ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands

T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

Exploitable Results by Third Parties

17039 XIVT

Project details

Project leader:	Gunnar Widforss
Email:	gunnar.widforss@pmir.se
Website:	https://www.xivt.org/

Name: MatAdd		
Input(s):	Main feature(s)	Output(s):
▪ Simulink codebase	▪ Clone detection in model-based simulink design ▪ Identify reusable patterns to reduce complexity and cost and facilitate testing and maintenance	▪ Demonstration of detected commonalities ▪ Result matrix

Unique Selling Proposition(s):	▪ Process large-scale industrial Simulink codebases and detect the commonalities and differences between its models.
Integration constraint(s):	▪ Matlab ▪ Simulink code base
Intended user(s):	▪ Currently an internal tool for Alstom.
Provider:	▪ Addiva/Alstom
Contact point:	▪ Daran Smalley
Condition(s) for reuse:	▪ Case-by-case negotiation

Latest update: 21.01.2022

Name: Sketcher 3D		
Input(s):	Main feature(s)	Output(s):
▪ XML from FeatureIDE ▪ Library of 3D parts	▪ Quick generation of 3D overview of selected components for a robot cell	▪ XML for Process Simulate import
Unique Selling Proposition(s):	▪ Quick overview if a 3D arrangement of components is valid for the specific case ▪ Rearrangement is possible with few overhead	
Integration constraint(s):	▪ Need a state-of-the-art computer hardware ▪ Standardized export file from FeatureIDE required	
Intended user(s):	▪ Can be used to create a quote for customers by the seller ▪ Can be used to check the feasibility	
Provider:	▪ EKS InTec GmbH	
Contact point:	▪ Holger Hämmerle holger.haemmerle@eks-intec.de	
Condition(s) for reuse:	▪ License and permission required	
	Latest update: 31.01.2022	

Name: ARRINA - Association and Recommendation for Requirements in Natural Language		
Input(s):	Main feature(s)	Output(s):
▪ Existing Natural Language (NL) Customer Requirements and Design Specifications linked between them ▪ Input requirements written in NL	▪ Automated extraction and interpretation of requirements and specifications ▪ Main requirements and specifications ▪ Representation of the knowledge within a subsystem ▪ Analysis and correlation of requirements and specifications for a Top-K recommendations	▪ Top-K recommendations for requirements to specifications ▪ Knowledge graph representative of the knowledge within a PCS ▪ Associations rules among requirements and among specifications of a subsystem

Unique Selling Proposition(s):	▪ Extracts specifications and recommends which are the most adequate for customer requirements ▪ ARRINA recommends specifications for requirements with an accuracy of 80% ▪ ARRINA achieves a reduction of testing time (mapping specifications for requirements) of 85%
Integration constraint(s):	▪ Python 3 ▪ Graphviz ▪ pandas ▪ NLTK ▪ mlxtend ▪ openpyxl ▪ xlrd
Intended user(s):	▪ Requirements engineers ▪ Software architects ▪ Project managers
Provider:	▪ FCUL: Faculty of Sciences of University of Lisboa
Contact point:	▪ Ibéria Medeiros (ivmedeiros@fc.ul.pt)
Condition(s) for reuse:	▪ open-source restricted and permission required
	Latest update: 02.02.2022

Name: CorCA - Correction of C/C++ Automatically		
Input(s):	Main feature(s)	Output(s):
▪ C/C++ source code of a program to test	▪ Identifies buffer overflows in C/C++ programs and confirm their existence ▪ Removes BO by correcting the code and validate the corrections made ▪ Generate a new release of the program	▪ New version of the program without vulnerabilities ▪ A report with the vulnerabilities found and fixed

Unique Selling Proposition(s):	▪ Identify BO and generate their exploits ▪ Reduce the rate of false positives produced by static analysis ▪ Produce fixes, repair the code, and validate the correctness and effectiveness of the fixes in an automated way ▪ achieves an accuracy and precision close to 100%
Integration constraint(s):	▪ Python 3 ▪ AFL ▪ Flawfinder ▪ pycparser
Intended user(s):	▪ Software developers ▪ Software testers
Provider:	▪ FCUL: Faculty of Sciences of University of Lisboa
Contact point:	▪ Ibéria Medeiros (ivmedeiros@fc.ul.pt)
Condition(s) for reuse:	▪ open-source restricted and permission required
	Latest update: 02.02.2022

