Model-Based Security Testing
Results from Industrial Case Studies

Ina Schieferdecker, Axel Rennoch
Fraunhofer FOKUS

Our testing background

- Automated test execution:

  TTCN-3 – Testing and Test Control Notation

  standardization at ETSI since 1998
- Automated test design:

  UTP – UML Testing Profile

  standardization at OMG since 2001
- **Test tools** development at FOKUS and Testing Technologies
- **Test suites** development and testing with numerous industrial partners
- Test automation, TTCN-3 and **MBT syllabi and certificates** with GTB

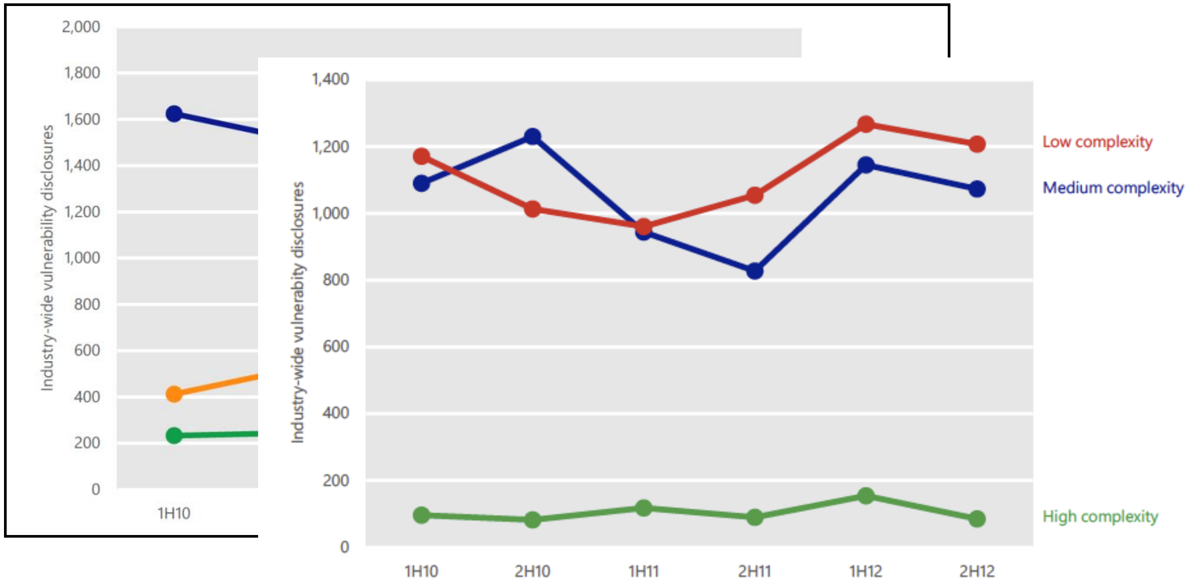
Outline

- Introduction and Overview
- Security Testing Improvement Profiles and Industrial Case Studies
- Details of Giesecke & Devrient Case Study
- Security Testing Approach and Traceing
- Summary

Introduction & Relevance

Vulnerabilities & software faults

- Most software vulnerabilities arise from common causes and the top 10 cause account for about 75% of all software vulnerabilities
- More than 90% of the vulnerabilities are caused by known causes
- The number of vulnerabilities being discovered in applications is far greater than the number of vulnerabilities discovered in operating systems
- Due to SEI and to McAfee, majority of security breaches is due to software faults



Source : Microsoft Security Intelligence Report
Volume 14 - July through Dec, 2012

Introduction & Relevance
Challenges

- Security engineering is increasingly challenged by the **openness**, **dynamics**, and **distribution** of networked systems

- Most verification and validation techniques for security have been developed in the framework of static or known configurations, with full or well-defined control of each component of the system

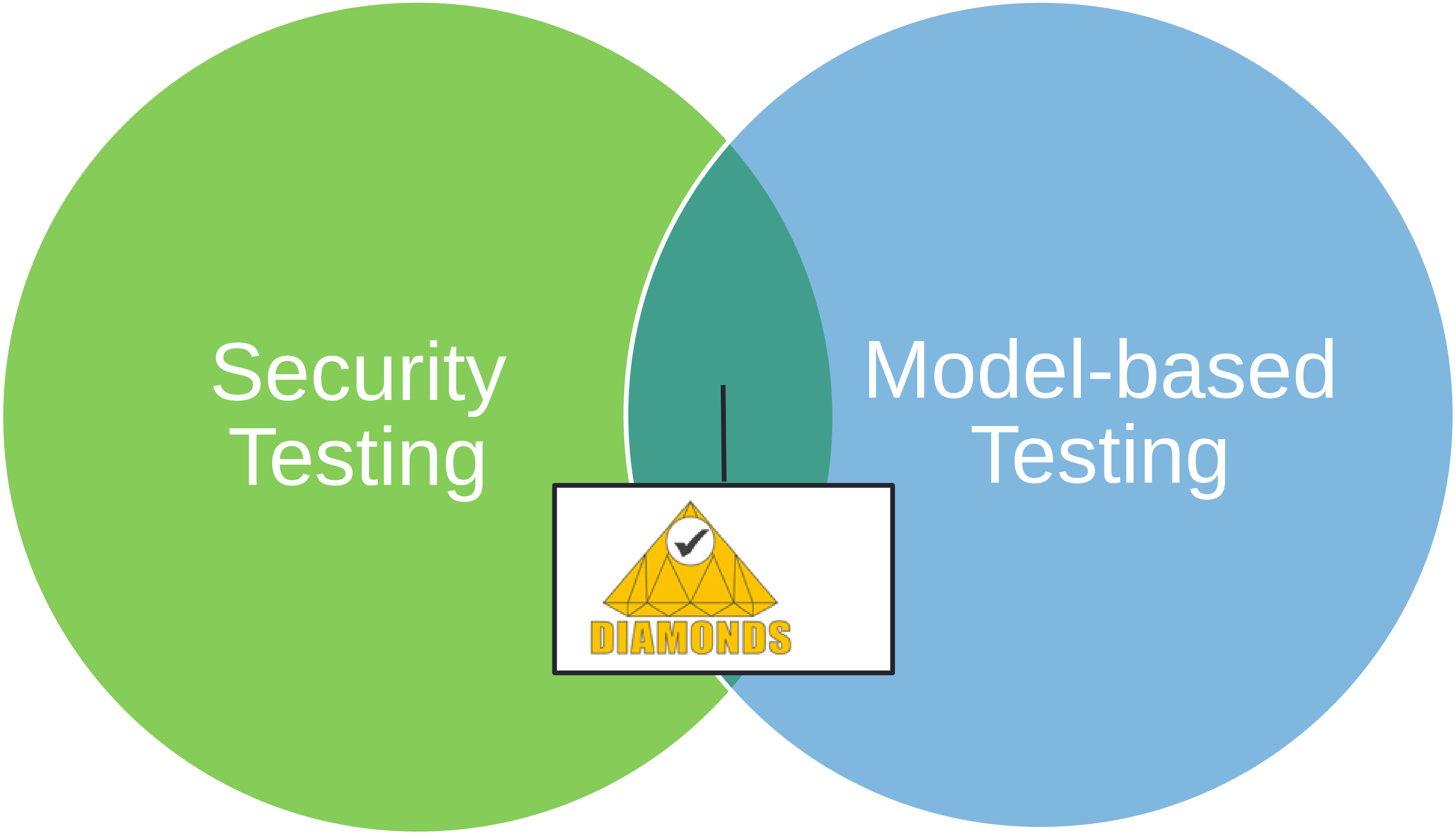
- This is not sufficient in networked systems, where control and observation of remote (sub) systems are dynamically invoked over the network

- DIAMONDS – **Development and Industrial Application of Multi-Domain Security Testing Technologies** – challenges the:

  → *Combination of active and passive security testing*
  → *Usage of fuzz tests (for unknown issues) and functional tests (for security measures)*
  → *Combination of risk analysis and test generation*
  → *Integration of automated test generation, test execution and monitoring*

Introduction & Relevance

Combination of approaches



Security Testing

Model-based Testing

Introduction & Relevance

Efficient and automated security testing

DIAMONDS will enable efficient and automated security testing methods of industrial relevance for highly secure systems in multiple domains.

Overall Objectives:

- Model-based security test methods and test patterns
- Automatic monitoring techniques
- Open source platform for security test tool integration

Business Impact:

- Experience reports from different industrial case studies
- Novel integration of testing, security and risk analysis
- Pre-standardization work

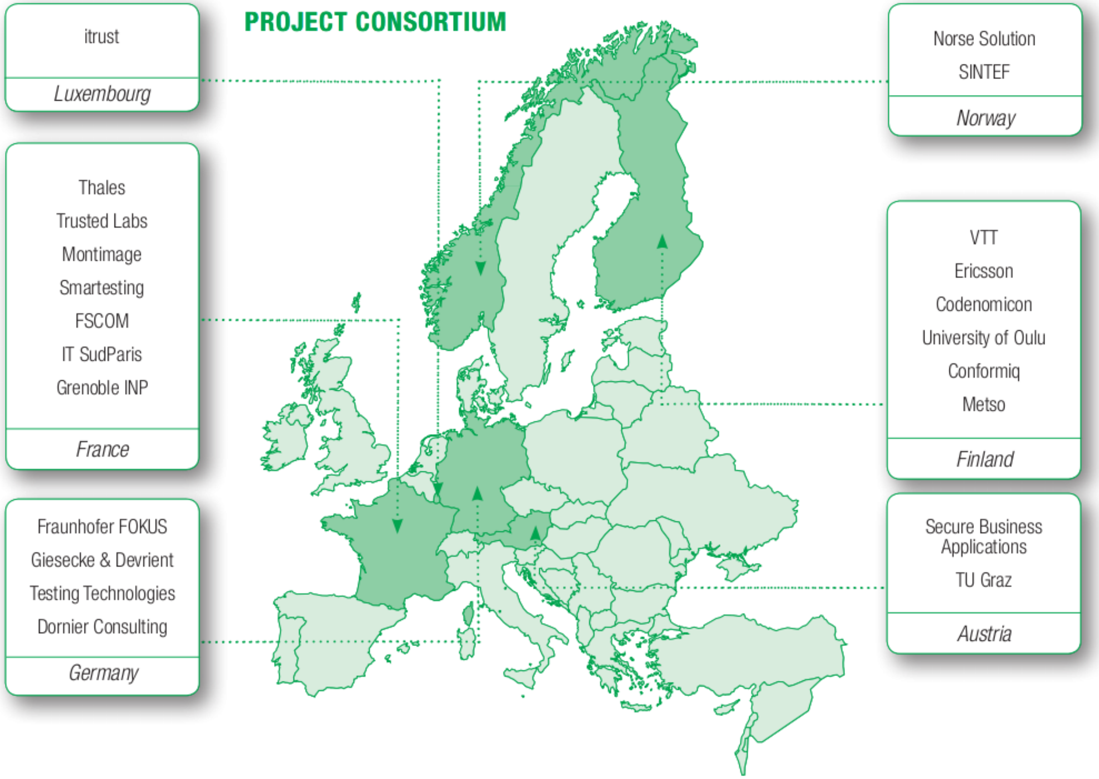
DIAMONDS Project

In six countries

Project Duration: October 2010 – June 2013

Project Partner:

- Large companies (6)
- Small companies (10)
- Universities (3)
- Research institutes (4)



PROJECT CONSORTIUM

itrust
Luxembourg

Thales
Trusted Labs
Montimage
Smartesting
FSCOM
IT SudParis
Grenoble INP
France

Fraunhofer FOKUS
Giesecke & Devrient
Testing Technologies
Dornier Consulting
Germany

Norse Solution
SINTEF
Norway

VTT
Ericsson
Codenomicon
University of Oulu
Conformiq
Metso
Finland

Secure Business Applications
TU Graz
Austria

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

DIAMONDS Achievements

Valuable results in fast track

- Successful fast exploitation (3 new commercial products, 3 open source products, 10 product updates)
- Adaptation of techniques in the productive environment by Metso, G&D, Thales etc.
- DIAMONDS contributed to the standardization initiatives at ETSI and ISO
- 8 case study experience reports and 11 innovation sheets
- 4 book chapters, 4 journal papers, 102 scientific or industrial papers or presentations, etc.
- DIAMONDS won the ITEA Exhibition award two times
- DIAMONDS tutorial with 7 DIAMONDS talks at the ICST 2013 with appr. 70 participants

