

The logo for Nokia Siemens Networks, featuring the company name in a bold, black, sans-serif font. Below the text is a stylized graphic consisting of a series of parallel lines that form a wavy, ribbon-like shape. The lines are colored in a gradient from purple on the left to yellow on the right.

**Nokia Siemens
Networks**

The background of the slide features abstract, flowing shapes in shades of purple, orange, and yellow, creating a dynamic and modern aesthetic.

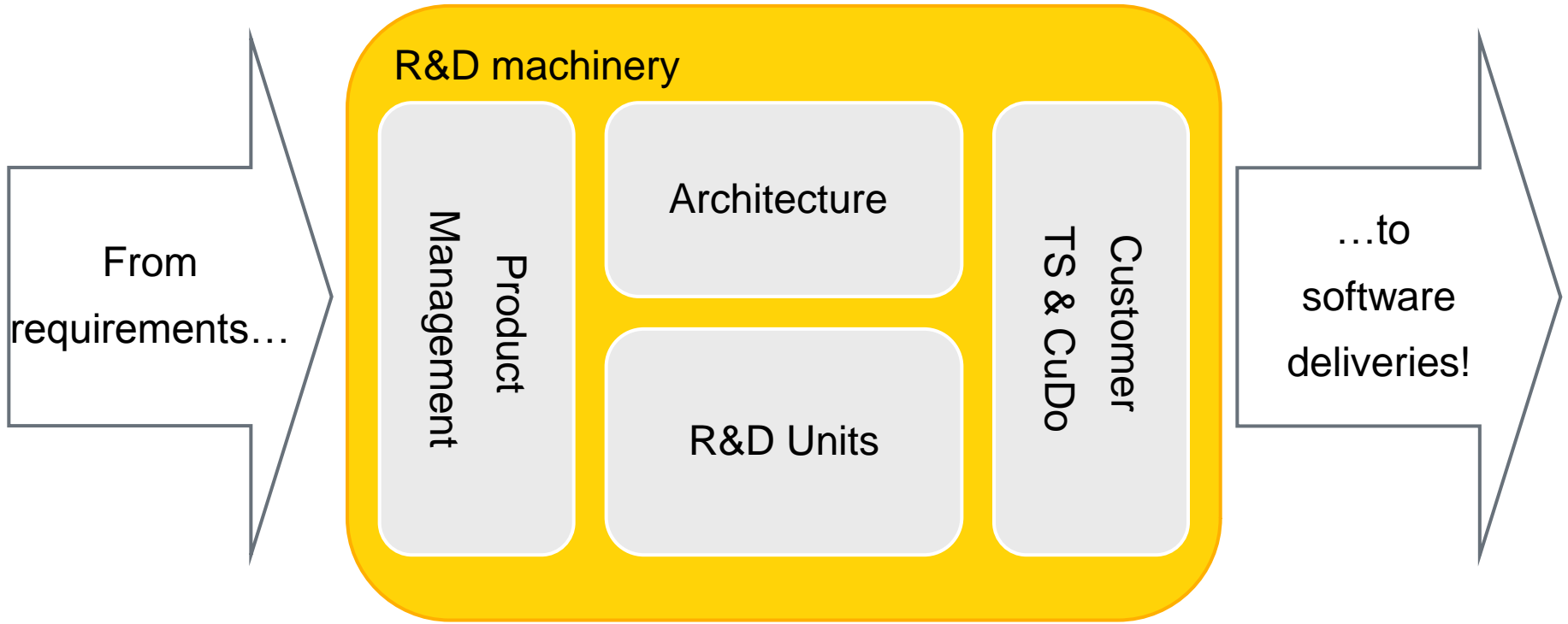
Model Based Testing deployment in Telecommunication project

Szilárd Széll – Business Improvement Manager

Gábor Federics – GSM-R Test Manager

MBTUC11 – Berlin, October 18-20, 2011

NSN Budapest - R&D center production chain



1.400.000.000 subscriber
connected worldwide

Challenges in Testing

- Huge amount of existing (legacy) functionality
 - More regression testing need
- Shorter release cycle
 - Less time for testing
- More customization
 - More specific function for smaller targets
- Economy
 - Pressure on cost effectiveness
- Quality goals
 - No quality sacrifice

Speed up testing

Test Automation

So make it faster, let's automate

1. Automate Test Execution

- Make executable test case scripts
- Schedule test run
- Collect result



2. Automate Test result analysis

- Compare test outcome with expected result
- Report test result
- Store/Archive result



