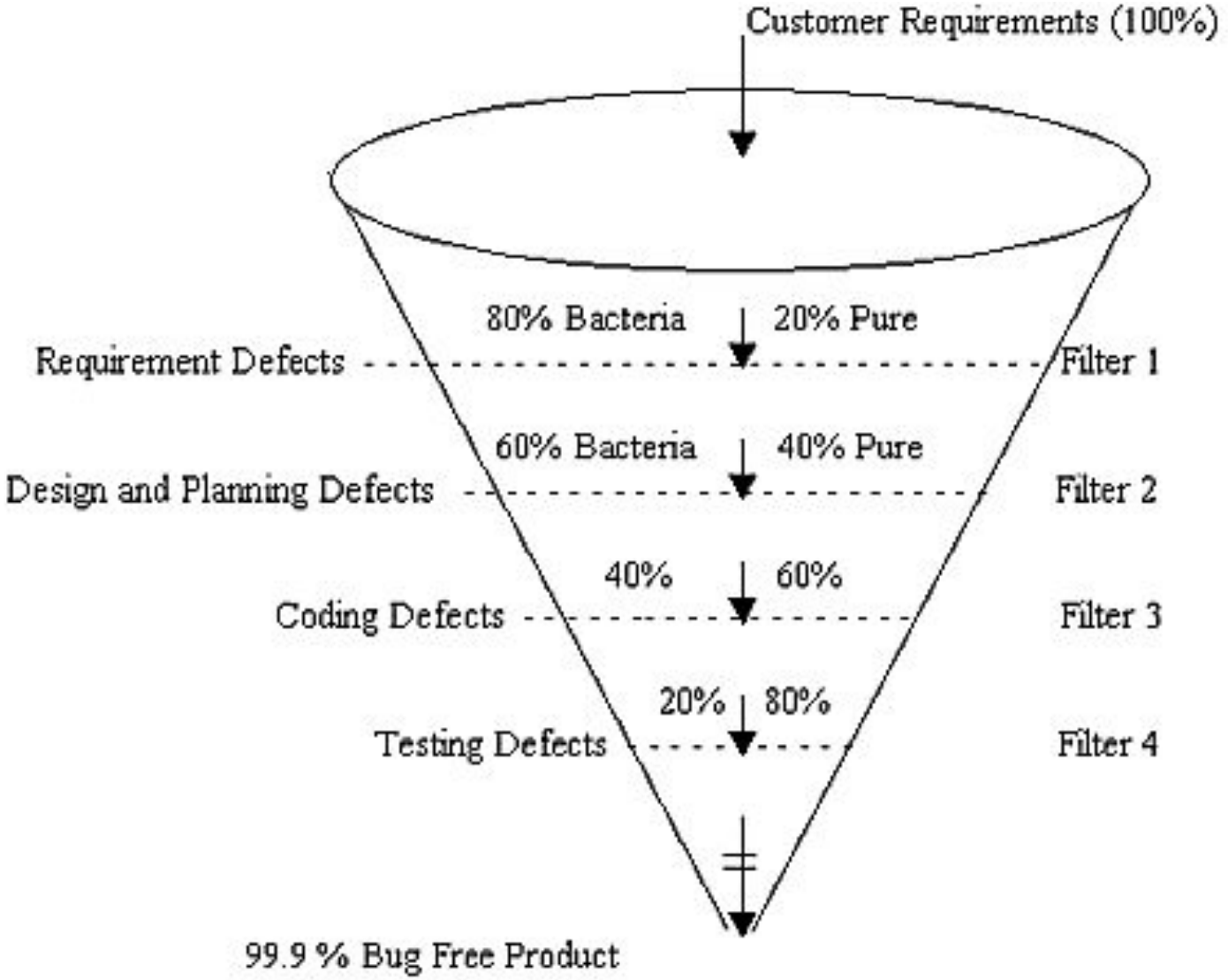
Functional Requirements	Non Functional Requirements
Defined at a component level.	Applied to a system as a whole.
Helps you verify the functionality of the software.	Helps you to verify the performance of the software.
Functional Testing like System, Integration, End to End, API testing, etc are done.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done.
Usually easy to define.	Usually more difficult to define.
Example **1)** Authentication of user whenever he/she logs into the system. **2)** System shutdown in case of a cyber-attack. **3)** A Verification email is sent to user whenever he/she registers for the first time on some software system.	**Example** **1)** Emails should be sent with a latency of no greater than 12 hours from such an activity. **2)** The processing of each request should be done within 10 seconds **3)** The site should load in 3 seconds when the number of simultaneous users are > 10000

What is Defect?

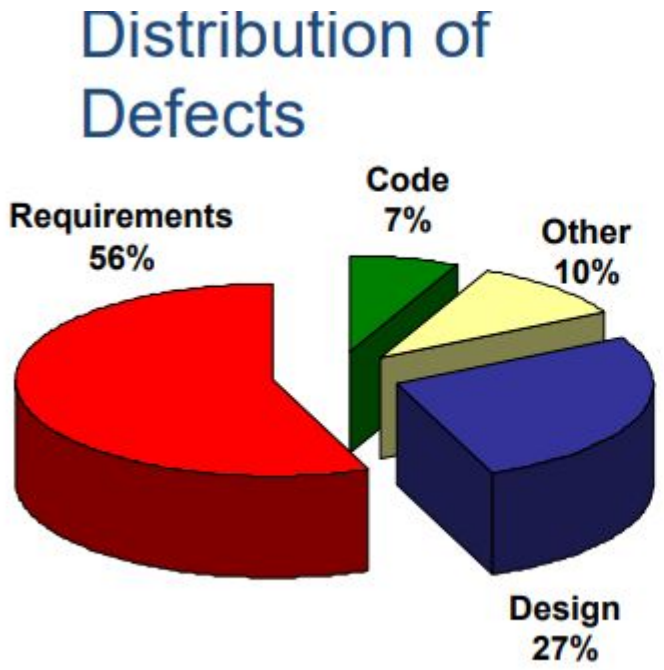
- Defect is deviation from customer requirement.
- Mostly defects are found in the software after software is shipped to the customer at production site.

