9th UCAAT User Conference on Advanced Automated Testing

Best Practices of Agile Testing in Cloud Software Products

Presented by:

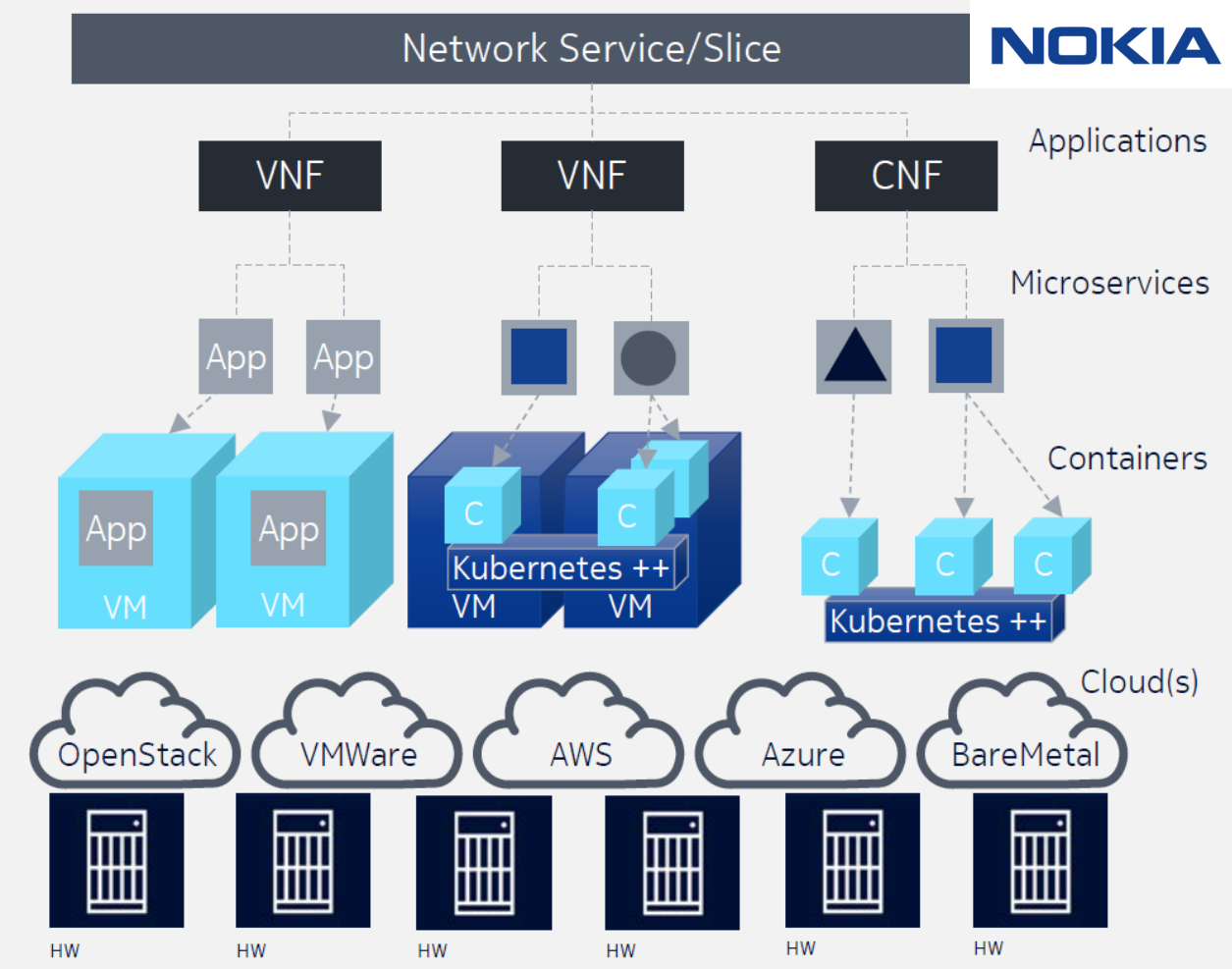
NOKIA

15/09/2022

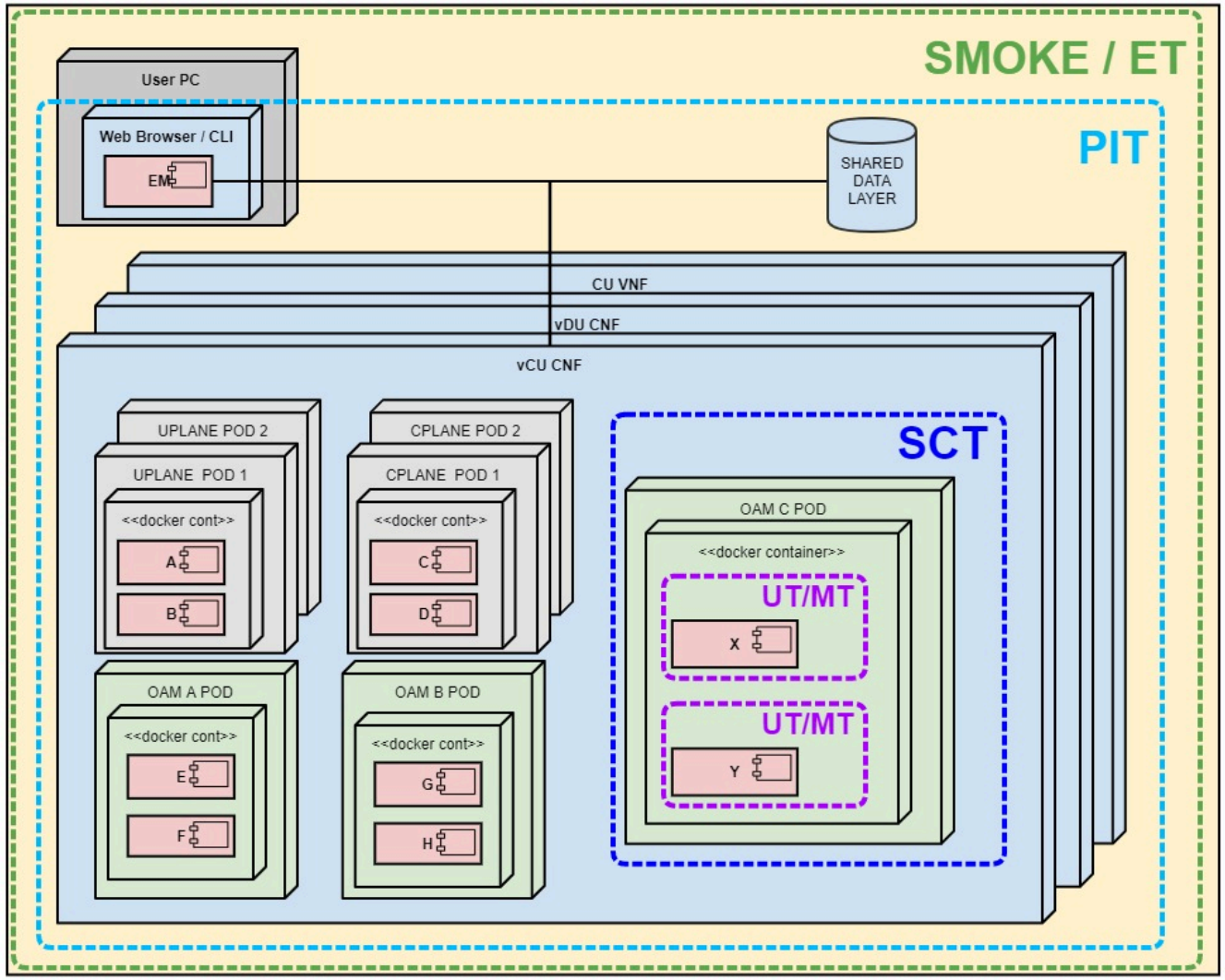
Agenda

