ONAP
OPEN NETWORK AUTOMATION PLATFORM

Discussion on the test strategy evolution in ONAP

K.Kuzmicki, M.Richomme (Integration)

Prague, 15th of January 2020

Discussion on the test strategy evolution in ONAP

Very ambitious tests had been announced in the past (S3P, chaos monkey). However health check still does not guarantee a good integration, verified end to end use cases are sometimes very big and hard to automate and/or to reproduce in different environments. It is today not trivial to know which components are really used/covered through existing use cases. Basic Integration tests in the target environment are missing to guarantee a good quality of the solution.

More basic tests should be provided by the components to offer a better coverage of their APIs, interfaces and features. CSIT tests are not enough as they are run on non representative target environment and robot healthcheck tests are sometimes to high level to re

Status

- Current healthcheck tests are not enough to be sure that the component is working
- CSIT tests are most of the time not run on target environment and look more to functional tests than to integration tests
- Stability and resiliency tests are limited
- Lots of use cases have been validated since Amsterdam version, some have been partially automated, few fully automated (integrated in CI)
- There is no load tests
- There is security tests
- There is no Tests KPI
- There is notion of test coverage – some components are never tested from an end to end perspective

Organization of the tests (Frankfurt)

category

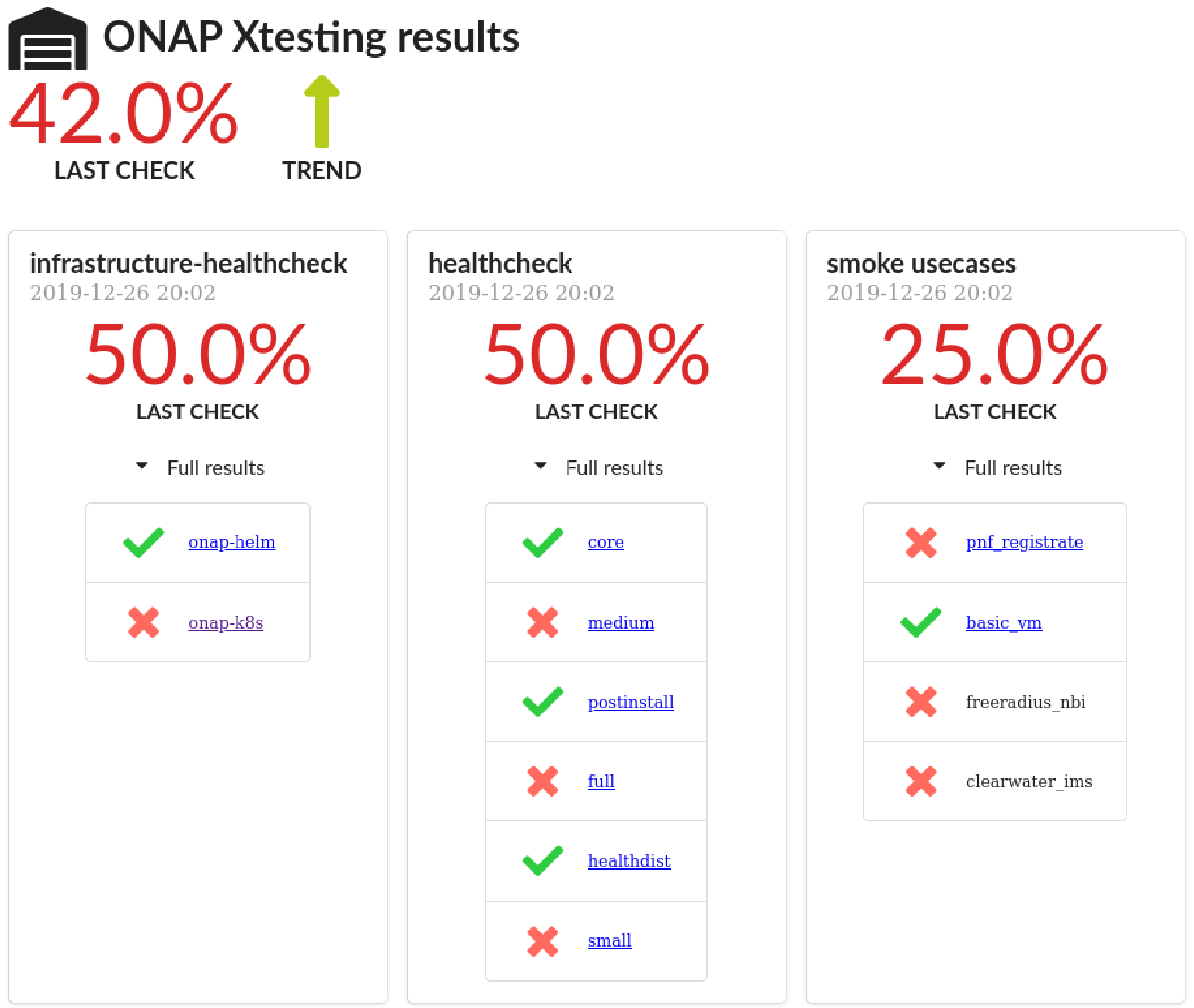
infrastructure-healthcheck	healthcheck	smoke-usecases	candidate-usecases	benchmarking	security
Infra-health** • onap-k8s • onap-helm	**Health*** • Core • Small • Medium • Full • Postinstall • Init • init_customer • Healthdist	**Smoke-robot*** • vfwcl • pnf_registrate • vCPE • vFWk8s • CDS • ... **Smoke-onaptest** • basic_vm • freeradius_nbi • clearwater_vim • basic_multicloud • basic_cds	**Candidate** • 5G-OOF-SON • basic_CL • basic_policy	**Benchmarking** • Stability72hr	**Security** • Postinstall