DIAMONDS Innovative Results

… and their application to case studies

- Risk Based Testing **(Banking, Automotive):**
  - Test-based risk assessment (SINTEF)
  - Risk-based security testing with security test pattern (FOKUS)
- Advanced Fuzz Testing **(Banking, Radio Protocols, Automotive, Telecom):**
  - Model-based behavioural fuzzing (FOKUS)
  - Model inference assisted evolutionary fuzzing (INPG)
- Active Testing Techniques **(Banking, Radio Protocols)**
  - Model-based security testing from behavioral models and test purposes (SMARTESTING)
  - Integration of model-based test generation and monitoring (MONTIMAGE, SMARTESTING, FSCOM)
- Autonomous Testing Techniques **(Radio Protocols, Industrial Automation):**
  - Passive symbolic monitoring + distributed intrusion detection (IT)
  - Static binary code analysis for vulnerability detection (INPG)
  - Model-based security monitoring for both testing and operation - DevOpsSec* (MONTIMAGE)
- Open Source Tools for Security Testing (**Banking, Automotive, Radio Protocols**):
  - Tracebility platform for risk-based security testing (FOKUS)
  - Malwasm (iTrust), MMT_Security (MONTIMAGE)

(*) DevOpsSec: term introduced by Gartner Research (« Hype Cycle for Application Security », July 2012)

Outline

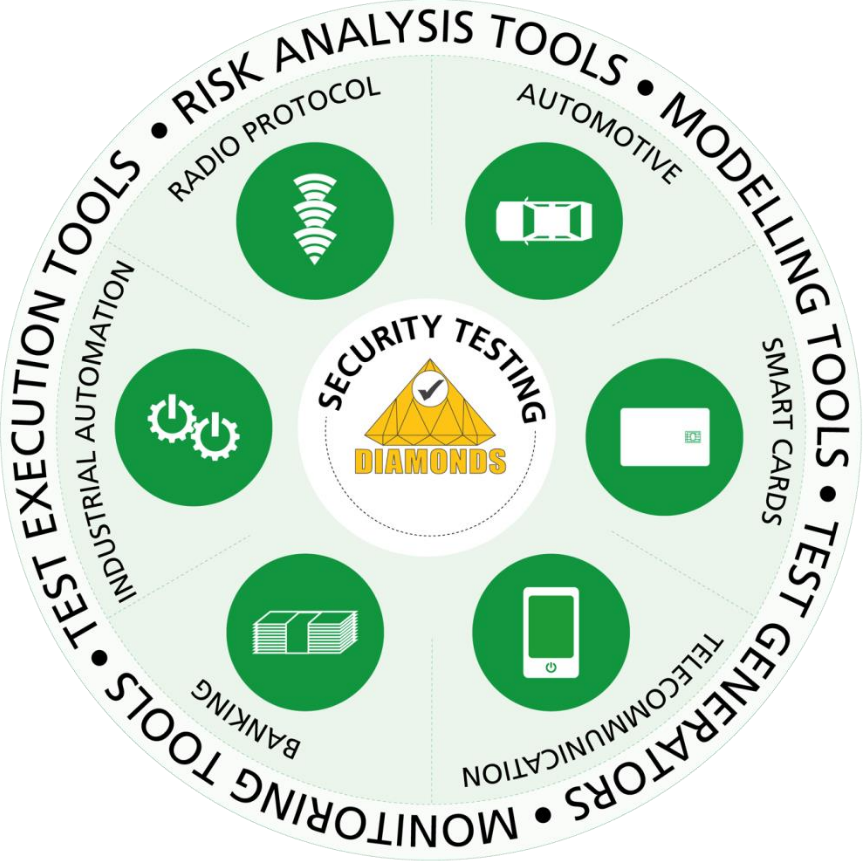
- Introduction and Overview
- Security Testing Improvement Profiles and Industrial Case Studies
- Details of Giesecke & Devrient Case Study
- Security Testing Approach and Traceing
- Summary

Case Studies

Six industrial domains

Security testing solutions for six industrial domains in 8 case studies

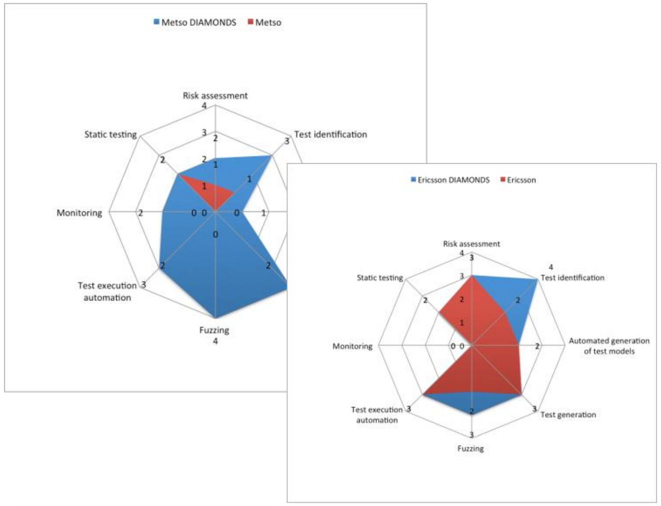
- Banking
- Automotive
- Radio protocols
- Smart cards
- Telecommunication
- Industrial automation

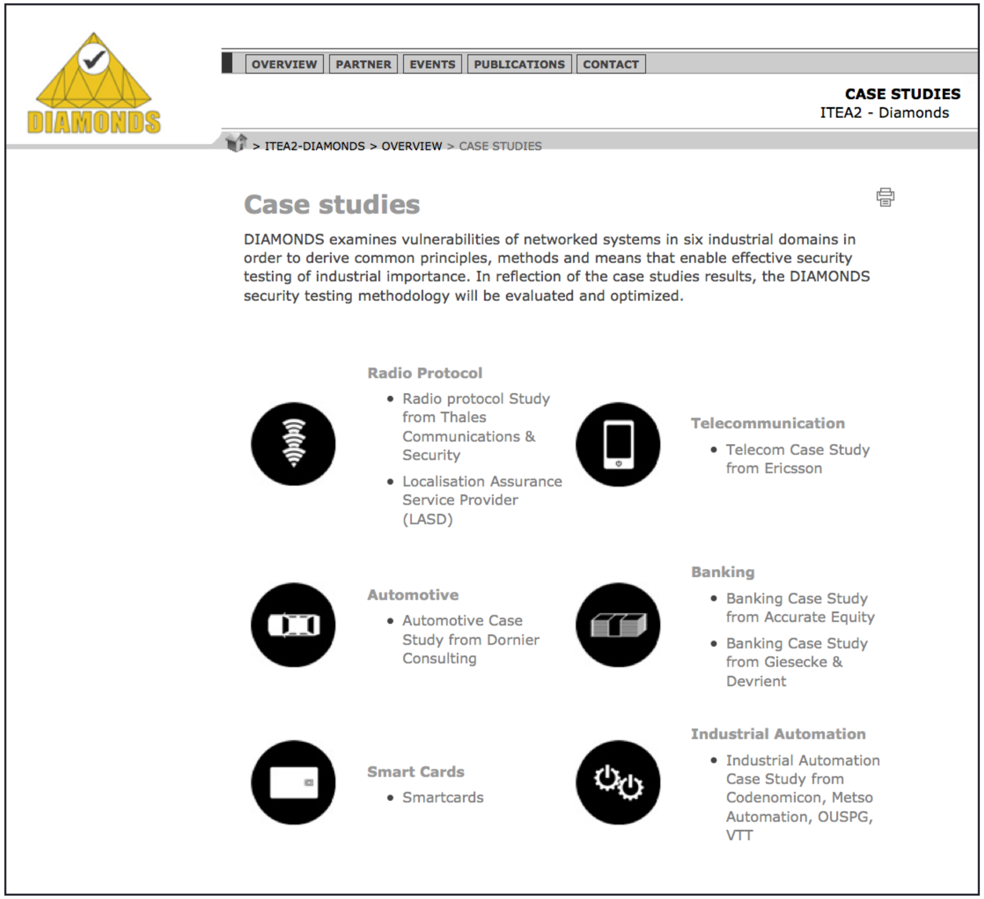


Industrial Impact

8 successful case studies and STIP evaluations

- Collection of the experiences and results for all case studies
  - Case study experience sheets
  - Available at DIAMONDS web site
- STIP Evaluation
  - Shows progress in all case studies





CASE STIDIES
ITEA2 - Diamonds

> ITEA2-DIAMONDS > OVERVIEW > CASE STUDIES

Case studies

DIAMONDS examines vulnerabilities of networked systems in six industrial domains in order to derive common principles, methods and means that enable effective security testing of industrial importance. In reflection of the case studies results, the DIAMONDS security testing methodology will be evaluated and optimized.

Radio Protocol

- Radio protocol Study from Thales Communications & Security
- Localisation Assurance Service Provider (LASD)

Telecommunication

- Telecom Case Study from Ericsson

Automotive

- Automotive Case Study from Dornier Consulting

Banking

- Banking Case Study from Accurate Equity
- Banking Case Study from Giesecke & Devrient

Smart Cards

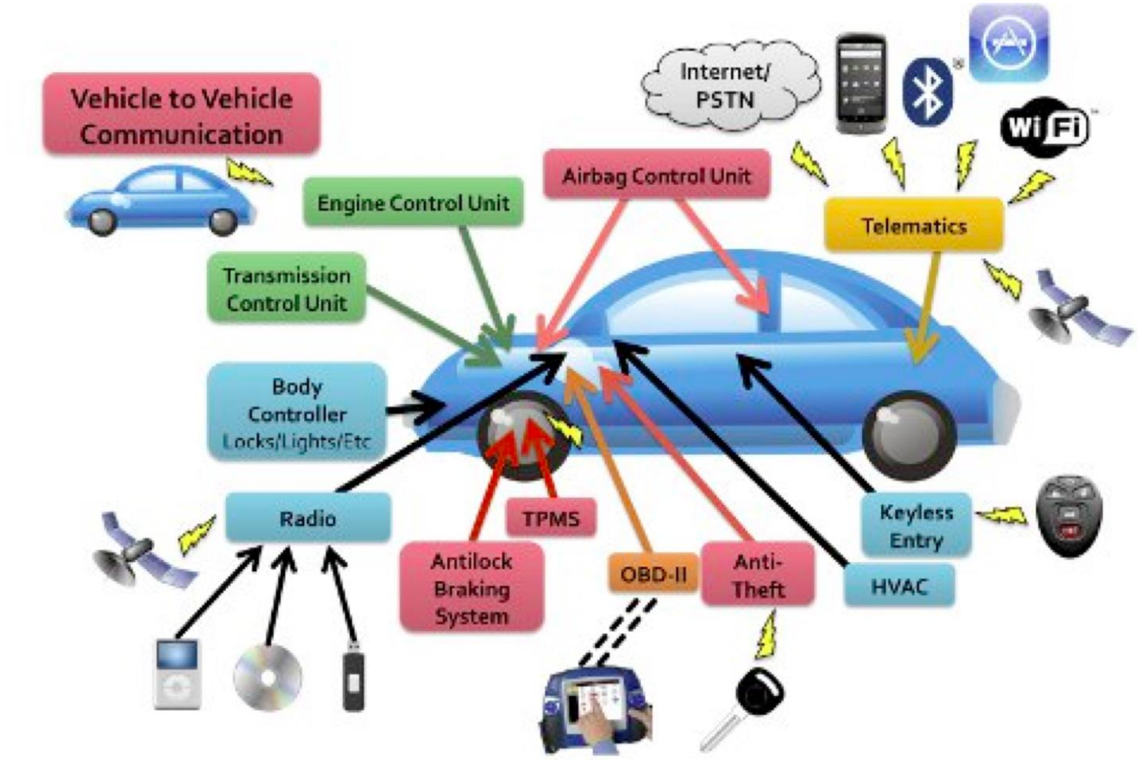
- Smartcards

Industrial Automation

- Industrial Automation Case Study from Codenomicon, Metso Automation, OUSPG, VTT

Dornier Consulting
Case Study I in Germany

- As Information and Communication Technology (ICT) systems become more and more part of our daily lives, current and future vehicles are more and more integrated into ICT networks.



