

Requirements Traceability

Software Requirements Engineering

By: Zohaib Ahmed

Traceability

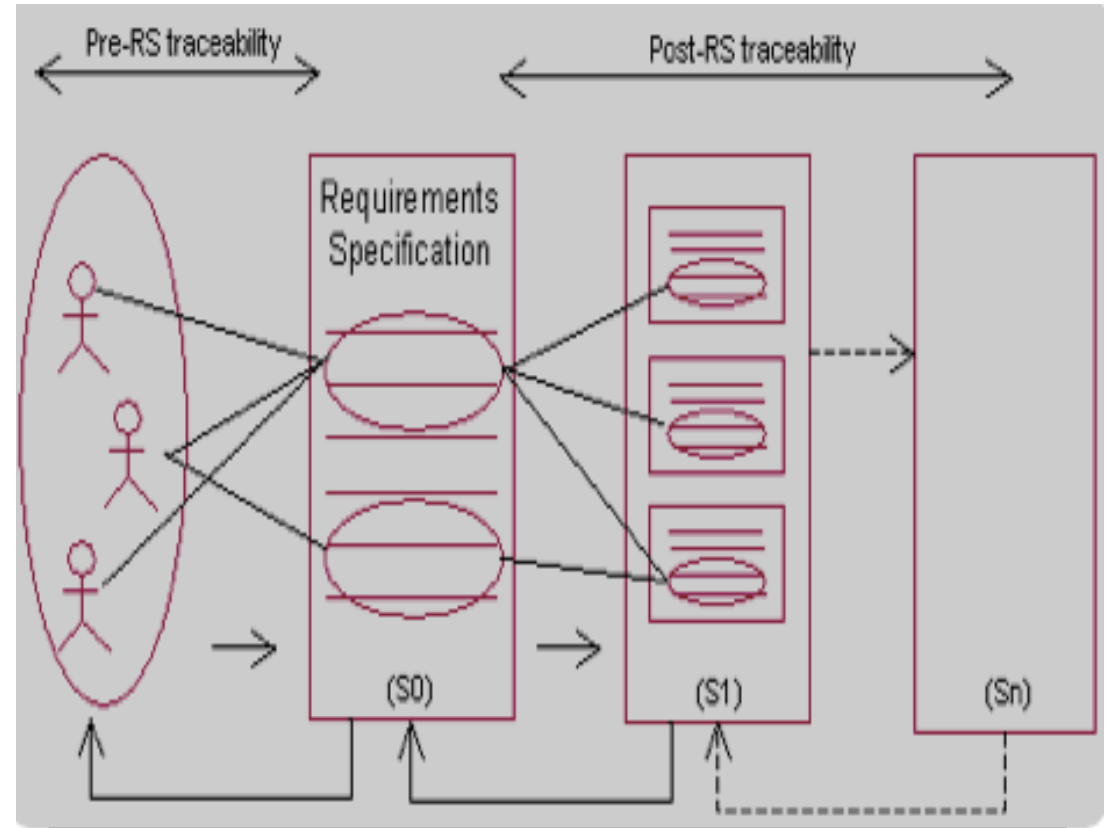
- **Traceability** comprises of two words i.e. trace and ability
- Trace means to find someone or something and ability means to a skill or capability or talent to do something
- Therefore, traceability simply means ***the ability to trace the requirement, to provide better quality, to find any risk, to keep and verify the record of history and production of an item or product by the means of documented identification***
- Due to this, it's easy for the suppliers to reduce any risk or any issue if found and to improve the quality of the item or product
- So, it's important to have traceability rather than no traceability
- Using traceability, finding requirements, and any risk to improve the quality of the product becomes very easy

Requirements Traceability

- According to ISO/IEC/IEEE 24765:2010, traceability is the
‘Discernable association among two or more logical entities, such as requirements, system elements, verifications and tasks.’
- The Guide to Business Analysis Body of Knowledge BABOK v3 concretes this definition for the special case of Requirements Traceability:
‘The ability for tracking the relationships between sets of requirements and designs from the original stakeholder need to the actual implemented solution.’

Pre and Post-Requirements Traceability

- According to BABOK v3, the relationship between requirements and their source – the stakeholders with their needs – should be able to be understood in both directions. This type of traceability is known as **Pre-Requirements Traceability**
- Likewise, the relationships between requirements, the draft, code and the tests should be able to be followed in both directions. This is referred to as **Post-Requirements Traceability**
- Both forms of requirements traceability help answer the many questions that emerge over the course of the development process
 - Examples of questions of pre-requirements traceability: to whom is the requirement important, and how? How will the need be met?
 - Examples of questions of post-requirements traceability: has a requirement been changed? What is effected by the changes to the requirement? Are artefacts like test cases lacking?