Name: DeltaFuzzer		
Input(s):	Main feature(s)	Output(s):
▪ C/C++ source code of a program to test	▪ targeted fuzzing of a specific code location ▪ Identifies buffer overflows in C/C++ programs with confirmation with their existence	▪ A report with the vulnerabilities found and their exploits

Unique Selling Proposition(s):	▪ Identify BO for a specific target code location and provide their exploits ▪ Increases code coverage by 30% ▪ Achieves an accuracy and precision close to 100%
Integration constraint(s):	▪ AFL ▪ LLVM ▪ cmake ▪ ninja ▪ gcc ▪ Python 3
Intended user(s):	▪ Software developers ▪ Software testers
Provider:	▪ FCUL: Faculty of Sciences of University of Lisboa
Contact point:	▪ Ibéria Medeiros (ivmedeiros@fc.ul.pt)
Condition(s) for reuse:	▪ close-source
	Latest update: 02.02.2022

Name: XIVT – Feature-Tree		
Input(s):	Main feature(s)	Output(s):
▪ Feature Library ▪ Feature-Information	▪ Fast configuration of variant-rich hardware robot systems	▪ XML for Sketcher import
Unique Selling Proposition(s):	▪ Fast configuration of necessary hardware components for a robot system with multiple technology applications ▪ Reconfiguration is possible with a small amount of know-how	
Integration constraint(s):	▪ Requires modern computer hardware ▪ Requires basic knowledge of robotic systems	
Intended user(s):	▪ Can be used to lay out a robot system with little information in detail ▪ Can be used to configure hardware variants	
Provider:	▪ FFT Produktionssysteme GmbH & Co. KG	
Contact point:	▪ Martin Kraft (martin.kraft@fft.de)	
Condition(s) for reuse:	▪ License and permission required	
	Latest update: 02.02.2022	

Name: XIVT – Plant Designer		
Input(s):	Main feature(s)	Output(s):
▪ Standard Process Simulate Library ▪ XML from Sketcher	▪ Reducing the gap in the tool chain between planning department and simulation department	▪ Configured simulation environment
Unique Selling Proposition(s):	▪ Quick import of the configuration from Sketcher (Planning departure) ▪ Creation of an initial state of the simulation environment	
Integration constraint(s):	▪ Requires modern computer hardware ▪ Standardized export file from Sketcher required	
Intended user(s):	▪ Can be used to quickly import the configuration of a body in white plant as the starting point of simulation department	
Provider:	▪ FFT Produktionssysteme GmbH & Co. KG	
Contact point:	▪ Martin Kraft (martin.kraft@fft.de)	
Condition(s) for reuse:	▪ License and permission required	
	Latest update: 02.02.2022	

Name: MESIR - Model-based Engineering and Simulation for Industrial Robots with ROS		
Input(s):	Main feature(s)	Output(s):
▪ Domain knowledge ▪	▪ Modeling support by UML Profile ▪ Code generation for ROS	▪ UML Model ▪ ROS packages
Unique Selling Proposition(s):	▪ MESIR allows rapid development of simulation environments for industrial robotic production cells ▪ MESIR reduces the number of languages and formats to learn and ensures consistency among artifacts ▪ MESIR is designed to work with MBPLE (see below) and is part of the XIVT Workbench by Fraunhofer FOKUS	
Integration constraint(s):	▪ Developed system based on Robot Operating System (ROS) ▪ Eclipse platform incl. Eclipse Papyrus	
Intended user(s):	▪ Engineers in system integration for robotic production cells	
Provider:	▪ Fraunhofer FOKUS	
Contact point:	▪ Niels Hoppe <niels.hoppe@fokus.fraunhofer.de>	
Condition(s) for reuse:	▪ Planned to be published as open-source software (EPL 2.0 license) ▪ Commercial support and consulting by Fraunhofer FOKUS	
	Latest update: 02.02.2022	