3. Automate Test Design

- Test design specification
- Selecting best test techniques
- Find optimal coverage
- Document test cases



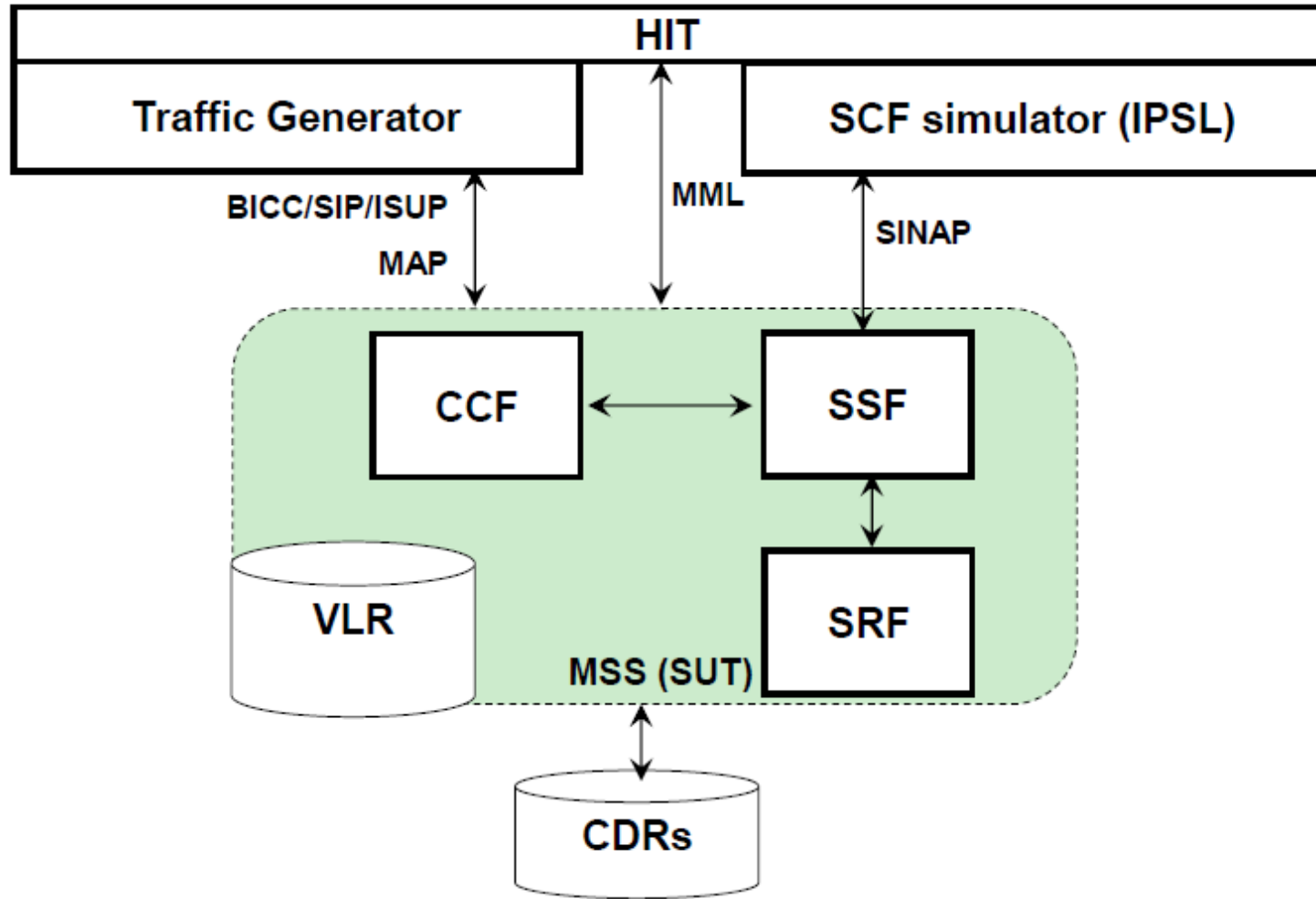
History of MBT in NSN Budapest

- First study on MBT in MSS during D-MINT project - 2008
 - Due to the existing partnership solution provided by Conformiq is used
- Initial presentation of MBT and pilot decision - 2009
- Pilot and Business case creation - 2010
 - Phase 1: clean the pipe - feasibility
 - Phase 2: ROI calculation - payback
- Decision and deployment in GSM-R - 2011