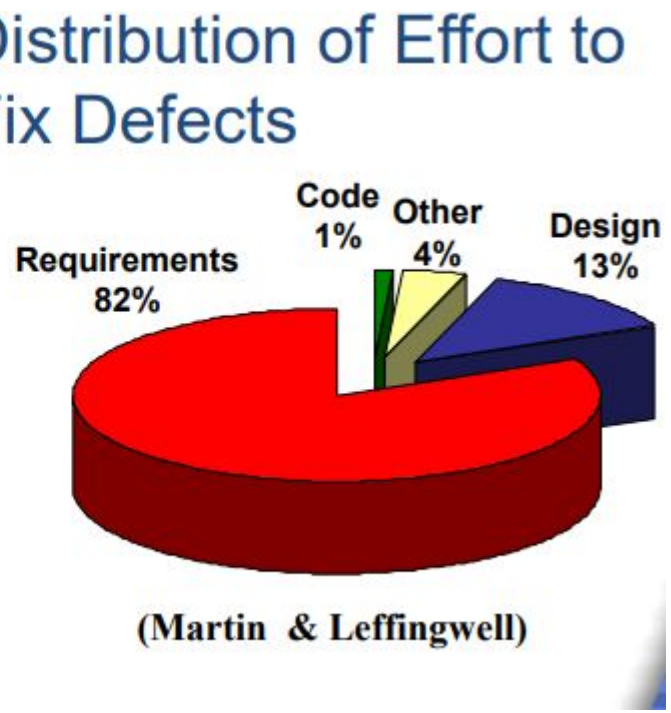
- Example:
- In online shopping, the option of searching a debit card for making payment is missing.

Defects


Customer Requirements (100%)
80% Bacteria
20% Pure
Requirement Defects
Filter 1
60% Bacteria
40% Pure
Design and Planning Defects
Filter 2
40%
60%
Coding Defects
Filter 3
20%
80%
Testing Defects
Filter 4
99.9 % Bug Free Product

Distribution of Defects





Failure Root Cause

- *Standish Group in 1994 study three most commonly factors that cause projects to be challenged*

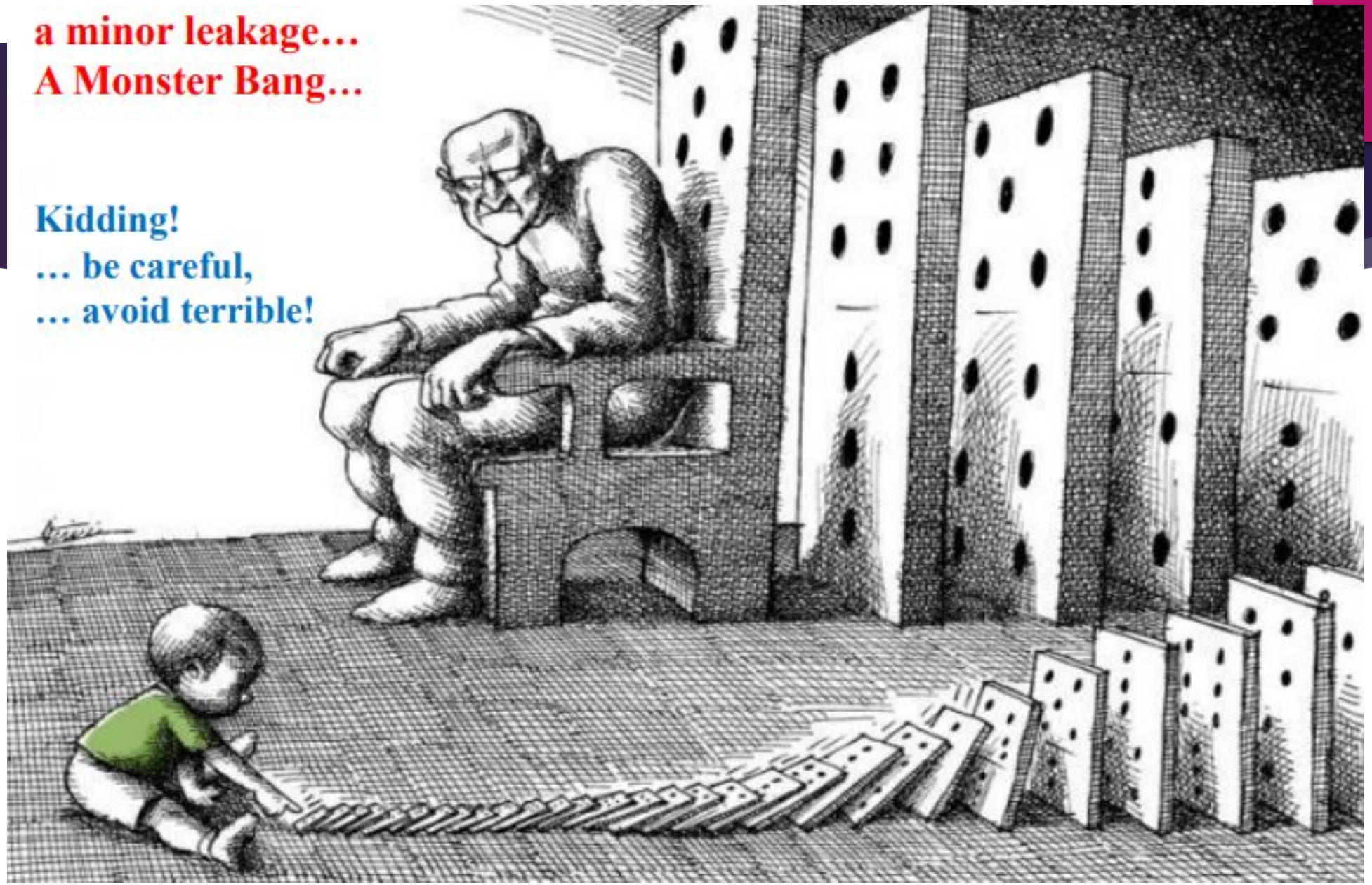
  - Lack of User Input
  - Incomplete Requirements & Specification
  - Changing Requirements and Specification

Failure Root Cause

- Standish Group Study in 1994:
- Observation for Failure: Why 1/3 Projects run in trouble, it is directly related to
  - Requirements Elicitation
  - Requirement Documenting
  - Requirement Management
- Standish Survey: Coding issue were a “non-problem.

a minor leakage…
A Monster Bang…

Kidding!
… be careful,
… avoid terrible!



Impact of Requirement Defects

- Leakage into other phases
  - Design, Code, Implementation, Maintenance etc.
- Impact
  - re-requirement,
  - re-design,
  - re-code,
  - Re-testing,
  - re-implementation,
  - re-deployment, re-training .
  - increase cost, extra support & service cost.

What are Missing Quality Requirements?

In a real-time scenario, more budgets mean more quality. This is both theoretically and practically true.

