ONAP

OPEN NETWORK AUTOMATION PLATFORM

Integration & Use Cases

March 2020

Integration & Use Cases

- Historical reminders
- Integration scope
- Integration and Use Cases
  - Resources
  - Support
  - Automation

Foreword

This slide deck reflects my vision based on my experience as PTL for the Frankfurt release.

It is a WIP deck...open to modification

If consensus is found, it could be transformed into an official documentation « Use Case guide »

The presenation is probably a bit late in Frankfurt Life Cycle but not real formal way was defined so far, this proposal aims to provide a framework for future releases

Historical reminders

Historical reminders

Integration project approved on june 2017*

Integration is responsible for ONAP cross-project system integration, CI/CD, and all related end-to-end release use cases testing with VNFs necessary for the successful delivery and industry adaption of the ONAP project as a whole.

4 PTLS since 2017 : H.Chen (Huawei), Yang Xu (huawei), Brian freeman (ATT), Morgan Richomme (Orange)

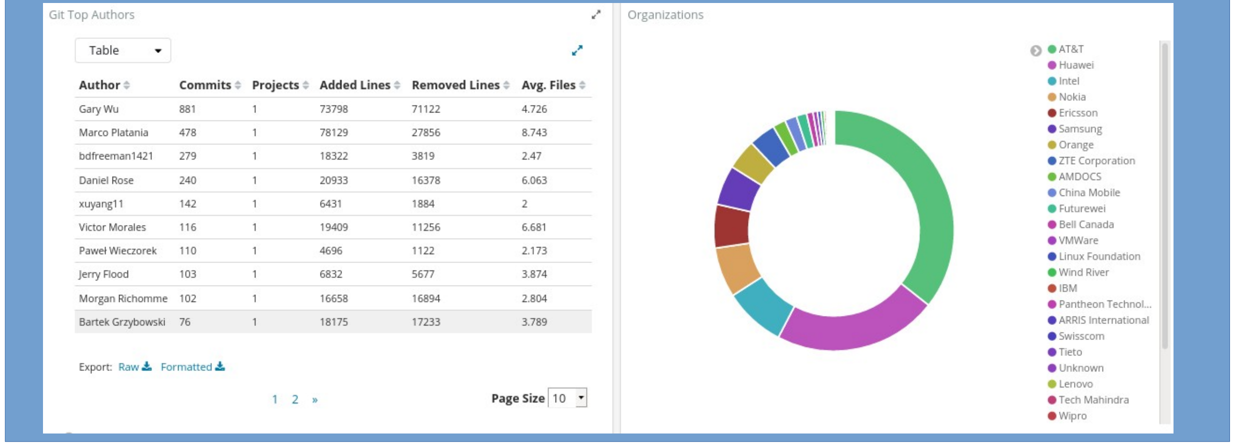
Several managed repositories :

- Integration
- Integration/*
- Testsuite
- Testsuite/*
- Demo

* : https://wiki.onap.org/display/DW/Approved+Projects

Historical reminders

Integration Last 3 years bitergia statistics



Git Top Authors

Table

Author	Commits	Projects	Added Lines	Removed Lines	Avg. Files
Gary Wu	881	1	73798	71122	4.726
Marco Platania	478	1	78129	27856	8.743
bdfreeman1421	279	1	18322	3819	2.47
Daniel Rose	240	1	20933	16378	6.063
xuyang11	142	1	6431	1884	2
Victor Morales	116	1	19409	11256	6.681
Paweł Wieczorek	110	1	4696	1122	2.173
Jerry Flood	103	1	6832	5677	3.874
Morgan Richomme	102	1	16658	16894	2.804
Bartek Grzybowski	76	1	18175	17233	3.789