Testing Techniques

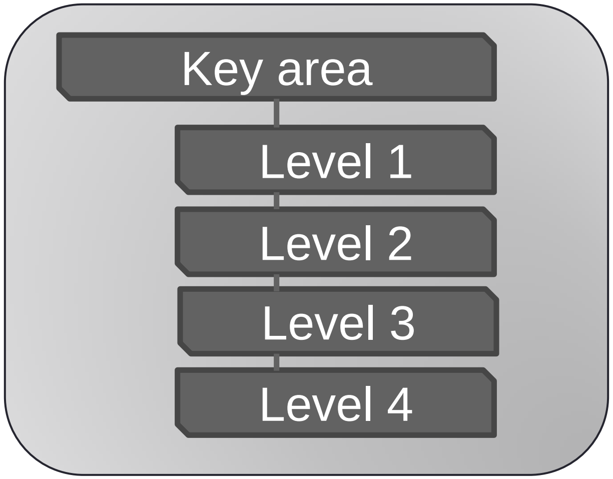
- Risk analysis with CORAS
- Fuzzing
- Symbolic execution and Parametric Trace Slicing
- Security monitoring

Evaluation of the DIAMONDS Case Studies
Security Testing Improvement Profiles (STIP)

Security Testing Improvement Profiles (STIP) enables an objective, detailed analysis and evaluation of your testing process

- Provide an objective, detailed analysis and evaluation of our research & development
- Show how out tools & techniques fit together
- Provide recommendations for other on how to pragmatically integrate our results to improve security testing processes on hand.
- Structure the order and target of the optimization steps

- Analysis with respect of the key areas
- Levels are used to assign a degree of progress to each key area
- Each higher level is better than its prior level in terms of time (faster), money (cheaper) and/or quality (better).



Evaluation of the DIAMONDS Case Studies
STIP key areas

Key area	Description
Security risk assessment	Security risk assessment is a process for identifying security risks.
Security test identification	Test identification is the process of identifying test purposes and appropriate security testing methods, techniques and tools.
Automated generation of test models	For model-based security testing (e.g. fuzzing, mutation based testing) various kinds of models are required, which can be either created manually or generated automatically.
Security test generation	Security test generation is about the automation of security test design.
Fuzzing	Fuzzing is about injecting invalid or random inputs in order to reveal unexpected behave or to identify errors and expose potential vulnerabilities.
Security test execution automation	The automation of security test execution conducts the automatic application of malicious data to the SUT, the automatic assessment of the SUT's state and output to clearly identify a security flaw, and the automatic control of the test execution with respect to different kind of caverages.
Security passive testing/ security monitoring	Security monitoring based on passive testing consists of detecting errors, vulnerabilities and security flaws in a system under test (SUT) or in operation by observing its behavior (input/output) without interfering with its normal operations.
Static security testing	Static security testing involves analysing application without executing it. One of the main components is code analysis.
Security test tool integration	Tool integration is the ability of tools to cooperate with respect to data interchange

Evaluation of the DIAMONDS Case Studies

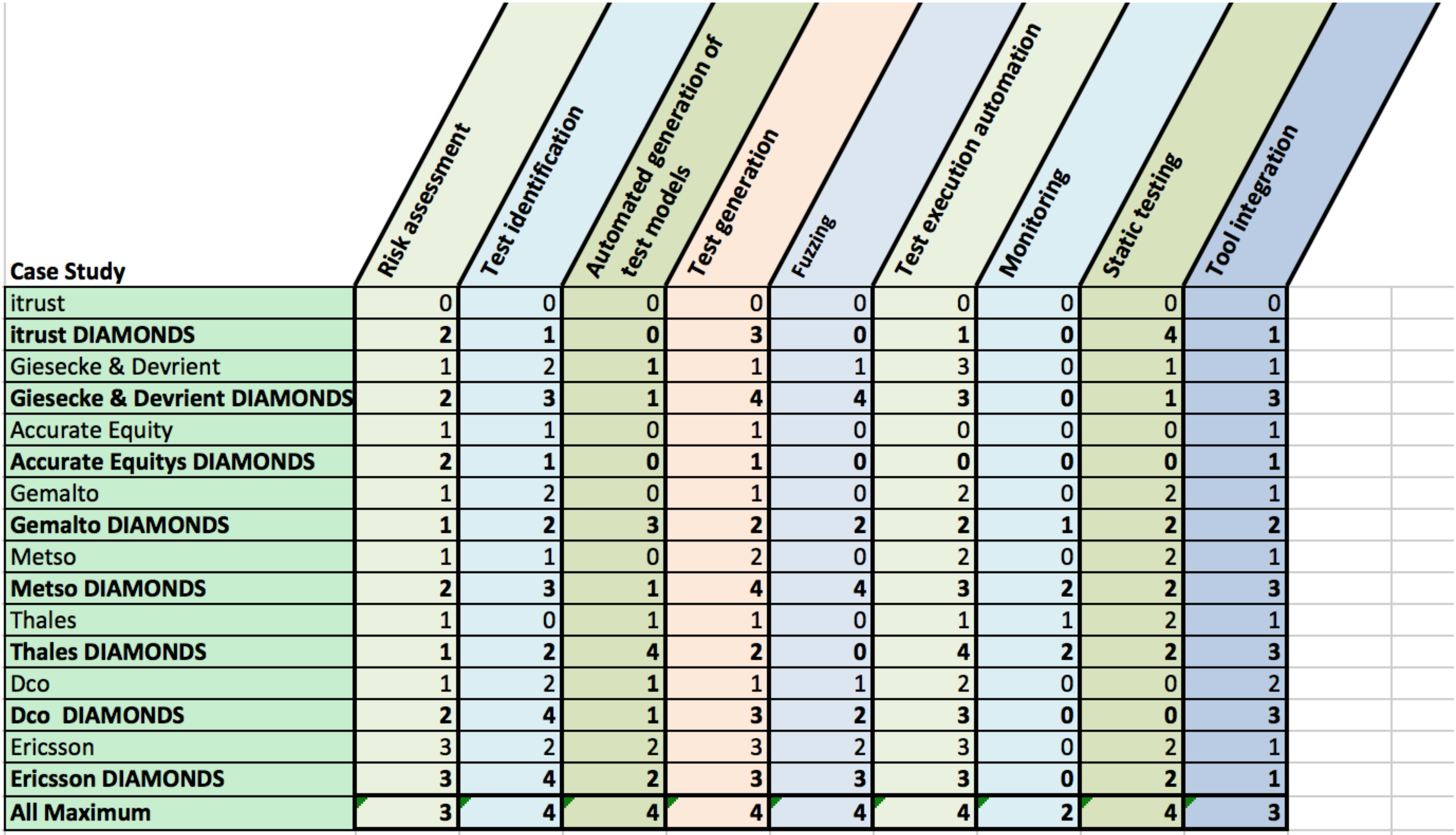
STIP level definition

Key area: Risk Assessment

#	Name	Description
L1	Informal security risk assessment	At this level, the security risk assessment is conducted in an unstructured manner without a specific notation/language for document risk assessment results or a clearly defined process for conducting the security risk assessment.
L2	Model-based security risk assessment	At this level, the security risk assessment is conducted in an unstructured manner without a specific notation/language for document risk assessment results or a clearly defined process for conducting the security risk assessment.
L3	Model and test-based security risk assessment	At this level, the security risk assessment is conducted with a language for documenting assessment results and a clearly defined process for conducting the assessment.
L4	Automated model and test-based security risk assessment	At this level, the model-based security risk assessment is uses testing for verifying the correctness of the risk assessment results.

Evaluation of the DIAMONDS Case Studies

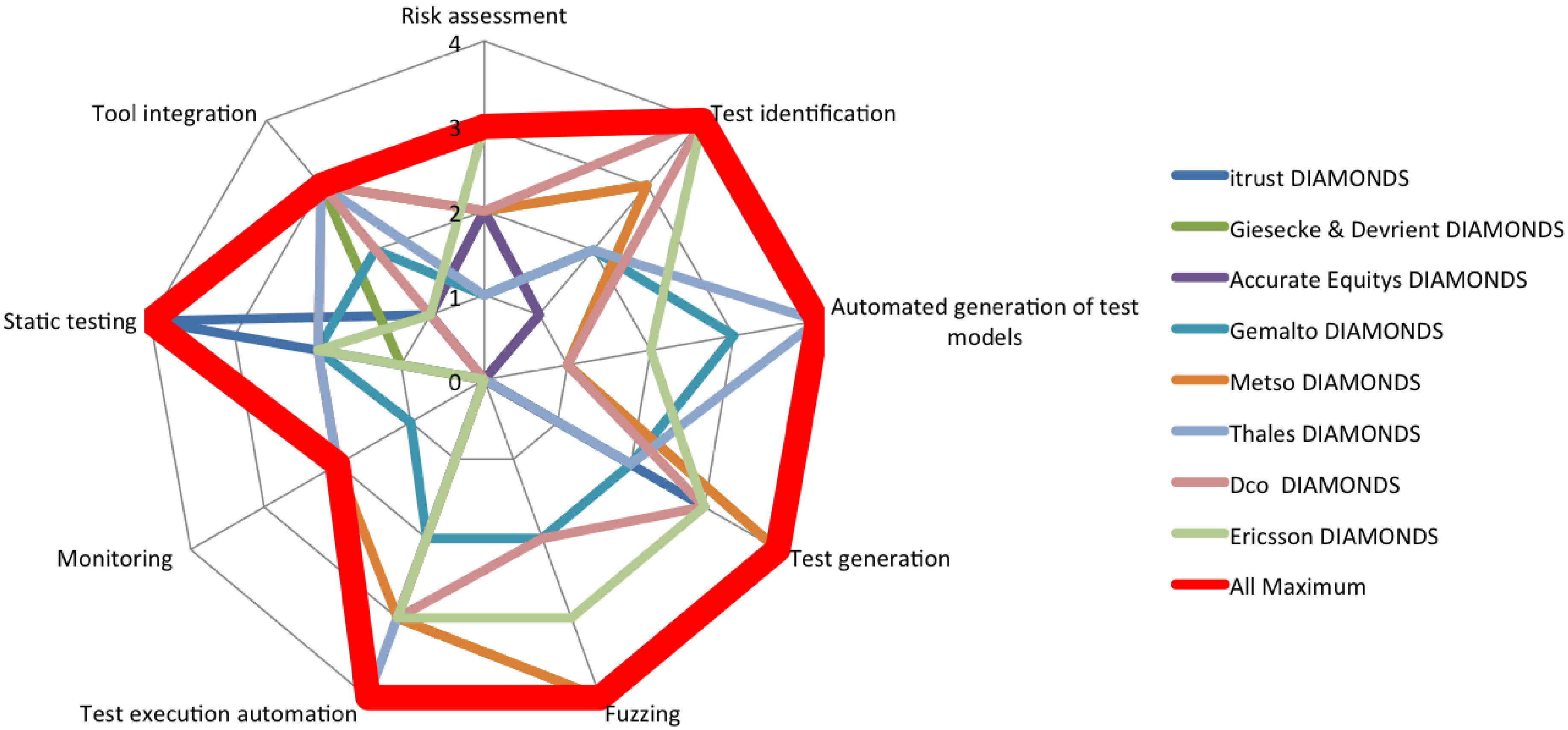
STIP results for the international case studies



Case Study	Risk assessment	Test identification	Automated generation of test models	Test generation	Fuzzing	Test execution automation	Monitoring	Static testing	Tool integration
itrust	0	0	0	0	0	0	0	0	0
itrust DIAMONDS	**2**	**1**	**0**	**3**	**0**	**1**	**0**	**4**	**1**
Giesecke & Devrient	1	2	1	1	1	3	0	1	1
Giesecke & Devrient DIAMONDS	**2**	**3**	**1**	**4**	**4**	**3**	**0**	**1**	**3**
Accurate Equity	1	1	0	1	0	0	0	0	1
Accurate Equitys DIAMONDS	**2**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**1**
Gemalto	1	2	0	1	0	2	0	2	1
Gemalto DIAMONDS	**1**	**2**	**3**	**2**	**2**	**2**	**1**	**2**	**2**
Metso	1	1	0	2	0	2	0	2	1
Metso DIAMONDS	**2**	**3**	**1**	**4**	**4**	**3**	**2**	**2**	**3**
Thales	1	0	1	1	0	1	1	2	1
Thales DIAMONDS	**1**	**2**	**4**	**2**	**0**	**4**	**2**	**2**	**3**
Dco	1	2	1	1	1	2	0	0	2
Dco DIAMONDS	**2**	**4**	**1**	**3**	**2**	**3**	**0**	**0**	**3**
Ericsson	3	2	2	3	2	3	0	2	1
Ericsson DIAMONDS	**3**	**4**	**2**	**3**	**3**	**3**	**0**	**2**	**1**
All Maximum	**3**	**4**	**4**	**4**	**4**	**4**	**2**	**4**	**3**

