

# Multi-site testing strategy using TTCN-3

Vikas Pratap Singh, David Alvarez Barragan, Vipul kumar  
Andy Rauland, Dr. Andreas Ulrich, Zhang Chao

### Topics:

- (A) Project Description.
- (B) Test Strategy and Test Process
- (C) Multi-site principles and strategy
- (D) Experiences

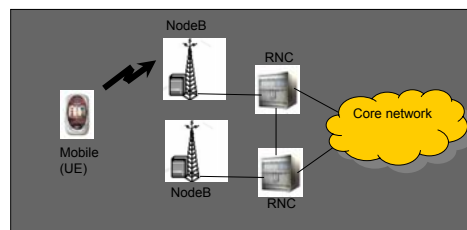


## Project Description (1/2)

### The RNC Product

#### UMTS

- Standards body : 3GPP
- Broadband mobile access for voice (TDM) and data (statistical multiplexing)
- Guaranteed QoS, can use IP for signaling



#### RNC: Radio Network Controller

- Owns and controls the radio resources (NodeB)
- Uses specialized procedures (e.g. handover control, admission control, power control...)



## Project Description (2/2)

### Testing RNC product

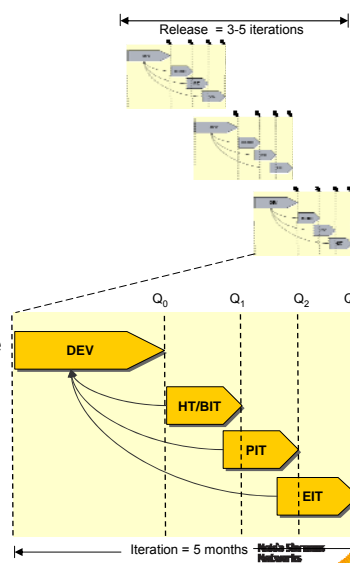
- 206 SYs multi-site test project. (448 SYs overall project)
- Test Strategy defined for complete project.
- Test System based on TTCN-3 technology. (reused & extended from previous project).
- Project schedules driven by test efforts.
- ~ 50% project efforts on test area.
- Test Data collection (TTCN-3 template writing) requires intensive domain know-how.
  - Typically, messages consist of some hundreds of information elements, on several levels of nesting structures. Since this being the first version of the product, we didn't have a history available.



## Test Strategy: (1/3)

### Overview

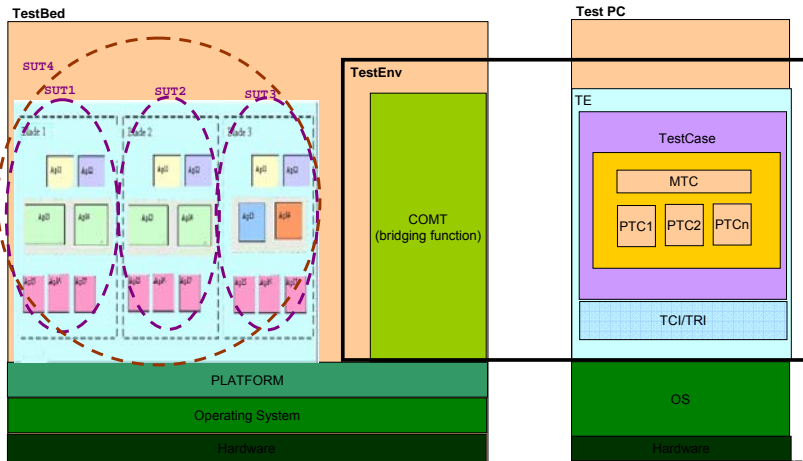
- A project release contains a set of features offered to the customer. Typically it is realized by means 3-5 iterations, that undergo every test phase in a timeframe of 5 months.
- Three test phases executed in sequence: Host / Blade Test, Plane Test, Entity Testing . (V model based)
- Each phase increases the scope of testing with respect to the previous phase , allowing a progressive integration into a complete tested product.
- Bug reports to development teams from each test phase
- A single Test Environment shall support all phases of testing aiming for reusability of TTCN-3 code. (templates, middleware and TCs)
- A common TTCN-3 middleware shall be the basis for all TTCN-3 test cases. (Abstraction layer, hiding low level details of the TCs.)



