VNF Requirements Project Overview

The intent of the VNF Requirement and Guidelines project is to inform VNF providers of the standards, specifications, and guidelines to which they should adhere when targeting the ONAP platform. These requirements and guidelines will support the ONAP Architecture Principles, and ensure a consistent experience for VNF providers across the VNF lifecycle. See the VNF Requirements Charter for more information.

Objectives of This Guide

This guide targets people that may want to contribute content to the project or have a better understanding of the standards that apply to the requirements. These standards will attempt to address the following concerns:

- Ensure the content is appropriate for inclusion within the project deliverables
- Guide contributors on the proper sub-project the provided content should target (e.g. guidelines, requirements, etc.)
- Establish a consistent format for all content
- Standardize the metadata collected for each requirement

Requirement Structure and Repositories

The VNF Requirements Project comprises of several different document repositories (i.e. sub-projects) that provide guidance to VNF Providers from various perspectives. The following table outlines the repository where the content is stored, the location of the latest published version of the content on the official ONAP documentation site, the description of the content. When contributing content, please familiarize yourself with the various projects and target the content to the appropriate repository.

<table>
<thead>
<tr>
<th>Repo</th>
<th>Description</th>
<th>Deliverable document Title(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/guidelines</td>
<td>This includes objectives and motivations for the VNF Requirements work as well as forward looking, narrative text for use in prototype RFP text.</td>
<td>• ONAP VNF Guidelines</td>
</tr>
<tr>
<td>/requirements</td>
<td>This includes formalized, uniquely numbered requirements that outline the requirements and specifications to which a VNF provider should adhere. Requirements will strive to be discrete and testable where possible, and will follow the guidance of RFC 2119 of requirement key words (e.g. SHOULD, MAY, etc.) for all numbered requirements.</td>
<td>• ONAP VNF Requirements</td>
</tr>
<tr>
<td>/usecases</td>
<td>Documents VNF specific use cases in support of ONAP E2E use cases illustrating behavior, sequences of operation, variants, error conditions, etc. Not every Use Case supported by ONAP will be documented in this section. Instead, key Use Cases that require specific coordination with the VNF can be added here to better describe the VNF’s responsibilities as a participant in the Use Case.</td>
<td>• ONAP VNF Provider Use Cases</td>
</tr>
</tbody>
</table>
This expands the use case template structure to supply the additional fields necessary to describe a test scenario. There may be multiple test case descriptions associated with a single use case.

General Standards

The following standards apply to all sub-projects within the VNF Requirements Project.

- As VNF Requirements project is an ONAP project, all content must adhere to the general documentation standards defined in Creating Documentation section of the ONAP Developer Guide.
- All content must be written in reStructuredText (RST) with all warnings and errors resolved.
- Wherever possible, let RST handle numbering of content. This includes ordered lists, section numbers, footnotes, etc. This ensure content can be re-arranged easily with less likelihood of breaking numbering conventions.

Guideline Standards

The VNF and PNF Guidelines are the highest level guidance provided by the VNF Requirements project. This section is best suited to include content that may provide context or forward-looking statements instead of concrete, verifiable requirements to which a VNF Provider should adhere. This text is strictly narrative in format, and does not include enumerated requirement and associated metadata as is the case in the VNF and PNF Requirements section of the document.

If a substantial change is to be proposed in this area of the requirements, then it should be initiated as a proposal with a corresponding JIRA ticket. Refer to VNFRQTS How to Contribute for more details.

Requirement Standards

VNF Requirements are the foundation of the project and as such have the most stringent content standards. The requirements form the basis of the specifications that a VNF provider must adhere to in order to successfully onboard, deploy, and operate a VNF on ONAP. The goal of this section is to provide independent, enumerated, and generally verifiable requirements to which a VNF should adhere to be managed on ONAP and meet ONAP's architectural principles.

As these are requirements, they should follow general expectations for what constitutes a good requirement:

- **Unambiguous** - It should have a single, objective interpretation. It should avoid subjective terms that would be open for misinterpretation. It should also include sufficient text or references to ensure how it should comply with a directive where applicable.
- **Concise** - An individual requirement should be specific about a single aspect, and strive to be as short as possible without sacrificing clarity. In general, a requirement should strive to be a single sentence where possible.
- **Verifiable** - Compliance with the requirement should be possible with some form of test, inspection, or demonstration.
- **Consistent** - Requirements must not introduce conflicting or contradictory guidance.
- **Feasible** - The requirement must be reasonably implementable by existing technology and practices. Forward looking guidance and aspirational goals are best suited for the Guidelines section of the document.
- **Observable** - The requirement should lend itself to externally observable behavior where ever possible. Guidance on internal implementation of meeting specific non-functional requirements should likely be included in guidance or as supplementary text.

Requirements that meet these criteria will have a specific format within the requirements document which will consist of a requirement statement and the metadata.

Requirement Statement Standards

Here is an example requirement statement:

*R-18725* - The VNF **MUST** handle the restart of a single VNFC instance without requiring all VNFC instances to be restarted.