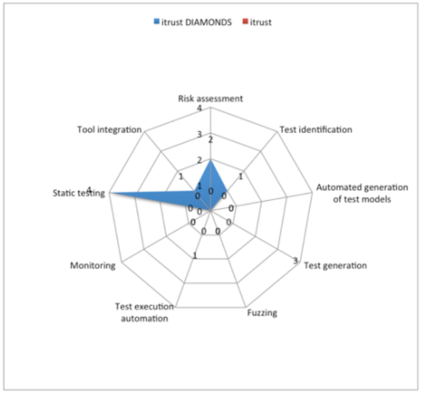
Evaluation of the DIAMONDS Case Studies

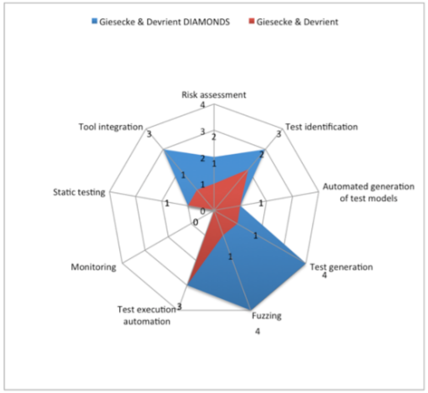
Progress in all case studies

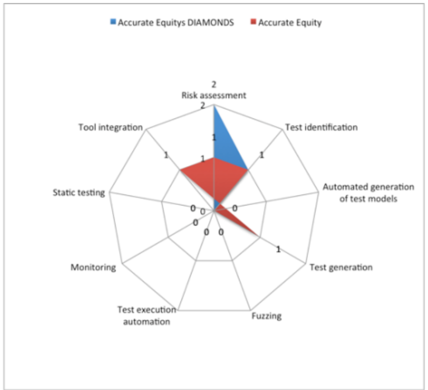


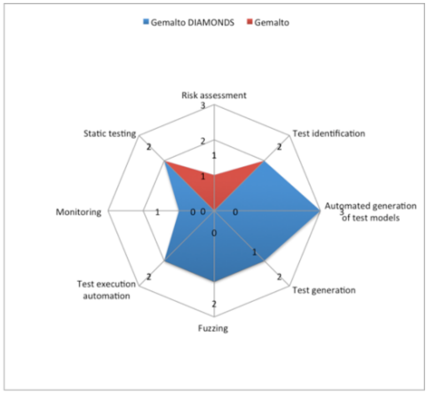
Evaluation of the DIAMONDS Case Studies

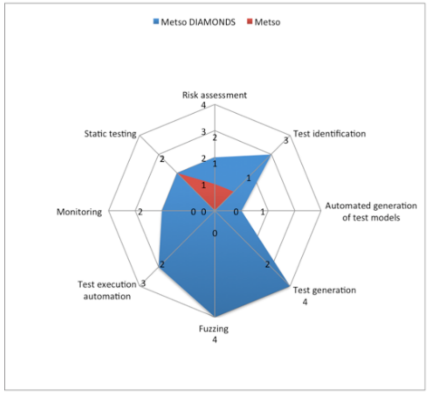
Progress in all case studies

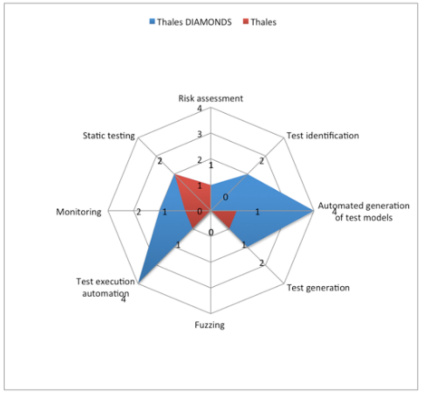


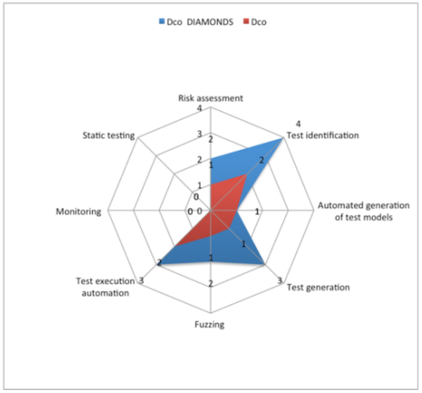


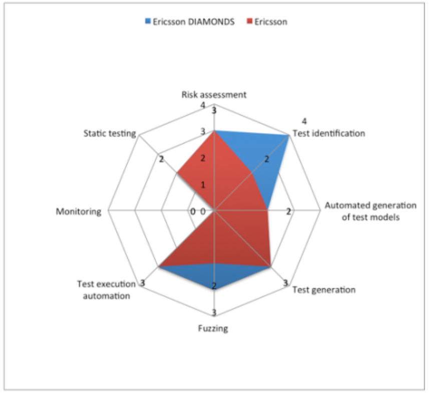












Outline

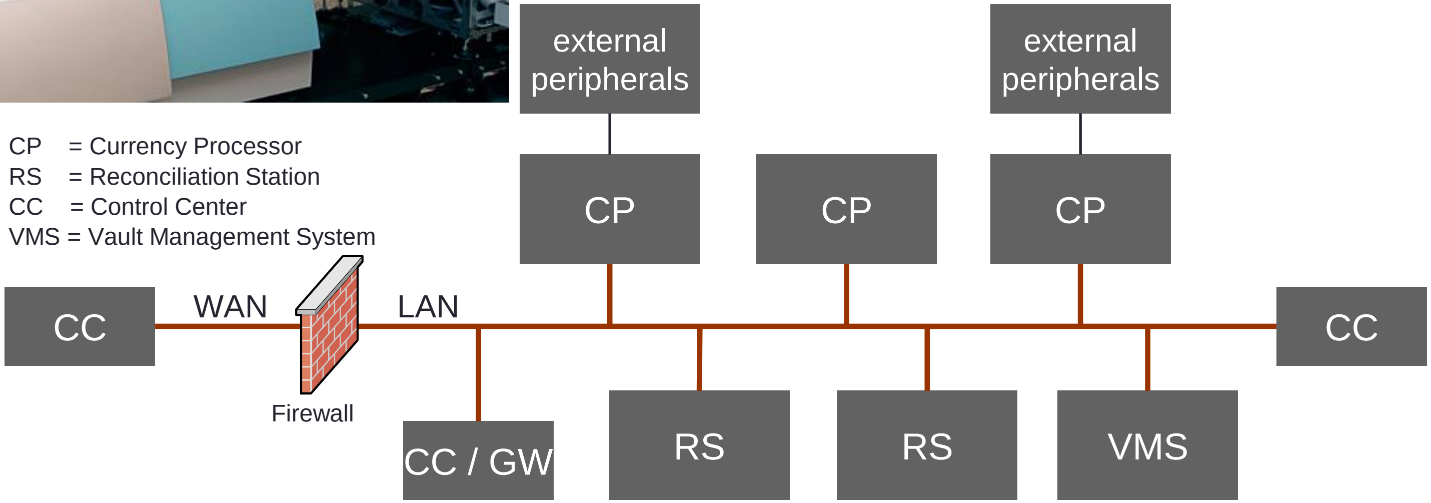
- Introduction and Overview
- Security Testing Improvement Profiles and Industrial Case Studies
- Details of Giesecke & Devrient Case Study
- Security Testing Approach and Traceing
- Summary

Giesecke & Devrient

Case Study II in Germany

Banknote processing machine that counts, sorts and assesses banknotes by their currency, denomination, condition and authenticity.

CP = Currency Processor
RS = Reconciliation Station
CC = Control Center
VMS = Vault Management System



Giesecke & Devrient

Giesecke & Devrient

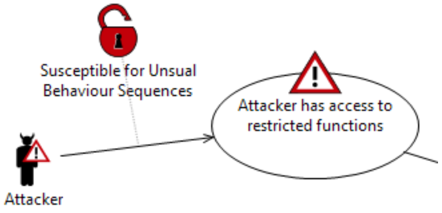
Case study characterization

- Security challenges
  - **Restricted access to functions:** The access to functions is restricted to authorized users.
  - **Operation system access restriction:** The access to the operation system, i.e. file system, or process monitor is restricted to authorized users.
  - **Prevent Admin Hijacking:** Hijacking an administrator account is used to get the privileges of an administrator account as a user that is not assigned to the administrator group.
  - **Prevent infiltration/manipulation of software:** Software manipulation can be used to fake data or to provoke errors on the currency processor application.
  - **Prevent manipulation of application configuration:** Manipulation could possibly change the classification of banknotes.

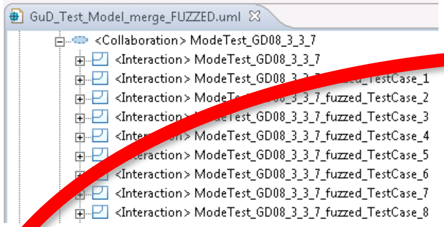
Giesecke & Devrient

DEMO: Online MBBF

CORAS Risk Analysis
Deliverable D1.WP2*

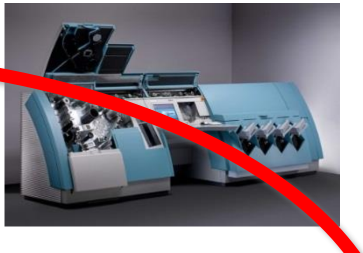

Susceptible for Unsual Behaviour Sequences
Attacker has access to restricted functions
Attacker

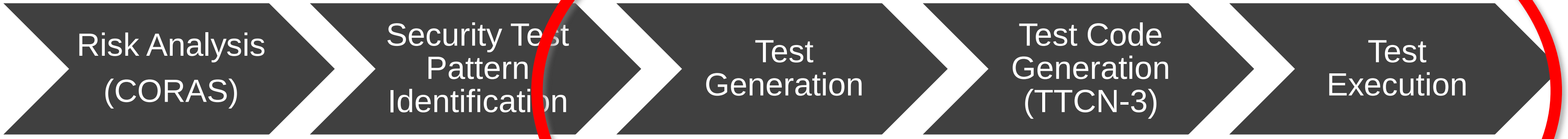
Behavioural Fuzzing
Deliverable D2.WP2* (see also next slide), D3.WP2*



```
<Collaboration> ModeTest_GD08_3_3_7
  <Interaction> ModeTest_GD08_3_3_7
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_1
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_2
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_3
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_4
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_5
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_6
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_7
  <Interaction> ModeTest_GD08_3_3_7_fuzzed_TestCase_8
```

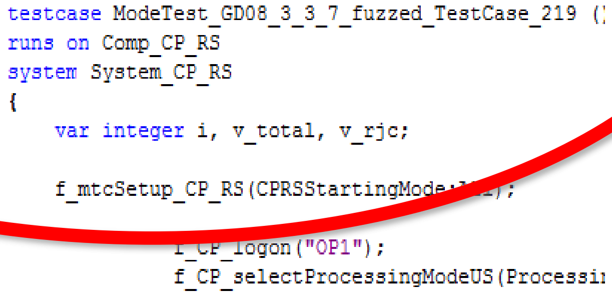
Data Fuzzing with TTCN-3
Deliverable D3.WP3*




Risk Analysis (CORAS)
Security Test Pattern Identification
Test Generation
Test Code Generation (TTCN-3)
Test Execution

Pattern name	Usage of Unusual Behavior Sequences
Context	Test pattern kind: Behavior Testing Approach(es): Prevention
Problem/Goal	Security of information systems is ensured in many cases by a strict and clear definition of what constitutes valid behavior sequences from the security perspective on those systems. For example…
Solution	Test procedure template: 1. … 2. …
Known uses	Model-based behavioural fuzzing of sequence diagrams is an application of this pattern

Security Test Pattern Catalogue
Deliverable D3.WP4.T1*



```
testcase ModeTest_GD08_3_3_7_fuzzed_TestCase_219 (
runs on Comp_CP_RS
system System_CP_RS
{
    var integer i, v_total, v_rjc;

    f_mtcSetup_CP_RS(CPRSStartingMode:[illegible]);

                f_CP_logon("OP1");
                f_CP_selectProcessingModeUS(Processi
```

*project deliverables are available at www.itea2-diamonds.org “publications”

Giesecke & Devrient

Results

- **Focus on risks related to**
  - unauthorized access
  - machine/configuration modification

- **Until now, no weaknesses were found**
  - confidence in the security of the system is strengthened

- **Metrics**
  - different security levels depending on the covered risks/vulnerabilities by
    - **number of test cases (one or more) per risk/vulnerability**
      unauthorized access, configuration modification: more
    - **number of test methods to generate these test cases**
      data fuzzing and behavioural fuzzing: 2 test methods