Name: MBPLE - Model-based Product Line Engineering		
Input(s):	Main feature(s)	Output(s):
▪ UML Product Line Model	▪ Extraction of variability information from UML model ▪ Configuration and instantiation of UML model	▪ Feature Model ▪ UML Product Model

Unique Selling Proposition(s):	▪ MBPLE lowers the entry-barrier to model-based product line engineering through automation and a non-intrusive modeling approach ▪ MBPLE is based on proven open-source software (FeatureIDE) ▪ MBPLE is designed to work with MESIR (see above) and is part of the XIVT Workbench by Fraunhofer FOKUS
Integration constraint(s):	▪ Eclipse platform incl. Eclipse Papyrus and FeatureIDE
Intended user(s):	▪ Engineers in system integration for robotic production cells
Provider:	▪ Fraunhofer FOKUS
Contact point:	▪ Niels Hoppe <niels.hoppe@fokus.fraunhofer.de>
Condition(s) for reuse:	▪ Planned to be published as open-source software (EPL 2.0 license) ▪ Commercial support and consulting by Fraunhofer FOKUS
	Latest update: 02.02.2022

Name: QATS - Quality Assessment of Test Suites		
Input(s):	Main feature(s)	Output(s):
▪ UML model of the system under test ▪ Product instantiation procedure ▪ Test suite	▪ Creation of model mutants for mutation testing ▪ Analysis of mutation testing results	▪ Test suite quality metrics

Unique Selling Proposition(s):	▪ QATS adapts the known and proven concept of mutation-testing to the specifics of software product lines ▪ QATS is part of the XIVT Workbench by Fraunhofer FOKUS
Integration constraint(s):	▪ Eclipse platform incl. Eclipse Papyrus
Intended user(s):	▪ Test engineers
Provider:	▪ Fraunhofer FOKUS
Contact point:	▪ Niels Hoppe <niels.hoppe@fokus.fraunhofer.de>
Condition(s) for reuse:	▪ Planned to be published as open-source software (EPL 2.0 license) ▪ Commercial support and consulting by Fraunhofer FOKUS
	Latest update: 02.02.2022

Name: CLSTR - Cloud-based Simulation and Testing for Robotics		
Input(s):	Main feature(s)	Output(s):
▪ System under test ▪ Test goals	▪ Keyword-driven testing library for robotics ▪ Configuration and tools for Continuous Integration (CI) process	▪ Keyword-based test suite ▪ CI pipeline ▪ Test results

Unique Selling Proposition(s):	▪ CLSTR allows simple implementation of simulation-based testing ▪ CLSTR allows cost-effective scaling of simulation-based testing
Integration constraint(s):	▪ System under test based on Robot Operating System ▪ GitLab, Docker
Intended user(s):	▪ (Test) engineers in system integration for robotic production cells
Provider:	▪ Fraunhofer FOKUS
Contact point:	▪ Niels Hoppe <niels.hoppe@fokus.fraunhofer.de>
Condition(s) for reuse:	▪ Planned to be published as open-source software (Apache 2.0 license) ▪ Commercial support and consulting by Fraunhofer FOKUS
	Latest update: 02.02.2022

Name: XIVT Robotics Demonstrator		
Input(s):	Main feature(s)	Output(s):
▪ N/A	▪ Configurable setup of two robotic arms with a selection of different tools ▪ Implementation of example processes	▪ N/A

Unique Selling Proposition(s):	▪ ▪ Based entirely on open-source software and affordable hardware
Integration constraint(s):	▪ Solutions to be demoed based on Robot Operating System (ROS)
Intended user(s):	▪ Researchers and engineers in industrial robotics
Provider:	▪ Fraunhofer FOKUS
Contact point:	▪ Niels Hoppe <niels.hoppe@fokus.fraunhofer.de>
Condition(s) for reuse:	▪ Planned to be published as open-source software (GPL 3.0 license) and documentation (Creative Commons license) ▪ Commercial support and consulting by Fraunhofer FOKUS
	Latest update: 02.02.2022