Here are the additional standards the requirement must adhere to in the context of this project:

- The requirement must be uniquely numbered (ex R-XXXXX). Please refer to VNFRQTS How to Contribute for more information on how requirement numbers are assigned.
- The requirement must use RFC 2119 keywords (MUST | MUST NOT | SHOULD | SHOULD NOT | MAY), and these keywords must be in uppercase and in bold. In RST, bold is achieved by wrapping the text in double asterisks (ex: **MUST**)
- The requirement should generally start off with the subject of the requirement and refer to one of the following, and then further refine from there
  - VNF
  - PNF
  - VNF or PNF
  - VNF Provider
  - PNF Provider
  - VNF or PNF Provider
  - VNF Heat Package
VNF CSAR Package
PNF CSAR Package
VNF or PNF Package
VNF Documentation Package
PNF Documentation Package
VNF or PNF Documentation Package

Example: The VNF Heat Package's base module **MUST** contain...

- The requirement should apply only to a single aspect of its intended requirements target, and not combine multiple independent statements into a single requirement.

**Requirement Metadata Standards**

In addition to the statement itself, this project captures assorted metadata about the requirement about the requirement for a variety of purposes:

- Supporting use cases such as dependency tracking, searching, filtering, and reporting.
- Exporting requirements in a machine-readable format for use by other projects such as the VNF Validation Project.
- Generating different formats and outputs without duplicating requirement content (ex: appendices, tables, CSV files, etc.)
- Providing traceability within the document between requirements, test cases, and other items within the documents.

This project makes use of an Sphinx extension called *sphinxcontrib-needs* project to in support of these objectives. Requirements will still be maintained in the reStructuredText file, but they will be formatted according to work with the *sphinxcontrib-needs* extension.

Here is an example of a requirement that adheres to the standards.

You may use the VNFQTS Requirement Generation Tool to generate properly RST formatted requirements with unique ID numbers.

**Requirement Example**

```bash
.. req::
   :id: R-01334
   :keyword: MUST
   :target: VNF
   :links: R-01335

The VNF **MUST** conform to the NETCONF RFC 5717, Partial Lock Remote Procedure Call
```

The following table outlines the proposed standard metadata elements that will be associated with the requirements. This list may change over time.

**Table 2: Requirement Metadata**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Required vs. Optional</th>
<th>Data Type</th>
<th>Valid Values/Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Required</td>
<td>String</td>
<td>R-#####</td>
<td>The unique requirement ID for this requirement. See VNFQTS How to Contribute for more details. On a new requirement, this attribute can be left off and the tox -e docs or check.py script generate and ID and populate this field.</td>
</tr>
<tr>
<td>target</td>
<td>Required</td>
<td>String</td>
<td>VNF, PNF, VNF or PNF, VNF DOCUMENTATION PACKAGE, PNF DOCUMENTATION PACKAGE, VNF or PNF DOCUMENTATION PACKAGE, VNF PROVIDER, PNF PROVIDER, VNF or PNF PROVIDER, VNF CSAR PACKAGE, PNF CSAR PACKAGE, VNF or PNF CSAR PACKAGE, VNF HEAT PACKAGE</td>
<td>The component to which the requirement applies.</td>
</tr>
<tr>
<td>keyword</td>
<td>Required</td>
<td>String</td>
<td>MUST, MUST NOT, SHOULD, SHOULD NOT, MAY</td>
<td>The RFC 2119 keyword for the requirement</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>introduced</td>
<td>Optional String</td>
<td>lower case release name (ex: beijing, casablanca)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>updated</td>
<td>Optional String</td>
<td>lower case release name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>impacts</td>
<td>Optional List of String</td>
<td>Comma separated list of ONAP components (ex: so, sdc)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>validation_mode</td>
<td>Optional String</td>
<td>static, stand_alone, in_service</td>
<td></td>
<td></td>
</tr>
<tr>
<td>validated_by</td>
<td>Optional List of String</td>
<td>Comma separated list: vvp,vnfsdk, sdc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Use Case Standards**

The use case section of the document affords a way to provide a VNF Provider a VNF-centric view of certain use cases. It is not required that every use case supported by ONAP be documented in this section. Instead, key use cases that require discrete actions and coordination with the VNF can be described here to provide a clearer understanding of potentially multi-step complex interactions.

Formal requirements **must not** be defined in this section, but instead they should be defined in the appropriate requirements section of the document. Those requirements should detail out the any specific management, monitoring, or other capabilities the VNF must provide to support a given use case.

The use case section can be referenced by the requirements to provide additional context and better illustrate the interactions.

At this stage the format for use cases is not rigid, but is should comprise of the following elements:

- A sequence diagram that shows the API-level interactions between any users, ONAP components, and the VNF itself to effectively coordinate the use case. Multiple diagrams can be provided if appropriate.
- A description of the workflow
- An enumerated list of VNF impacts that detail out the specific concerns for the VNF provider and provide references to key requirements

**Test Case Standards**

TODO