Giesecke & Devrient

Exploitation

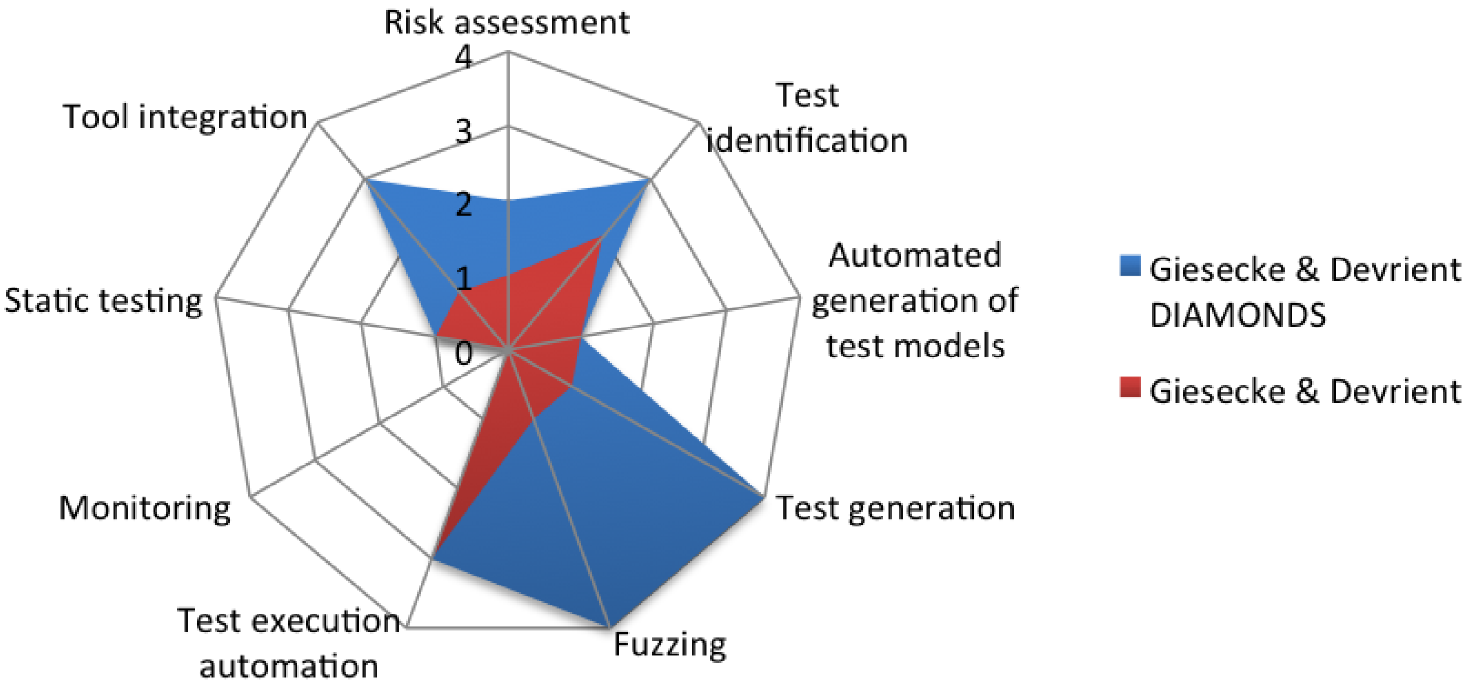
- **CORAS method for risk analysis has been proved of value**
  - graphical modelling
  - specification of assets to be protected

- **Saved resources due to**
  - reuse of functional test cases and
  - reuse of test execution environment for non-functional security testing
  - integration of data fuzzing in the TTCN-3 execution environment
    - keeps the behavioural model clean and concise
    - allows easy combination of data and behavioural fuzzing

- **Standardization of DIAMONDS results provides certification options for products with security requirements**

Giesecke & Devrient
Summary

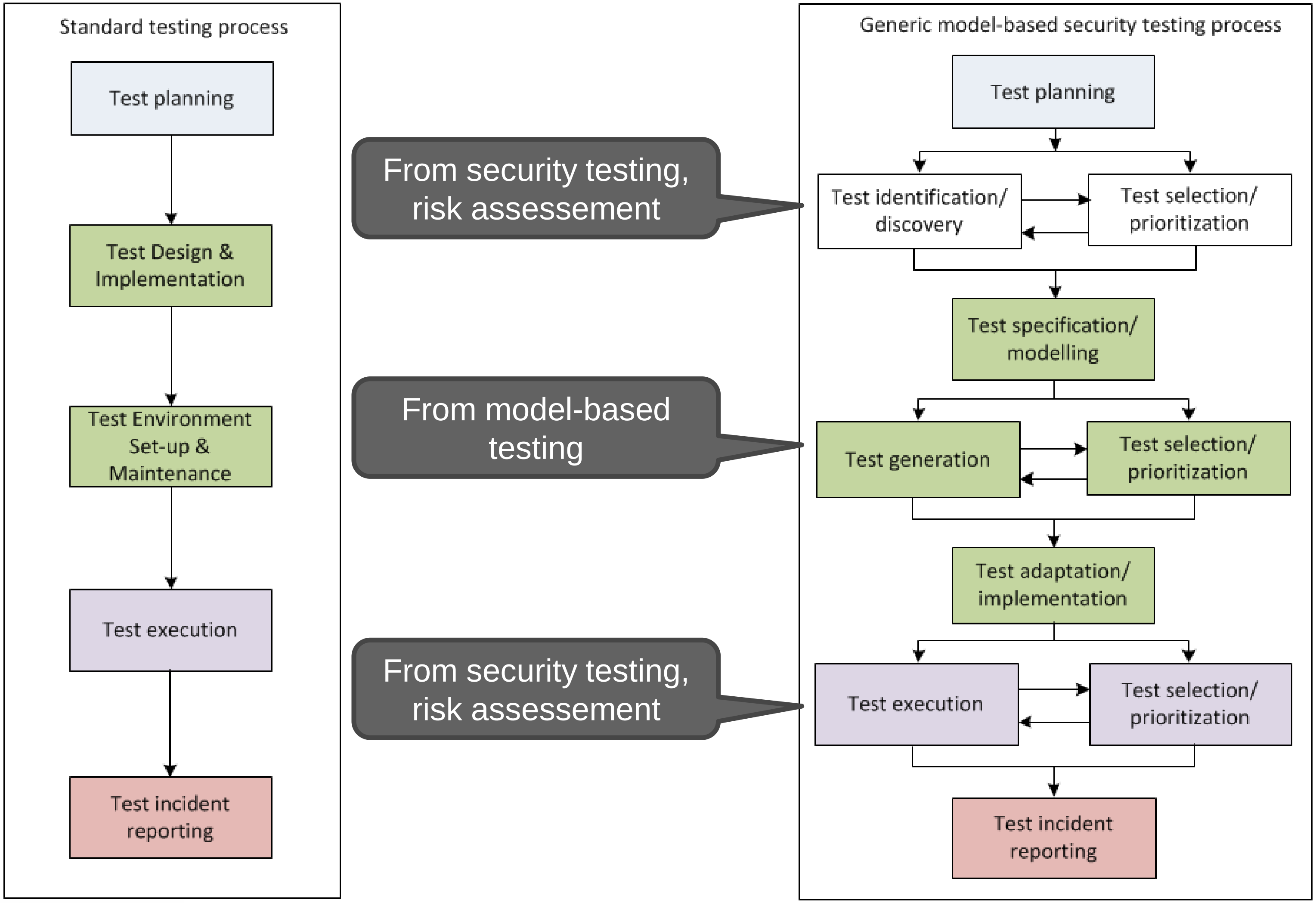
- **Improvement gains according to our STIP:**



Outline

- Introduction and Overview
- Security Testing Improvement Profiles and Industrial Case Studies
- Details of Giesecke & Devrient Case Study
- Security Testing Approach and Traceing
- Summary

The DIAMONDS Process for Model-Based Security Testing

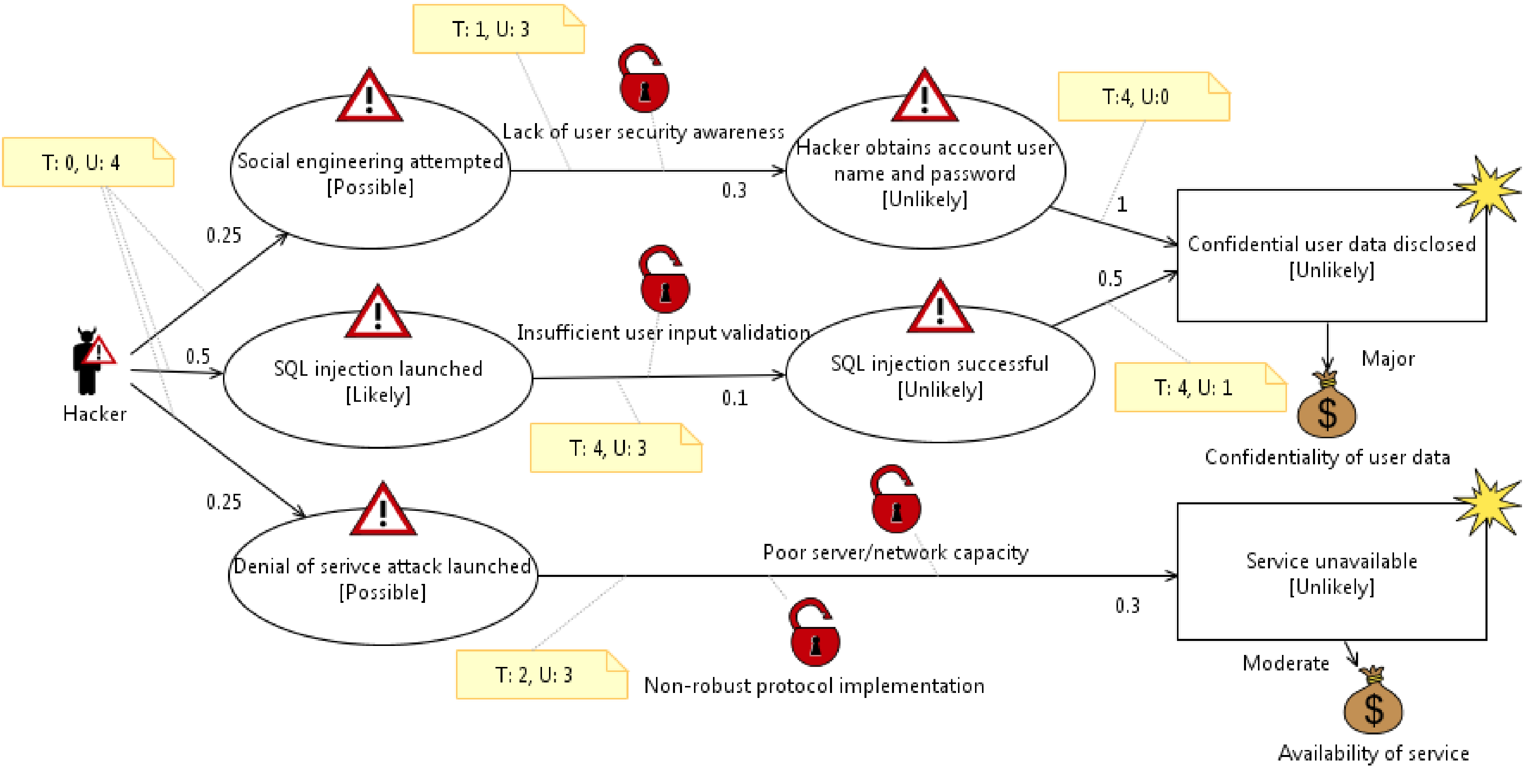

Standard testing process
Test planning
Test Design & Implementation
Test Environment Set-up & Maintenance
Test execution
Test incident reporting
Generic model-based security testing process
Test planning
Test identification/ discovery
Test selection/ prioritization
Test specification/ modelling
Test generation
Test selection/ prioritization
Test adaptation/ implementation
Test execution
Test selection/ prioritization
Test incident reporting

From security testing, risk assessement

From model-based testing

From security testing, risk assessement

Test Prioritization Exemplified


T: 1, U: 3
T: 0, U: 4
Social engineering attempted [Possible]
Lack of user security awareness
Hacker obtains account user name and password [Unlikely]
T:4, U:0
0.3
1
0.25
Confidential user data disclosed [Unlikely]
0.5
Hacker
0.5
SQL injection launched [Likely]
Insufficient user input validation
SQL injection successful [Unlikely]
0.1
T: 4, U: 3
T: 4, U: 1
Major
Confidentiality of user data
0.25
Denial of serivce attack launched [Possible]
Poor server/network capacity
Service unavailable [Unlikely]
0.3
T: 2, U: 3
Non-robust protocol implementation
Moderate
Availability of service

