Increased transparency on software modules included in an ONAP release

*** PROPOSAL FOR DISCUSSION ***

Amy Zwarico  Pawel Pawlak  Eric Debeau  Cedric Ollivier  Andreas Geissler  Timo Perala  Sylvain Desbureaux

Kenny Paul  David McBride  Catherine Lefevre

Targets

⚠️ ONAP must not depend on unmaintained software modules / projects.

Increase transparency on software modules (repositories) included in an ONAP release to enable effective and efficient ...

- security management
- release management
- documentation management

Provide reliable information for all stakeholder (rel mgr, projects, oom, sec, doc) ...

- which software module (repository) is included (was used to build the docker container)
- which version of the software module is included (was used to build the docker container)

Improve the lifecycle management of software modules/projects.

Provide transparent and reliable knowledge about the ...

- relationship between ONAP Release  Docker Container & Version  Repository & Version  Project
- dependencies (deploy time & runtime) among ONAP projects
- dependencies (runtime) between ONAP projects and use cases
- current lifecycle state of ONAP projects and repositories (incl. optimized maintenance of this information)

Enable a reliable and prompt decision making in case of unmaintained projects/repos.

Current Situation

- A deployable ONAP release consists of 'maintained' an 'unmaintained' software modules (used to build the docker container).
- Only 'projects' are managed in the release process
- But a 'project' consists of 1-n repositories. Repositories are not managed in the release process.
  - Question: If a project is (un)maintained, are all repositories (un)maintained?
- Only 'maintained' projects are managed in the release process.
- But "unmaintained" software modules (docker container) are added to the list of deployed software modules in the end of the release process (oom)
- Unmaintained, but 'participating' projects (not repositories!) of the 'Istanbul' release are:
  - Application Authorization Framework AAF
  - Application Controller APPC
  - External API
  - External System Register ESR
  - Logging
  - MUSIC
  - Portal
  - Virtual Infrastructure Deployment VID
- The management of 'release participation' ...
  - is limited (projects managed, not repositories)
  - has distributed responsibilities (relmgr, oom, sec, doc)
  - and is not synced and done in a collaborative, end-to-end manner
- Information about the lifecycle state of projects/repositories and their release participation is distributed, manually maintained and not in sync, e.g.
  - GIT: repository state (active | read only; Kenny? LFIT?)
  - INFO.YAML: project state (inherited to every repo of the project; project lead)
  - WIKI: release participation (release manager)
  - WIKI: project lifecycle state (Kenny)
  - WIKI: documentation tracking per release (doc team)
  - RTD: security vulnerabilities / package updated (sec team)
  - RTD: Interactive ONAP Architecture Diagram
  - Helm charts?
  - Jenkins Jobs?
- There is no transparency whether use cases depend on unmaintained software modules (repositories)
Proposal

The information about software modules included in an ONAP release must be managed ...  
* by the release manager (responsible) and stakeholder (contributing)  
* on repository level  
* from the start of the release process (planning phase)  
* to the end of the release process  
  * building the deployable release  
  * building the release specific documentation  
* consistently (one single source) for all stakeholder  
* with a high grade of automation

Dependencies between ONAP projects and use cases must be managed ...  
* by the Architecture Subcommittee (incl. continuous documentation) together with the TSC (incl. the decision how to continue with an unmaintained project)  
* on project level

To finally participate in an ONAP release, a software module ...  
* must be in the 'maintained' state  
* must have created a release branch  
* must have updated their documentation to reflect the current state of development  
* must have up-to-date release notes (on sub-project level or in the main project)

Stakeholders are only using/managing maintained software components/projects.

Use cases are based only on functionality provided by maintained projects/repositories.

The release process ensures the compliance of the above topics and provides appropriate means to check and react in a timely manner.

Additional Information

* Excel Spreadsheet: onap_tables_211128.xlsx  
* conf.py (istanbul) showing the current situation for documentation:

```python
from docs_conf.conf import *

# Change the line below from 'latest' to '<Name_of_the_new_ONAP_Release>' after  
# you have create the new branch for the 'doc' project.
branch = 'istanbul'
doc_url = 'https://docs.onap.org/projects'  
master_doc = 'index'

intersphinx_mapping = {}  

# Mapping to 'latest' if this files is used in 'latest' (master) branch.  
# Changed to '<Name_of_the_new_ONAP_Release>' after you have create the new  
# branch for the 'doc' project.
#
# Istanbul
intersphinx_mapping['onap-aai-aai-common'] = '{}/onap-aai-aai-common/en/%s'.format(doc_url) % branch, None)  
intersphinx_mapping['onap-aai-sparky-be'] = '{}/onap-aai-sparky-be/en/%s'.format(doc_url) % branch, None)  
intersphinx_mapping['onap-ccsdk-apps'] = '{}/onap-ccsdk-apps/en/%s'.format(doc_url) % branch, None)  
intersphinx_mapping['onap-ccsdk-cds'] = '{}/onap-ccsdk-cds/en/%s'.format(doc_url) % branch, None)  
intersphinx_mapping['onap-ccsdk-features'] = '{}/onap-ccsdk-features/en/%s'.format(doc_url) % branch, None)  
intersphinx_mapping['onap-ccsdk-distribution'] = '{}/onap-ccsdk-distribution/en/%s'.format(doc_url) % branch, None)  
intersphinx_mapping['onap-ccsdk-oran'] = '{}/onap-ccsdk-oran/en/%s'.format(doc_url) % branch, None)
```