Putting in more money for quality of the software product will result in low probability of product failure and may save a lot of financial resources as the high quality product will be immune to threats.

Prioritize areas which are non-negotiable.

- For example, a software product with excellent User Interface (UI) but with no firewall for database security will face more threats. So adding a firewall to ensure Database is secure is more important than spending budget on cosmetic changes in UI.

Missing Quality Requirements

Lack of quality in any product can lead to massive losses but when we talk about lack of quality in software products, we can expect catastrophe. One such scenario occurred when Hackers access personal information associated with at least a half billion Yahoo accounts. This incident was report in 2016 but occurred sometime in late 2014.

What was the ramification? Prior to the announcement of the breach, Verizon negotiated and decided to purchase Yahoo for $4.8 billion and this deal was to be closed in March 2017. But later in February 2017, Verizon and Yahoo announced that the deal will still go forward, but dropping the sale price by $350 million

.

On the other side, user’s confidential information including email, credit card details, bank account details and many others hit the market putting millions of users on stake

Cost Analysis based Approach

Missing Quality in Software Application has direct impact on People and Organizations as seen by the example mentioned previously.

According to **Eppler and Helfert principles** the costs are classified in two categories:

Direct Cost of missing Quality

In-Direct Cost of missing Quality:

Cost Analysis based Approach

Direct Cost of missing Quality

Direct Costs, as the name suggest, are directly linked to the missing quality. The direct costs are effects that are easily observable/measureable and they occur immediately after any unfortunate event. Examples includes; financial loss & physical injury and related. In short, direct costs are visible and measureable.

Cost Analysis based Approach

Indirect Cost of missing Quality

Indirect Costs are invisible cost of missing quality and hence difficult to calculate. It is also, sometime, difficult to realize or identify as they occur after a long time of the incident. Example includes: Loss of market share or reputation, loss of market and shareholders trust and investment. Opposed to the direct cost, these are invisible as they may remain hidden for pretty long time, may have long-term impact as well.

Scenario of Nokia serves a good example, its CEO said in May-2016 in his farewell speech: **"We didn't do anything wrong but somehow we Lost".**

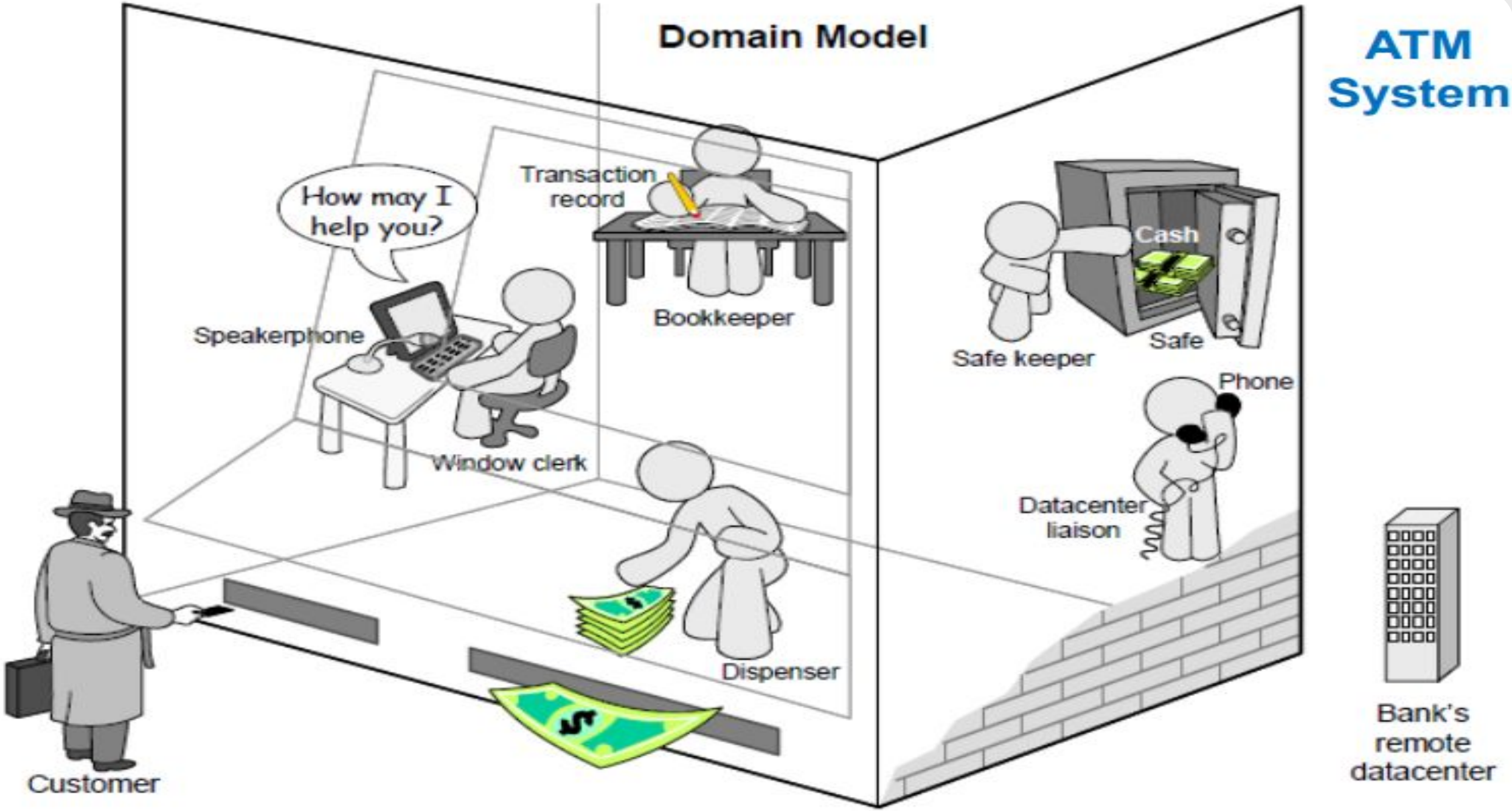
RE Related Activities

1. **Functional Requirements**: describes what a software system should do
2. **Non-Functional:** place constraints on how the system will do so.
3. **Business Processes**: procedure or event with the purpose of reaching a goal
4. **Scope:** required tasks to accomplish, boundary of system
5. **Goals**: An observable and measurable end result having one or more objectives to be achieved within a more or less fixed timeframe
6. **Stakeholders:** person, group or organization that has interest or concern in an organization