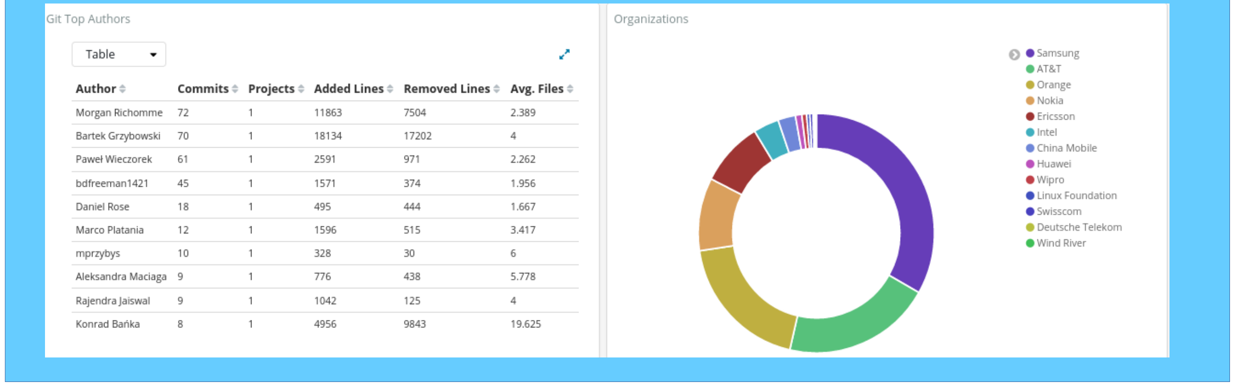
Export: Raw Formatted

1 2 »

Page Size 10

Organizations

Integration Last 6 months bitergia statistics



Git Top Authors

Table

Author	Commits	Projects	Added Lines	Removed Lines	Avg. Files
Morgan Richomme	72	1	11863	7504	2.389
Bartek Grzybowski	70	1	18134	17202	4
Paweł Wieczorek	61	1	2591	971	2.262
bdfreeman1421	45	1	1571	374	1.956
Daniel Rose	18	1	495	444	1.667
Marco Platania	12	1	1596	515	3.417
mprzybys	10	1	328	30	6
Aleksandra Maciaga	9	1	776	438	5.778
Rajendra Jaiswal	9	1	1042	125	4
Konrad Bańka	8	1	4956	9843	19.625

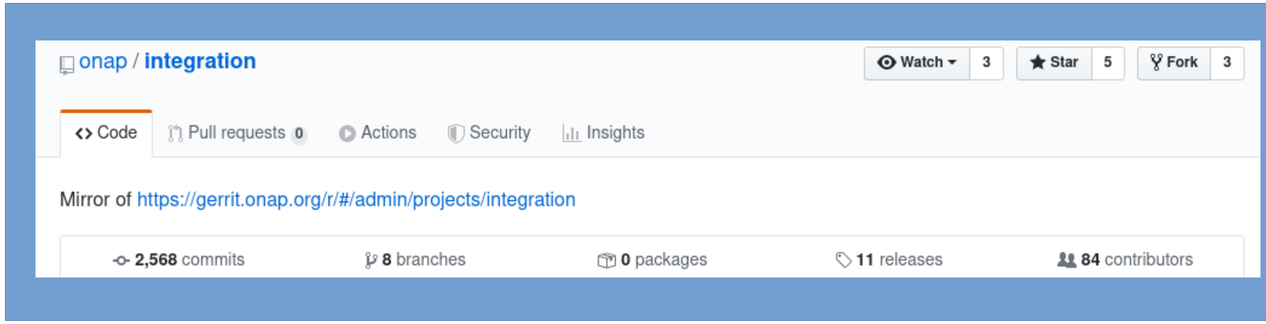
Organizations

List of committers (+2/merge rights on the different repositories) :

- Christophe Closset
- Daniel Rose
- Catherine Lefevre
- Marco Platania
- Brian Freeman
- Morgan Richomme
- Bartek Grzybowski
- Marcin Przybysz
- Eric Multanen

- Regular active contributors
- Pawel Wieczorek
- Krzysztof Kuzmicki
- Andreas Geisler
- Lukasz Rajewski
- And many more (71 authors over the last 6 months)

