Best Practices of Agile Testing in Cloud Software Products

Presented by:

NOKIA
Agenda

• Introduction: Testing Telco Cloud Software Products
• Testing Levels in Nokia Cloud Software Products
• Best Practices of Agile Testing
• Tools for Automated testing of Nokia Cloud Software
• Q&A
Introduction
Testing Telco Cloud Software Products

Network Service/Slice

- VNF
  - App
  - App
- VNF
  - Kubernetes ++
  - VM
- CNF
  - Applications
  - Microservices
  - Containers

Cloud(s)
- OpenStack
- VMWare
- AWS
- Azure
- BareMetal

Testing of Trustworthy Systems
#UCAAT
Testing Levels in Nokia Cloud Products

- **UT/MT** – tests classes, methods, logic and algorithms of each unit.
- **SCT (Software Component Test)** – tests functional behavior of each microservice or Virtual Network Function Component (VNFC).
- **PIT (Plane Integration Test)** – tests black-box functional behavior of integrated VNFCs (subsystem) in simulated VNF/CNF environment.
- **Smoke Test** – tests released builds for core functionalities of subsystem in real VNF/CNF environment.
- **ET (Entity Test)** – tests new functionalities of subsystem in mixed real and simulated VNF/CNF environment.
Left-shift Testing and Automation
Best Practices of Agile Testing

Automated and manual tests
- Hardware environment / Real BTS

Fully Automated tests
- Software environment

- ROBOT / Python / Selenium
- TTCN3
- Viper/Python & TTCN3
- Gtest / MOCK
- I&V / SyVe
- SMOKE / Entity tests (ET)
- Plane Integration Tests (PIT)
  (Multi Component Tests MCT)
- Software Component Tests (SCT)
- Unit Tests (UT) / Module Tests (MT)

- System Functional Specification
- Entity Functional Specification
- Plane Functional Specification
- Component Functional Specification
- SW Design

- Hardware environment / Real BTS
- Software environment
Feature Test Strategy
Best Practices of Agile Testing

Feature Development Process

FS
- Feature screening
- Planning
  - Customer Requirements
  - Feasibility study
  - Materials Refinement
  - Providing EE
  - Feature Planning

CFAM
- Common Feature Analysis Module
- Describe feature from top to bottom
  - System Functional Spec
  - Entity Functional Spec
  - Design Impact Analysis
  - Plane/Component Functional Spec

Dev.
- Development & testing
  - Code delivery + UT/MT/SCT
  - Plane Level Tests
  - Entity Level Tests
  - System Level Tests

Maint.
- Maintenance

Feature Owner Team
FOT meeting
Feature Test Strategy

Testing of Trustworthy Systems
Optimizing Estimations and Planning
Best Practices of Agile Testing

• Detailed and Accurate estimations
  • Using template for estimation analysis
  • Checking estimation gaps from previous features to improve

• Split Feature into Sub-features
  • Split Tasks into Sub-tasks
• Feature Owner Team (FOT) creates Feature Test Strategy to optimize feature delivery
  • Left-shift testing practices
  • Optimize full coverage of requirements into different testing levels

• Virtual Feature Owner Team (FOT) regular meeting to mitigate risks
  • Clarification of Acceptance criteria
  • Removal of blocking points
  • Keeping track of feature deliverables
Faster Feedback Loop
Best Practices of Agile Testing

DevOps in Telco Cloud

WHAT

- Agile development
- Continuous integration
- Continuous delivery
- Continuous deployment

HOW

- Microservices & containers
- Automated workflows
- Continuous SW delivery & verification
- Digital delivery: Core AppStore

DevOps
Tools for Automated Testing of Nokia Cloud Software

Static TTCN-3 Test Generator

Most of the features require different configurations and base Start-up on Air with particular number of Radio Units, Radio Units Types and number of Cells.

We develop a simple tool which can generate testcases based on user inputs, and automatically create ttcn3 tests and configuration files to be executed.
Coding guidelines

It defines clear rules on the proper usage of TTCN3 and guidelines for testing Cloud software products with examples straight from the repository. The documentation is written in reStructuredText format stored in GIT.

TTCN3 Linter

This is command-line-based tool written in C++. It allows static code analysis based on coding guidelines rules. It is executed as part of test code review pipeline and lists the lines that violate the rules.
Tools for Automated Testing of Nokia Cloud Software

**Rain**
This is web-based tool. There are two main aspect of Rain:

- **Manual log analysis** part of Rain is an easy-to-use tool which merges, parses and presents logs in a readable form.

- **Automatic log analysis** where we can deploy our own plugin which automatically analyses logs. The plugin display results of analysis appropriately.

**Logan**
This is TTCN3 test log analyzer that performs analysis of the logs of the executed tests. It provide the ability to generate reports about performance, durations, sequences, events, and message frequencies.
Tools for Automated Testing of Nokia Cloud Software

Stability Monitoring (A-10) Tool

Deploys data analytics algorithm on finding patterns over the failed test cases from CI/CD regression.

Tool can crawl the Jenkins jobs and extract failed tests from each build in a given range, or for a set of predefined builds. It processes and filters data to map tests to the common points of failure.
Tracker is a tool enabling easy tracking of integrations and introduces multiple tools for CI.

It can monitor the:

- Software integrations (Timeline, status of SW releases)
- KPI (Branches availability, SW releases, Integration Tests & feedback loop statistics)
- Branches (List of branches, branch status)
Thank you!

Gemmilyn Chu
gemmilyn.chu@nokia.com

Daniel Ardelean
Daniel.ardelean@nokia.com

Mariusz Lont
mariusz.lont@nokia.com

Piotr Czermak
piotr.czermak@nokia.com