



eXcellence In Variant Testing



Project References

| | | | |
|--------------------|---|------------------|-----------|
| Project Acronym | XIVT | | |
| Project Title | eXcellence In Variant Testing | | |
| Project Number | 17039 | | |
| Project Start Date | November 1, 2018 | Project Duration | 36 months |
| Project Manager | Gunnar Widforss, Bombardier Transportation, Sweden | | |
| Website | https://www.xivt.org/ | | |

Document References

| | | | |
|---------------------|--|----------------|----------------------|
| Work package | WP2: Knowledge-Based Test Optimization | | |
| Work package lead | Mehrdad Saadatmand, RISE Research Institutes of Sweden (mehrdad.saadatmand@ri.se) | | |
| Deliverable | D2.6: Test optimization tool set: final version | | |
| Deliverable type | Software (SW) | | |
| Dissemination level | Public (including description of private artifacts) | Date & Version | Mar 29, 2022 V1.0 |
| Mapped tasks | T2.1: Knowledge-based testing for product variants T2.2: Test optimization criteria for variant-intensive products T2.3: Test object- and feature- based optimization T2.4: Test prioritization and selection for variant-intensive systems | | |

Executive Summary

This deliverable reports on the final list and versions of the tools and software artifacts that are developed and extended in the XIVT project in the scope of WP2 for optimization of the test process of variant-intensive systems. It includes a set of publicly available and also private artifacts. While the public ones can be openly accessed through the provided links, to get access to the private ones, respective partner can be contacted. The partners reserve the right to change the licensing and accessibility model of their solutions and tools at any time.

Tools

1. VARA: Variability Aware Reuse Analysis (RISE)

VARA is variability-aware requirements reuse analysis method which aims to automate the product line assets reuse analysis and thus helps teams achieve quick and quality delivery of software systems. VARA takes customer requirements as input and uses state-of-the-art natural language processing and machine learning algorithms to predict existing product line assets that can be reused to realize the customer requirements.

VARA is under development for further improvements in close collaboration with Alstom as part of their XIVT use case solution. VARA will help teams reduce the time to market by at least 20 days. Currently, VARA recommends reuse with around 82% of average accuracy. Furthermore, qualitative evaluation of VARA shows that the recommendations produced by VARA are perceived as useful and insightful by engineers.

Webpage: N/A

Source or Binary Link: privately maintained - contact for access

Instruction manual for the tool: N/A, Confidential

Type: Closed Source

Version: 0.2

Updates wrt D2.2: In this update, VARA has undergone some interface changes, making it more modular. An installation manual and user manual is now also provided with the tool. In addition, the accuracy is improved to around 82% by a direct match with product line requirements.

Additional Info: N/A

2. MBRP: Model-Based Requirements Prioritization (RISE)

MBRP is an open-source tool for modeling and prioritization of software requirements that takes in natural language requirements, dependencies, and stakeholder's preferences as input and produces a ranked list of requirements using PageRank algorithm. It uses a meta-model-based approach to help the requirements analyst in modeling the requirements, stakeholders, and inter-dependencies between

requirements. The model instance is then processed by a modified PageRank algorithm to prioritize the given requirements. An experiment was conducted, comparing the modified PageRank algorithm's based MBRP's efficiency and accuracy with five existing requirements prioritization methods. Besides, we also compared our results with a baseline prioritized list of 104 requirements prepared by 28 human subjects. Results showed that our MBRP was able to prioritize the requirements more effectively and efficiently than the other prioritization methods.

Webpage: N/A

Source or Binary Link: <https://github.com/a66as/mbrp>

Instruction manual for the tool: (same as above)

Type: Open Source under the GNU General Public License v3.0

Version:

Updates wrt D2.2: No updates

Additional Info: N/A

3. SEAFOX (MDH)

SEAFOX is a combinatorial test generation and measurement tool that supports the generation of test suites using pairwise, base choice and random strategies. For pairwise generation, SEAFOX uses the IPOG algorithm as well as a first pick tie breaker. A developer using SEAFOX can automatically generate test suites needed for a given program variant after manually providing the input parameter range information based on the defined behavior written in the specification.

Webpage: N/A

Source or Binary Link: <https://github.com/acn18/DVA331-SEAFOX-02>

Instruction manual for the tool: (same as above)

Type: Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY- NC-ND 3.0)

Version: 0.1

Updates wrt D2.2: No updates

Additional Info: N/A

4. NALABS (MDH)

NALABS is a tool for measuring bad smells of textual requirements in terms of variability, quality, logical complexity and understandability by applying a set of different measures.

Webpage: N/A

Source or Binary Link: <https://github.com/eduardenoiu/NALABS>

Instruction manual for the tool: (same as above)

Type: MIT License

Version: 0.1

Updates wrt D2.2: No updates

Additional Info: N/A

5. ReForm: NLP-based Requirement Formalization (ifak)

This tool will investigate in analysis of textual requirements with knowledge-based techniques including natural language processing and machine learning to extract models for variant rich systems. It will parse the requirements which are written in natural language in order to extract the relevant information and create requirement models.

Webpage: N/A

Source or Binary Link: https://github.com/ifak-prototypes/nlp_reform

Instruction manual for the tool: (same as above)

Type: Open Source

Version: 0.2

Updates wrt D2.2: The code has been restructured and modularized. Rules have been extended, e.g. a synonym dictionary was added, Boolean values can be handled, etc. A new algorithm for pronoun resolution using AllenNLP is used.

Additional Info: N/A

6. ifakVBT: Test case generation and variant traceability (ifak)

This tool will create a specification model (state machine) from requirement models and automatically generate test cases. Furthermore, it produces a linkage between the test cases and the corresponding requirements, features and variants of a specific model. It will use prioritization information of requirements to prioritize test cases automatically.

Webpage: N/A

Source or Binary Link: privately maintained - contact for access

Instruction manual for the tool: N/A, Confidential

Type: Closed Source

Version: 0.2

Updates wrt D2.2: Traceability information was added to test cases. Performance of test case generation was improved. Minor errors were fixed.

Additional Info: N/A

7. ARRINA (FCUL)

SRXCRM evolved to a recommendation system, now called ARRINA (*Association and Recommendation for Requirements in Natural Language*). ARRINA is a system based on Natural Language Processing (NLP) that is able to process design specifications and customer requirements, both written in unstructured natural language and belonging to two different, but related, domains, in order to prioritize customer requirements for testing software and map them to design specifications in an automated manner. The ARRINA model establishes a pipeline of NLP and Information Retrieval methods and a weight similarity metric to process design specifications and customer requirements.

Webpage: N/A

Source or Binary Link: privately maintained - contact for access

Instruction manual for the tool: (same as above)

Type: Closed Source

Version: 1.0

Updates wrt D2.2: A recommendation system was defined and implemented with a weight metric. No updates, it is the first release version

Additional Info: N/A

8. RiSco: Risk-based test scoring (QAC)

This tool is under development and aims at prioritizing test cases based on risk criteria (e.g. Safety/Security). It is built around a Knowledge-Based System (KBS) that require domain-specific knowledge bases to be collected. It receives abstract test cases as input and search test cases for patterns indicative of such risks, assigns risk scores to the test cases as pertinent.

Webpage: N/A

Source or Binary Link: privately maintained - contact for access

Instruction manual for the tool: (same as above)

Type: Closed Source

Version: 0.1

Updates w.r.t D2.2: No updates