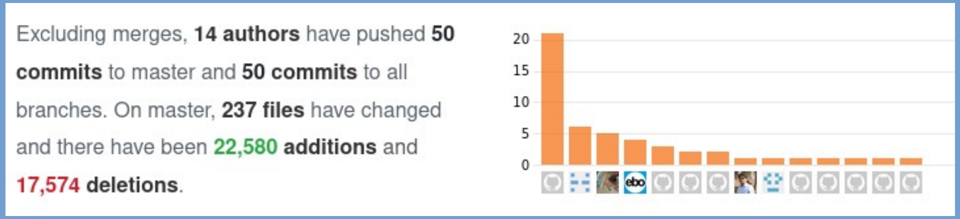
Historical reminders



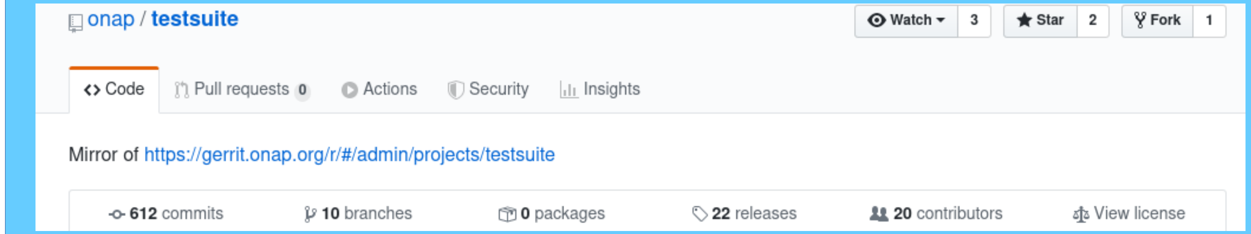
onap / **integration**

Mirror of https://gerrit.onap.org/r/#/admin/projects/integration

2,568 commits · **8** branches · **0** packages · **11** releases · **84** contributors



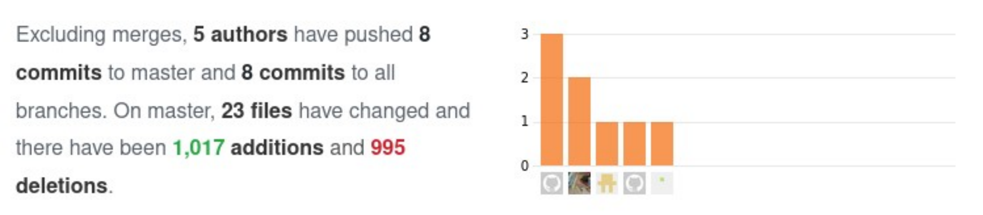
Excluding merges, **14 authors** have pushed **50 commits** to master and **50 commits** to all branches. On master, **237 files** have changed and there have been **22,580 additions** and **17,574 deletions**.



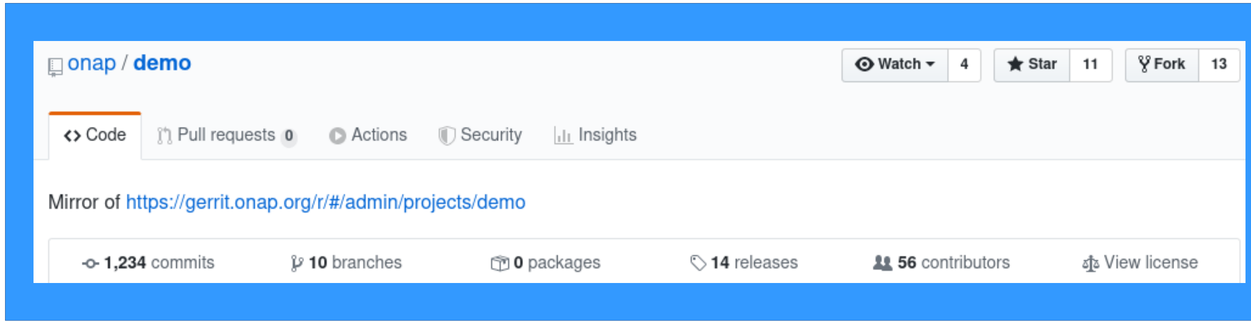
onap / **testsuite**

Mirror of https://gerrit.onap.org/r/#/admin/projects/testsuite

612 commits · **10** branches · **0** packages · **22** releases · **20** contributors · View license



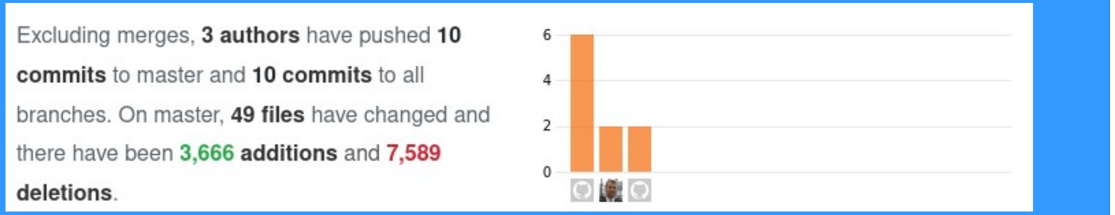
Excluding merges, **5 authors** have pushed **8 commits** to master and **8 commits** to all branches. On master, **23 files** have changed and there have been **1,017 additions** and **995 deletions**.



onap / **demo**

Mirror of https://gerrit.onap.org/r/#/admin/projects/demo

1,234 commits · **10** branches · **0** packages · **14** releases · **56** contributors · View license



Excluding merges, **3 authors** have pushed **10 commits** to master and **10 commits** to all branches. On master, **49 files** have changed and there have been **3,666 additions** and **7,589 deletions**.