Name: BBTO - Budget Based Testing Optimization Tool		
Input(s):	Main feature(s)	Output(s):
▪ Product variants ▪ Annotated test cases (variant under test, cost, criticality, etc.)	▪ Prioritization and selection of test cases based on budget ▪ Prioritization and selection of product variants for testing based on budget	▪ Selection of product variants and test cases ▪
Unique Selling Proposition(s):	▪ BBTO offers multi-objective optimization with various strategies and different priorities ▪ Besides test cases, BBTO selects and prioritizes variants to actually execute tests on	
Integration constraint(s):	▪ Easy to integrate web services ▪ JSON or XML data exchange with field name mapping import / export services for high flexibility	
Intended user(s):	▪ Test engineers	
Provider:	▪ Fraunhofer FOKUS	
Contact point:	▪ Johannes Viehmann <johannes.viehmann@fokus.fraunhofer.de>	
Condition(s) for reuse:	▪ Proprietary, contact Fraunhofer FOKUS	
	Latest update: 02.02.2022	

Name: ReForm - Automated Requirement Formalization		
Input(s):	Main feature(s)	Output(s):
▪ Requirements in natural language ▪ Design Specifications in natural language	▪ Automated signal detection ▪ Automated parameter detection ▪ Logic extraction ▪ Model recommendations with user interaction	▪ Requirement Models ▪ Links between model and textual entities
Unique Selling Proposition(s):	▪ Automated generation of requirement models with an average accuracy of 90% ▪ High support for the manual requirements verification process ▪ Fast and easy-to-use tool with the ability to automate multiple steps in a model-based testing pipeline	
Integration constraint(s):	▪ Python 3 ▪ spaCy ▪ NLTK ▪ JavaScript	
Intended user(s):	▪ Requirements engineers ▪ Software testers	
Provider:	▪ IFAK Institute for Automation and Communication	
Contact point:	▪ Robin Gröpler, robin.groepler@ifak.eu	
Condition(s) for reuse:	▪ Planned to license commercially on request ▪ Support and consulting by IFAK	
	Latest update: 31.01.2022	

Name: ifakVBT - Model-Based Test Case Generation		
Input(s):	Main feature(s)	Output(s):
▪ Specification model as UML sequence diagrams or UML state machine	▪ Automatic generation of abstract test cases ▪ Selection of graph-based coverage criteria (all paths, all transitions, etc.) ▪ Requirements-based variant management	▪ Abstract test cases as UML sequence diagrams ▪ Traceability information

Unique Selling Proposition(s):	▪ Automatic test case generation with graph-based coverage criteria of 100% ▪ Editable and easy-to-use textual and visual representation of specification models and test cases ▪ Integration with other IFAK tools for requirement formalization (ReForm), test prioritization, and test execution
Integration constraint(s):	▪ C++ ▪ JavaScript
Intended user(s):	▪ Software testers ▪ Software engineers
Provider:	▪ IFAK Institute for Automation and Communication
Contact point:	▪ Karsten Meinecke, karsten.meinecke@ifak.eu
Condition(s) for reuse:	▪ Planned to license commercially on request ▪ Support and consulting by IFAK
	Latest update: 31.01.2022

Name: NALABS/RCM		
Input(s):	Main feature(s)	Output(s):
▪ Existing Natural Language Requirements and their links to implementing software ▪ New Natural Language requirements to be implemented	▪ Automated Bad Smells and Complexity Detector	▪ Bad smells and complexity indicators

Unique Selling Proposition(s):	▪ Detects requirements containing potential mistakes or indicators or overcomplexity
Integration constraint(s):	▪ C# ▪ .NET
Intended user(s):	▪ Requirements engineers ▪ Software testers ▪ Software architects ▪ Project managers
Provider:	▪ Mälardalen University
Contact point:	▪ Eduard Paul Enoiu – eduard.paul.enoiu@mdh.se
Condition(s) for reuse:	Licensing and permission required
	Latest update: 02.02.2022