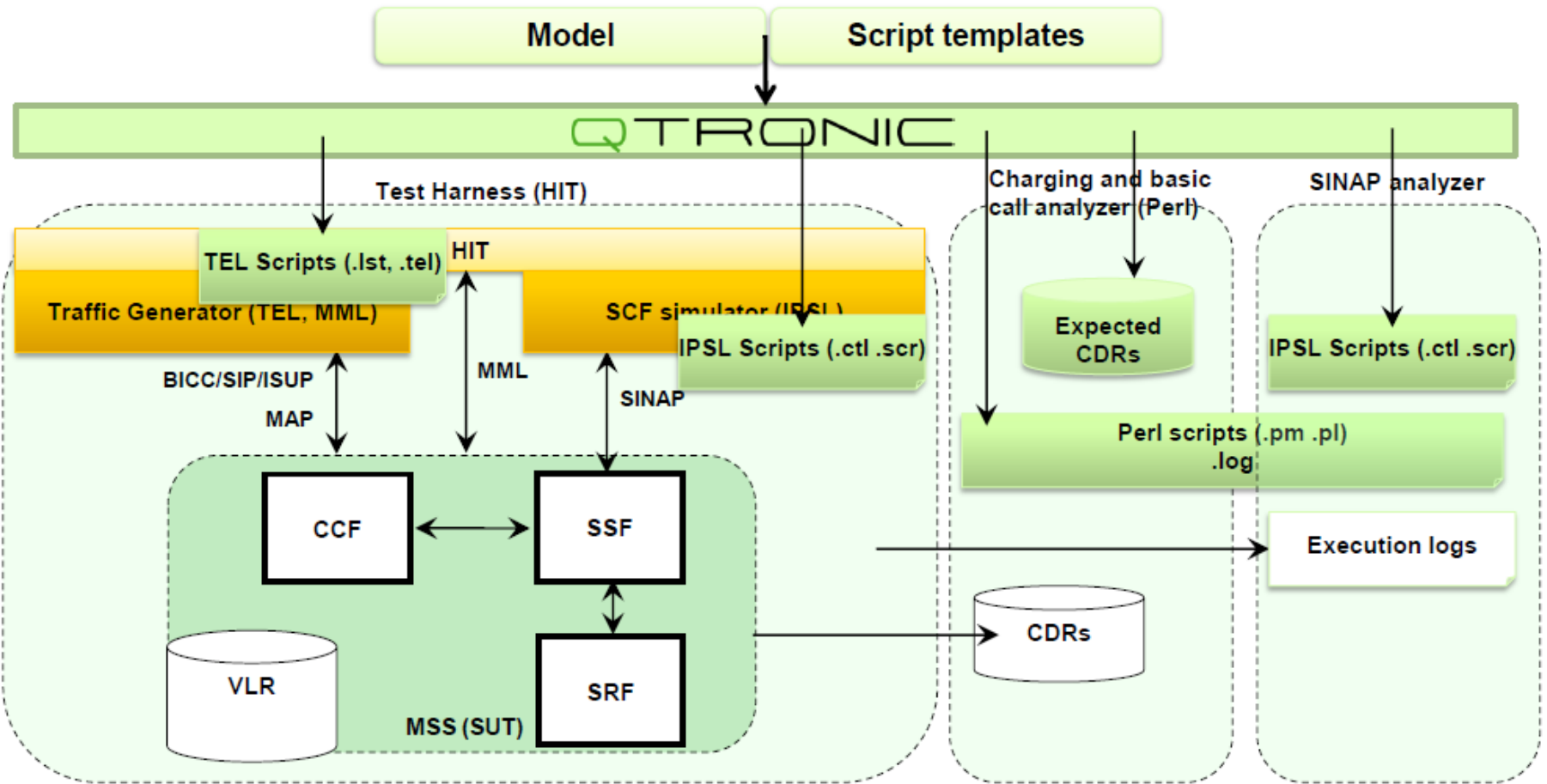
Pilot phase 1 - Project Scope and Goals

- Goals of the project
 - Introduce the concept of Automated Test Design to Intelligent Network Scrum team
 - Demonstrate that Conformiq Designer suite can be integrated with existing test harness in short term (3 weeks)
 - Create reusable assets of Automated Test Design
- What was the scope of the project
 - SUT: Release 4 Open Mobile Softswitch (MSS)
 - Testing objective: SINAP Charging
 - SUT accessed through Man Machine Interface (MML) using HIT test script language, and SINAP interfaces (IPSL protocol test tool)