Historical reminders

ONAP Testing group reporting

http://testresults.opnfv.org/onap-integration/index.html

Several labs :

- Windriver lab (>Amsterdam):
  - use cases (SB-00, SB-01,..)
  - Developers (VM AAF, ...)
- Orange Lab (>Dublin):
  - Gating (gating 1,2,3,4)
  - CI Daily (master and stable)
  - OpenLab (last stable)
- DT Lab (>El Alto):
  - CI Daily Master
- E// Lab (>El Alto)
  - CI Daily Master
- (China Mobile lab : use case)

- A real time support chan : https://team.onap.eu

Integration scope

Integration project mission

« Integration is responsible for

- ONAP cross-project system integration
- CI/CD,
- and all related end-to-end release use cases testing with VNFs necessary for the successful delivery and industry adaption of the ONAP project as a whole. »

ONAP cross-project system integration

oparent management (java dependencies) – concretely managed by Pam (Policy)

Java11 docker for components :
https://gitlab.com/onap-integration/docker/onap-java/container_registry

Follow-up of the CSIT jenkins page dealing with project functional page (docker compose and jjb created historically by Integration team for the projects)

Hosting different pages during the release (e.g. for Frankfurt*)

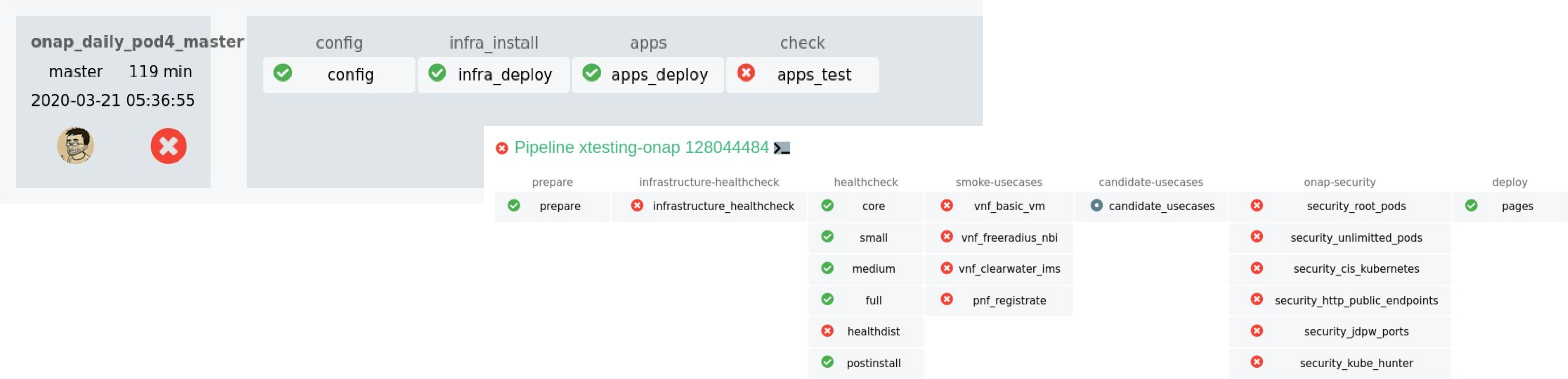
- 0: Integration Weather Board for Frankfurt Release – check list for the PTL
- 1: Frankfurt Release Integration Test Blocking Issues – high and highest Jiras
- 2: Frankfurt Release Integration Testing Status – follow-up of use case integration

*: https://wiki.onap.org/display/DW/Integration+F+Release

CI-CD

CIST run Daily

- End to End CI/CD through 2 types of chains
- CI Daily chains (Master and Stable) : everyday we redeploy and test an ONAP solution based on OOM Master / last Stable
- Gating : deployment/test of a full ONAP solution on any OOM and SO patchset submission



onap_daily_pod4_master

master 119 min

2020-03-21 05:36:55

✗

config	infra_install	apps	check
✓ config	✓ infra_deploy	✓ apps_deploy	✗ apps_test

✗ Pipeline xtesting-onap 128044484

prepare	infrastructure-healthcheck	healthcheck	smoke-usecases	candidate-usecases	onap-security	deploy
✓ prepare	✗ infrastructure_healthcheck	✓ core	✗ vnf_basic_vm	○ candidate_usecases	✗ security_root_pods	✓ pages
		✓ small	✗ vnf_freeradius_nbi		✗ security_unlimitted_pods	
		✓ medium	✗ vnf_clearwater_ims		✗ security_cis_kubernetes	
		✓ full	✗ pnf_registrate		✗ security_http_public_endpoints	
		✗ healthdist			✗ security_jdpw_ports	
		✓ postinstall			✗ security_kube_hunter	

and all related end-to-end release use cases testing with VNFs necessary for the successful delivery and industry adaption of the ONAP project as a whole. »

Management of the robot VM included in OOM and used to perform healthchecks + automation part of some use cases

Deployment of the ONAP solutions in Windriver for the use cases.

