SmartDelta

Automated Quality Assurance and Optimization in Incremental Industrial Software Systems Development

D2.5 - SmartDelta Methodology Implementation Toolset

Submission date of deliverable: Dec 31, 2024
Edited by: SmartDelta Partners

Project start date	Dec 1, 2021
Project duration	36 months
Project coordinator	Dr. Mehrdad Saadatmand, RISE Research Institutes of Sweden
Project number & call	20023 - ITEA 3 Call 7
Project website	https://itea4.org/project/smartdelta.html & https://smartdelta.org/

Contributing partners	WP2 partners, and solution providers of the project
Version number	V1.0
Work package	WP2
Work package leader	Juan Miguel Gómez Berbis
Dissemination level	Public
Description	The D2.5 SmartDelta Methodology Implementation Toolset delivers the final version of SmartDelta's comprehensive software solutions, showcasing a suite of tools designed to efficiently manage product deltas across different phases of the SmartDelta methodology.

Executive Summary

The SmartDelta Methodology is a cornerstone outcome of SmartDelta, providing stakeholders with foundational guidelines to engage with SmartDelta solutions and workflows. As part of this methodology, a suite of specialized tools has been developed to manage product deltas across various phases of the methodology. These tools are designed to address the unique challenges of managing product deltas ensuring adaptability and efficiency.

In WP2, these tools have been meticulously aligned with each phase of the methodology, demonstrating their value and applicability for different development activities ranging from requirements engineering to system level quality visualization. This deliverable showcases the tools developed under WP2 for implementing the SmartDelta methodology, highlighting their roles in facilitating delta management.

Table of contents

SmartDelta Methodology

The SmartDelta Methodology provides a structured approach to incremental software development that emphasizes adaptability to change. It enables companies to address essential aspects of delta management within the software engineering process, offering targeted solutions for various stages. The main objective of the SmartDelta Methodology is to provide a comprehensive overview that guides companies in managing software deltas, enabling incremental development of their software systems according to their unique requirements and strategic objectives.
By focusing on both ***Version Deltas*** (incremental updates within a product's lifecycle) and ***Variant Delta*** (customizations for specific applications), the SmartDelta Methodology is designed to address the evolving needs of complex software systems. It guides the development lifecycle through six key stages. Each stage incorporates delta-aware practices, ensuring that every modification meets the product's quality expectations. The *SmartDelta Methodology* contains several process stages that are part of a software development lifecycle:

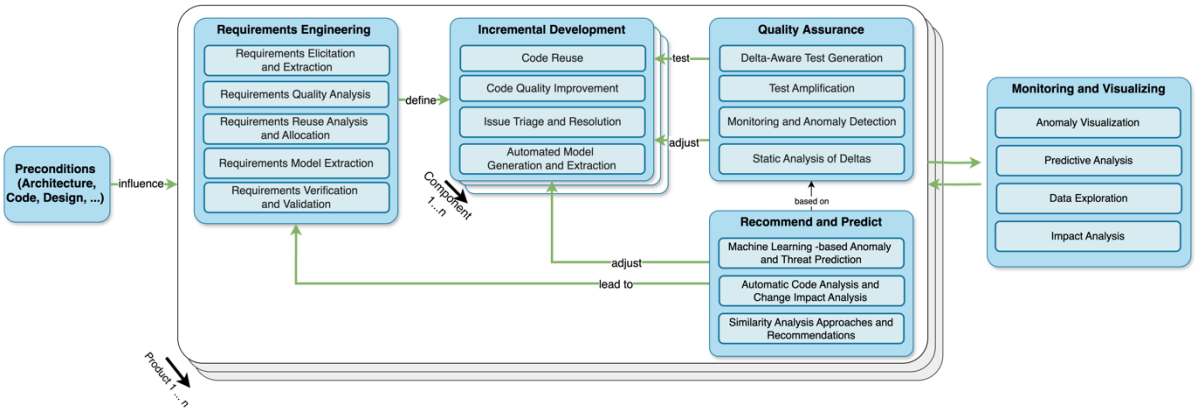

Figure 1 SmartDelta Methodology

In this deliverable, each tool developed in SmartDelta is mapped to a specific phase and technical area of the methodology, ensuring targeted solutions for managing deltas throughout the workflow. This document introduces and details these tools.