Name: SEAFOX		
Input(s):	Main feature(s)	Output(s):
▪ .xlxs files ▪ FBD IEC 61131-3 program (PLCOpen XML)	▪ SEAFOX is the only available combinatorial test suite generation and selection tool for industrial control software.	▪ Test Cases (.xlxs)

Unique Selling Proposition(s):	▪ SEAFOX supports the generation of test suites using pairwise, base choice and random strategies. ▪ SEAFOX was used in several studies in order to support testing of industrial programs and fault detection. ▪ A tester using SEAFOX can automatically generate test suites needed for a given industrial IEC program after manually providing the input parameter range information based on the defined behavior written in the specification.
Integration constraint(s):	▪ C# ▪ .NET
Intended user(s):	▪ Software testers
Provider:	▪ Mälardalen University
Contact point:	▪ Eduard Enoiu - eduard.paul.enoiu@mdh.se
Condition(s) for reuse:	▪ Publicly available
	Latest update: 02.02.2022

Name: XIVT End-to-End Tool Chain		
Input(s):	Main feature(s)	Output(s):
▪ Depends on services of the tool chain user intends to use	▪ An intuitive user interface that would allow various services required for variability testing to be easily configured, launched and managed.	▪ Depends on the services of the tool chain the user intends to use

Unique Selling Proposition(s):	▪ Utilize state of the art methods and solutions for optimum variability test of select use cases in a cohesive, user friendly manner
Integration constraint(s):	▪ Custom plugins for integration via RESTful web services
Intended user(s):	▪ Quality Engineers tasks with variability testing of complex systems in Railway, Industrial Manufacturing, Automotive & Telecom
Provider:	▪ mobileLIVE
Contact point:	▪ noel.kirthiraj@mobilelive.ca
Condition(s) for reuse:	▪ License to be discussed ▪ Custom plugins to integrate with proprietary systems can be developed for fixed cost to be discussed
	Latest update: 30.01.2022

Name: MES Test Manager® (MTest)		
Input(s):	Main feature(s)	Output(s):
▪ Compilable Simulink/TargetLink/Stateflow software models ▪ Requirements specification documents ▪ Parameter files	▪ Requirements based variants testing of Simulink/Targetlink/Stateflow Software models ▪ Variants testing ▪ Requirements based assessment generation ▪ Requirements based test sequence generation	▪ Generated test stimuli and test assessments ▪ Test results ▪ Test metrics

Unique Selling Proposition(s):	▪ Requirements based assessment and test sequence generation (based on semi formal requirements syntax MARS)
Integration constraint(s):	• Only works for Matlab/Simulink software models from R2013b onwards • Windows OS
Intended user(s):	▪ Software Testers ▪ Test Managers ▪ Requirements Engineers
Provider:	▪ Model Engineering Solutions GmbH
Contact point:	▪ Sebastian Baronick
Condition(s) for reuse:	▪ Licensing required
	Latest update: 24.01.2022

Name: Recorder Tester v2.0		
Input(s):	Main feature(s)	Output(s):
▪ C/C++ code to test ▪ Build scripts etc ▪ Build flags to test	▪ Variant testing of C/C++ libraries with respect to build flag combinations. ▪ Test optimization using static analysis to eliminate redundant test cases. ▪ Parallel test execution in containers for horizontal scaling in the cloud.	▪ Test results

Unique Selling Proposition(s):	▪ Increase test coverage for reusable C/C++ code with many build flags.
Integration constraint(s):	▪ C/C++ toolchain (most compilers supported) ▪ Docker ▪ Jenkins
Intended user(s):	▪ Currently an internal tool. Possibility to commercialize for general C/C++ developers. Especially suitable for providers of reusable C/C++ libraries intended for 3rd party use, such as communication stacks and real-time operating systems.
Provider:	▪ Percepio
Contact point:	▪ Johan Kraft
Condition(s) for reuse:	▪ Case-by-case negotiation
	Latest update: 21.01.2021