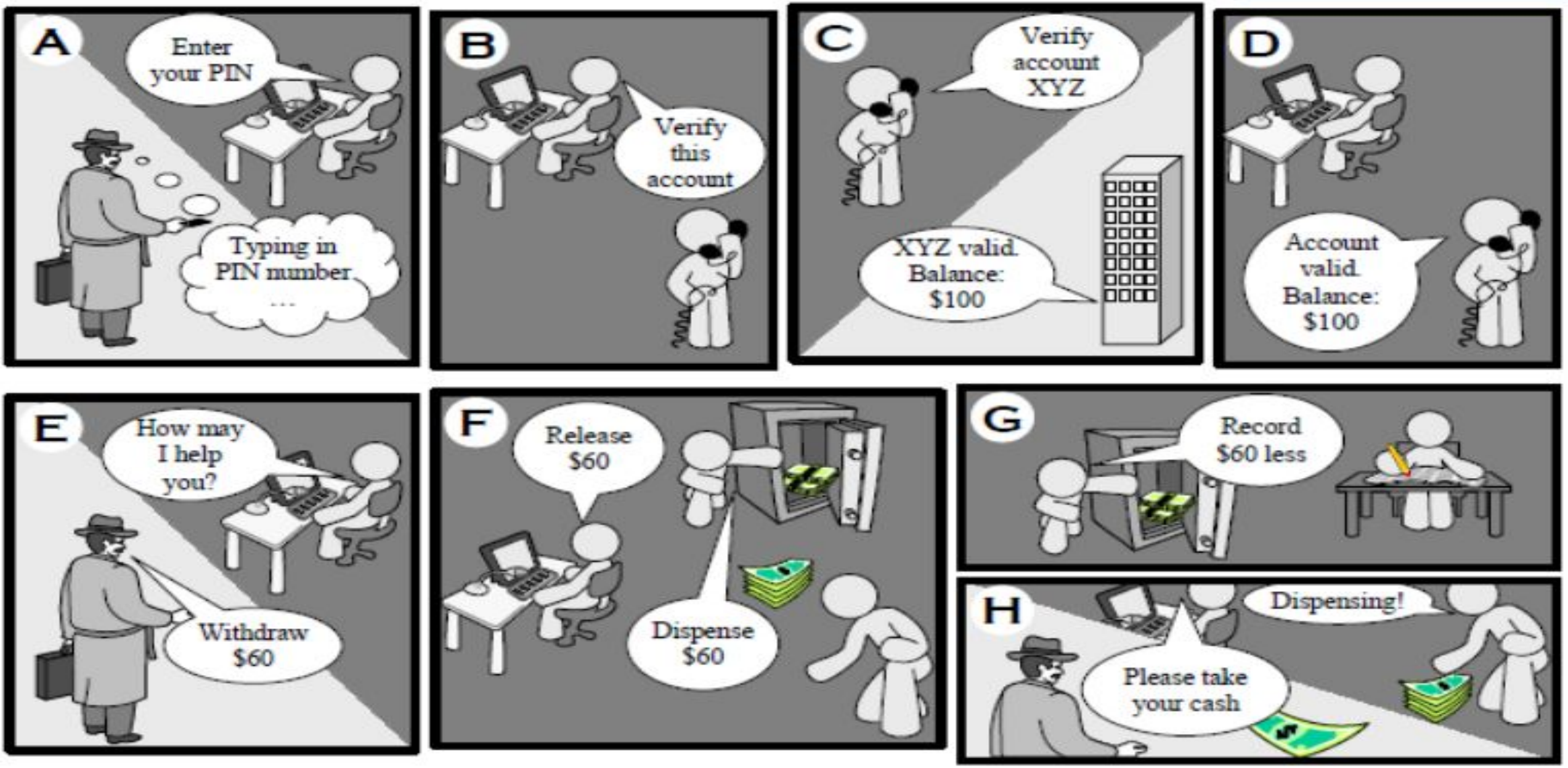
7. **Sources:** someone or something that provides what is wanted or needed i.e. human, documents, context, situational factors, application types etc.
8. **Feasibility study:** a feasibility study is an analysis of the viability of an idea

RE Related Activities

9. **GUI**: Graphical User Interface
10. **Traceability:** concerned with documenting the relationships between different development artifacts (i.e. requirements and other artifacts).
11. **Measureable**: objectives should be measurable and achievable
12. **Domain**: area, business, discipline, field, realm, sphere
13. **Prototyping:** An easily modified and extensible model (representation, simulation or demonstration), partial or approximation of final product, useful for clarifying requirement.



Imagined static structure of ATM shows internal components and their roles.

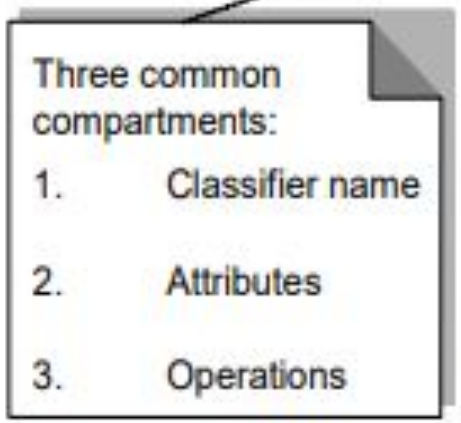


Dynamic interactions of the imagined components during task accomplishment.