For additional information regarding the tools outlined in this deliverable, please refer to the related documents listed below:
D1.6 SmartDelta in Industrial Environments - Use Case Report
D3.5 Delta-oriented Quality Assurance Toolset: Final version
D3.4 Delta-oriented Quality Assurance Methodology: Final version
D4.5 SmartDelta Quality Optimization and Recommendation Methodology
D5.5 SmartDelta Visualization Dashboard: Final Version

Tools

Below are some of the tools created by the project partners to support the guidelines defined in the SmartDelta Methodology,

1. SoHist v2

SonarQube has some limitations when it comes to historical code analysis. To address these, SoHist was developed. It features an enhanced version comparison, providing visualizations, and in its second version, it allows users to benchmark their project's quality metrics against a large set of comparable projects.

Delta aspect: The term "Delta" in SoHist represents the difference between two software versions in terms of their code quality metrics. This difference is analyzed using SonarQube.
More information can be found in: WP3, WP4, WP5.
Partner: UIBK, c.c.com
Contact Person: benedikt.dornauer@uibk.ac.at

2. Intelligent Issue Management Support Tool (INIMASU)

Tool for assessing the health of a software product under continuous development by looking manly at the upcoming issues and the reactions these provoke. Consoludation of both previously as II=Issue Intelligence listed tools from FOKUS.

Delta aspect: The tool does time series analysis over the changes made to a software repository - the development of the response time to issues is a good indicator for the overall health of a software system.
More information can be found in: WP1, WP4, WP5.
Partner: FOKUS
Contact Person: abhishek.shrestha@fokus.fraunhofer.de, johannes.viehmann@fokus.fraunhofer.de

3. Call Graph Delta Analyzer

A static code analysis tool.

Delta aspect: It identifies the delta among call graphs models that are derived from various products (versions).
More information can be found in: WP3, WP4, WP5.
Partner: Özyeğin University (Subcontracted by Erste)
Contact Person: hasan.sozer@ozyegin.edu.tr

4. Mut4SLX

Mutation Testing for Simulink and Stateflow.

Delta aspect: Not explicitly, but one can do trend analysis on the mutation coverage.
More information can be found in: WP3.
Partner: UAntwerpen
Contact Person: serge.demeyer@uantwerpen.be

5. Code Similarity Investigator

The tool Code Similarity Investigator (CSI) computes similarity between code sections. It transforms code into Code Property Graphs (CPGs) and performs graph algorithms to calculate the similarity.

Delta aspect: Changed sections are the starting point for similarity and relatedness.
More information can be found in: WP4.
Partner: TWT
Contact Person: emanuel.ramneantu@twt-gmbh.de

6. Graph Similarity Recommender

Tool to compare state machines. Given a state machine, it recommends a similar one.

Delta aspect: Analysis and visualisation of deltas between compared graphs.
More information can be found in: WP4.
Partner: TWT
Contact Person: merve.kutlu@twt-gmbh.de

7. SmartMetrics

Analyszes git repositories and calculates semantic infomation along with some basic metrics.

Delta aspect: Dashboard which compares two git hashes (side by side), but this is not the main feature (developmend is on hold/in backlog).
More information can be found in: WP4.
Partner: Akkodis
Contact Person: andreas.dreschinski@akkodis.com

8. SmartTrace

Semantic search of Software Artefacts, based on DB created by SmartMetrics.

Delta aspect: No explicit delta features, works not on product level, but on artefact level. Selecting old git commits allows historical search and comparing search results.
More information can be found in: WP4.
Partner: Akkodis
Contact Person: andreas.dreschinski@akkodis.com

9. PyLC

An automated search-based PLC testing framework called PyLC has been developed. PyLC facilitates the unit testing of the PLC programs in Structured Text (ST) and Function Block Diagram (FBD) programming languages of the IEC61131-3 standard by automatically translating a PLC program to Python and generating test cases for it using the Pynguin [footnote: https://pynguin.readthedocs.io/en/latest/index.html] test generator tool in Python.