Provide Support to the use case project to finalize the integration in an integration lab

But Integration is

NOT LF IT

- Repository creation / Nexus / Gerrit / Jenkins

NOT a use case project

- does not write the code of the simulators/test scenarios
- does not define requirements for the projects
- does not define the architecture/interconnection solutions

NOT responsible of the project tests, only of the integration of them + its own tests

NOT responsible of the certificate renewals

NOT responsible of the security expectations (and possible waivers)

- Xfail use
- Use of java11 docker

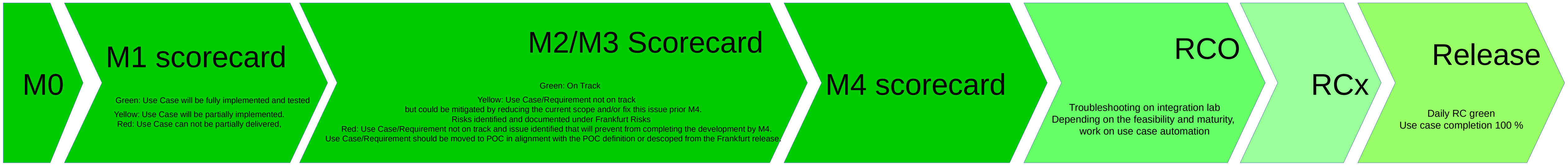
NOT responsible of the docker version of the components (reference is OOM Master, override.yaml with staging is only for pre-tests)

Integration & use cases

Integration and use cases

TSC Prioritization (Ranking)
RANK #0 – Special GO - quick wins, fully covered by involved companies
RANK #1 – TSC Must Have – Mandatory for the release
RANK #2 – Continuity - Items continued from previous releases
RANK #3 – PTL Go – items that PTLs is OK to include since team has bandwidth
RANK #4 – NO GO – items not approved for various reasons

Use cases



M0

M1 scorecard

Green: Use Case will be fully implemented and tested
Yellow: Use Case will be partially implemented.
Red: Use Case can not be partially delivered,

M2/M3 Scorecard

Green: On Track
Yellow: Use Case/Requirement not on track
but could be mitigated by reducing the current scope and/or fix this issue prior M4.
Risks identified and documented under Frankfurt Risks
Red: Use Case/Requirement not on track and issue identified that will prevent from completing the development by M4.
Use Case/Requirement should be moved to POC in alignment with the POC definition or descoped from the Frankfurt release.

M4 scorecard

RCO

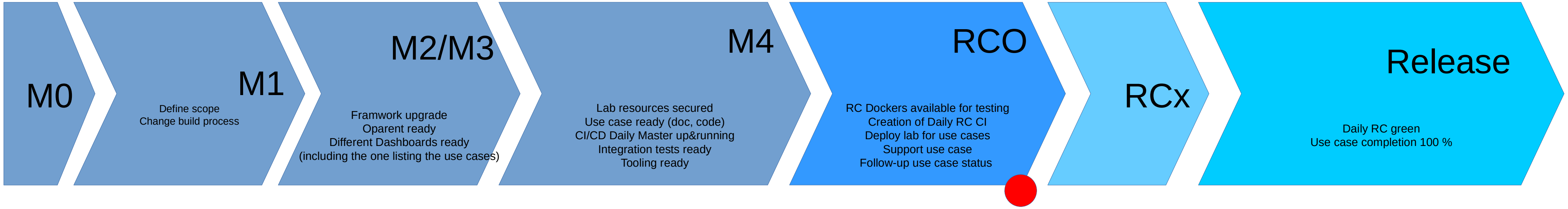
Troubleshooting on integration lab
Depending on the feasibility and maturity,
work on use case automation

RCx

Release

Daily RC green
Use case completion 100 %

integration



M0

M1

Define scope
Change build process

M2/M3

Framwork upgrade
Oparent ready
Different Dashboards ready
(including the one listing the use cases)

M4

Lab resources secured
Use case ready (doc, code)
CI/CD Daily Master up&running
Integration tests ready
Tooling ready