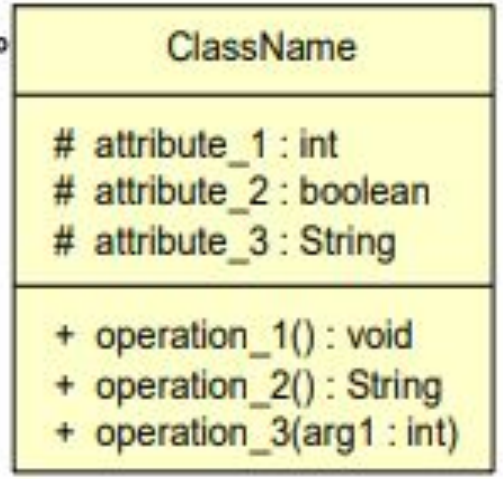
ATM System

UML = Unified Modeling Language

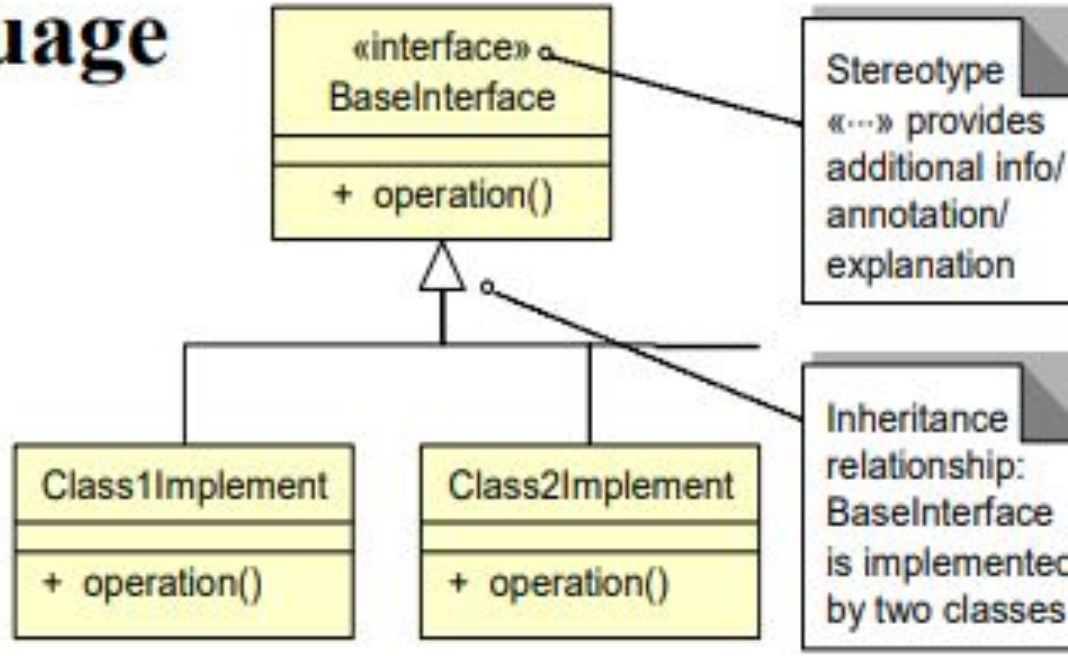
Actor



Comment

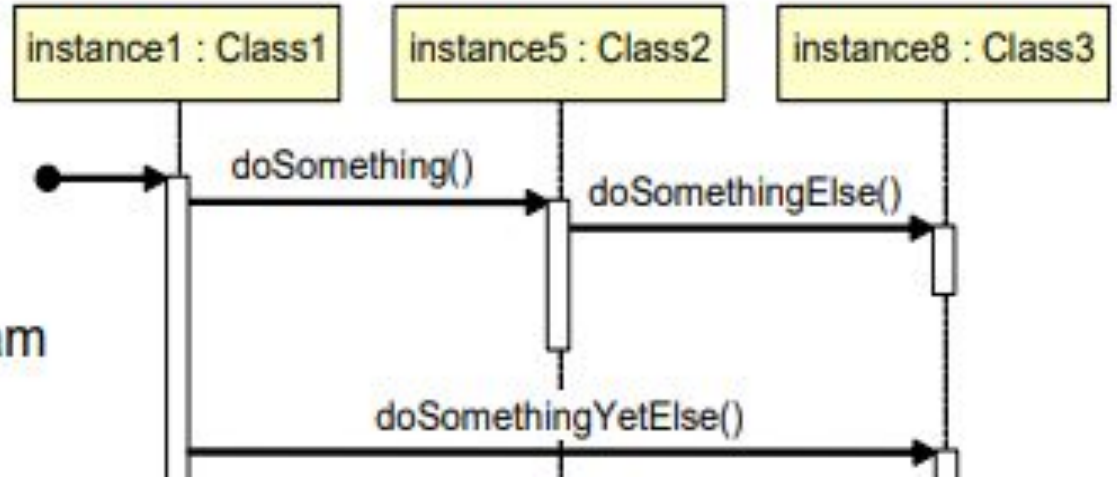


Software Class



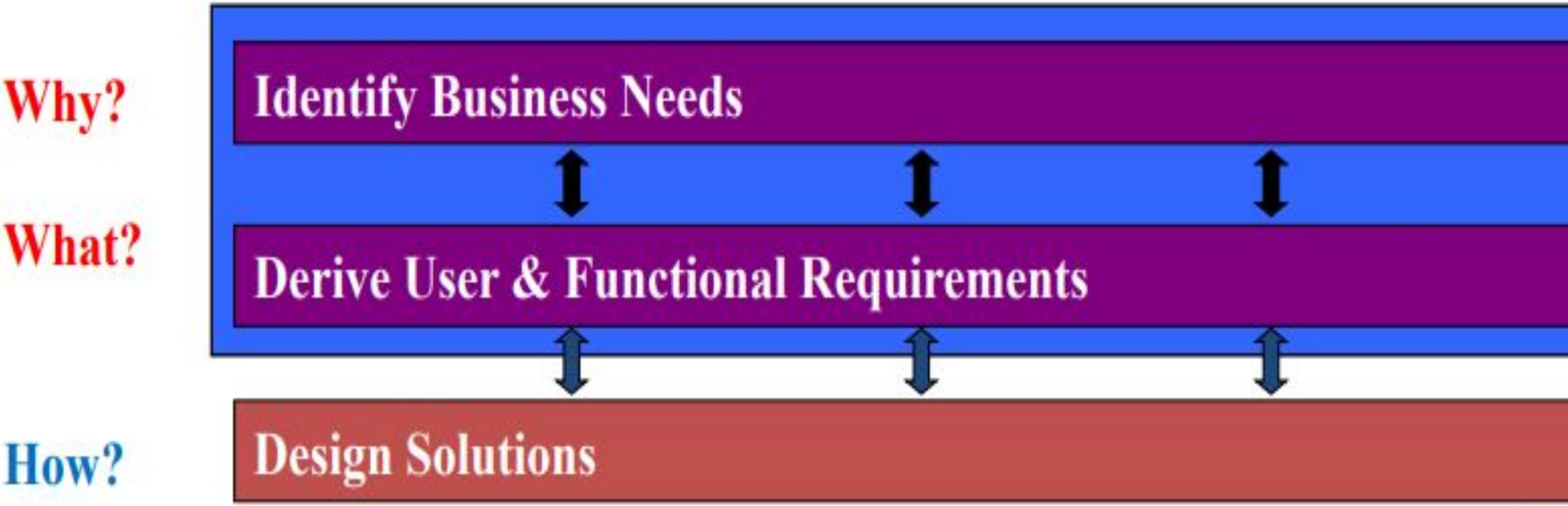
Software Interface Implementation

Interaction Diagram



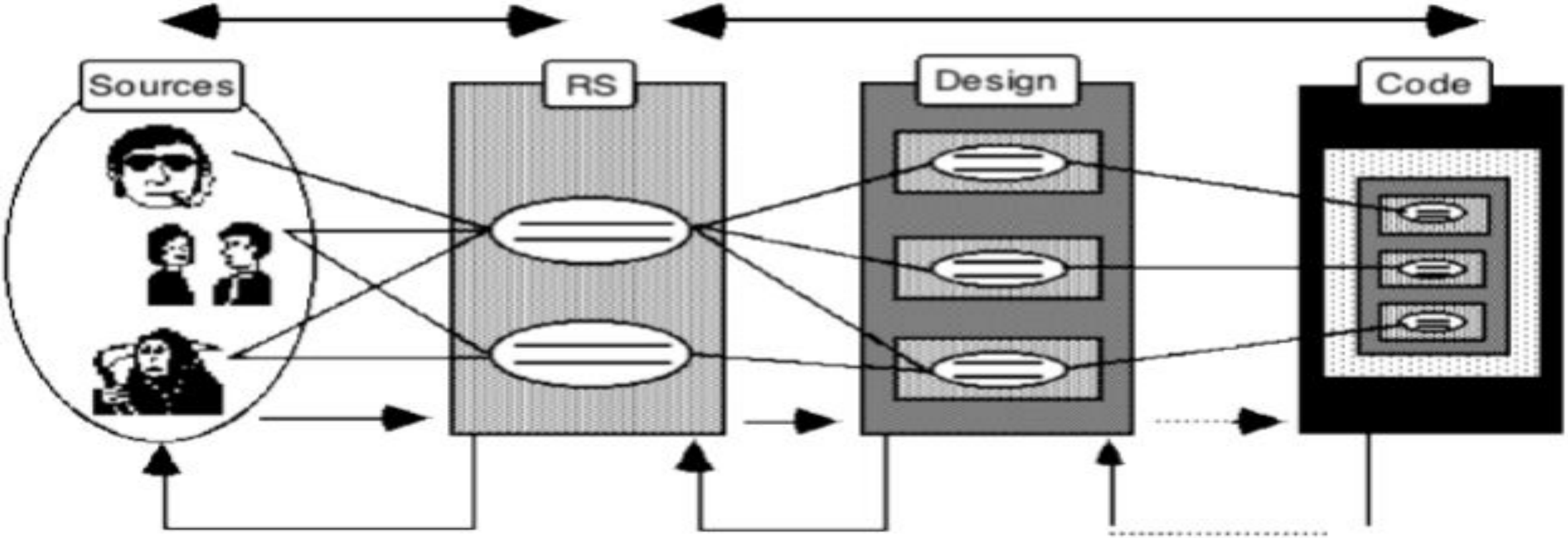
RE Process and Related Activities:

Why? What? How?



What , How, Why?

A Simplified Picture



Requirements traceability matrix

- A requirements traceability matrix is a document that demonstrates the relationship between requirements and other artifacts. It's used to prove that requirements have been fulfilled. And it typically documents requirements, tests, test results, and issues.

Requirements Traceability Matrix

Project Name	<Project Name here>	Created On	3-Oct-11	Reviewed On	4-Oct-11
Release No	<Project Release>	Created By	<Creater's Name>	Reviewed By	<Reviewer's Name>
Version	<Doc version>				

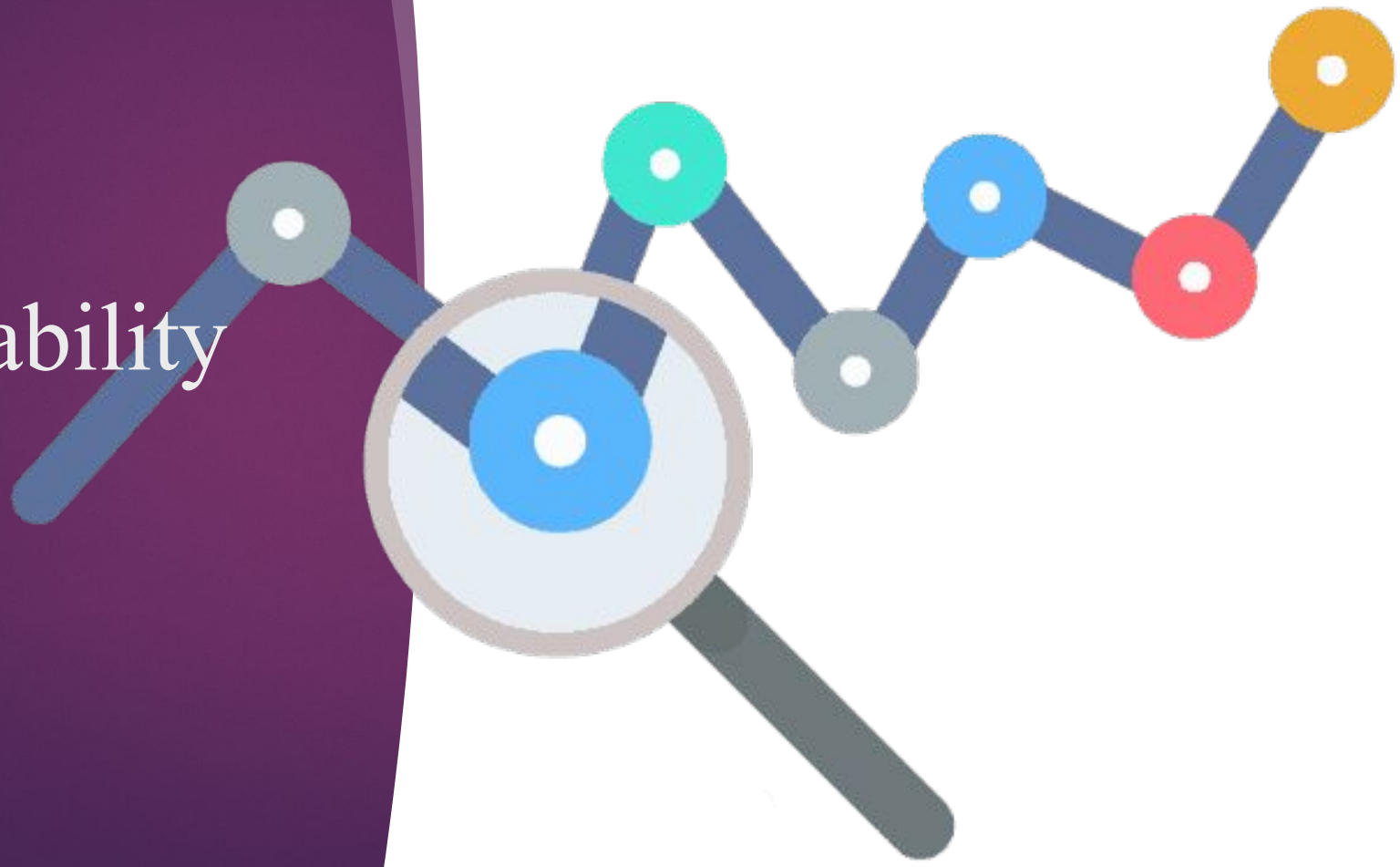
ID	Requirement ID	Requirement Description	Status	Design Document	Code Module	TestCase ID	Test Case Name	User Manual	Tested On/ Verification
001	UC 1.0	Testing Requirement Description here. It should not be more than 2-3 lines	Status	DM-001	CM-001	TC-001	ProjName_UCID_TestCase Name	Section 4.5	Pending