Delta aspect: PyLC handles incremental changes by validating transformations through automated test generation and translation rules, ensuring that product variations maintain consistency.
More information can be found in: WP3.
Partner: MDU
Contact Person: eduard.enoiu@mdu.se

10. DRACONIS

DRACONIS is a static analysis framework designed for automating the review of block-based, safety-critical software, particularly in domains like railway systems. It automates design rule checks for graphical development tools such as Simulink and Function Block Diagrams (FBD).

Delta aspect: DRACONIS applies round-trip analysis, where generated code is re-analyzed and mapped back to the model level, ensuring consistency across product deltas.
More information can be found in: WP3, WP5.
Partner: MDU, ALSTOM
Contact Person: eduard.enoiu@mdu.se

11. SEAFOX

SEAFOX is a combinatorial testing tool that automates test case generation for industrial control software, particularly those developed in the CODESYS environment for PLC systems. It supports various combinatorial methods such as Pairwise, Base Choice, and Random testing.

Delta aspect: SEAFOX can be extended to handle different versions of industrial programs, supporting additional data types and integration with existing tools like CfUnit for incremental testing.
More information can be found in: WP3, WP5.
Partner: MDU
Contact Person: eduard.enoiu@mdu.se

12. Architecture Analysis and Visualization Tool

Visualize sequence diagrams, log similarity, file relations and heatmaps.

Delta aspect: Visualizes differences between two state machines computed through GSR tool.
More information can be found in: WP5.
Partner: FOKUS
Contact Person: abhishek.shrestha@fokus.fraunhofer.de

13. Smellyzer

Smellyzer tool is designed to elevate software development quality by identifying and addressing inefficiencies within the code review and bug tracking processes. By pinpointing detrimental patterns, such as large changesets, sleeping reviews, and unassigned bugs, Smellyzer provides teams with actionable insights to refine their practices. This results in improved traceability, expedited issue resolution, and a streamlined development workflow, ensuring a more cohesive and efficient software development lifecycle.

Delta aspect: Delta aspect for your work in SmartDelta is the focused lens through which you assess the incremental quality shifts—identified as “smells”—in the code review and bug management processes. Each delta you examine provides a clearer view of how day-to-day changes shape the long-term health and efficiency of the development lifecycle.
More information can be found in: WP3.
Partner: ARCELIK, Bilkent
Contact Person: eraytuzun@cs.bilkent.edu.tr

14. Relink

ReLink, a prediction-based PR-issue linking tool with additional visualization features. ReLink differs from existing solutions with its availability as a fully functional web application and highly balanced prediction metrics. Missing links between PRs and issues are determined based on a final confidence score normalized between 0 and 100, where a higher score indicates a stronger possibility.

Delta aspect: Delta aspect for ReLink within the SmartDelta framework involves examining how newly introduced or adjusted PR-issue links affect the overall quality landscape as the product incrementally evolves. These deltas provide actionable insights into traceability, predictability, and the health of development workflows
More information can be found in: WP
Partner: ARCELIK, Bilkent
Contact Person: eraytuzun@cs.bilkent.edu.tr

15. PieR

PieR (Pull Request Classification tool) that analyses incoming pull requests for a given repository to classify them based on the root cause of where the error originated, such as the requirement phase, design phase, or implementation phase. The tool also categorizes pull requests into various types, including "bug fix," "new feature," "security update," and others.

Delta aspect: Delta aspect within SmartDelta is about interpreting every new set of pull requests as incremental indicators of shifting quality. This enables teams to proactively manage their development process and ensure that, as the system changes, they maintain or enhance overall quality.
More information can be found in: WP4.
Partner: ARCELIK, Bilkent
Contact Person: eraytuzun@cs.bilkent.edu.tr

16. K2

K2 is a state space exploration engine based on abstract state machines implemented in Java. It is able to execute rules of an abstract state machine according to some scenario configurations to explore the state space of executable specifications. Test cases are then generated from the explored finite state machine according to some coverage criteria. K2 is also capable of synthesizing test data required to enable rules. The generated test suites are represented in a proprietary format inspired by the UML Testing Profile. K2 offers textcode generators for Robot and JUnit.