- 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



Testing Levels in Nokia Cloud Products

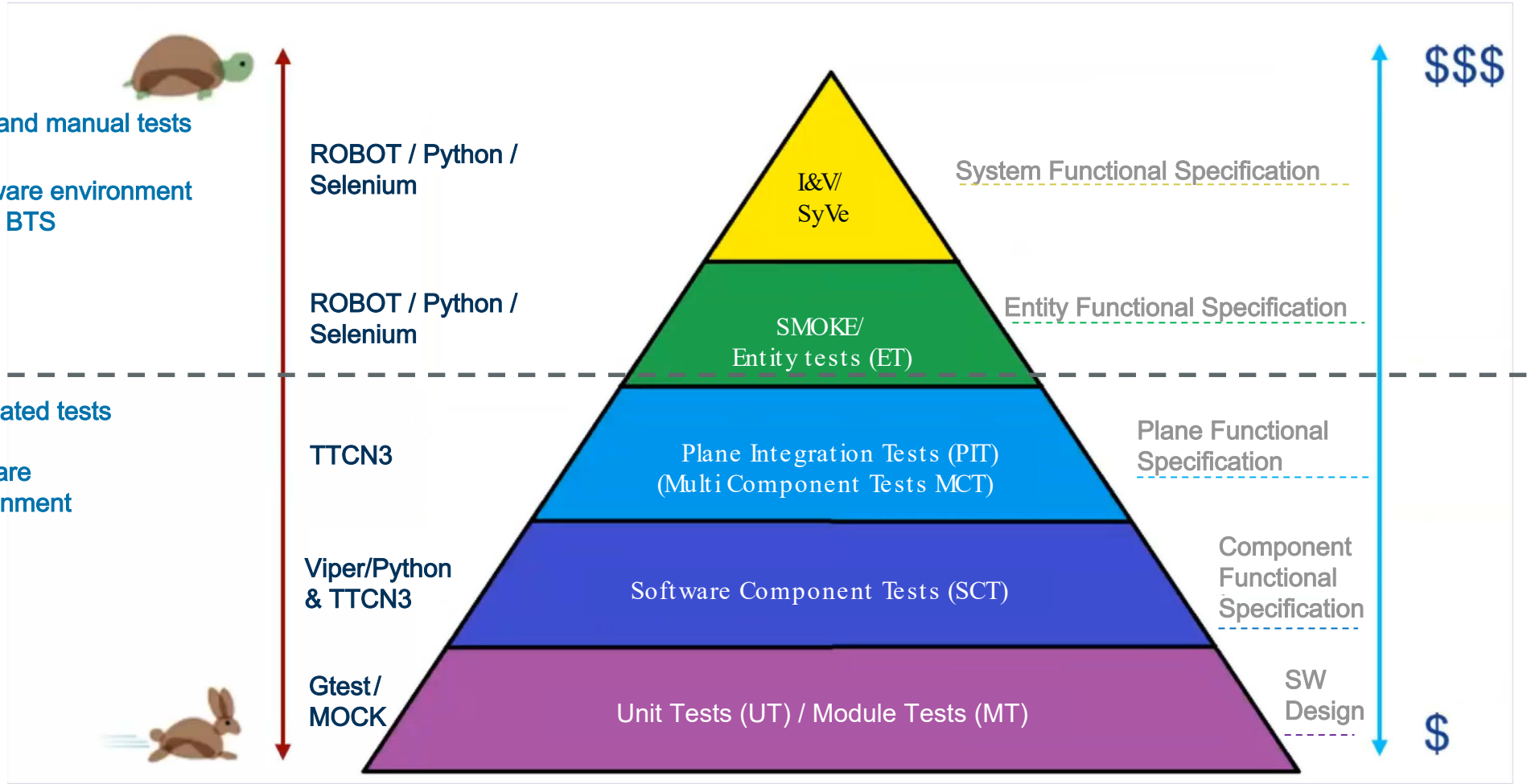


- **UT/MT** – tests classes, methods , logic and algorithms of each unit.
- **SCT (Software Component Test)** – tests functional behavior of each microservice or Virtual Network Function Component (VNFC).
- **PIT (Plane Integration Test)** – tests black-box functional behavior of integrated VNFCs (subsystem) 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.