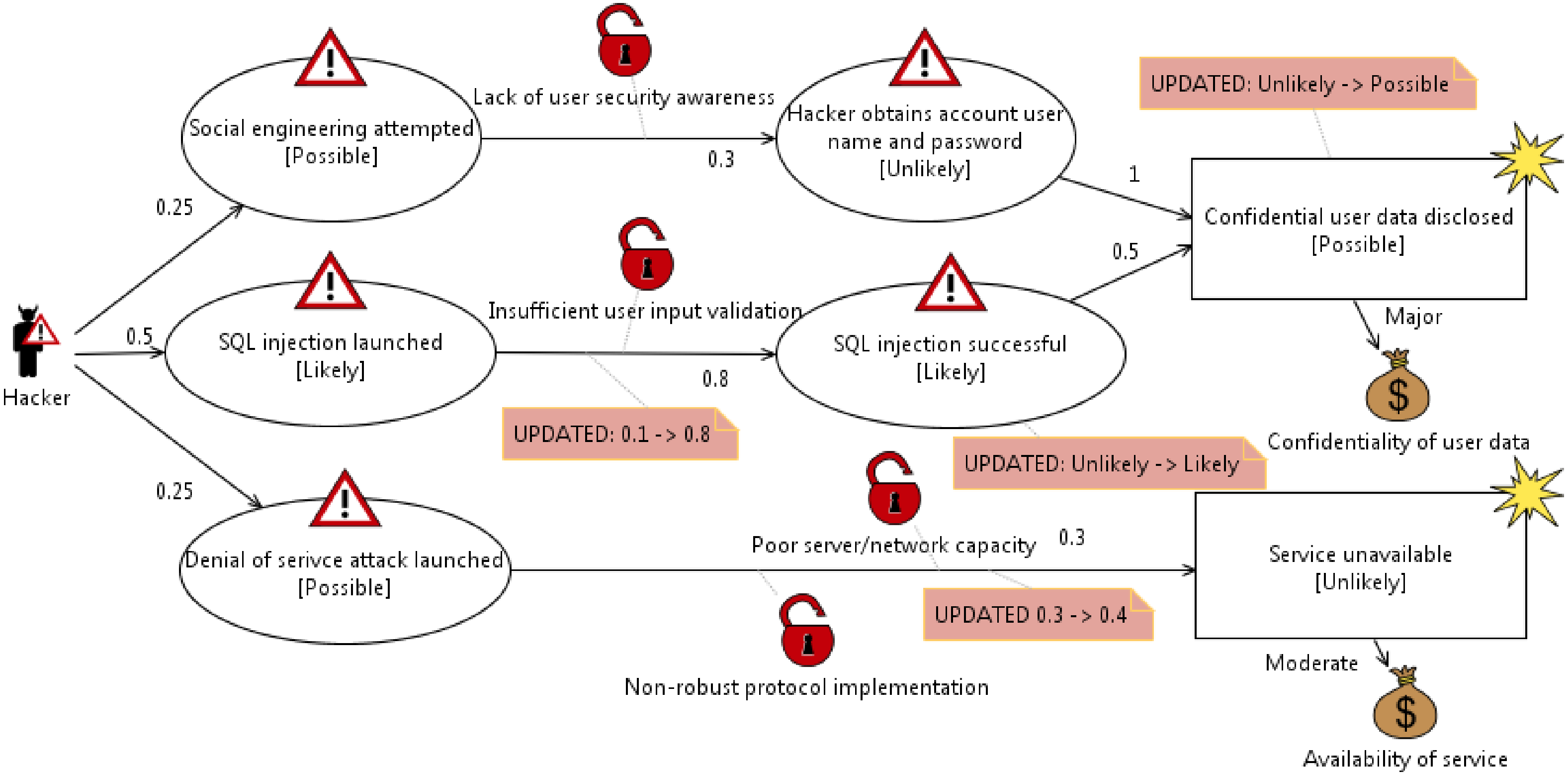
Prioritization is based on

- Testability (T)
- Uncertainty (U)
- Severity (S)

Test Prioritization Exemplified (cont.)

Id	Test scenario	S	T	U	Priority
TS5	SQL injection launched leads to SQL injection successful with conditional likelihood 0.1, due to Insufficient user input validation.	3	4	3	36
TS6	Denial of service attack launched leads Service unavailable with conditional likelihood 0.3, due to Poor server/network capacity and Non-robust protocol implementation.	3.2	2	3	19.2
TS4	Social engineering attempted leads to Hacker obtains account user name and password with conditional likelihood 0.3, due to Lack of user security awareness.	1.5	1	3	4.5
TS1	Hacker initiates Social engineering attempted with likelihood 0.25.	2.5	0	4	0
TS2	Hacker initiates SQL injection launched with likelihood 0.5.	2.5	0	4	0
TS3	Hacker initiates Denial of service attack launched with likelihood 0.25.	2.5	0	4	0
TS7	Hacker obtains account user name and password leads to Confidential user data disclosed with conditional likelihood 1.	1	4	0	0
TS8	SQL injection successful leads to Confidential user data disclosed with conditional likelihood 0.5.	2	4	0	0

Risk Validation and Treatment Exemplified


Lack of user security awareness
Social engineering attempted [Possible]
0.3
Hacker obtains account user name and password [Unlikely]
UPDATED: Unlikely -> Possible
1
Confidential user data disclosed [Possible]
0.25
0.5
Hacker
Insufficient user input validation
SQL injection launched [Likely]
0.8
SQL injection successful [Likely]
0.5
Major
Confidentiality of user data
UPDATED: 0.1 -> 0.8
UPDATED: Unlikely -> Likely
0.25
Denial of serivce attack launched [Possible]
Poor server/network capacity
0.3
Service unavailable [Unlikely]
UPDATED 0.3 -> 0.4
Non-robust protocol implementation
Moderate
Availability of service

Traceability Platform for RBST
Description

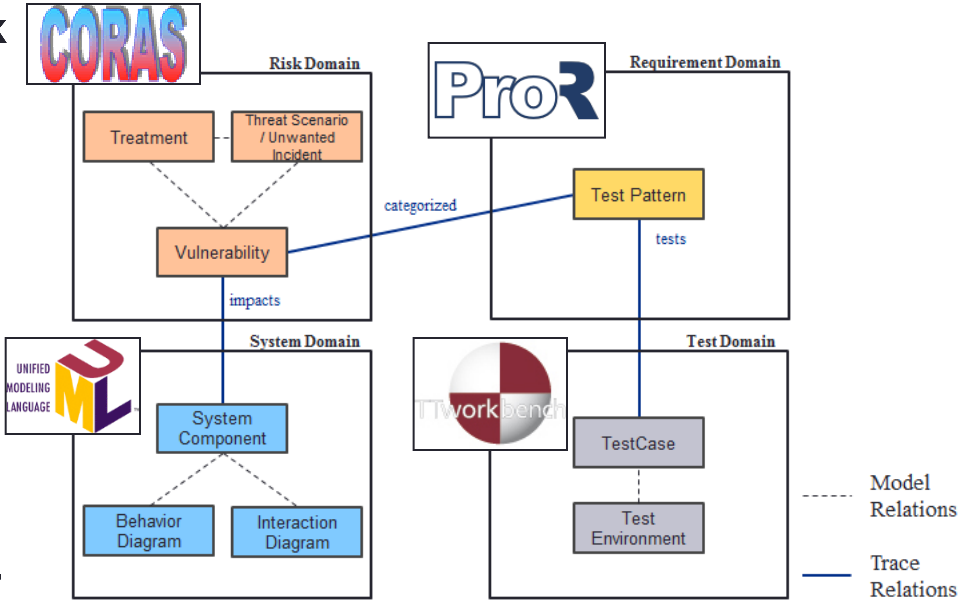
Dedicated traceability support for risk based security testing.

Enables traceability between security testing artefacts.

- Risk model elements (threats, vulnerabilities, unwanted incidents)
- UML model elements
- Security test cases, test pattern and test results
- Security requirements

Allows for interaction/combination of different security engineering and testing tools

- Follow traces from security threats, vulnerabilities and their associated risks to testing artefacts
- basis to determine coverage/completeness metrics (e.g. risks coverage)

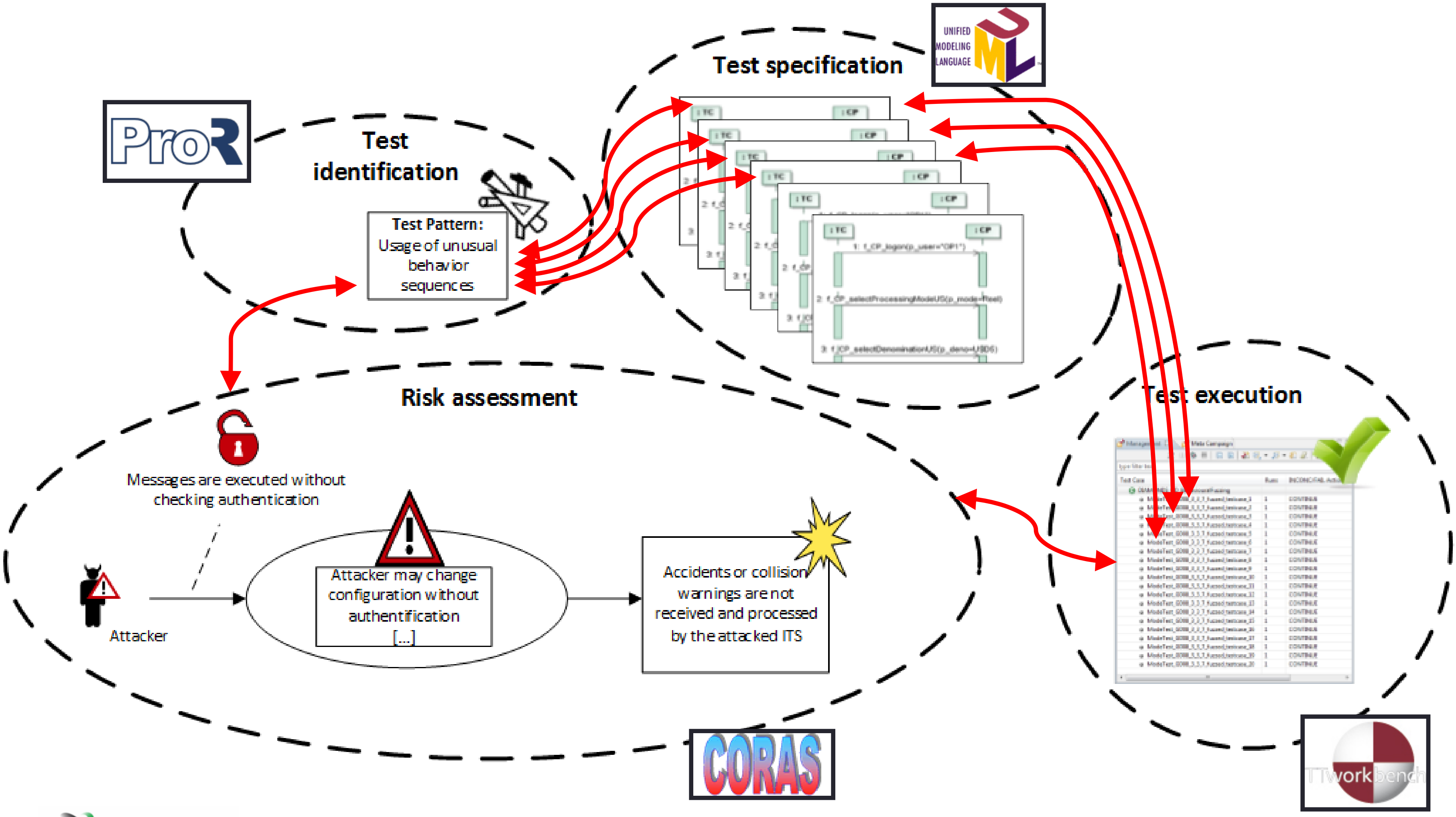


Fully integrated in **Eclipse**

Based on **open source tool CREMA**

Traceability Platform for RBST

Demo: CORAS, Papyrus, ProR and TTworkbench



Test identification

Test Pattern:
Usage of unusual behavior sequences

Test specification

Test execution

Risk assessment

Messages are executed without checking authentication

Attacker

Attacker may change configuration without authentification
[...]