Delta aspect: Regenerates the set of tests required for a new version based on a formal model, i.e., the abstract state machine. The Delta is reflected in the model as changes from one version to another and implicitly in reducing the need for impact analysis for each new release.
More information can be found in: WP3.
Partner: FOKUS
Contact Person: marc-florian.wendland@fokus.fraunhofer.de

17. CBTS

CBTS stands for change-based test selection. It is a prototype that analyzes and classifies the modifications made to the code for a new release to assess whether the modification is prone to side-effects. CBTS is used to select an adequate regression test suite to safeguard changes according to the potential of a test case to detect an unwanted side-effect.

Delta aspect: Selects a subset of regression tests for a new release according to the changes made (the Delta) and the possibility to uncover unwanted side effects accidentally introduced during the modification.
More information can be found in: WP3.
Partner: FOKUS
Contact Person: marc-florian.wendland@fokus.fraunhofer.de

18. Jazure

The tool enables us to link git requests and Jira work items to create insightful reports or dashboards.

Delta aspect: Links Data Point.
More information can be found in: WP3.
Partner: ARCELIK, Bilkent
Contact Person: omercan.devran@beko.com

19. Metric Dashboard

Qlik-based system designed to monitor IT software processes and quality metrics. It aggregates data from multiple sources, including Azure DevOps, SonarQube, Jira, Statuspage, and Google Analytics, allowing a comprehensive overview of development progress and software health. This integration provides actionable insights, supports continuous improvement, and enhances decision-making around development quality and team efficiency.

Delta aspect: Represents and Compares Deltas
More information can be found in: WP5.
Partner: ARCELIK, Bilkent
Contact Person: omercan.devran@beko.com

20. DIA4M

Within DIA4M there is “Delta” tracking in both time (in the sense of version) and space (in the sense of variant). It allows to compare deltas for two different versions of the same microservice and for two different microservices.

Delta aspect:
- Log comparison for microservice versions and variants
- Comparing resources such as memory, cpu for microservice versions
- Latency and trend comparison for microservice versions

More information can be found in: WP3, WP4, WP5
Partner: ORION/NETRD
Contact Person: hakan.kilinc@orioninc.com

21. DETANGLE

The analysis suite does a holistic technical debt analysis over time by correlating KPIs like high maintence with different root causes like architecture quality, code quality, process quality and knowledge distribution aspects to identify refactoring hotspots.

Delta aspect: The term delta refers to changes of code on code entity (e.g. class, method, function), file, folder and system levels between months, quarters and years (or other periods ot time like Sprints). These changes are evaluated to derive the code, architecture and process quality metrics.
More information can be found in: WP3, WP4, WP5.
Partner: Cape of Good Code, Dakik, TURK BANK
Contact Person: wuchner@capeofgoodcode.com

22. GHS anomaly detection and Offense Prioritization tools

Our apps were developed to detect Event per Second (EPS) anomaly and to prioritize offences inside QRadar (SEIM) system. Our apps utilize machine learning to identify anomaly in EPS and predict impact score of incoming offenses inside QRadar. This enables analysts to find suspicious events and sort offenses based on score and triage most impactful offenses faster.

Delta aspect: The term Delta in our app is training the ML models with different dataset from QRadar as new eps and offenses are being generated continuously. Retraining with different delta of these data makes our app robust to newer eps and offenses.
More information can be found in: WP4.
Partner: Glasshouse Systems and Ontario Tech University)
Contact Person: akamul.azim@ontariotechu.net

23. SONATA

SONATA is a web-based tool that optimizes software testing by analyzing Java code repositories and recommending relevant test cases. Using knowledge graphs and semantic matchmaking, it identifies similarities with past projects to automate test recommendations, reducing time and ensuring comprehensive coverage for new developments or deltas.