NOKIA

Left-shift Testing and Automation

Best Practices of Agile Testing



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

NOKIA

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

Strategizing and synchronizing Execution
Best Practices of Agile Testing

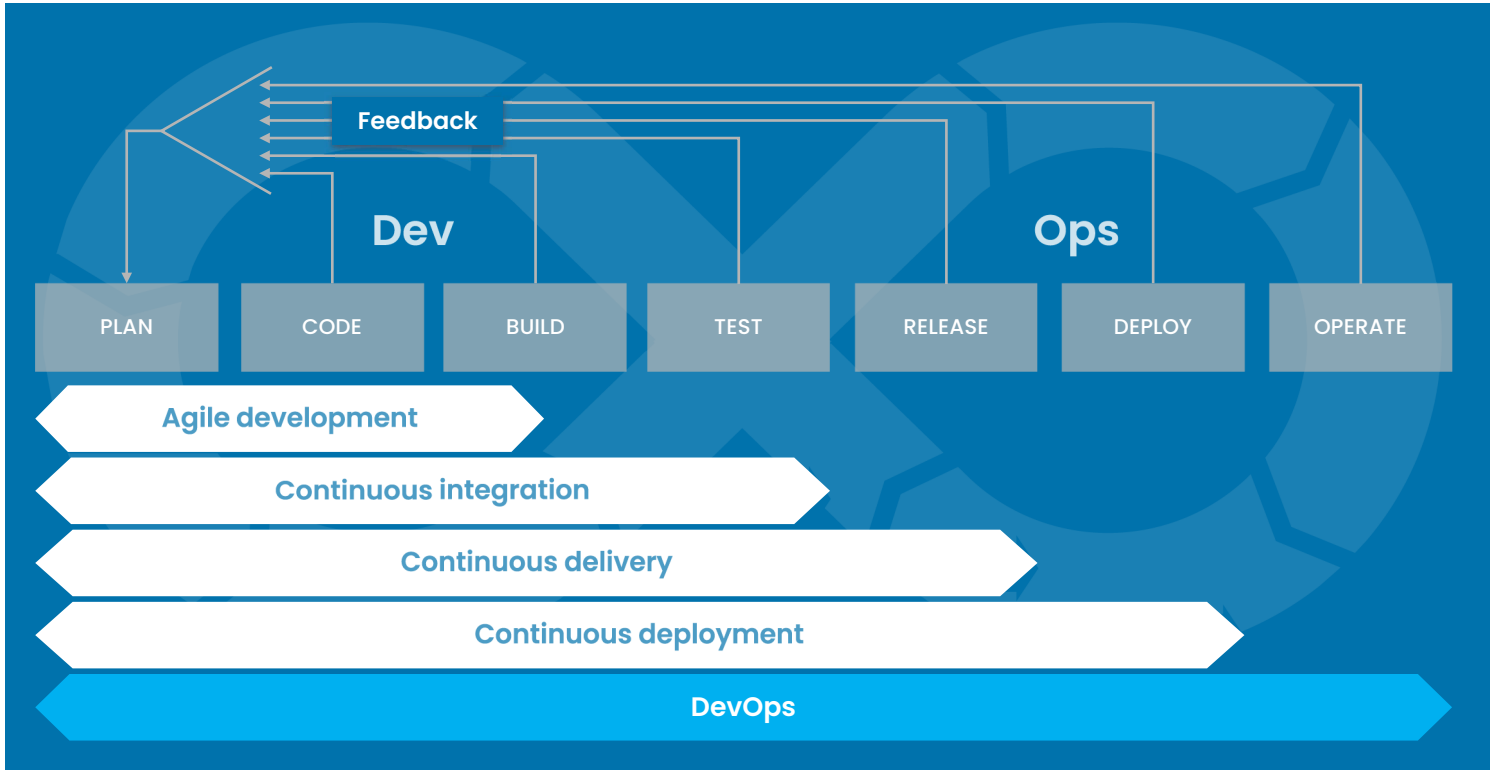
- Feature Owner Team (FOT) creates Feature Test Strategy to optimize feature delivery
  - Left-shift testing practices
  - Optimize full coverage of requirements into different testing levels