Accidents or collision warnings are not received and processed by the attacked ITS

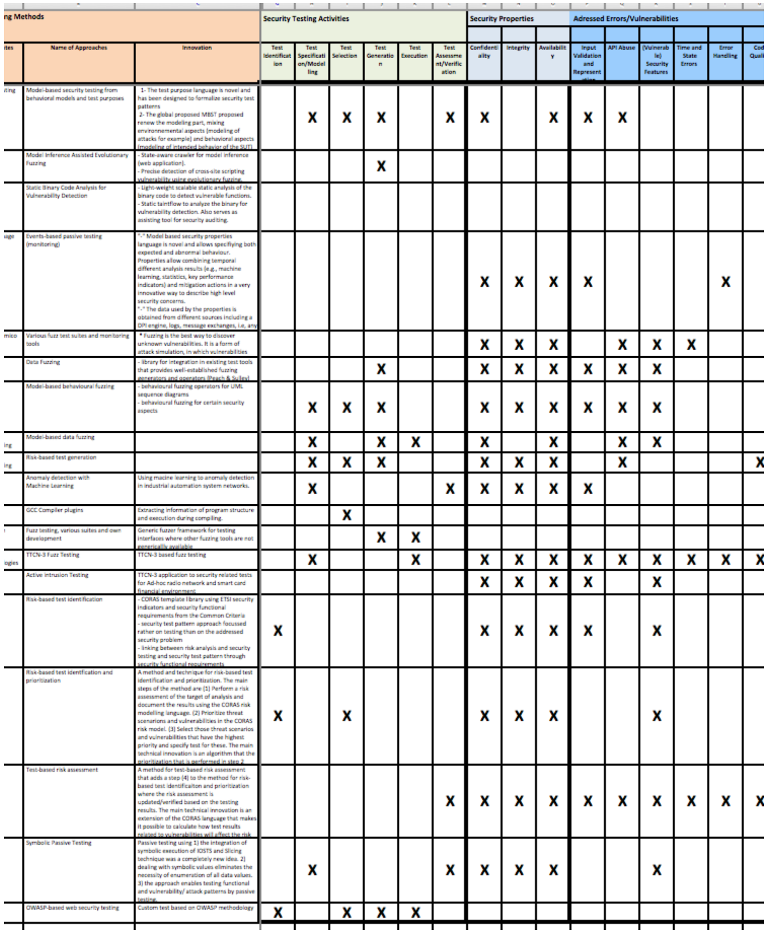
ProR
UNIFIED MODELING LANGUAGE
: TC
: CP
1: t_CP_logon(p_user="OP1")
2: t_CP_selectProcessingModeUS(p_mode=Reel)
3: t_CP_selectDenominationUS(p_deno=USD5)
CORAS
TTworkbench

Outline

- Introduction and Overview
- Security Testing Improvement Profiles and Industrial Case Studies
- Details of Giesecke & Devrient Case Study
- Security Testing Approach and Traceing
- Summary

Techniques Overview

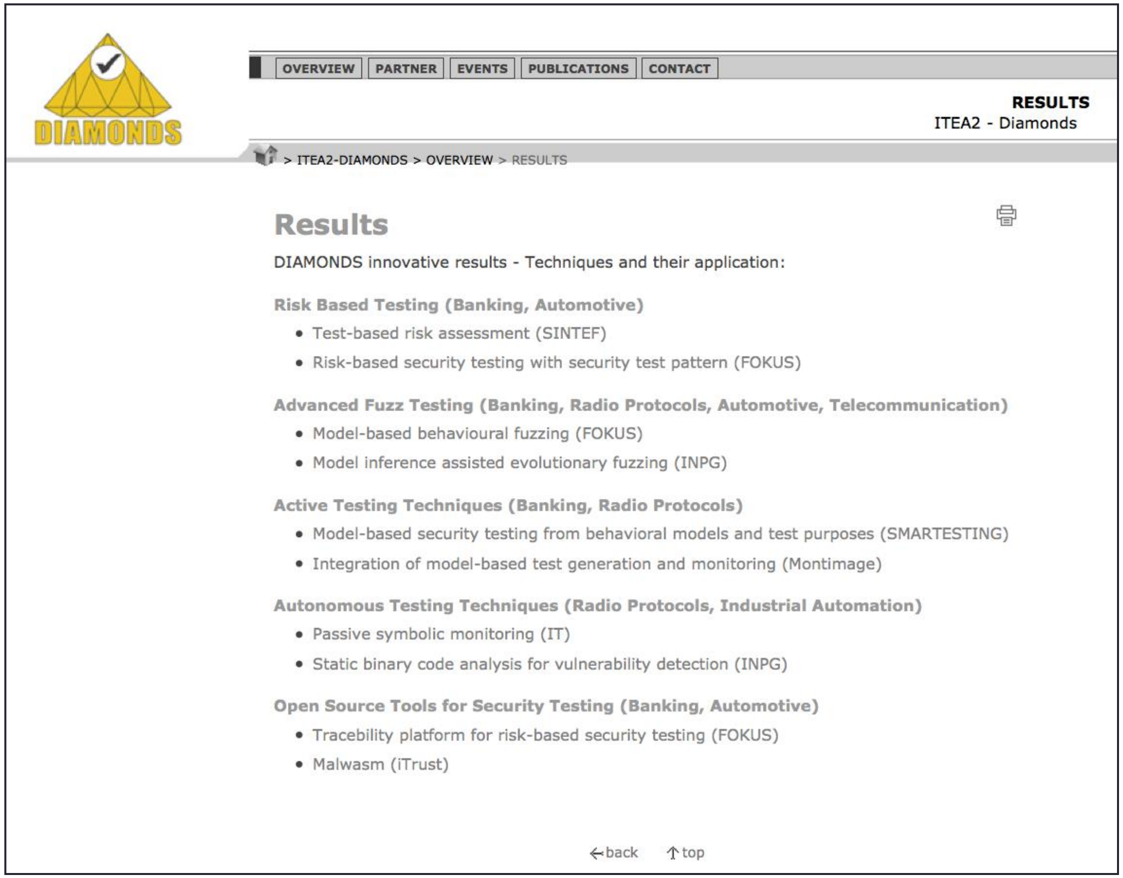
- **17 different techniques** developed
- Techniques cover **all phases of a security testing process** (test identification, test specification/modeling, test generation, test execution, test assessment)
- Techniques cover **all security properties** (confidentiality, availability, integrity)
- Techniques cover **all kinds of vulnerability classes** (input validation, API abuse, security features, time and state error, error handling)



ng Methods			Security Testing Activities						Security Properties			Adressed Errors/Vulnerabilities					
tes	Name of Approaches	Innovation	Test Identification	Test Specification/Modelling	Test Selection	Test Generation	Test Execution	Test Assessment/Verification	Confidentiality	Integrity	Availability	Input Validation and Represent	API Abuse	(Vulnerable) Security Features	Time and State Errors	Error Handling	Cod Quali
ting	Model-based security testing from behavioral models and test purposes	1- The test purpose language is novel and has been designed to formalize security test patterns 2- The global proposed MBST proposed renew the modeling part, mixing environnemental aspects (modeling of attacks for example) and behavioral aspects (modeling of intended behavior of the SUT)		X	X	X		X	X		X	X	X				
	Model Inference Assisted Evolutionary Fuzzing	- State-aware crawler for model inference (web application). - Precise detection of cross-site scripting vulnerability using evolutionary fuzzing.				X											
	Static Binary Code Analysis for Vulnerability Detection	- Light-weight scalable static analysis of the binary code to detect vulnerable functions. - Static taintflow to analyze the binary for vulnerability detection. Also serves as assisting tool for security auditing.															
uage	Events-based passive testing (monitoring)	"-" Model based security properties language is novel and allows specifying both expected and abnormal behaviour. Properties allow combining temporal different analysis results (e.g., machine learning, statistics, key performance indicators) and mitigation actions in a very innovative way to describe high level security concerns. "-" The data used by the properties is obtained from different sources including a DPI engine, logs, message exchanges, i.e, any							X	X	X	X				X	
mico	Various fuzz test suites and monitoring tools	* Fuzzing is the best way to discover unknown vulnerabilities. It is a form of attack simulation, in which vulnerabilities							X	X	X		X	X	X		
	Data Fuzzing	- library for integration in existing test tools that provides well-established fuzzing generators and operators (Peach & Sulley)				X			X	X	X	X	X	X			
	Model-based behavioural fuzzing	- behavioural fuzzing operators for UML sequence diagrams - behavioural fuzzing for certain security aspects		X	X	X			X	X	X	X	X	X			
ing	Model-based data fuzzing			X		X	X		X		X		X	X			
ing	Risk-based test generation			X	X	X			X	X	X		X				X
	Anomaly detection with Machine Learning	Using macine learning to anomaly detection in industrial automation system networks.		X				X	X	X	X	X					
	GCC Compiler plugins	Extracting information of program structure and execution during compiling.			X												
t	Fuzz testing, various suites and own development	Generic fuzzer framework for testing interfaces where other fuzzing tools are not genericallly available				X	X										
logies	TTCN-3 Fuzz Testing	TTCN-3 based fuzz testing		X			X		X	X	X	X	X	X	X	X	X
	Active intrusion Testing	TTCN-3 application to security related tests for Ad-hoc radio network and smart card financial environment							X	X	X	X		X			
	Risk-based test identification	- CORAS template library using ETSI security indicators and security functional requirements from the Common Criteria - security test pattern approach focussed rather on testing than on the addressed security problem - linking between risk analysis and security testing and security test pattern through security functional requirements	X						X	X	X	X		X			
	Risk-based test identification and prioritization	A method and technique for risk-based test identification and prioritization. The main steps of the method are (1) Perform a risk assessment of the target of analysis and document the results using the CORAS risk modelling language. (2) Prioritize threat scenarions and vulnerabilities in the CORAS risk model. (3) Select those threat scenarios and vulnerabilities that have the highest priority and specify test for these. The main technical innovation is an algorithm that the prioritization that is performed in step 2	X		X				X	X	X			X			
	Test-based risk assessment	A method for test-based risk assessment that adds a step (4) to the method for risk-based test identificaiton and prioritization where the risk assessment is updated/verified based on the testing results. The main technical innovation is an extension of the CORAS language that makes it possible to calculate how test results related to vulnerabilities will affect the risk						X	X	X	X	X	X	X	X	X	X
	Symbolic Passive Testing	Passive testing using 1) the integration of symbolic execution of IOSTS and Slicing technique was a completely new idea. 2) dealing with symbolic values eliminates the necessity of enumeration of all data values. 3) the approach enables testing functional and vulnerability/ attack patterns by passive testing.		X				X	X	X	X			X			
	OWASP-based web security testing	Custom test based on OWASP methodology	X		X	X	X										

Innovation Sheets

- Collection of the innovative DIAMONDS techniques
- Common structure
  - Technique description
  - State of the art
  - Advances beyond the state of the art
  - Exploitation and application to case studies
- Available at DIAMONDS web site



Results

DIAMONDS innovative results - Techniques and their application:

Risk Based Testing (Banking, Automotive)

- Test-based risk assessment (SINTEF)
- Risk-based security testing with security test pattern (FOKUS)

Advanced Fuzz Testing (Banking, Radio Protocols, Automotive, Telecommunication)

- Model-based behavioural fuzzing (FOKUS)
- Model inference assisted evolutionary fuzzing (INPG)

Active Testing Techniques (Banking, Radio Protocols)

- Model-based security testing from behavioral models and test purposes (SMARTESTING)
- Integration of model-based test generation and monitoring (Montimage)

Autonomous Testing Techniques (Radio Protocols, Industrial Automation)

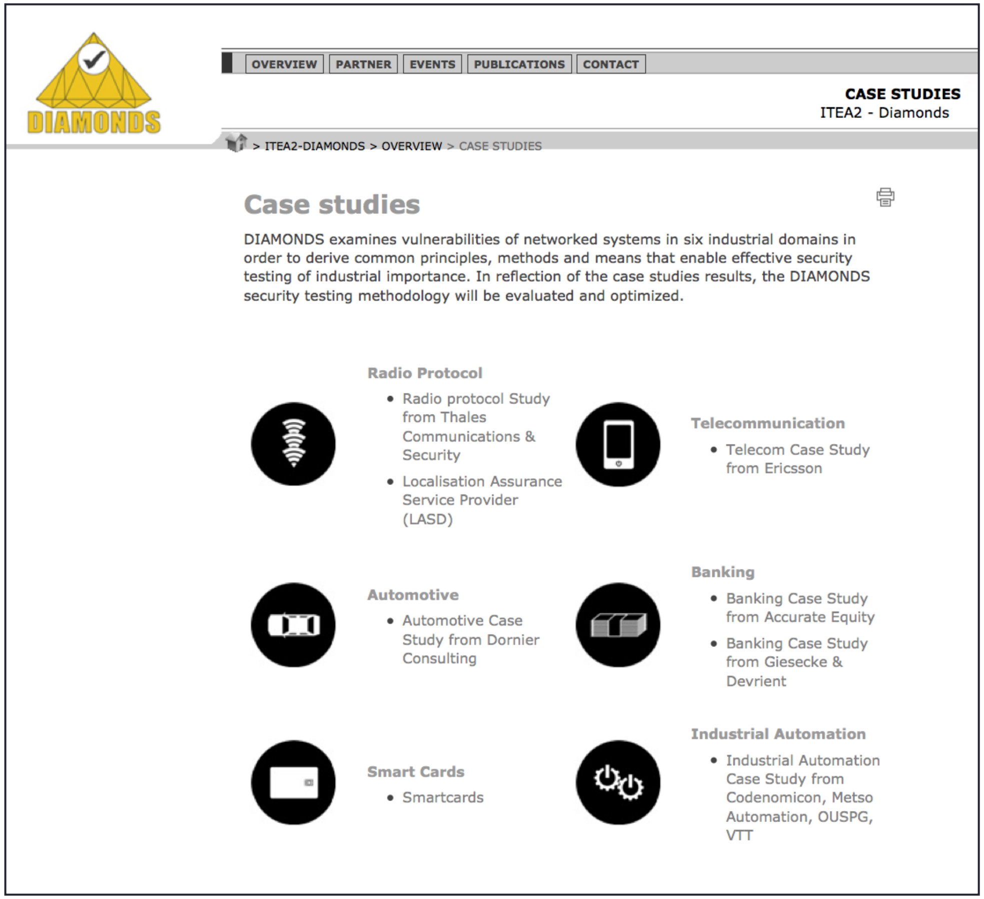
- Passive symbolic monitoring (IT)
- Static binary code analysis for vulnerability detection (INPG)