002	UC 1.1	Testing Requirement Description here. It should not be more than 2-3 lines	Approved	DM-002	CM-002	TC-002 TC-003	N.A	Section 4.6	Verified
003	UC 1.2	Testing Requirement Description here. It should not be more than 2-3 lines	Status	DM-003	CM-003	TC-004 TC-006 TC-007 TC-008		Section 5.7	Verified
004	UC 1.3	Testing Requirement Description here. It should not be more than 2-3 lines	Approved	DM-004	CM-004			Section 6.8	In-progress
005	UC 1.4	Testing Requirement Description here. It should not be more than 2-3 lines	Approved	DM-005	CM-005			Section 7.9	Not Verified
006	UC 1.5	Testing Requirement Description here. It should not be more than 2-3 lines	TBD	DM-006	CM-006			Section 4.10	Verified
007	UC 1.6	Testing Requirement Description here. It should not be more than 2-3 lines	Approved					Section 4.11	Not Verified
008	UC 1.7	Testing Requirement Description here. It should not be more than 2-3 lines	TBD					Section 4.12	Pending
009	UC 1.8	Testing Requirement Description here. It should not be more than 2-3 lines	TBD					Section 4.13	Not Verified
010	UC 1.9	Testing Requirement Description here. It should not be more than 2-3 lines	Approved					Section 4.14	Pending