- 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



HOW

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

NOKIA

Tools for Automated Testing of Nokia Cloud Software

Static TTCN-3 Test Generator

Most of the features require different configurations and base Start -up to onAir 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.

```
module Startup_N_Demo2 {
    //Common used imports, generated automatically
    private import from MVDuBtsomMtc all;
    private import from MVDuBtsomSystem all;
    private import from StartupVDuFunctions all;
    private import from StartupFunctions {function f_vDuStartupToOnAir};
    private import from CommonFunctions {function f_setupTestcaseFlow};
    private import from EnvironmentConfiguration {modulepar OMK3_ROOT; type FlexiConfiguration};
    private import from MLogger {const PRINT_DEBUG};
    private import from VDU_N_5AHCA all;

    const float TC_TIME := 300.0; //this is default, generated automatically.

    testcase test() runs on CVDuBtsomMtc system CVDuBtsomSystem {
        var FlexiConfiguration v_configuration:= VDU_N_5AHCA.f_createConfiguration();
        StartupFunctions.f_vDuStartupToOnAir(
            va_configuration := v_configuration,
            va_scfVariant := SCF_VARIANT
        );

        setverdict(pass);
        MVDuBtsomMtc.f_teardown();
    }

    control {
        f_setupTestcaseFlow({···
        });

        execute(test(), TC_TIME);
    }
}
```

```
/*
* This is a automatically generated file
* Configuration is by default with FDD
* You can generate code for N radios, given as keyboard input
* Serial Number is hardcoded and static for now
*/

module VDU_N_5AHCA {
    import from ConfigurationCommonFunctions    all;
    import from EnvironmentConfiguration        all;

    function f_createConfiguration() return FlexiConfiguration {
        var FlexiConfiguration v_ret := ConfigurationCommonFunctions.f_makeEmptyConfiguration();
        v_ret.name := "VDU_N_5AHCA";
        v_ret.eCpriRadios[0] := ConfigurationCommonFunctions.f_createECpriRadio("AHCA", "L1174907823", 0);
        v_ret.eCpriRadios[1] := ConfigurationCommonFunctions.f_createECpriRadio("AHCA", "L1174907823", 1);
        v_ret.eCpriRadios[2] := ConfigurationCommonFunctions.f_createECpriRadio("AHCA", "L1174907823", 2);
        v_ret.eCpriRadios[3] := ConfigurationCommonFunctions.f_createECpriRadio("AHCA", "L1174907823", 3);
        v_ret.eCpriRadios[4] := ConfigurationCommonFunctions.f_createECpriRadio("AHCA", "L1174907823", 4);

        return v_ret;
    }
}
```

NOKIA

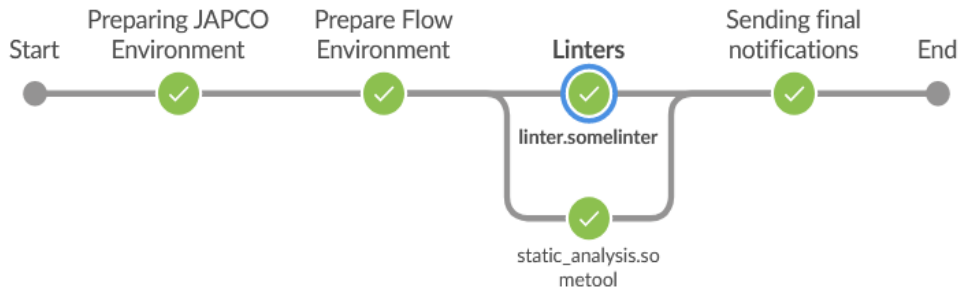
Tools for Automated Testing of Nokia Cloud Software

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.

```
root@                          ttcn3Linter
/                                                    :FormalValuePar:function parameter name should start with prefix va_:HIGH

/                                                    .FunctionDef:function name should follow camelCase format and start with prefix f_:HIGH
```