Name: Risk-based test Scoring (RiSco)		
Input(s):	Main feature(s)	Output(s):
▪ Abstract test suite ▪ Domain-specific requirements (including operational modes and weather conditions) ▪ Risk criterion ▪ A list of scored components based on defined risk criterion ▪ A list of scored operational modes and weather conditions based on defined risk criterion	▪ Test case scoring based on a risk criterion ▪ Scoring operational situations based on a risk criterion	▪ Scored abstract test cases

Unique Selling Proposition(s):	▪ Service for prioritizing test cases based on risks associated with component's functional failure
Integration constraint(s):	▪ Abstract test cases are described in a set of system components and defined in three parts of "Pre-condition set" + Input set"à "Impacted set"
Intended user(s):	▪ Software Testers ▪ Test Managers
Provider:	▪ QA Consultants
Contact point:	▪ Farsam Farzadpour (ffarzadpour@qaconsultants.com)
Condition(s) for reuse:	▪ Case-by-case negotiation
	Latest update: 26.01.2022

Name: Human-Readable test scripts (HuBle)		
Input(s):	Main feature(s)	Output(s):
▪ Abstract test suite ▪ Natural language generator (A list of statements to define system actions)	▪ Converting machine-readable test scripts into human readable test scripts (test steps) ▪ Parser to converts abstract test cases to an internal representation ▪ Enabling manual test execution for automated generated test cases	▪ Abstract test suite presented in human-readable test scripts

Unique Selling Proposition(s):	▪ Service for generating human-readable test scripts from automated generated machine-readable test scripts.
Integration constraint(s):	▪ Abstract test cases are described in a set of system components and defined in three parts of “Pre-condition set” + Input set”à “Impacted set”
Intended user(s):	▪ Software Testers ▪ Test Managers
Provider:	▪ QA Consultants
Contact point:	▪ Farsam Farzadpour (ffarzadpour@qaconsultants.com)
Condition(s) for reuse:	▪ Case-by-case negotiation
	Latest update: 26.01.2022

Name: Tool for requirements-based variability modelling (BeVAR)		
Input(s):	Main feature(s)	Output(s):
▪ Project documents ▪ Product line model (encodes relevant information about the product)	▪ Create core model of the product model ▪ Create Δ-models of the product model	▪ Product line variance (UML file)

Unique Selling Proposition(s):	▪ Service for modelling the variability in the product
Integration constraint(s):	▪ Manual development of model in Eclipse environment using GUI ▪ Product line model defined in an customized XLST format
Intended user(s):	▪ Software Testers ▪ Test Managers
Provider:	▪ QA Consultants
Contact point:	▪ Farsam Farzadpour (ffarzadpour@qaconsultants.com)
Condition(s) for reuse:	▪ Case-by-case negotiation
	Latest update: 26.01.2022

Name: Industrial internet of things finder (IIotFinder)		
Input(s):	Main feature(s)	Output(s):
▪ Abstract test cases ▪ List of IIOT components identifiers	▪ Identify test cases with IIOT systems	▪ Labelled abstract test cases

Unique Selling Proposition(s):	▪ Service for identifying test cases with industrial internet of things system
Integration constraint(s):	▪ Abstract test cases are described in a set of system components and defined in three parts of “Pre-condition set” + Input set”à “Impacted set”
Intended user(s):	▪ Software Testers ▪ Test Managers
Provider:	▪ QA Consultants
Contact point:	▪ Farsam Farzadpour (ffarzadpour@qaconsultants.com)
Condition(s) for reuse:	▪ Case-by-case negotiation
	Latest update: 26.01.2022