## Test Strategy: (2/3)

Definition of SuT and TestSystem boundaries for each test phase.

For each test phase identified in the Test Strategy, the SuT boundaries towards TestSystem are determined.



5

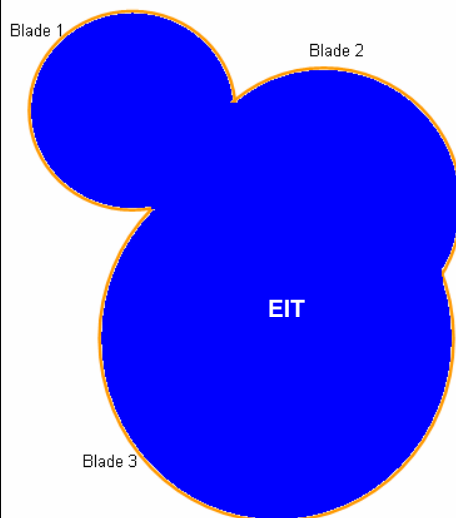
© Nokia Siemens Networks

TTCN-3 user's conference - 2007



## Test Strategy (3/3)

Mapping of Features to Test Phases.



**HT:**

- Testing of features on host machine.
- Focus on features affecting only 1-2 blades.

**BIT:**

- Testing of features on the ATCA blade. (real HW).
- Focus on features with strong PF and HW dependencies (Eg. Bring-up, Restart, Switchover, Contexts, ...)

**PIT:**

- Testing of features on real HW (all blades).
- Focus on features affecting 2-3 blades.

**EIT:**

- Overall testing of the product as an entity.
- Focus on customer scenarios first)

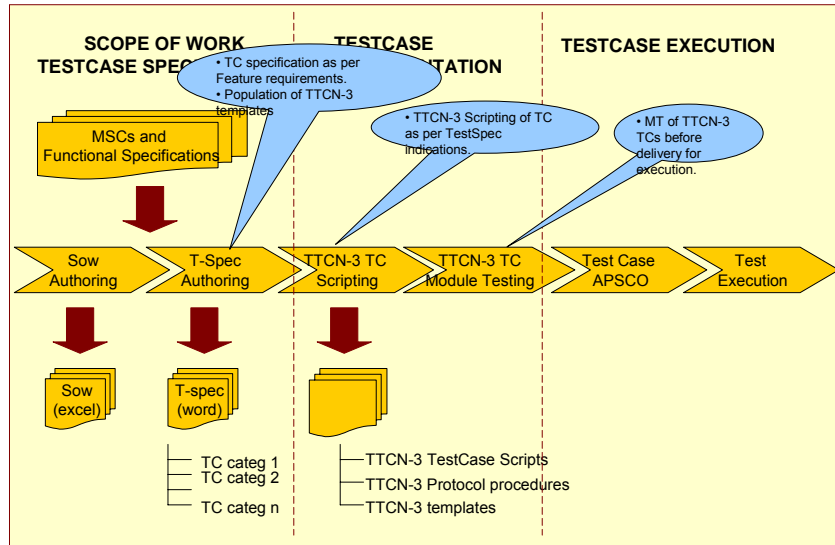
6

© Nokia Siemens Networks

TTCN-3 user's conference - 2007



## Test Process definition From SoW until TTCN-3 TC execution



7

© Nokia Siemens Networks

TTCN-3 user's conference - 2007

## The TTCN-3 based Test System

A long path ahead ...

- We have formed a centralized Test Environment team that designed and implemented a TTCN-3 based test system to assist the testing process.
- TestEnv has been successfully used over 3 years by ~70 testers on a big multi-site project for ~600 TTCN-3 TCs.

**Jan' 07** An improved test efficiency (16%) was achieved on the TestEnv and test processes.

In **Apr' 06** Architecture-Test team was founded. (Improvements phase)

**Nov' 05 - Apr' 06** → TestEnv was progressively extended along with the iterative cycles of the Product development. Initial Performance problems were identified and solved for the most complex SuTs.

In **Nov'05** the first release of TestEnv was provided to the testers for their official testing.