Test architecture view of the scope of Phase 1



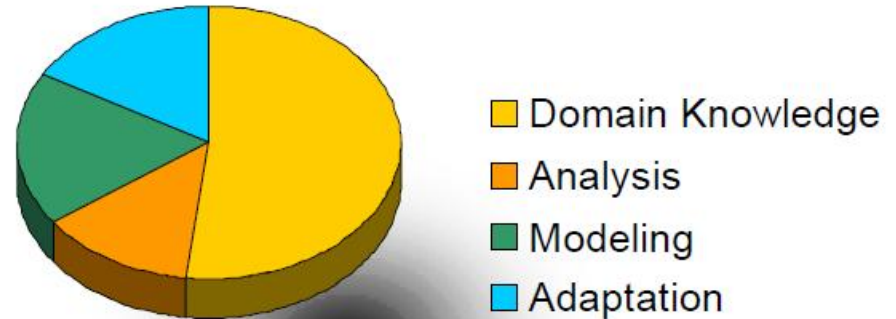
Test Architecture with ATD



Results from Pilot

During the pilot we reached

- Resources spent
 - Obtain domain knowledge 128 h
 - Work with model 45 h
 - Analyze system + logs 32h
 - Backend work 41 h
- MBT Training: 7 Certified Conformiq Technology Associate
- Conformiq Designer is integrated with existing Test Automation framework in 3 weeks
- Reusable components ready
 - call state machine
 - template based reusable backend for different test tools (HIT, IPSL, CHA analyzer, Basic MSS Analyzer)



Phase 1 conclusion

- Integrating Conformiq Designer into our Test Harness was done with reasonable effort
- **MBT enforces understanding (thereby indirectly enforcing cooperation between teams) and documenting correct system behavior in form of models**
- Due to the complex domain MBT with proper model architecture is very likely to **increase test design efficiency due to possibility for incremental changes.**
- In Automated Test Design both test cases and execution scripts are generated with guaranteed traceability mapping at once. This avoids multi-stage human error

GO for phase 2

Phase 2 – additional goals

- By comparison to previously available manually generated test suites (or appropriate estimates), assess what gains Automated Test Design can provide in the areas of
 - Productivity
 - test case quality
 - requirement coverage
 - requirements traceability
 - and ease of maintenance

