DOC Challenges and Proposals

• Intention of this document
• Challenges
  • Content
    • Software related content (Wiki, RTD)
    • Projects related content (Wiki)
  • Process
    • Process description
    • Writing
    • Converting (Wiki-to-RST)
    • Building (RST)
• Proposals
  • Create and approve guidelines for ...
  • Usage
    • Wiki
    • RTD
  • Content
    • RTD: ONAP Software related
    • Wiki: ONAP (sub)projects and development related
    • Responsibilities
  • Process
  • Presentation
• Approach
  • For content in responsibility of the doc project
  • For content in responsibility of the (sub)projects
  • To be discussed
• Questions
• Related Tickets
• Wiki-RTD Guideline Presentation
• Document History

Intention of this document
With the listed observations and described proposals we would start the discussion of how we should continue in the documentation project. We are aiming an aligned view on the fields of improvements and their prioritization.

Challenges

Content

Software related content (Wiki, RTD)
• hard to find due to the split between Wiki and RTD
• still available (and publicly editable) in the Wiki even after transfer to RTD
• partly outdated
• partly incomplete
• partly missing at all
• E2E doc partly missing
  • even in the case all single (functional) projects would be well documented, we need end-to-end documentation that explains (user) processes / use cases holistically (spanning over several ONAP modules)
• no QA in the release process

Projects related content (Wiki)
• partly outdated project profiles
• relation to a specific release unclear
  • project scope might evolve over time; no info about this
• activity state unclear

Process

Process description
• Wiki: "Documenting ONAP" partly outdated
• RTD: "Dev Guides - Creating Documentation" partly outdated
Writing

- writing doc in RST format is not fully accepted
- it is an error prone, mainly manual process
- Wiki: Easy to create doc here - but afterwards content often remains only in the Wiki

Converting (Wiki-to-RST)

- no simple solution to convert Wiki content automatically to the RST format
- Wiki as a dev environment for doc?
  - Problem: linking to other doc elements (rst) possible?

Building (RST)

- approx. 1500 build errors (@RTD only, elalto-verify-13.11.2019)
- warnings are ignored
- failed builds are not recognized
- currently one large build which includes all project doc as submodules
- expertise for doc tooling & process is no longer available
- no QA in the release process

Proposals

Create and approve guidelines for ...

Usage

Wiki

- writing documentation in RST format and using gerrit for reviews remains the first choice
- as an alternative the Wiki can be used for development and collaboration on documentation
- doc must be released afterwards in RST format for RTD

RTD

- the only source for released documentation
- ambition: no build errors
  - at least no errors which are causing content not showing up at RTD

Content

RTD: ONAP Software related

- alligned high-level-view on chapters, subchapters and to-be-addressed readerships
  - ONAP overview documentation
  - ONAP architecture
  - ONAP developer guide
  - ONAP user guide (E2E)
  - ONAP (sub)project guide
    - release notes
    - architecture
    - logging & diagnostics
    - human interfaces
    - admin guide
    - deployment / installation
    - config
    - user guide
    - api (consumed, offered)
  - ONAP admin guide
  - ONAP release notes
  - all grouped by release
- doc requirements (must-haves) depending on the type of doc (functional, non-functional, E2E, ...)
  - to be proved in the dev process
  - TODO: develop doc requirements / templates

Wiki: ONAP (sub)projects and development related

- project management (meetings, plans, milestones, members, ...)
- project dev guides
- ongoing activities and discussions
• Exception: "dev environment" for documentation

• page header information names ...
  • targeted release
  • status
    • under development
    • release candidate
    • released (incl. link to RTD)
• OPTION 1: delete content after "ready for release" at RTD
• OPTION 2: mark as "released at RTD" and keep for further "development"
• Drawback: manual conversion to RST!

Responsibilities

• doc project
  • overall doc structure and build process
  • ONAP user guide (E2E)
    • managed by doc project
    • done together by doc project and subject matter experts
  • ONAP doc dev guide
  • ONAP release notes
  • ONAP overview documentation
  • (sub)projects
    • (sub)project release notes
    • ONAP (sub)product guide
  • OPEN: ONAP architecture?

Process

• QA for content and build in place
• available documentation (from the earlier release) will be validated and approved for the next release during the development process
  • result: if content is available in the doc for the specific release, then it is definitely valid for this release
• write (Wiki) & convert (RST) as easy as possible: evaluate "Confluence2MD" and create a (semi)automatic solution
• build doc asynchronous / per project and get rid of the monolithic build / submodules

Presentation

• doc can be easily searched and filtered
• easy understandable overall structure of chapters and subchapters
• clearly addressing the type of readership (user, developer, admin, manager ...)
• RTD & sphinx theme
  • doc must be appealing to the readership
  • well defined set of fonts and colors

Approach

For content in responsibility of the doc project

• analyse existing Wiki content
  • TODO: check assessment of Eric Debeau
  • TODO: statistics to show status of doc (outdated, incomplete, missing, ...)
• mark existing Wiki pages which are not yet migrated to RTD
• delete existing Wiki pages which are migrated to RTD ?
• define & implement QA process
• update "Contribution" chapters in Wiki and RTD depending on the results of this activity

For content in responsibility of the (sub)projects

• analyse existing Wiki content
• mark existing Wiki pages which are not yet migrated to RTD
• OPT: delete existing Wiki pages which are migrated to RTD
• define & implement QA process

To be discussed

• TODO: search for an expert in tooling / doc writing that can support
• TODO: support from LF?
• TODO: evaluate tooling and best-practice process for open source documentation and increase acceptance of developers
• TODO: reactivate / update doc team liaisons (as we had it in earlier times)!
• PROPOSAL: Complete restart with the above discussed QA processes in place to build up a new documentation and using only validated / approved and therefore valid content. Even if we are expecting, that the upcoming doc releases will be incomplete or empty in large areas in the beginning.
PROPOSAL: Under context of E2E doc for users, an initial proposal is being discussed to start documenting subset deployment (2020-01-16 ver., 2020-01-23 ver.)

Questions

What is the value of more and more ONAP features without having them - and other, important basics - not well documented?

Example: SDC usage

Are we (developers, PTL, members of the doc project, TSC, ...) able to master the current challenges for ONAP doc without spending much (!) more time and effort for documentation?

Are we willing to spend much (!) more time and effort for documentation?

How can "we" incentivize "good" documentation?

Related Tickets

DOC-573 - Getting issue details... STATUS

ONAPARC-274 - Getting issue details... STATUS

Wiki-RTD Guideline Presentation

Proposal (updated 23.1.2020)

ONAP-Documenta...-RTD-v0.4.pptx

Document History

2020-01-07 Ticket added to "Related Tickets"
2019-12-19 Added proposal to evaluate "Confluence2MD" for semi-automatic Wiki-2-RST conversion
2019-12-11 Added "Related Tickets"; Minor editorial changes
2019-12-10 Initial doc based on the (modified) mindmap. Alignment to be done.
2020-01-17 Added TSC/PTL draft slides for Guideline