Delta aspect: The optimization of test generation and reuse for incremental software changes. By leveraging knowledge graphs and semantic analysis, SONATA efficiently identifies and recommends test cases tailored to the specific changes or deltas in the software, ensuring quality and reducing manual effort.
More information can be found in: WP3, WP5
Partner: Izertis, UC3M
Contact Person: ypabon@inf.uc3m.es

24. YATAP A Tool for Change Impact Analysis

Change impact tool explores the tentative effects of a change in other parts of a system. These changes can be on requirements/features, architectural, infrastructure etc.

Delta aspect: YATAP only monitors product deltas in product, software module and code construct levels. However, it shows them in graphs and timeseries together with other changes within the system, to make it easier to identify the cause of these deltas.
More information can be found in: WP4, WP5.
Partner: Erste
Contact Person: oguz@erstesoftware.com

25. Modernization Toolkit

The Modernization Toolkit Analyzer (MTK Analyzer) and Dragonfly help transition applications to the latest **Vaadin version** by analyzing and transforming Java source code. Operating as Eclipse or Maven plugins, they parse code into an Abstract Syntax Tree (AST) and apply predefined rules to update API references. The tools provide transformation coverage summaries and updated source code as output.

Delta aspect: The tools identify changes needed to align Java code with new **Vaadin API** versions, applies predefined transformation rules to modify code using AST, and outputs the updated code or a summary of the changes.
More information can be found in: WP4, WP5.
Partner: Vaadin
Contact Person: yuriy@vaadin.com

26. AirOPs

A tool for analyzing QA metrics on cloud-based airfield software.

Delta aspect: This tool identifies software bugs on different cloud-hosted software builds and provide predictions
More information can be found in: WP4.
Partner: Ontario Tech University, Team Eagle
Contact Person: akramul.azim@ontariotechu.ca. lindac@team-eagle.ca

27. NALABS

NALABS is a tool designed to detect "bad smells" in natural language requirements and test specifications. It supports quality assurance by automatically identifying vague or poorly structured specifications, helping to improve maintainability and clarity.

Delta aspect: NALABS handles product variations by continuously analyzing new requirements to ensure consistency and clarity across updates.
More information can be found in: WP3, WP5.
Partner: MDU, ALSTOM, ADDIVA
Contact Person: eduard.enoiu@mdu.se

28. GW2UPPAAL

GW2UPPAAL is a hybrid tool that combines Model-Based Testing (MBT) using GraphWalker with model checking using UPPAAL. It transforms GraphWalker models into UPPAAL timed automata for automated analysis, focusing on verifying reachability and deadlock properties.

Delta aspect: GW2UPPAAL supports incremental changes by re-analyzing updated models and generating new test cases based on the modified requirements.
More information can be found in: WP3.
Partner: MDU
Contact Person: eduard.enoiu@mdu.se

29. TIGER +

The Model-Based Test Script Generation Framework (TIGER) is developed for generating concrete test scripts from abstract test cases using GraphWalker, an open-source Model-Based Testing (MBT) tool.

Delta aspect: The TIGER framework is used for incremental development and delta testing due to its ability to adapt test generation to changes in the model and test optimization.
More information can be found in: WP3.
Partner: Mälardalen University (MDU)
Contact Person: eduard.enoiu@mdu.se

30. ReForm

The Automatic Requirements Formalization (ReForm) tool transforms textual requirements into formal models, such as state machines, to streamline system design. By leveraging advanced language models (LLMs) and retrieval techniques (RAG), it simplifies the process of interpreting and structuring complex requirements, such as those from ISO 15118. ReForm accelerates requirements engineering, reduces manual effort, and improves the accuracy and quality of system models.

Delta aspect: Adapt state machine models due to requirement changes.
More information can be found in: WP4.
Partner: IFAK, Akkodis
Contact Person: robin.groepler@ifak.eu

31. AISA

The Automatic Issue Similarity Analysis (AISA) tool enhances issue management by automatically identifying related problems using advanced language models. It reduces redundancies and improves efficiency by detecting duplicates and interlinked issues across products, versions, and variants.