Coding guidelines

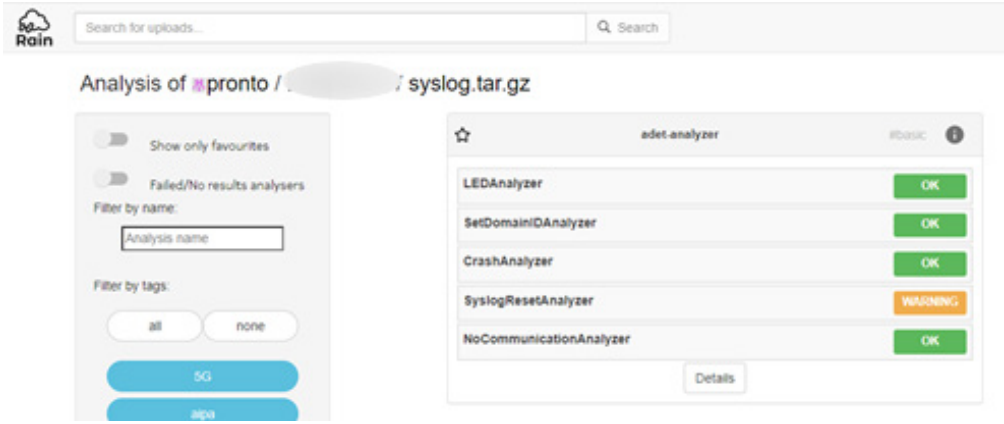
It defines clear rules on the proper usage of TTCN3 and guidelines for testing Cloud software products with examples straight from the repository. The documentation is written in reStructuredText format stored in GIT.

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.

```
root@i...: logan k3log     path_to_log                          performance messages
Time format: [H]H:MM:SS[.UUUUUU]
Percent Sum Average Median nCalls <Min,Max> Name
26.4154% 0:34:23.308649 0:02:08.956790 0:02:08.979582 16 <0:02:08.650618,0:02:09.159795> Message 1
11.8096% 0:15:22.444281 0:01:42.493809 0:01:30.011393 9 <0:00:04.022531,0:02:49.879414> Message 2
11.5585% 0:15:02.832183 0:02:08.976026 0:02:00.172095 7 <0:00:00.039938,0:05:00.798319> ...
6.9492% 0:09:02.803739 0:01:48.560747 0:01:45.373469 5 <0:00:01.259754,0:03:45.423288> ...
6.9328% 0:09:01.520992 0:02:15.380248 0:02:15.377737 4 <0:00:45.347751,0:03:45.417767> Message n
```

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 filter data to map tests to the common points of failure.

Common Issues	TC	No. f…	Builds	Is St…
[redacted]	[redacted]	1	16933	N
		1	16947	N
		1	16960	N
		1	16960	N
		1	16960	N
		2	16960, 1…	N
		1	16960	N
		1	16960	N
		1	16960	N
		1	16960	N
		2	16961, 1…	N
		1	16961	N
		1	16961	N
		1	16962	N
		1	16962	N
		1	16963	N
		1	16963	N
		1	16963	N
		1	16963	N
		1	16963	N
[redacted]	[redacted]	1	16934	N
		1	16936	N
		1	16944	N
[redacted]	[redacted]	1	16935	N
		2	16945, 1…	N
		1	16959	N
[redacted]	[redacted]	1	16937	N
[redacted]	[redacted]	1	16944	N
		1	16945	N