Classifications of Requirements Traceability

1. Forward to Requirements

When customer needs evolve, requirements may have to be adjusted in response. By making these adjustments, project teams can keep pace with changes in customers priorities, introductions of new business rules, and modifications of existing rules, among other events.

2. Backward From Requirements

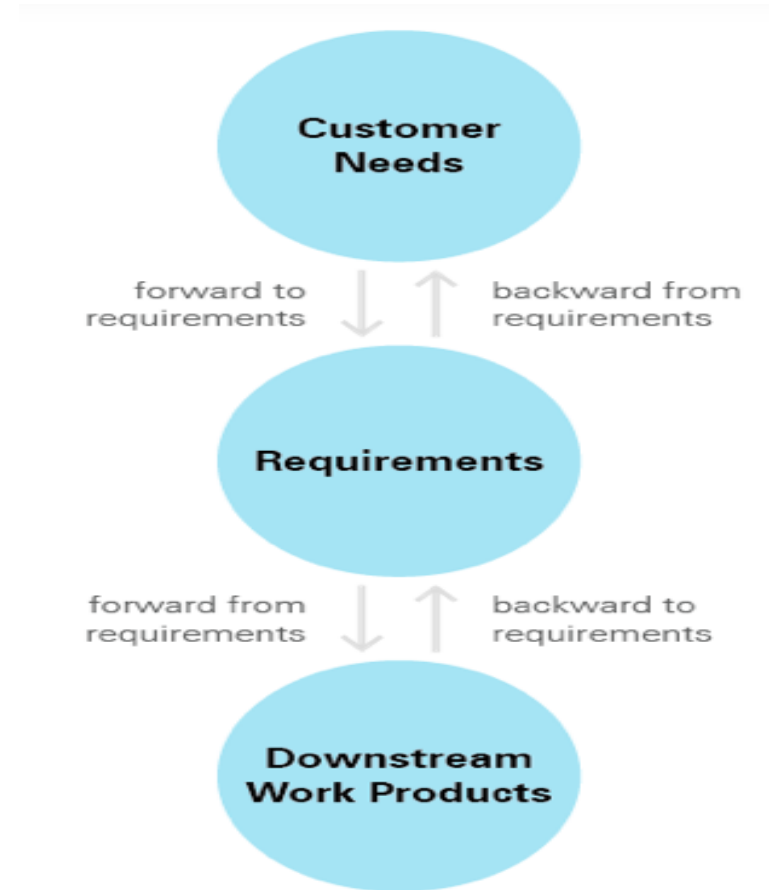
Tracking backward from requirements can provide clarity into the origin of each derived requirement. For instance, a requirements management tool could show the link between the derived requirement, the requirement it came from, and the customer use case being addressed.

3. Forward From Requirements

Once derived requirements begin flowing into downstream deliverables during product development, it's possible to draw trace relationships between requirements and their corresponding elements. This type of link provides assurance that every requirement is satisfied by a particular component.

4. Backward to Requirements

Finally, this type of link allows for visibility into why certain features were created. Consider how most applications include lines of code that don't directly relate to stakeholder requirements. Even so, it is important to know why a software engineer wrote that code in the first place.



Types of Traceability

- **Source traceability** – These are basically the links from requirement to stakeholders who propose these requirements.
- **Requirements traceability** – These are the links between dependent requirements.
- **Design traceability** – These are the links from requirement to design.
- **Testing traceability** – These are the links between requirements and test cases, which ensure that each requirement has been properly tested.
- **Code traceability** – These are the links between the requirements and the actual code that is developed to implement those requirements.
- **Version traceability** – These are the links between different versions of software or documents, which allow for tracking of changes and updates over time.
- **Release traceability** – These are the links between the requirements and the specific software release or version in which they were implemented.
- **Risk traceability** – These are the links between risks identified in the project and the mitigating actions taken to address those risks.
- **Business traceability** – These are the links between project requirements and overall business goals and objectives.
- **Quality traceability** – These are the links between requirements, design, testing, and implementation, which ensure that quality is maintained throughout the software development process.

Traceability Matrix

- Traceability matrix is generally used to represent the information of traceability
- For mentioning the traceability of small systems, usually the traceability matrix is maintained
- If one requirement is dependent upon another requirement then in that row-column cell 'D' is mentioned
- If there is a weak relationship between the requirements, then corresponding entry can be denoted by 'R' as shown in the figure.

Requirement ID	A	B	C	D	E	F
A		D			R	
B			D			
C				R		
D			D			R
E						
F	R			D		

Traceability Matrix Structure