Legend

Use case already exists and integrated
Use case already exists but integration in progress
Use case not created yet

* : Health and smoke-robot tests are launched as K8s jobs (with internal url)
** : Infra-health run kubectl commands

Current tests executed on Gating/Daily CI chains



ONAP Xtesting results

42.0% LAST CHECK

TREND ↑

infrastructure-healthcheck
2019-12-26 20:02
50.0%
LAST CHECK
Full results

✓	onap-helm
✗	onap-k8s

healthcheck
2019-12-26 20:02
50.0%
LAST CHECK
Full results

✓	core
✗	medium
✓	postinstall
✗	full
✓	healthdist
✗	small

smoke usecases
2019-12-26 20:02
25.0%
LAST CHECK
Full results

✗	pnf_registrate
✓	basic_vm
✗	freeradius_nbi
✗	clearwater_ims

Open Discussion on the evolution of the test strategy

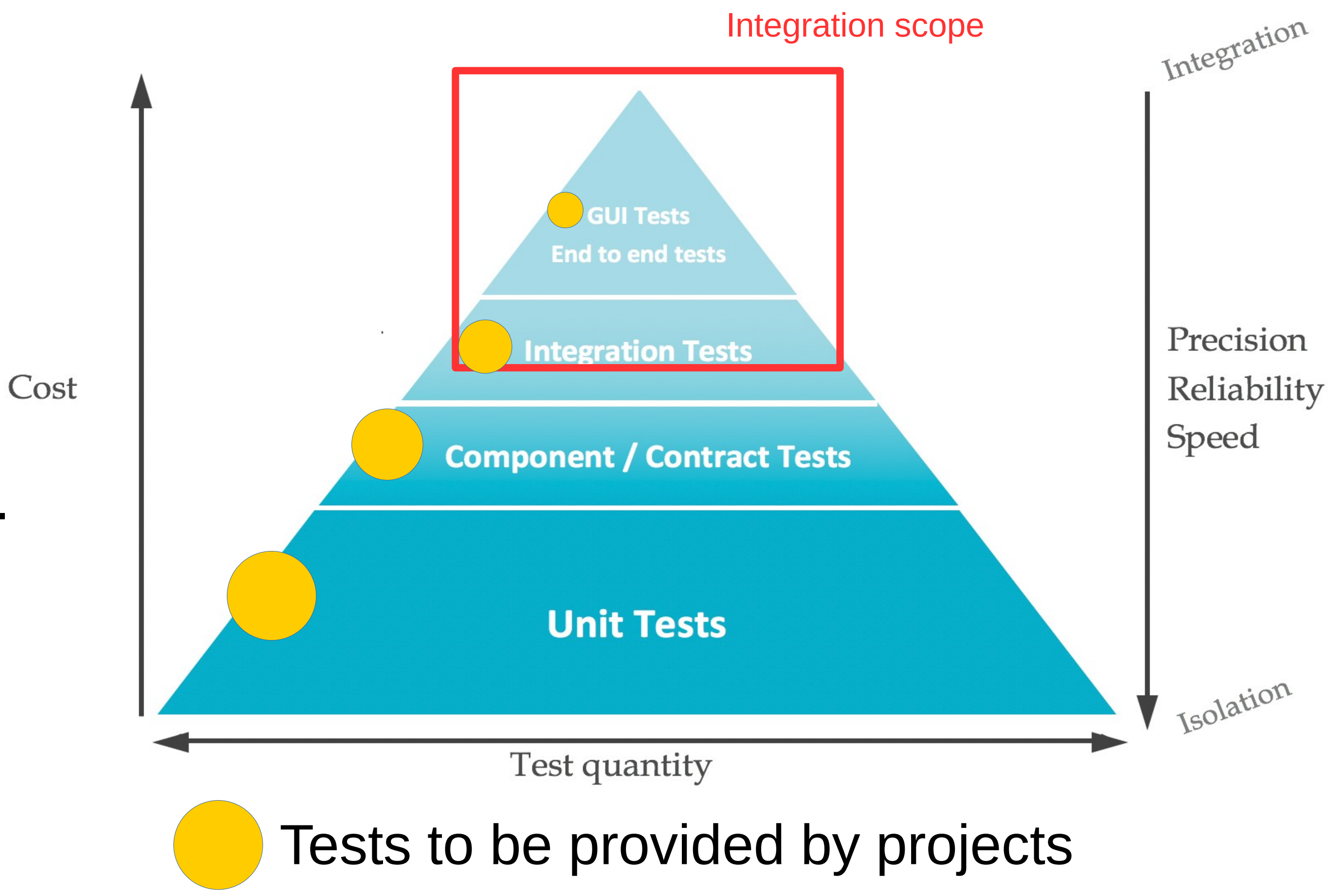
Role of Integration

- Today we validate ONAP mainly through use cases that induce complexity that has nothing to do with ONAP : We do not really test ONAP, we test use case X, Y, Z using ONAP and deduce that ONAP is working as expected
- Wiki use case list is not adapted
  - Declarative results in a wiki (may work on lab A but not on lab B)
  - Test plans not clear : use JIRA test plan for each use case ?
  - All use cases not maintained from one version to another but still in the wiki from a version to another (maybe misleading)
  - Documentation not always good enough (and not maintained) to replay the use case in other env : use cases sometimes seen as a quick win for a release but not seen as samples delivered as part of the release (no code quality verification on the test code)

Integration versus projects

Each project must develop
- unit tests
- Functional/Component tests
- API /contract tests (including bench)
- Integration tests
- GUI tests (if GUI)

Integration team must
- Run integration tests
- Develop specific integration tests (e.g. with infra)
- End-to-End tests
- Stability tests
- Resiliency tests
- Load tests



Integration versus projects

- Projects must provide better healthcheck tests
  - Not acceptable to have component not working with healthcheck PASS
  - 1 new feature => 1 new test
  - N pods => N+ tests
  - Healthcheck is not just a check that a web server is answering
- Projects not covered by tests should be considered as extensions and not be part of the official release (simplification + time saving): TSC decision ?
- When projects deal with GUI, they should provide a way to test it

Integration versus projects

- Pairwise testing