intersphinx_mapping['onap-cli'] = ('{}/onap-cli/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-cps'] = ('{}/onap-cps/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-cps-ncmp-dmi-plugin'] = ('{}/onap-cps-ncmp-dmi-plugin/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-cps-cps-temporal'] = ('{}/onap-cps-cps-temporal/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-dcaegen2'] = ('{}/onap-dcaegen2/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-dmaap-message-router-messageservice'] = ('{}/onap-dmaap-message-router-messageservice/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-dmaap-buscontroller'] = ('{}/onap-dmaap-buscontroller/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-dmaap-datarouter'] = ('{}/onap-dmaap-datarouter/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-holmes-engine-management'] = ('{}/onap-holmes-engine-management/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-holmes-rule-management'] = ('{}/onap-holmes-rule-management/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-integration'] = ('{}/onap-integration/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-modeling-etsicatalog'] = ('{}/onap-modeling-etsicatalog/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-multicloud-framework'] = ('{}/onap-multicloud-framework/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-multicloud-k8s'] = ('{}/onap-multicloud-k8s/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-msb-apigateway'] = ('{}/onap-msb-apigateway/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-msb-swagger-sdk'] = ('{}/onap-msb-swagger-sdk/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-oom'] = ('{}/onap-oom/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-oam-plugin'] = ('{}/onap-oam-plugin/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-so'] = ('{}/onap-so/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnc'] = ('{}/onap-vnc/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-guidelines'] = ('{}/onap-vnfrqts-guidelines/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-requirements'] = ('{}/onap-vnfrqts-requirements/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-testcases'] = ('{}/onap-vnfrqts-testcases/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-usecases'] = ('{}/onap-vnfrqts-usecases/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-vcspolicy'] = ('{}/onap-vnfrqts-vcspolicy/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-vmpolicy'] = ('{}/onap-vnfrqts-vmpolicy/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vnfrqts-vmprofile'] = ('{}/onap-vnfrqts-vmprofile/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vvppolicy'] = ('{}/onap-vvppolicy/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vvppolicy-oam'] = ('{}/onap-vvppolicy-oam/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vvp-documentation'] = ('{}/onap-vvp-documentation/en/{}.format(doc_url) % branch, None)

# Mapping to an older version of documentation for those projects who are part of the new ONAP release but have not created a branch for it.
# This can happen e.g. for 'unmaintained' (but still required) projects.

# Honolulu
branch = 'honolulu'
intersphinx_mapping['onap-dmaap-dbcapi'] = ('{}/onap-dmaap-dbcapi/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-externalapi-nbi'] = ('{}/onap-externalapi-nbi/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-vid'] = ('{}/onap-vid/en/{}.format(doc_url) % branch, None)

# Guillin
branch = 'guillin'
intersphinx_mapping['onap-portal'] = ('{}/onap-portal/en/{}.format(doc_url) % branch, None)

# Frankfurt
branch = 'frankfurt'
intersphinx_mapping['onap-appc'] = ('{}/onap-appc/en/{}.format(doc_url) % branch, None)
intersphinx_mapping['onap-appc-deployment'] = ('{}/onap-appc-deployment/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-music'] = ('{}/onap-music/en/%s'.format(doc_url) % branch, None)

# Mapping to 'latest' (master) version of documentation for those projects who
# are part of the new ONAP release but never have created a branch.
#
# Latest
branch = 'latest'
intersphinx_mapping['onap-saf-authz'] = ('{}/onap-saf-authz/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-saf-sms'] = ('{}/onap-saf-sms/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-aai-event-client'] = ('{}/onap-aai-event-client/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-aai-esr-gui'] = ('{}/onap-aai-esr-gui/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-aai-esr-server'] = ('{}/onap-aai-esr-server/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-ccsdk-dashboard'] = ('{}/onap-ccsdk-dashboard/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-ccsdk-platform-plugins'] = (
    '{}/onap-ccsdk-platform-plugins/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-logging-analytics'] = ('{}/onap-logging-analytics/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-modeling-toscaparsers'] = ('{}/onap-modeling-toscaparsers/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-msb-discovery'] = ('{}/onap-msb-discovery/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-msb-java-sdk'] = ('{}/onap-msb-java-sdk/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-multicloud-azure'] = ('{}/onap-multicloud-azure/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-music-distributed-kv-store'] = ('{}/onap-music-distributed-kv-store/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-oparent-cia'] = ('{}/onap-oparent-cia/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-osd'] = ('{}/onap-osd/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-sdc-sdc-distribution-client'] = ('{}/onap-sdc-sdc-distribution-client/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-sdc-sdc-workflow-designer'] = ('{}/onap-sdc-sdc-workflow-designer/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-sdc-sdc-tosca'] = ('{}/onap-sdc-sdc-tosca/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-sdc-sdc-docker-base'] = ('{}/onap-sdc-sdc-docker-base/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-so-libs'] = ('{}/onap-so-libs/en/%s'.format(doc_url) % branch, None)
intersphinx_mapping['onap-vfc-nfvo-driver-vnfm-svnfm'] = ('{}/onap-vfc-nfvo-driver-vnfm-svnfm/en/%s'.format(doc_url) % branch, None)

linkcheck_ignore = [
    'http://localhost',
    'https://example.com',
    'about:config',
    # this URL is not directly reachable and must be configured in the system hosts file.
    'https://portal.api.simpledemo.onap.org:30225/ONAPPORTAL/login.htm',
    # anchor issues
html#release-non-functional-requirements',
]

html_last_updated_fmt = '%d-%b-%y %H:%M'

def setup(app):
    app.add_css_file("css/ribbon.css")

* "Unmaintained" Meeting Notes (incl. slide deck from David McBride)
- Jakarta documentation page
- Session planned for the "2022 LFN Developer Event" January 2022