Requirements			Testing		Deviations
Marketing Requirements	**Product Requirements**	**System Level Specifications**	**Test Cases**	**Test Runs**	**Issues**
URS-305	→ US-313 - Warning Tones **Risk-Level:** Low As a nurse I would like to be able to have audible alerts from the nurse's station that will let me know when I am not in the patient's room and something happens to the WysiPump. As a team we need to gather information for these audible tones	→ PHYS-168 - Audible Tone eSigned & Approved ***Component:*** *Hardware* The WysiPump needs to be able to communicate without having to be connected to the WysiStation as they are stand alone products. The WysiPump has 5 distinct tones that can be played though the imbedded speaker at various intervals and scale. • One Tone - on one second off one second. The device needs to be set for this medication. • Two Tones - two fast tones followed by a 5 second silence (repeated constantly) The device is out of medication • Three Tones - three fast tones followed by a 5 seconde silence (repeated constantly). - The device lost BPL connection to the WysiStation • Three Tones - three long tones followed by a 5 second silence (repeated constantly). - The device gained BPL connection to the WysiStation • Four Tones - four fast tones followed by a 5 second silence (repeated constantly) - The device is clogged • Five Tones - five fast tones followed by a 5 second silence (repeated constantly) - The device is low on battery and will shut down in 20 minutes • SOS - three short tones, three long tones, three short tones (repeated constantly) - The device is out of power and is shutting down in 5 minutes	→ TP-20 - TC<-Req - (PHYS-168) -Audible Tone eSigned & Ready Verified to Test ***Type:*** *User Acceptance* ***Estimated Time:*** 00:02:00 ***Last Modified:*** *9/29/2016 3:46 PM*	→ TR-54 - TR<-TC<-Req - (PHYS-168) -Audible Tone Passed ***Variants:*** *Sanxo DTX-1200 Vendor C* ***Actual Time:*** *0.25 - 0.25*	
				→ TR-55 - TR<-TC<-Req - (PHYS-168) -Audible Tone Passed ***Variants:*** *Sanxo DTX-1200 Vendor A* ***Actual Time:*** *0.30 - 0.30*	
				→ TR-56 - TR<-TC<-Req - (PHYS-168) -Audible Tone Failed Code does not work; Cat-5820 (add new medication bag as the old one is still being infused. ***Variants:*** *Sanxo DTX-1200 Vendor B* ***Actual Time:*** *0.34 - 0.34*	↔ ISS-81 - ISS<-TR<-TC<-Req - (PHYS-168) -Audible Tone -Can't add new medication bag as the old one is still being infused. In Development ***Development Notes:***
				→ TR-3367 - TR<-TC<-Req - (PHYS-168) -Audible Tone In Progress wont allow ***Variants:*** *Samchung SC-66* ***Actual Time:*** -	↔ ISS-86 - ISS<-TR<-TC<-Req - (PHYS-168) -Audible Tone -wont allow In Development ***Development Notes:***
				→ TR-3368 - TR<-TC<-Req - (PHYS-168) -Audible Tone Not Started ***Variants:*** *KyroChip KC-400* ***Actual Time:*** -	
				→ TR-3369 - TR<-TC<-Req - (PHYS-168) -Audible Tone Not Started ***Variants:*** *Sanxo DTX-1200* ***Actual Time:*** -	
	→ URS-316 - Nurses Station **Risk-Level:** Low The WysiPump will connect to the nurse's station to alert of certain changes to the system such as: • low medication bags • empty medication bags • unplugged device • battery low	→ SYS-145 - Notification Needs Review ***Component:*** *Patient Button: Software* The device shall have the ability to notify when medication bags are low and the patient is low on PCA controlled medication.The device shall beep when medication becomes low and shall beep three times every five minutes when medication becomes critically low.			

Who Needs Requirement Traceability?

- Every industry that produces software or hardware could use requirement traceability. But it's a pressing need for industries with something to prove.
- Heavily regulated industries need traceability to prove compliance. These are typically quality- and safety-critical industries.