RCO

RC Dockers available for testing
Creation of Daily RC CI
Deploy lab for use cases
Support use case
Follow-up use case status

RCx

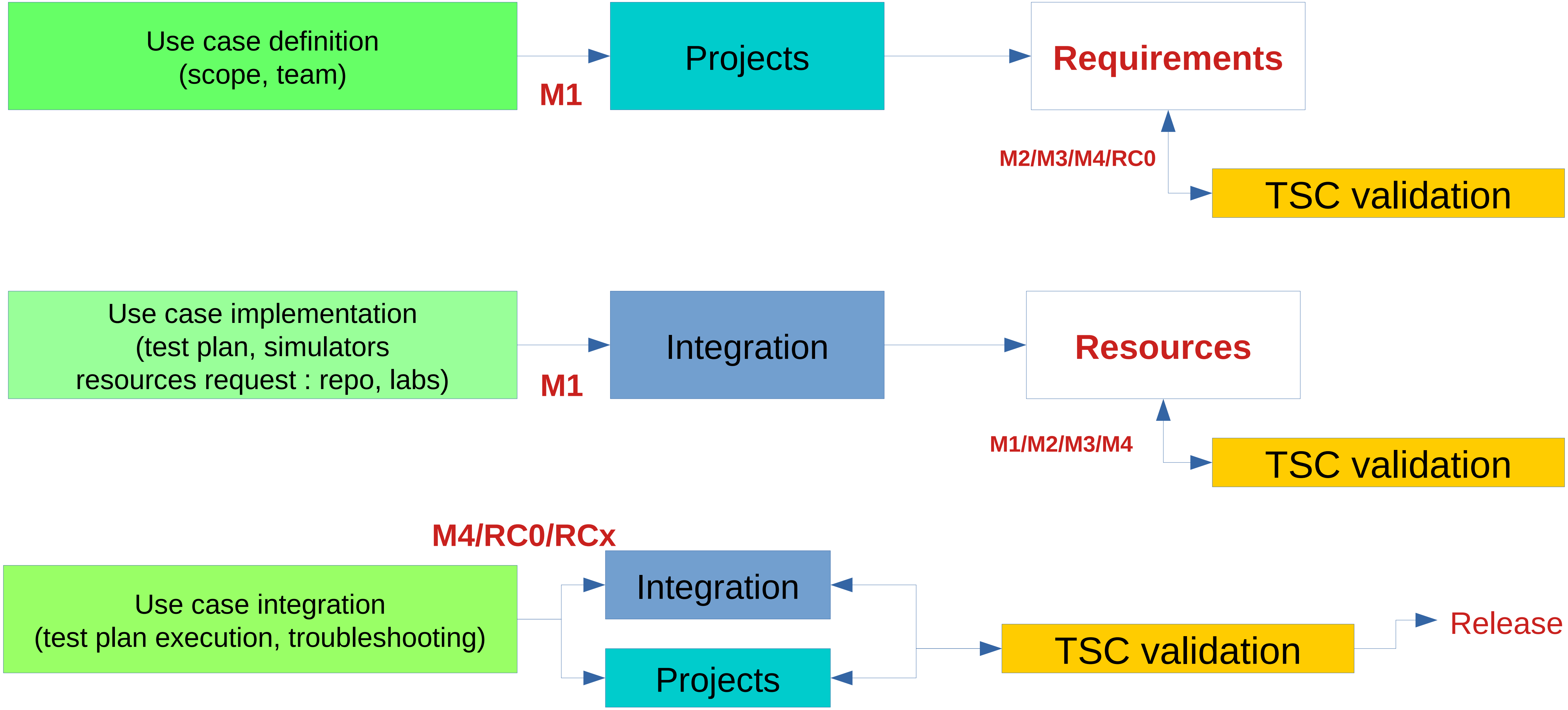
Release

Daily RC green
Use case completion 100 %

Target ONAP
Available for use cases

Src : https://wiki.onap.org/display/DW/Frankfurt+Release+Requirements
https://wiki.onap.org/display/DW/Frankfurt+Deliverables+by+Milestone

Use case journey (seen from Integration perspective)



Use case definition (scope, team)

M1

Projects

Requirements

M2/M3/M4/RC0

TSC validation

Use case implementation (test plan, simulators resources request : repo, labs)

M1

Integration

Resources

M1/M2/M3/M4

TSC validation

M4/RC0/RCx

Use case integration (test plan execution, troubleshooting)

Integration

Projects

TSC validation

Release

Chronology of a use case

1) The use case is defined (scope, requirements, resources)
2) The use cas is run on a local lab (usually the use case team)
3) The use cae is documented (M4) so it could be in theory replayed on another lab
4) The use case is run in an Integration lab (with rc dockers) – runnability must be assesable
5) The test plan of the use case is fully executed in the RC target env (during Rcx, 100 % expected for the release)

Optionnaly - the use case can be integrated in CI chains to be part of the automated verification of the future releases to ensure the stability of the solution (use cases become part of the samples provided with ONAP).