Name: VARA		
Input(s):	Main feature(s)	Output(s):
▪ Existing Natural Language Requirements and their links to implementing software ▪ New Natural Language requirements to be implemented	▪ Automated Variability-Aware Reuse analysis and recommendation	▪ Software components that are candidates for reuse in new projects
Unique Selling Proposition(s):	▪ Identifies reuse opportunities across variants and projects ▪ VARA identifies and recommend reuse with around 82% average accuracy	
Integration constraint(s):	▪ Python 3 ▪ spaCy ▪ scikit-learn	
Intended user(s):	▪ Requirements engineers ▪ Software testers ▪ Software architects ▪ Project managers	
Provider:	▪ RISE Research Institutes of Sweden (in collaboration with Mälardalen University)	
Contact point:	▪ Muhammad Abbas – muhammad.abbas@ri.se	
Condition(s) for reuse:	Licensing and permission required	
	Latest update: 02.02.2022	

Name: Experience report on production line adoption in industry		
Input(s):	Main feature(s)	Output(s):
▪ N/A	▪ Article reporting on the results of XIVT project investigation regarding adoption of product line engineering concepts in industry: “Product Line Adoption in Industry: An Experience Report from the Railway Domain”	▪ Report on the benefits as well as challenges and open research questions in adoption of product line engineering in industry

Unique Selling Proposition(s):	▪ Providing a real-world industrial investigation ▪ Helping companies in their journey towards adoption of product-line engineering concepts
Integration constraint(s):	N/A
Intended user(s):	▪ Software engineers ▪ Software architects ▪ Researchers in software engineering ▪ Project managers
Provider:	▪ RISE Research Institutes of Sweden ▪ Mälardalen University ▪ Alstom (Bombardier Transportation)
Contact point:	▪ Muhammad Abbas - muhammad.abbas@ri.se
Condition(s) for reuse:	▪ Publicly available article
	Latest update: 02.02.2022

Name: Otomat v4.0		
Input(s):	Main feature(s)	Output(s):
▪ Voice recordings describing the intended test scenario ▪ List of applications to be tested	▪ Voice recording facility in the user interface ▪ Automatic STT conversion and input for user approval ▪ Parallel test execution ▪ Scheduled/prioritized test execution	▪ Automated, runnable test cases ▪ Test reports
Unique Selling Proposition(s):	▪ Zero-code test scenarios ▪ Test clause by voice ▪ No need to technical knowledge about SW test automation	
Integration constraint(s):	▪ The tool is used remotely. There is no constraint for local machines. ▪ Selenium Grid must be active on the computer which Otomat works with	
Intended user(s):	▪ Software developer teams, analysts, testers in SW development companies, working on backend, frontend, and mobile developments	
Provider:	▪ Turkcell Teknoloji	
Contact point:	▪ Buket Sarac	
Condition(s) for reuse:	▪ Commercial license to be negotiated ▪ Selenium and Cucumber are free tools	
	Latest update: 27.01.2022	

Name: InnSpect		
Input(s):	Main feature(s)	Output(s):

▪ Set of Test Cases after Test Optimization	▪ Test automation Engine
	▪ Simulates human interactions with SuT (System under Test)
	▪ This engine is cross-device and cross-technologies allowing:
	  ▪ Web Interface testing
	  ▪ Mobile Apps
	  ▪ Desktop Apps
	  ▪ Web Services (API Testing)
	  ▪ Database validation
	  ▪ STB (Set-Top-Boxes) testing
		▪ Generate test execution results, namely evidences of each test step in every test case
		▪ When software fails, InnSpect captures a screenshot for evidence of actual result for debugging purposes

Unique Selling Proposition(s):	▪ Fully integrated solution to address multi-variant and configurable systems ▪ It addresses a comprehensive set of business domains and comprehensive set of technologies ▪ Business domains are achieved by easily integration with other XIVT modules for test optimization ▪ Comprehensive set of technologies are achieved by built-in components at InnSpect tool	
Integration constraint(s):	▪ InnSpect uses Windows and SQL Server as architecture ▪ Integrates with other modules through JSON files	
Intended user(s):	▪ Test Analysts – people who do test design ▪ Test Engineers – people who do test execution / create test automation scripts	
Provider:	▪ WinTrust	
Contact point:	▪ Filipe.carlos@wintrust.tech	
Condition(s) for reuse:	▪ Commercial license to be negotiated	
	Latest update: 31.01.2022	