# Software Traceability Concepts

In recent years, Software Traceability research has prominently focused on requirements traceability and aiming at studying how to describe and follow the life of a requirement in both forward and backward directions. Tracing links between software requirements and other project artifacts, can support number of software activities. Among many, most traceability research attempt to address the following activities:

* System Verification and Validation
* Change impact analysis
* Regulatory compliance
* Maintaining the traceability Link

# Traceability Tools and Features

Based on traceability research review paper, the below are the list of traceability tools developed by researchers in the past.

|  |  |  |
| --- | --- | --- |
| **Tool** | **Features** | **Trace Link** |
| Ecolabor | A tool for using hypermedia to maintain traceability between different artefacts. Provide features of creating, | Traces requirement Life Cycle |
| TOOR | A tool for recording traces between requirements. | Maintain Trace Between requirements and managing items |
| Poirot | Implements a probabilistic approach to dynamically generate traceability links | Semi-automated web-based traceability tool for heterogeneous software artifects |
| DOORS | A tool that extends on DOORS for change management | Change management |
| RESAT | A tool allows users to automatically access if a design description meet its requirement | Trace design description meets requirements |
| CREWS-EVE | Offers multimedia support and animation to visualize traceability to test cases. | Features for visualizing the traces maintained between artefacts |
| STRADA | Scenario-based Trace Detection and Analysis tool helps software engineers explore trace links to source code through testing. |  |

# Types of System Researcher Trace

Below are the types of software system trace by researchers in the past.

|  |  |
| --- | --- |
| **System Type** | **Description** |
| Information Systems | Which store, process and show data for their users |
| Safety-critical Systems | Real-time embedded system |
| Real-time embedded system | Which are subject to real-time constraints |
| Non-software system | Physical documents managed in an organization. |

# Frequent Trace Artefacts

|  |  |
| --- | --- |
| 1 | Requirements and Testing |
| 2 | Requirements and Design |
| 3 | Requirements and Code |
| 4 | Trace Between Requirement specifications Artifects |
| 5 | Requirements and Source |

# Traceability Challenge in Research & Practice

|  |  |
| --- | --- |
| **Challenges** | **%** |
| Lack of knowledge and understanding about traceability | 17.1% |
| Maintaining traceability when requirement evolve | 12.9% |
| Impact of human factors and Judgement in Traceability | 8.6% |
| Cost related to requirements traceability | 8.6% |
| Effective representation of tractability Information | 8.6% |
| Challenge face by practitioners in industrial project settings | 5.7% |
| Accessing the traceability maintained | 1.4% |

# Research Ideas

|  |  |
| --- | --- |
| Trace link | Trace Between Post- Requirements Spec/ Design Spec and Source Code |
| System Type | Information Systems |
| Trace Purpose | System Verification and Validation |
| Tool | Develop Visualizing Tools |
| Evaluate Tool | UTS Library Web Site as Case Study to evaluate the tool |