By default any new use case for a release is a candidate use case for CI integration.
If integrated it becomes a smoke use cases, Integration becomes responsible to maintain it over the releases.
Smoke use cases must be PASS on CI for release criteria. Note that all the use cases will not be integrated as we must keep the CI under control but we should select representative use case to cover as many features as possible.

Reminder today smoke use cases are

- basic_vm : onboarding/distribution/instantiation of a single VM through ONAP ugin VNF_API
- freeradius_nbi : idem basic VM but using northbound API of NBI to instantiate
- clearwater_ims : deployment of a vIMS clearwater
- pnf_registrate.

Focus on Automation

First of all, obviously, automation is possible ONLY if the use case can be executed manually determinastically (run anytime/ anywhere)

We distinguish several levels:

- 0 automation: the documentation associated with the use case is sufficient to perform the use case step by step
- Partial automation: you develop some bash/robot/python/go/whatever scripts to simplify some parts of the use case execution on an environement already providing the resources (tenants, networks, Public IP, ..) and the tooling.
- Advanced automation: you develop scripts to simplify the execution of the use cases including the setup/teardown of resources but some manual steps are still needed
- Full automation: your test can be run in 1 click on any environement. It means you can create the needed resources on Openstack/Kubernetes (setup), execute the tests, collect the results, clean the resources without manual operations in the the middle. Test framework can be used to help synchronising states / managing complexity if needed.

Focus on Automation

Automation is NOT possible if

- You use proprietary VNF under commercial licences
- You use specific tooling (commercial loader/emulators, RAN, physical equipments) that cannot be used by the other labs

Full automation is not possible if

- You do not have programmatic access to your cloud resources (K8s or openstack client with admin rights)

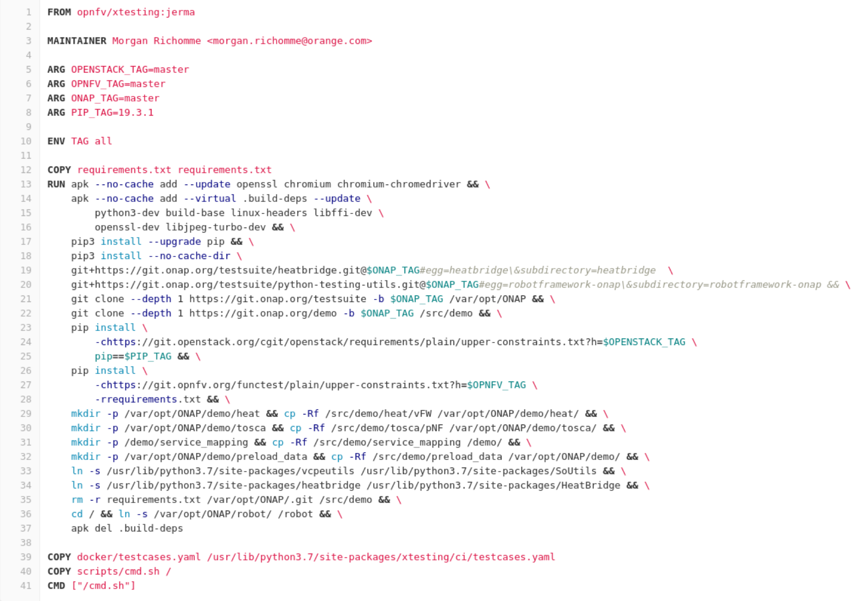
Integration in CI

Assuming that you reached the One click stage, it is possible to include your use case in ONAP CI.
It can be done as a gitlab-ci/jenkins stage (but then you need to get some skills in CI)
Or it can be done by embedding your use case in a xtesting docker (you will leverage the harmonization of inputs/outputs and ease the integration in any CI chain)

First integration can been done as « candidate use case » in integration/xtesting :
https://gitlab.com/Orange-OpenSource/lfn/onap/integration/xtesting