Delta aspect: The Delta aspect of the AISA tool lies in its ability to identify related issues across different versions of software, highlighting how changes or updates may have introduced, resolved, or altered problems..
More information can be found in: WP4.
Partner: IFAK, Software AG
Contact Person: robin.groepler@ifak.eu

32. AILA

The Automatic Issue Labeling Tool (AILA) streamlines the classification of software requirements and software issues by using advanced AI techniques to reduce manual effort and errors. It automatically assigns labels or tags to issues, e.g. for security or severity, enabling faster prioritization, targeted processing, and efficient assignment of tasks to appropriate teams.

Delta aspect: By leveraging continual learning techniques, AILA retains knowledge from previous versions while incorporating new data, ensuring consistent and accurate labeling despite changes in requirements or context.
More information can be found in: WP4.
Partner: IFAK in collaboration with Software AG
Contact Person: robin.groepler@ifak.eu

33. Telemetry Anomaly Analyzer

Tool to detect and visualize anomalies on OpenTelemetry based telemetry information.

Delta aspect: Can compare between release versions of the platform
More information can be found in: WP4, WP5.
Partner: Hoxhunt
Contact Person: janne.piironen@hoxhunt.com

SmartDelta Methodology – Tools Mapping

Tool Name	Owner (first position) and Partners (followed)	Requirements Engineering					Incremental Development				Quality Assurance				Recommend and Predict			Monitoring and Visualizing			
		Requirements Elicitation and Extraction	Requirements Quality Analysis	Requirements Reuse Analysis and Allocation	Requirements Model Extraction	Requirements Verification	Code Reuse	Code Quality Improvement	Issue Triage and Resolution	Automated Model Generation and Extraction	Delta-Aware Test Generation	Test Amplification	Monitoring and Anomaly Detection	Static Analysis of Deltas	Machine Learning -based Anomaly and Threat Prediction	Automatic Code Analysis and Change Impact Analysis	Similarity Analysis Approaches and Recommendations	Anomaly Visualization	Predictive Analysis	Data Exploration	Impact Analysis
SoHist v2	UIBK, c.c.com							**X**						**X**		**X**					
Intelligent Issue Management Support Tool (INIMASU)	FOKUS		X		X				X						X	X			X	X	X
Call Graph Delta Analyzer	Özyeğin University (Subcontracted by Erste)							X		X			X	X		X	X	X			X
Mut4SLX	UAntwerpen										X	X	X								
Code Similarity Investigator	TWT						X										X				

Tool Name	Owner (first position) and Partners (followed)	Requirements Engineering					Incremental Development				Quality Assurance				Recommend and Predict				Monitoring and Visualizing			
		Requirements Elicitation and Extraction	Requirements Quality Analysis	Requirements Reuse Analysis and Allocation	Requirements Model Extraction	Requirements Verification	Code Reuse	Code Quality Improvement	Issue Triage and Resolution	Automated Model Generation and Extraction	Delta-Aware Test Generation	Test Amplification	Monitoring and Anomaly Detection	Static Analysis of Deltas	Machine Learning -based Anomaly and Threat Prediction	Automatic Code Analysis and Change Impact Analysis	Similarity Analysis Approaches and Recommendations	Anomaly Visualization	Predictive Analysis	Data Exploration	Impact Analysis
Graph Similarity Recommender	TWT																X				
SmartMetrics	Akkodis						X														
SmartTrace	Akkodis						X														
PyLC	MDU										X	X									
DRACONIS	MDU, ALSTOM					X	X	X						X							
SEAFOX	MDU				X						X	X									
Architecture Analysis and Visualization Tool	FOKUS								X					X							
Smellyzer	ARCELIK, Bilkent								X				X						X		

