Versioning data content in production

The following rules shall be adopted for versioning My Kanta Pages PHR data content in production. The rules are based on the FHIR specification (Read more about the versioning (hl7.org)). The rules described in this document apply to the R4 version and later versions.

Semantic version

The Semantic Versioning specification recommend that versions be represented in three parts using the form MAJOR.MINOR.PATCH where, for the kind of artefacts described in this specification:

- MAJOR version increments when incompatible changes are made to the artifact

- MINOR version increments functionality is added to an artefact in a backwards-compatible manner

- PATCH version increments when minor changes are made that don't impact on implementations of the artefact (e.g. documentation clarifications)

For the resources that have canonicalURL we are using semantic versioning (Read more about the semantic versioning (semver)).

StructureDefinitions, CodeSystems and ValueSets: Using .version, canonical URL and .status

The solution distinguishes between patch, minor and breaking changes made to StructureDefinitions, CodeSystems and ValueSets.

- Minor (and patch) changes can be made by updating the existing version of the data content and keeping the same canonical URL.

- A breaking change means that a new separate version of the data content shall be created, with a new major version and a new canonical URL. The previous version of the data content shall stay in production use for a transition time, after which it will retire and won't be usable in production anymore (no new resources can be stored using this retired data content).

Whether a change is a minor one or a breaking change? This will be assessed during the process of accepting the change to the national data content.

The following rules regarding .version, .url and .status SHALL apply in the Finnish PHR data content:

.version

- A completely new StructureDefinition/CodeSystem/ValueSet SHALL have a version of 0.X.Y during development and testing phases, including during the approval testing of the application.

- In the first production deployment, the version SHALL be set to 1.0.0.

- If a patch is made after the first production deployment, the existing version can be updated and the minor version number SHALL be changed (e.g. 1.0.X).

- If a minor change is made after the first production deployment, the existing version can be updated and the minor version number SHALL be changed (e.g. 1.X.0).

- If a breaking change is made, a new version SHALL be created and the major version number SHALL be changed (e.g. 2.0.0), along with the canonical URL (see the following section .url).
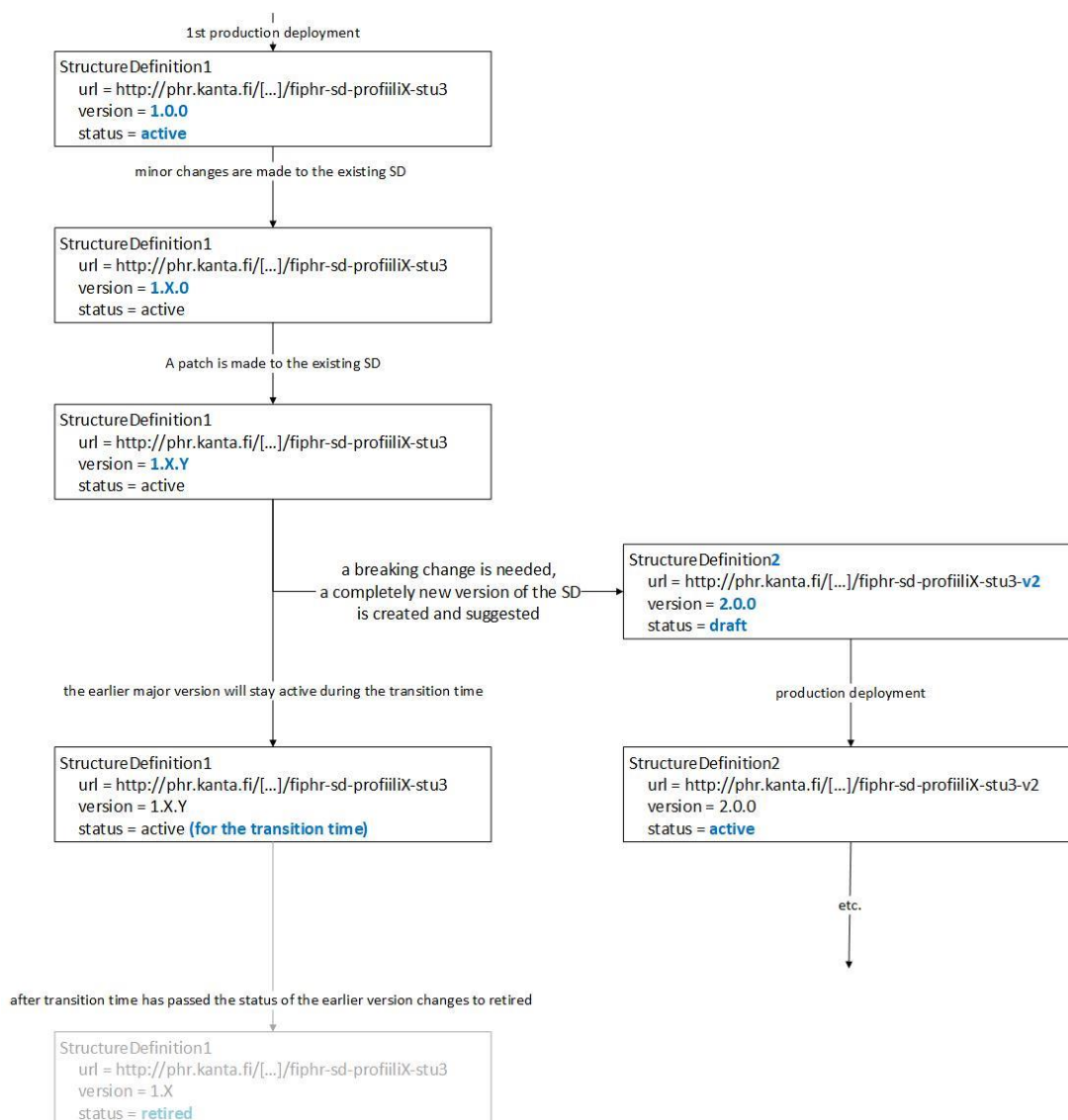
.url

- NB! The first production version (major version 1.0.0) SHALL have a canonical URL without the major version included in the URL. For example http://phr.kanta.fi/StructureDefinition/fiphr-bloodpressure-stu3. All subsequent versions with only minor changes (1.X.Y) SHALL maintain the same URL.

- All subsequent major versions (2.0.0, 3.0.0, 4.0.0, etc.) and their minor versions (2.1.0, 3.4.0, 4.2.0, etc.) SHALL include the major version of the profile in the URL itself. For example:
http://phr.kanta.fi/StructureDefinition/fiphr-bloodpressure-stu3-v2,
http://phr.kanta.fi/StructureDefinition/fiphr-bloodpressure-stu3-v3,
http://phr.kanta.fi/StructureDefinition/fiphr-bloodpressure-stu3-v4.

.status

- A completely new StructureDefinition/CodeSystem/ValueSet SHALL have the status 'draft' during development and testing phases, including during the approval testing of the application.

- When deployed in production, the status SHALL be 'active'.

- If a new version is created (a breaking change has been made), the status of the earlier version SHALL be set to 'retired' after a predefined transition time. The transition time will be available at Simplifier and in the implementation guide. (**NB!** Resource instances can refer only to profiles with status 'active' in the production environment. The server will reject instances that refer to profiles with another status.)

Questionnaires: Using version, url and status

The same kind of rules are used with Questionnaires. Minor changes can be made by updating the existing version and changing only the minor version of the Questionnaire. Breaking changes mean that a new version of the Questionnaire shall be created, and the major version as well as the Questionnaire.url shall change.