Until **Oct' 05**, the TestEnv analysis and implementation was done in parallel to product analysis. Prototype testing was used for the validation of every enhancement/extension in the TestEnv.

**Sept' 04:** ad-hoc TestEnv was developed for testing of early prototypes. (Started reusing previous project)

## Multi-site principles and strategy (1/5)

### Project sites (Test Involvement)

#### Cost effectiveness :

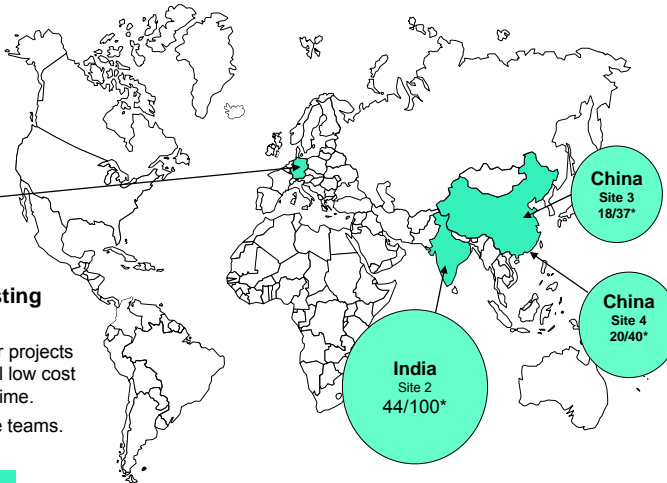
- Low Cost Target: 80%
- Siemens in-house development & test, following well established processes for SDLC.

**Germany:**  
Site 1  
20/53\*

#### Expansion driven by existing domain expertise:

- Core teams from other similar projects were made available at several low cost locations at different points of time.
- Site growth sustained by core teams.

\*no. of testers/total workforce



## Multi-site principles and strategy (2/5)

### On what basis is the work split done and how it works?

- **Domain expertise:** Different sites have specialized expertise in different areas. Based on this expertise, relevant parts of product are developed or tested there.
- **Self-contained clearly defined work-packages:** Complete responsibility given to sites. (blade-wise and test-phase wise).
- **Testers shall sit close to developers** (as far as possible).
- **Centralized Project Leadership** supported by **distributed local Project Management**.
- **Complexity of the SUT:** If the SUT is very complex then its testing can be distributed at different sites. Requires several reporting interfaces to PL
- **Common Dev teams** : for all blades report directly to PL.
- **Centralized common tasks.** (APSCO production, TestSystem, ...).
- **Multi-site virtual teams:** Architecture, A-test, Diagnosis, ...Reporting done directly to the PL.
- Take advantage of **Time-zone difference**.

## Multi-site principles and strategy (3/5)

### Reporting and support tooling.

- Weekly reporting to Project Leader in PSM (Project Steering Meeting)
- Test Progress Reporting:
  - TTCN-3 TC Specification: % completion. (incl. TTCN-3 template population)
  - TTCN-3 TC scripting: % completion.
  - TTCN-3 TC Module Testing: % completion.
  - TTCN-3 TC execution:
    - # TCs executed: planned vs passed.
    - # FRs raised: weekly delta. (closed vs. raised).
    - Schedule Variance and Effort Variance.
- TCM: Test Case Management. (Storage and scheduling of Test Cases)
- FRM: Fault Report Management. (Omni-tracker). FR handling.