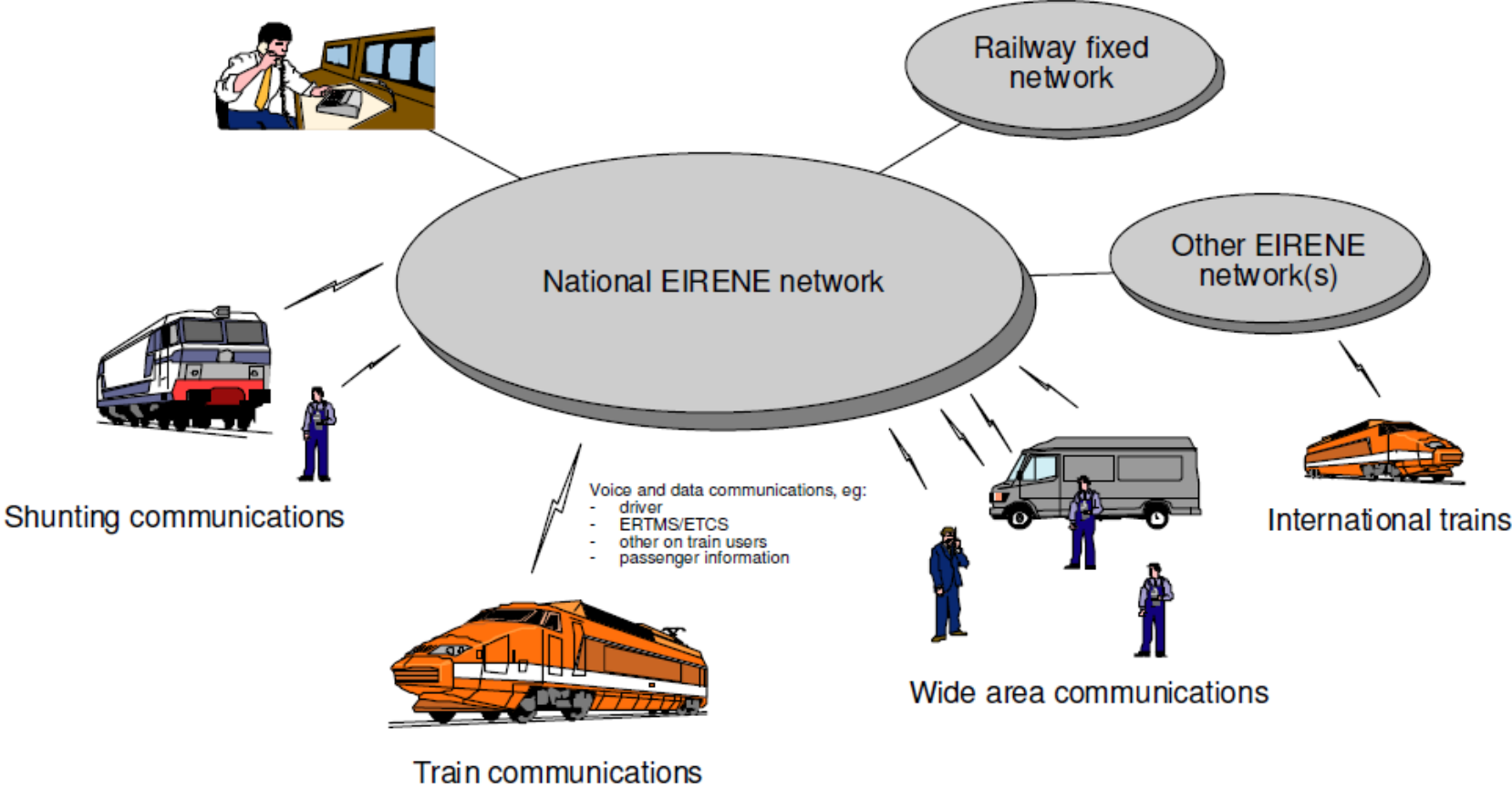
Phase 2 result – ROI and Business Case

- Productivity
 - 100% Requirement coverage with 2/3 of original test cases (based on 7 requirements)
 - 20 test cases → 14 test cases
 - Less time spent
 - 650 → 550 hours: 15% improvement during initial model based testing (creating reusable asset)
 - 278 → 172 hours: 40% improvement during incremental add-on built on top of existing reusable assets
 - Better functionality coverage
 - 3 minor bugs found during pilot in an already tested feature (in live usage already)
 - Documentation inconsistencies
 - Revealed challenges due to scattered documentation (common in Telco industry)

- Positive Business Case

**break even during the 2nd year
after ramp-up**

GSM-R in general



Deployment in GSM-R

- Goals of the deployment
 - Confirm MBT pilot results in real project environment in GSM-R program
 - Confirm business case
 - To show that technology can be adapted by MSS developers
 - Define Mode of Operation
 - Engineer feedback
- Scope of deployment
 - SUT: Open Mobile Softswitch for Railways
 - SUT accessed through Man Machine Interface (MML) using HIT test script language, H.248 and A interfaces (TTCN3 protocol test tool)
 - Test levels of deployment
 - Component Testing
 - Functional Testing

Deployment results

Work hour analysis

Reference	Component Test Ratio		
Reference program estimation (manual)	36,8%	for reference	Legacy reference data
Reference program reported (manual)	44,3%		
GSM-R Component1 historical expert opinion (manual)	40%		
GSM-R Component1 phase1 reported (MBT)	30%	Component 1	GSM-R
GSM-R Component1 phase2 reported (MBT)	28,3%		
GSM-R Component2 estimated (manual)	44,1%	Component 2	
GSM-R Component2 reported (MBT)	34,6%		

Deployment results

Code coverage analysis

Area	Coverage Points	Code Coverage		
Component1 (manual)	410	Phase1	80.7%	GSM-R
Component1 (MBT)	410	Phase1	87%	
	609	Phase2	92%	
Component2 (MBT)	119		97%	
Component2 Database (MBT)	862		93%	
Legacy example1 (manual)	20445		83.9%	Legacy reference data
Legacy example2 (manual)	20360		74.5%	

Deployment results

Fault findings in Component1

- Component1 testing done both manually and model based
- Fault findings:

Component1	Faults found	Code coverage
Manual testing	12	80.7%
Model based testing	12 + 9	87%