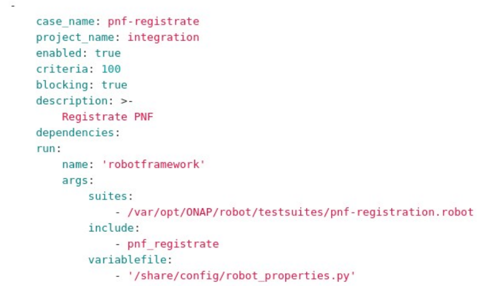
In the Dockerfile/requirements, you must add all you need to run your tests

- Librairies
- Scripts
- Config files



```
FROM opnfv/xtesting:jerma

MAINTAINER Morgan Richomme <morgan.richomme@orange.com>

ARG OPENSTACK_TAG=master
ARG OPNFV_TAG=master
ARG ONAP_TAG=master
ARG PIP_TAG=19.3.1

ENV TAG all

COPY requirements.txt requirements.txt
RUN apk --no-cache add --update openssl chromium chromium-chromedriver && \
    apk --no-cache add --virtual .build-deps --update \
        python3-dev build-base linux-headers libffi-dev \
        openssl-dev libjpeg-turbo-dev && \
    pip3 install --upgrade pip && \
    pip3 install --no-cache-dir \
    git+https://git.onap.org/testsuite/heatbridge.git@$ONAP_TAG#egg=heatbridge\&subdirectory=heatbridge  \
    git+https://git.onap.org/testsuite/python-testing-utils.git@$ONAP_TAG#egg=robotframework-onap\&subdirectory=robotframework-onap && \
    git clone --depth 1 https://git.onap.org/testsuite -b $ONAP_TAG /var/opt/ONAP && \
    git clone --depth 1 https://git.onap.org/demo -b $ONAP_TAG /src/demo && \
    pip install \
        -chttps://git.openstack.org/cgit/openstack/requirements/plain/upper-constraints.txt?h=$OPENSTACK_TAG \
        pip==$PIP_TAG && \
    pip install \
        -chttps://git.opnfv.org/functest/plain/upper-constraints.txt?h=$OPNFV_TAG \
        -rrequirements.txt && \
    mkdir -p /var/opt/ONAP/demo/heat && cp -Rf /src/demo/heat/vFW /var/opt/ONAP/demo/heat/ && \
    mkdir -p /var/opt/ONAP/demo/tosca && cp -Rf /src/demo/tosca/pNF /var/opt/ONAP/demo/tosca/ && \
    mkdir -p /demo/service_mapping && cp -Rf /src/demo/service_mapping /demo/ && \
    mkdir -p /var/opt/ONAP/demo/preload_data && cp -Rf /src/demo/preload_data /var/opt/ONAP/demo/ && \
    ln -s /usr/lib/python3.7/site-packages/vcpeutils /usr/lib/python3.7/site-packages/SoUtils && \
    ln -s /usr/lib/python3.7/site-packages/heatbridge /usr/lib/python3.7/site-packages/HeatBridge && \
    rm -r requirements.txt /var/opt/ONAP/.git /src/demo && \
    cd / && ln -s /var/opt/ONAP/robot/ /robot && \
    apk del .build-deps

COPY docker/testcases.yaml /usr/lib/python3.7/site-packages/xtesting/ci/testcases.yaml
COPY scripts/cmd.sh /
CMD ["/cmd.sh"]
```

You must reference your use case in testcases.yaml,
and precise your entry point (bash, python, robot,..)



```
-
    case_name: pnf-registrate
    project_name: integration
    enabled: true
    criteria: 100
    blocking: true
    description: >-
        Registrate PNF
    dependencies:
    run:
        name: 'robotframework'
        args:
            suites:
                - /var/opt/ONAP/robot/testsuites/pnf-registration.robot
            include:
                - pnf_registrate
            variablefile:
                - '/share/config/robot_properties.py'
```

Open Questions

Synchro of the use cases / releases roadmap

For the moment we set a dependency, shall we ? Use cases could be integrated in stable versions rather than on the next version.Use cases and releases could de desynchronized, it would avoid short time integration at the end of a release

- Possible if no new developments are required from projects
- Suppose a clean management of versions CI : today most of the efforts are on the next release, few fix coming from the projects (« it will be fixed in next release »)
- Suppose a clean way to communicate on use cases independently from the release

Use case repositories & artifacts

Today the documentation/code/scripts/simulators are most of the time hosted in integration/testsuite/demo. It would be better to dedicate specific repositories / use case to avoid mixing everything and clearly identify who is responsible of what. Work started with bbs.

Use case resource labs

Windriver ONAP labs are installed through Jenkins jobs created by Gary thanks to Marco and Brian. Both will not be so active in G. We need to find a solution to keep on offering the testing env to use cases.

Annex