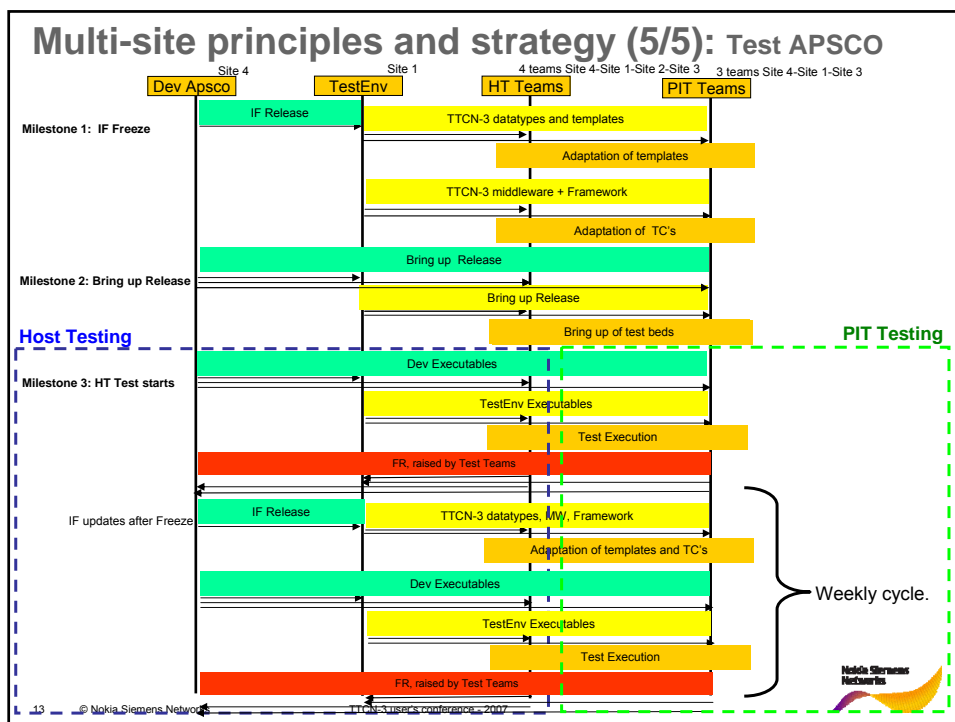
## Multi-site principles and strategy (4/5)

### Synchronization of test teams

- APSCO (Application Production System Co-ordination) is the process that rules the production of deliverables across different locations.
- All TTCN-3 code and APSCO deliverables are stored under configuration management (version control) system –Clearcase-, replicated strategically across sites.
- Types of APSCO releases used in the project:
  - Production APSCO releases — contains the software load to be tested.
  - TestEnv APSCO releases — contains all tooling required for testing.
    - TTCN-3 Data Types
    - Codec
    - Codec libraries
    - Framework
    - Tools
    - Default TTCN-3 templates
    - TTCN-3 middleware
  - Test APSCO releases — contains the Test Artifacts (TCs) required for the testing. (allow reusability across sites, teams and test phases).
    - TTCN-3 scripts
    - TTCN-3 Templates
    - Test executables (in the form of jar files)

SWload Label + TestEnv Label + TestCase Label = Unambiguous Testing





## Experiences(1/2)

- We used TTCN-3 in two large multi-site test projects for over 3 years.
- We faced difficulties:
  - Initial buggy TTCN-3 Test System and Test Cases.
  - Significant initial investment is required for starting a big TTCN-3 test project.
    - Construction of a suited Test Environment. (Incl. TCI, TRI extensions, ...)
    - Automation (additional needs).
    - TTCN-3 training (low market availability of TTCN-3 experts, despite of the increasing demand)
  - Big efforts on coordination and communication across sites. This is typical from multi-site projects and neutral from TTCN-3 perspective. (Good processes can help)
- It is worth investing on a TTCN-3 middleware architecture to base your TCs.
- Reusability of TTCN-3 code shall be considered –at different levels- from early stages of the project. (see table below)

	Goal	Achieved	Comments
TTCN-3 test system	Full	Full	
TTCN-3 middleware	Full	Full	
TTCN-3 templates	Full	High: across Test Phases Low: across HT Test Teams	Could be improved by means of: - Usage of refactoring techniques. - Better alignment/cooperation between test teams.
TTCN-3 test cases (code)	High	Low	Investigate better reusability of TTCN-3 code (besides TTCN-3 middleware)

## Experiences(2/2)

- TTCN-3 proves to be an excellent technology for testing in the telecom domain.
- Standard bodies and tool vendors need to continuously strive to meet industry needs.
  - multiple proprietary tools being developed in the SW industry to overcome limitations.
- Recommendations:
  - Test your TTCN-3 Test System. (TEVP= Test Environment Verification Process).
  - Test your TTCN-3 Test Cases (TC\_MT: Automated Module Testing of TTCN-3 TCs).
  - Settle a continuous improvement program for TTCN-3 (A-test).



***Thank you for your co-operation***