Tool Name	Owner (first position) and Partners (followed)	Requirements Engineering					Incremental Development				Quality Assurance				Recommend and Predict				Monitoring and Visualizing			
		Requirements Elicitation and Extraction	Requirements Quality Analysis	Requirements Reuse Analysis and Allocation	Requirements Model Extraction	Requirements Verification	Code Reuse	Code Quality Improvement	Issue Triage and Resolution	Automated Model Generation and Extraction	Delta-Aware Test Generation	Test Amplification	Monitoring and Anomaly Detection	Static Analysis of Deltas	Machine Learning -based Anomaly and Threat Prediction	Automatic Code Analysis and Change Impact Analysis	Similarity Analysis Approaches and Recommendations	Anomaly Visualization	Predictive Analysis	Data Exploration	Impact Analysis
Relink	ARCELIK, Bilkent								X				X		X						
PieR	ARCELIK, Bilkent								X										X		
K2	FOKUS									X	X	X									
CBTS	FOKUS										X					X					
Jazure	ARCELIK, Bilkent			X		X															
Metric Dashboard	ARCELIK, Bilkent																	X		X	X
DIA4M	ORION/NETRD								X				X		X			X		X	
DETANGLE	Cape of Good Code, Dakik, Kuveyt Türk Bank							X	X				X	X				X			

Tool Name	Owner (first position) and Partners (followed)	Requirements Engineering					Incremental Development				Quality Assurance				Recommend and Predict				Monitoring and Visualizing			
		Requirements Elicitation and Extraction	**Requirements Quality Analysis**	**Requirements Reuse Analysis and Allocation**	**Requirements Model Extraction**	**Requirements Verification**	**Code Reuse**	**Code Quality Improvement**	**Issue Triage and Resolution**	**Automated Model Generation and Extraction**	**Delta-Aware Test Generation**	**Test Amplification**	**Monitoring and Anomaly Detection**	**Static Analysis of Deltas**	**Machine Learning -based Anomaly and Threat Prediction**	**Automatic Code Analysis and Change Impact Analysis**	**Similarity Analysis Approaches and Recommendations**	**Anomaly Visualization**	**Predictive Analysis**	**Data Exploration**	**Impact Analysis**
GHS anomaly detection and Offense Prioritization tools	Glasshouse Systems and Ontario Tech University)												X		X			X	X		X
SONATA	Izertis, UC3M				X					X	X						X			X	
YATAP A Tool for Change Impact Analysis	Erste							X	X					X		X		X	X		
Modernization Toolkit	Vaadin						**X**			**X**						**X**	**X**				X
AirOPs	Ontario Tech University, Team Eagle							X	X							X			X		

Tool Name	Owner (first position) and Partners (followed)	Requirements Engineering					Incremental Development				Quality Assurance				Recommend and Predict				Monitoring and Visualizing			
		Requirements Elicitation and Extraction	**Requirements Quality Analysis**	**Requirements Reuse Analysis and Allocation**	**Requirements Model Extraction**	**Requirements Verification**	**Code Reuse**	**Code Quality Improvement**	**Issue Triage and Resolution**	**Automated Model Generation and Extraction**	**Delta-Aware Test Generation**	**Test Amplification**	**Monitoring and Anomaly Detection**	**Static Analysis of Deltas**	**Machine Learning -based Anomaly and Threat Prediction**	**Automatic Code Analysis and Change Impact Analysis**	**Similarity Analysis Approaches and Recommendations**	**Anomaly Visualization**	**Predictive Analysis**	**Data Exploration**	**Impact Analysis**
NALABS	MDU, ALSTOM, ADDIVA		**X**			**X**															
GW2UPPAAL	MDU				X						X										
TIGER +	MDU				**X**						**X**	**X**									
ReForm	IFAK, Akkodis				X					X											
AISA	IFAK, Software AG			**X**					**X**								**X**				
AILA	IFAK, Software AG			X					X								X				
TCG	IFAK										**X**										
Telemetry Anomaly Analyzer	Hoxhunt														X			X		X	X

SmartDelta

Summary

The SmartDelta project introduces a dedicated methodology for managing deltas in software development. Central to this methodology is a suite of carefully designed tools that facilitate its implementation and ensure its successful instantiation in different contexts. These tools equip stakeholders with the necessary capabilities to effectively manage diverse product deltas, empowering them to seamlessly incorporate the SmartDelta Methodology into their development processes.