NOKIA

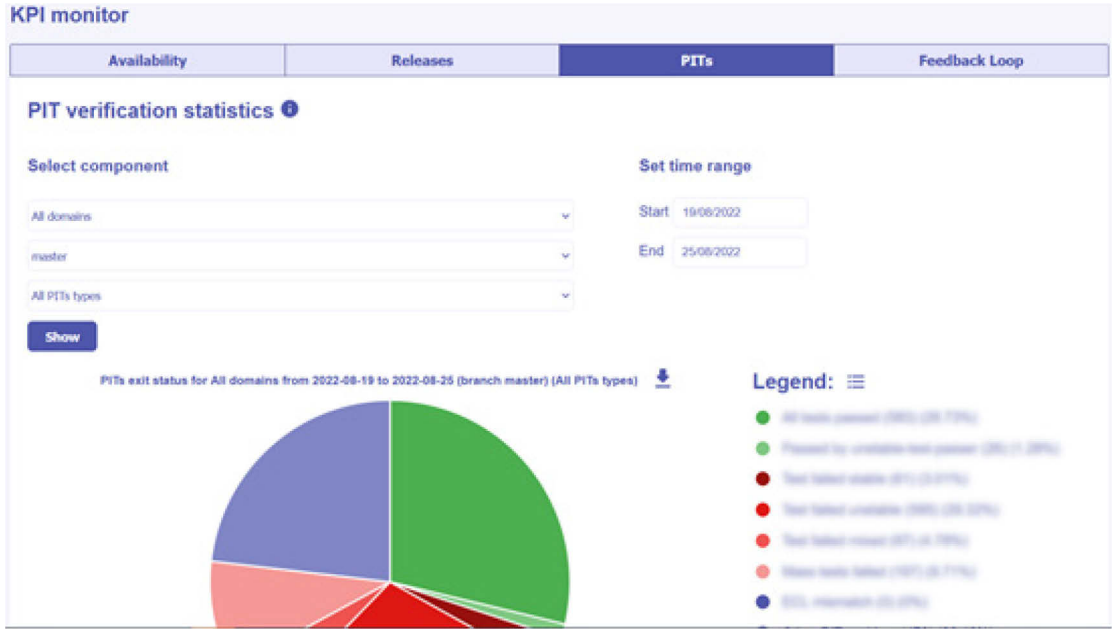
Tools for Automated Testing of Nokia Cloud Software

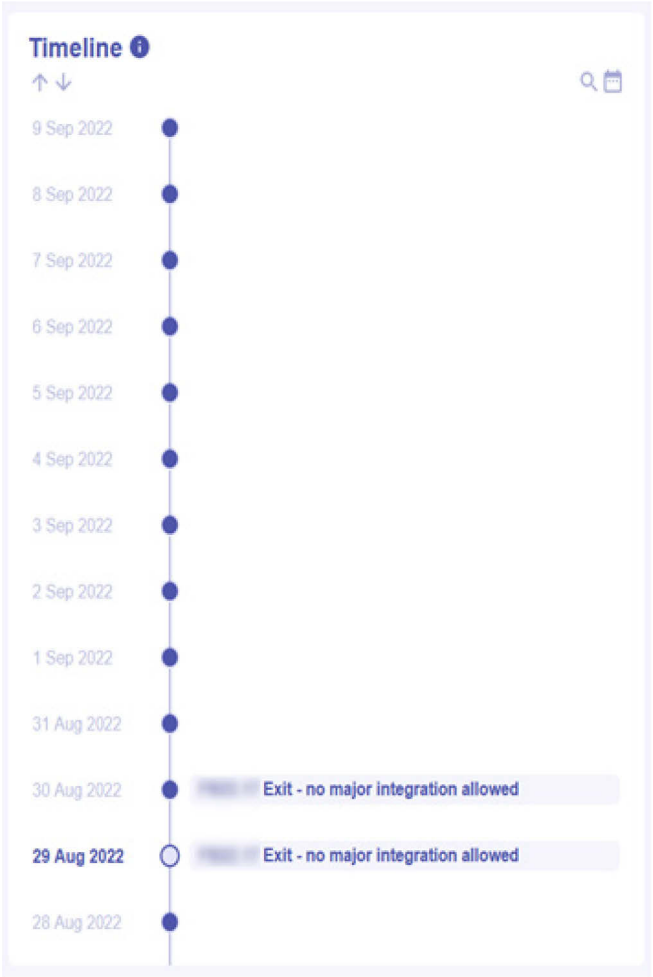
Tracker

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)

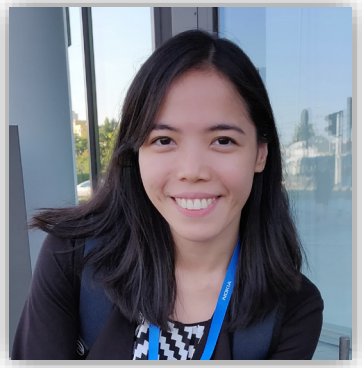




NOKIA

NOKIA

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