Creating traceability matrix



1. Define Your Goal

- Your first step when creating a traceability matrix in Excel — or creating a traceability matrix, period — is to define your goals.
- What do you want to deliver with your requirements traceability matrix?
- Here are some example goals:
- I want to create a traceability matrix to prove that I've met compliance requirements for my product.
- I want to create a traceability matrix to make sure that my requirements have been tested and passed before I ship.
- I want to create a traceability matrix so that I know which tests and issues are impacted if a requirement changes.
- By setting your goal before you begin, you'll make sure you're gathering the right information for your traceability matrix.

2. Define and Gather Your Artifacts

- You'll need to define which artifacts should be included, based on your goal(s).
- At its most basic, a traceability matrix should include:
  - Requirements
  - Tests
  - Test results
  - Issues
- Once you've defined your artifacts, you'll need to gather them. This might mean tracking down the most recent requirements document. Each requirement listed should have a unique requirements ID. And this ID should not change if your requirements are reordered.
- You'll also need to track down your test cases and results. If testing is in progress or completed, you'll need to find test statuses.
- If any tests have failed, you'll also need to gather any issues that may have been detected.

3. Create Your Requirements Traceability Matrix Template

- Once you've defined and gathered your documents, you're ready to make your template.
- You'll need to add a column for each of your artifacts. For a basic traceability matrix, your columns will be:
- • Column 1: Requirements
- • Column 2: Tests
- • Column 3: Test Results
- • Column 4: Issues
- Then, you'll be ready to start adding your artifacts in the columns you've created.

5.Update the Traceability Matrix — Constantly

- It’s one effort to create a traceability matrix. But it’s a full-time job to keep it updated and do it right.
- If a requirement changes, you’ll need to update the traceability matrix. Or there might be requirements you decided not to fulfill — and you’ll need a way to indicate that, too. If someone adds a test case, you’ll need to update the matrix. When a test run passes or fails, you’ll need to update it again. If an issue found in testing is resolved, you’ll need to update it yet again.
- Remember to keep a close eye on your requirement IDs. Those should stay the same, even if you reorder your requirements list or reuse a requirement.

Benefits of traceability matrix

Benefits of Using a Requirements Traceability Matrix

- There are six key benefits of using a requirements traceability matrix.
- You'll:
- • Get visibility across development.
- • Make better decisions (e.g., on requirements change).
- • Accelerate release cycles.
- • Rest easy knowing your requirements are fulfilled.
- • Prove compliance faster.
- • Pass audits without fear.

REQUIREMENT INSPECTION

REQUIREMENT INSPECTION HELPS IDENTIFY DEFECTS IN THE REQUIREMENTS EARLY ON.

Inspection Participants

- Appropriate inspectors of a requirements document include its author (typically a requirements analyst), another skilled requirements analyst, the project manager, and actual users or marketing representatives. Anyone who has to do work based on the document also brings a vital perspective.
- These downstream "victims" of the requirements include architects, designers, system test engineers, documentation writers, and support representatives.

- The user viewpoint is critical. Inspectors who are not users cannot accurately judge whether the specification correctly addresses the users' needs. Non-user inspectors often guess at what the users need and add extra requirements.

The Inspection Meeting

- During the inspection meeting, the reader describes his interpretation of each requirement in his own words. Such paraphrasing allows the other participants to compare their understanding with that of someone other than the author.
- Differences in interpretation can reveal omissions and surface assumptions. Inspection is a powerful method for uncovering ambiguities, in which different readers interpret a requirement in different ways.
- If the reader has difficulty describing a requirement, perhaps it is too complex, poorly expressed, or incorrect.


CLASS
ACTIVITY

Requirements for an online shopping website

- The requirement states that the "shopping cart should allow users to add items.
- The requirement states that the website should allow users to "sort products by price "
- The requirement states that the website should allow users to sort products by popularity.“
- The requirement says "the website should have a user-friendly interface,"
- The requirement only states that the website should "support multiple payment options,"
- The requirement states that the website "should load quickly,"
- The requirement states that the website should "support augmented reality product visualization,“

Flaws identified

- Ambiguous Requirements: The requirement states that the "shopping cart should allow users to add items," but it does not specify if there are any limitations on the number of items that can be added.
- Inconsistent Requirements: The requirement states that the website should allow users to "sort products by price," but later in the document, it says "sort products by popularity."
- Unclear Requirements: The requirement says "the website should have a user-friendly interface," but it does not specify how the interface should be structured or what specific functionality should be included.

- Incomplete Requirements: The requirement only states that the website should "support multiple payment options," but it does not specify what specific payment options should be supported, such as credit card, PayPal, or Apple Pay.
- Unverifiable Requirements: The requirement states that the website "should load quickly," but it does not specify a performance metric or benchmark to measure the website's load time.
- Invalid Requirements: The requirement states that the website should "support augmented reality product visualization," but it later becomes clear that the target market or user base does not have the necessary technology to use augmented reality features.

An attempt after flaws removal

- "The shopping cart should allow users to add an unlimited number of items."
- "The website should allow users to sort products either by price or popularity."
- "The website should have a user-friendly interface that is structured in a way that enhances the user experience and includes necessary functionality."

- "The website should support a variety of payment options such as credit card, PayPal, and Apple Pay."
- "The website should load in less than 2 seconds and this will be measured using [performance metric or benchmark of choice]."
- "The website should have alternative visualization options for products, as the target market or user base may not have the necessary technology for augmented reality features."

2. اَللّٰھُمَّ صَلِّ عَلٰی مُحَمَّدٍ وَعَلٰی آلِ مُحَمَّدٍ کَمَا صَلَّیْتَ عَلٰی اِبْرَاھِیْمَ وَعَلٰی آلِ اِبْرَاھِیْمَ اِنَّکَ حَمِیْدٌ مَجِیْدٌ اَللّٰھُمَّ بَارِکْ عَلٰی مُحَمَّدٍ وَعَلٰی آلِ مُحَمَّدٍ کَمَا بَارَکْتَ عَلٰی اِبْرَاھِیْمَ وَعَلٰی آلِ اِبْرَاھِیْمَ اِنَّکَ حَمِیْدٌ مَجِیْدٌ.

اے اللہ! اپنی رحمت نازل فرما محمد پر اور آل محمد پر جیسا کہ تو نے اپنی رحمت نازل فرمائی ابراہیم پر اور آل ابراہیم پر۔ بیشک تو تعریف والا اور بزرگی والا ہے۔ اے اللہ! برکت نازل فرما محمد پر اور آل محمد پر جیسا کہ تو نے برکت نازل فرمائی ابراہیم پر اور آل ابراہیم پر۔ بیشک تو تعریف والا اور بزرگی والا ہے۔

(صحیح البخاری:3370)