  - Not automated – sync time slot before a release to speed up troubleshooting
  - Responsability not fully clear / green/red light is declarative
  - Scenario not always clear
- API
  - Clear documentation of the API systematically (swagger like), version compatibility
  - Does each project perform benchmark on its APIs ? No figure found, no load on the API
- CSIT tests
  - Are functional tests most of the time (not integration tests)
  - Shall be moved to the projects for non target env (e.g. docker compose) ~ partially done csit repo used sometimes only to bootstrap but all the tet code is in teh projects. Improvement in El Alto (cleaning of non maintained jobs + Mail notification)
  - Should be done through Gating mechanism (for the components – but build chain to be understood for each component)

Integration versus use cases

- A use case may have different status
  - Candidate : documentation + manual tests => wiki, not part of the release (lower priority for Integration team)
  - Officially part of the release (documented, referenced,..): only it is fully integrated in the gates (CI Daily)

Integration versus use cases

- Keep it simple
  - More basic end to end tests are needed
    - Lots of candidate today..
      - GUI/Certificate validity/Basic Policy/CL/CDS/CDT/multicloud/..
    - Ensure components are really tested (needed before Gating integration)
    - Resources shall be planned by Projects + Integration
    - They must be integrated in CI

Integration versus use cases

- Complex use cases still needed
  - « Window » of ONAP
  - Shall be as automated as possible
  - Could be managed independently from the release cadence (tested on last stable rather than on master)

Integration new test domains

- Stability*
  - Today low visibility, run only once just before the release, scenario not fully clear
  - to be included in CI (weekly – today run at the very end of the validation)
- Resiliency* (weekly)
  - Today low visibility, run only once just before the release, scenario not fully clear
  - must be automated
  - random destruction of resources
  - // instantiations
- Load* (define KPIs..)
- Security
  - Sometimes done but output not clear/not properly shared
  - e.g. scan for post installation, check open ports, check docker vulnerabilities (harbor)...
- Operations
  - Version update
  - Password changes
  - Backup/restore
  - Monitoring/supervision

* : see presentation dedicated to stability/robustness/resiliency/load

ONAP

OPEN NETWORK AUTOMATION PLATFORM

Thank you!