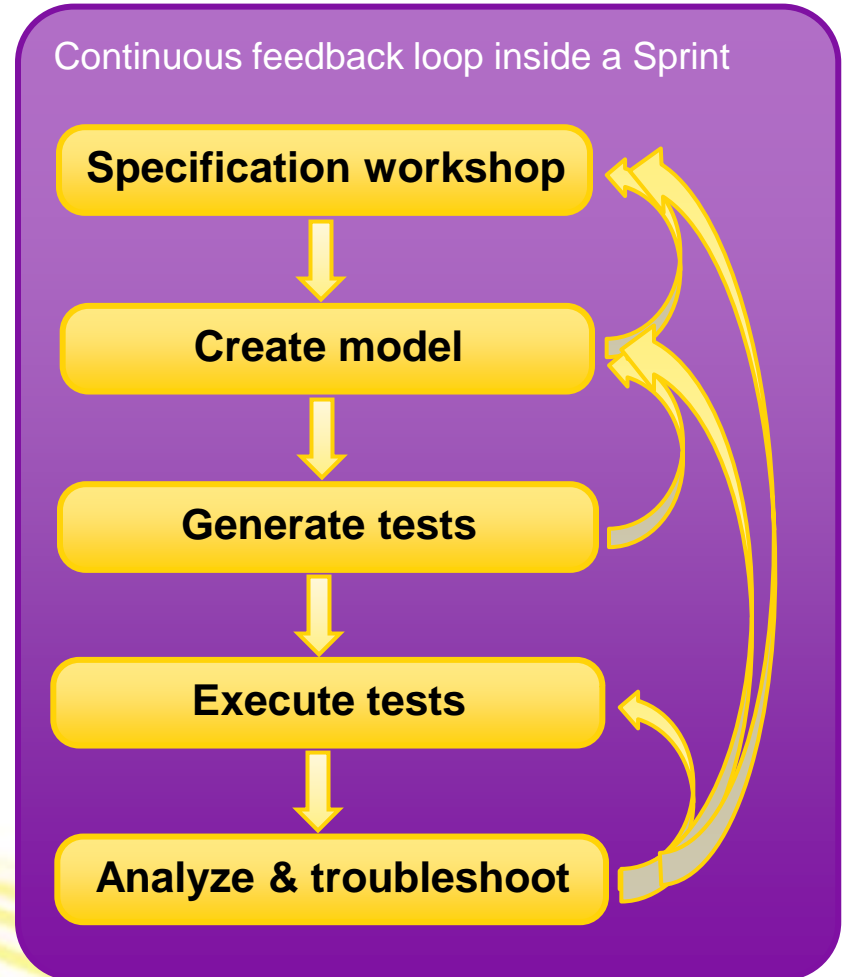
- According to root cause analysis results, 3 faults would be probably caught only during very late test phase

Deployment results Summary

- Very good experience with database modeling
 - 20-30% gain in speed
 - Outstanding level of code coverage reached
- Results with MML modeling dissatisfactory
 - MML: commands in hierarchical structure used for system configuration via telnet interface
- Good experience with state-chart based component modeling
 - 25% gain in speed
 - Higher code coverage compared to manual test planning
 - Addition faults found after manual component test ready
- Positive results expected from Functional Testing

Integration into Agile work practices

- Continuous communication sessions established in all areas
 - 2 review levels must be ensured for feedback
- Model review
 - With modelers, specifiers, lead designers
- TC review
 - With all the engineers in the effected domain
- New roles in Scrum team
 - Modeler
 - Backend scripiter
- Automatic documentation into QC



Engineer satisfaction survey

- „A nicely built model can be understandable for anyone. There is no need to check several other documents, only the model”
- „This method is requires a systematic approach, so it decreases the negative effect of human factor”
- „Modeling is fast and comfortable after having experience. Longer TC generation times can be obstacle”
- „During the model creating very deep knowledge can be gathered about the system. This is challenging and motivating”
- „It is also motivating that serious code defects were found with MBT (after manual testing was done)”

Lessons learned

- MBT technology can be adopted with significant improvement on complex projects
- Step-by-step Pilot and deployment approach is needed with
 - Clear goals
 - Go/No-Go decision criteria
 - Business sponsorship
- Strict reporting practices needed for Business case validation
 - Fault reporting
 - Effort reporting for different activities
- Technology support is needed for proper ramp-up
 - On-site / remote support as requested
 - Attending an all reviews
 - To review also scripting backend
- Train not only modelers
 - Specifiers (one per area)
 - Lead designers (one per area)
 - Test architects (one per area)



Questions?