Open Source Tools for Security Testing (Banking, Automotive)

- Tracebility platform for risk-based security testing (FOKUS)
- Malwasm (iTrust)

Case Study Experiences

- Collection of the experiences and results for all case studies
  - Case study experience sheets
  - Available at DIAMONDS web site
- STIP Evaluation
  - Shows progress in all case studies



CASE STUDIES
ITEA2 - Diamonds

> ITEA2-DIAMONDS > OVERVIEW > CASE STUDIES

Case studies

DIAMONDS examines vulnerabilities of networked systems in six industrial domains in order to derive common principles, methods and means that enable effective security testing of industrial importance. In reflection of the case studies results, the DIAMONDS security testing methodology will be evaluated and optimized.

Radio Protocol

- Radio protocol Study from Thales Communications & Security
- Localisation Assurance Service Provider (LASD)

Telecommunication

- Telecom Case Study from Ericsson

Automotive

- Automotive Case Study from Dornier Consulting

Banking

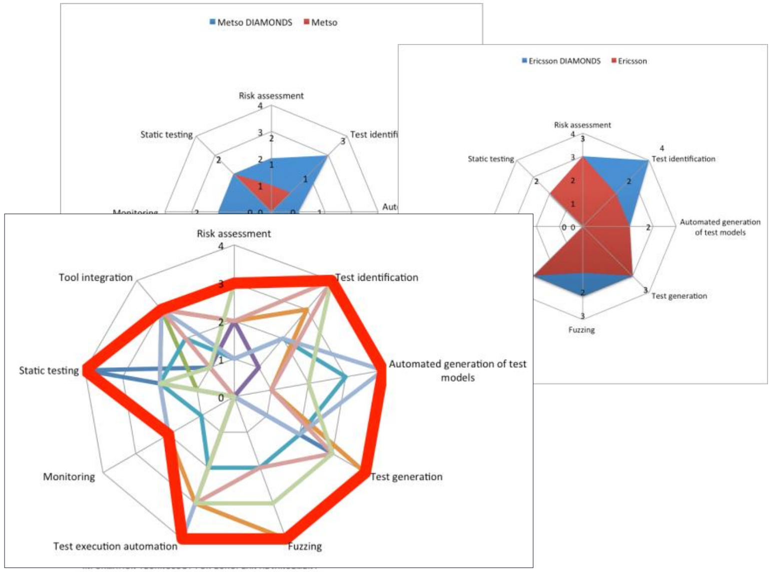
- Banking Case Study from Accurate Equity
- Banking Case Study from Giesecke & Devrient

Smart Cards

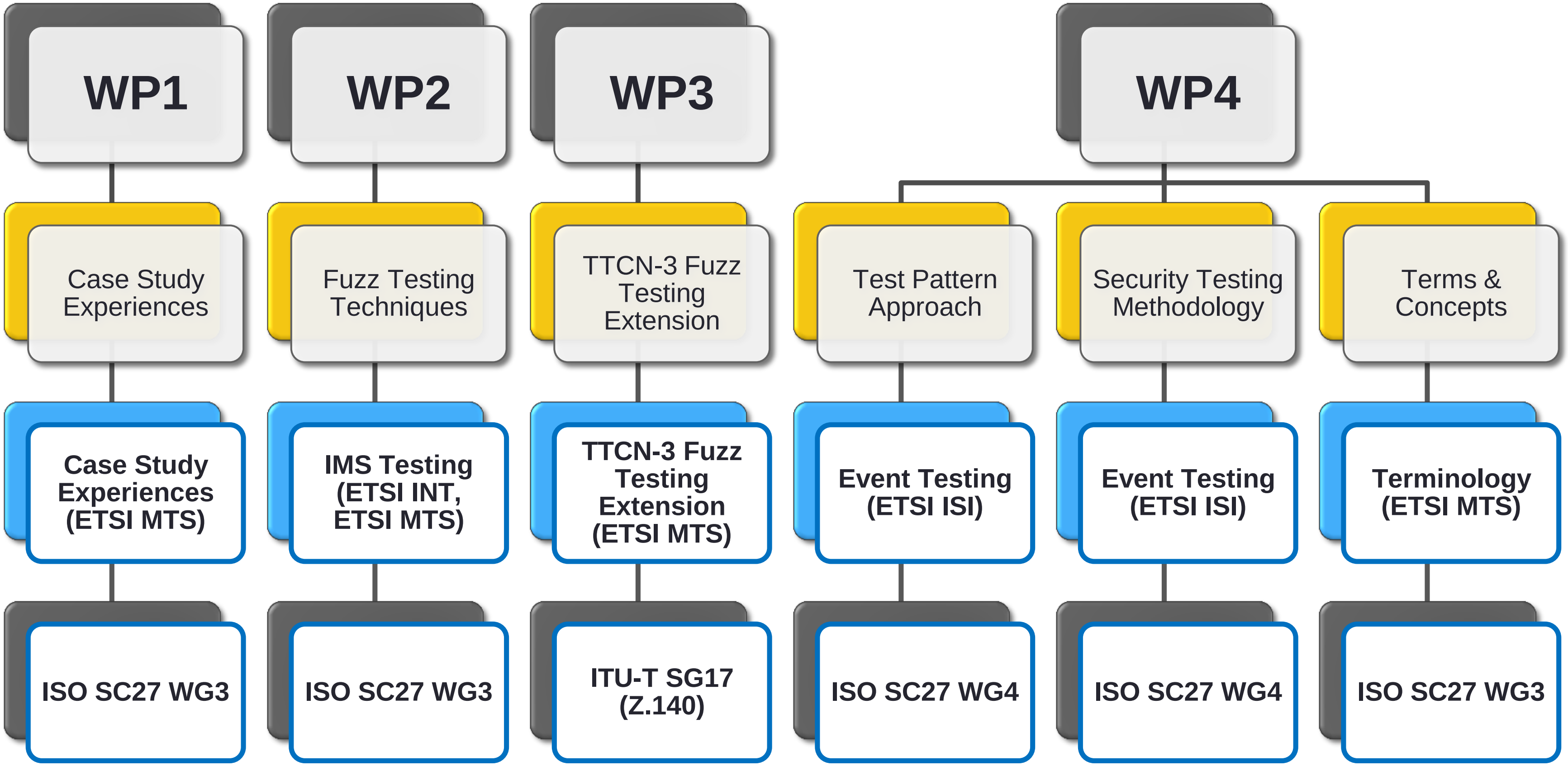
- Smartcards

Industrial Automation

- Industrial Automation Case Study from Codenomicon, Metso Automation, OUSPG, VTT



Results in Standardization



WP1

Case Study Experiences

Case Study Experiences (ETSI MTS)

ISO SC27 WG3

WP2

Fuzz Testing Techniques

IMS Testing (ETSI INT, ETSI MTS)

ISO SC27 WG3

WP3

TTCN-3 Fuzz Testing Extension

TTCN-3 Fuzz Testing Extension (ETSI MTS)

ITU-T SG17 (Z.140)

WP4

Test Pattern Approach

Event Testing (ETSI ISI)

ISO SC27 WG4

Security Testing Methodology

Event Testing (ETSI ISI)

ISO SC27 WG4

Terms & Concepts

Terminology (ETSI MTS)

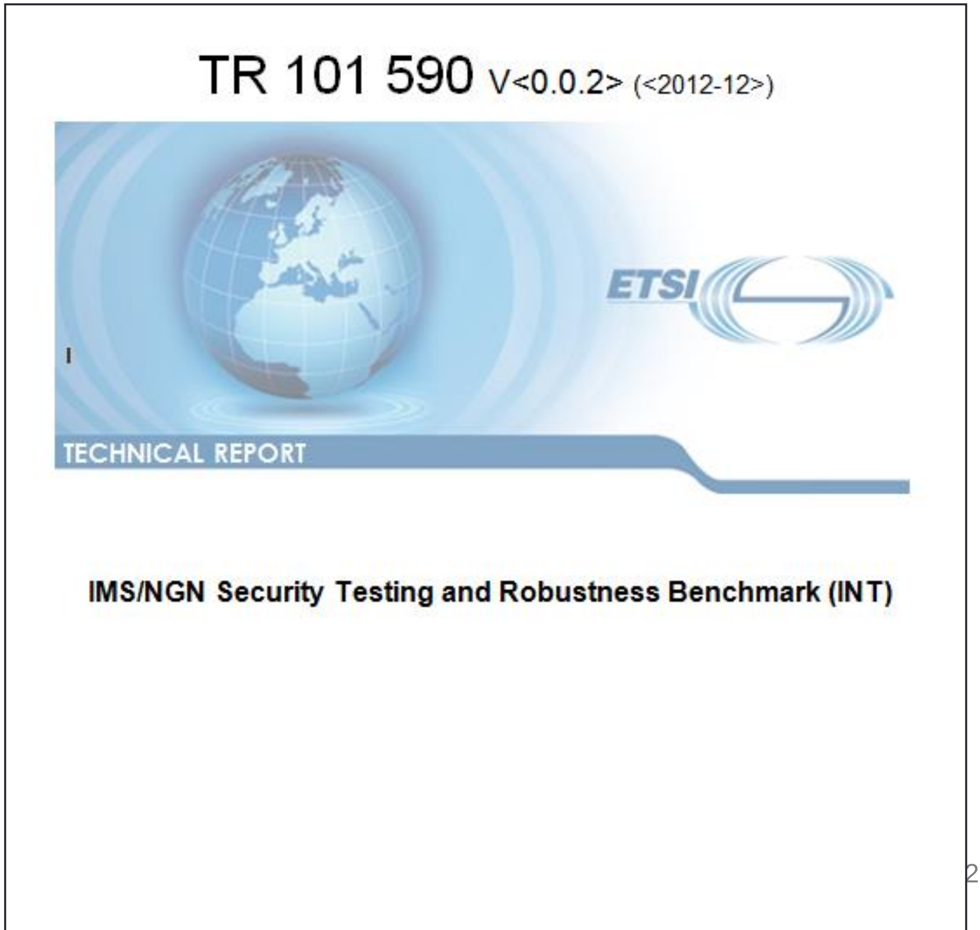
ISO SC27 WG3

ETSI INT

- **Technical Committee INT: Draft on Robustness testing in IMS (incl. Model-based and Mutation-based fuzzing)**

- Final draft Document has been approved as:

  TR 101 590 IMS/NGN Security Testing and Robustness Benchmark



TR 101 590 V<0.0.2> (<2012-12>)

TECHNICAL REPORT

IMS/NGN Security Testing and Robustness Benchmark (INT)

Summary

- Industry relevant subject
- Innovative approaches & methodology
- Effective tool solutions in industrial products
- Integration strategies for methods and tools
- Cross-country and cross-case study cooperation
- Experience reports on the case studies
- Standardization work

→ **DIAMONDS puts ground to make differences in security testing for the European industry!**

DIAMONDS

... in the sun

Thank you for your attention ! Questions ?

Prof. Dr.-Ing. Ina Schieferdecker
+49 (30) 3463-7241
ina.schieferdecker@fokus.fraunhofer.de

Axel Rennoch
+49 (30) 3463-7344
axel.rennoch@fokus.fraunhofer.de

FOKUS
Fraunhofer Institute for Open
Communication Systems FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

Tel: +49 (30) 34 63 – 7000
Fax: +49 (30) 34 63 – 8000

Web: www.fokus.fraunhofer.de
www.itea2-diamonds.org