FHIR IG Review

The following checklist can be used as a template for performing an Implementation Guide review – it guides the overall process for the review.

This slide deck gives details for reviewing FHIR IGs: 2019-09 FHIR IG Review.pptx

To use this checklist, copy the content to a Word Document for each IG Review.

Step #1: Orientation / Conceptual
- Is the purpose of the IG clearly explained?
- Have roles and use cases been defined?
- Does the IG have the appropriate dependencies? Is it reinventing existing content?
- Do the artifacts fit appropriately into the defined scope?
- Check stated Realm
- Check kind of IG – what expectations arise?

Step #2: Orientation / Technical
- Read note to balloters (if present)
- Check history notes
- Check qa.html

Step #3: Orientation / Approach
- Check kinds of exchanges
- APIs
- Document
- Messaging (o Check messaging protocol)
- Make list of kinds of information exchanged
- Check consistent with realm core expectations
- Make list of actors described in system
- Characterize as Producer | Consumer | Repository

Step #4: Capability Statement
- Check resources exposed. For each resource
  - Check Interactions
  - Check Search Parameters
  - Modifiers
  - Chaining
  - Combinations
- Conformance expectations clear
- Check System Profile
- Check Use case Profiles
- Check Global Profiles

Step #5: Profiles
Check resources/profiles exposed. For each resource
- Check Text Summary – human to human
- Check Differential as summary (consistent with text summary?)
- Check what other related profiles exist in other IGs
- For each element in the differential:
  - Check definitions – changed? Needs to be changed?
  - Check any mappings added
  - Check cardinality - elements should be 0..0 only if the inclusion of that element would be an error
  - Check must support (and check it’s meaning is defined)
  - Check Terminology Binding – extensible and preferred binding pose challenges for searching
- For each sliced element (including extensions), assure the generic short and definition have been replaced with real definitions
- Any orphan profiles?

Step #6: Extensions
Any extensions? For each extension:
• Is there a standard extension that could be used?
• Has the context of use been set (associating the extension with the type of resource it can be used with)
• If a simple extension, has the data type of the value been appropriately constrained? Has the extension’s extension been set to 0..0?
• Has the short and definition of the extension been set? (These are appear in the profile, so they are important)

Step #7: Value Sets / Code Systems

• Check Copyright (externally sourced content)
• Check Versioning
• For code systems: something that needed to be defined?

Step #8: Examples

• Check that all examples validate against profiles
• Check that examples have all information required for the use case
• Where there are slices, examples should populate every slice. Check that the QA report to see if the slices were uniquely distinguishable.

Step #9: Specific Requirements

• For each:
  • Defined / described?
  • Requirements described?
  • Do you agree?
• Security
• Error Handling
  • Specifying correct behaviour for operational failures
  • Mechanisms for handling erroneous data
• Audit / Provenance
• Consent / Privacy
• Test Cases / Conformance testing support
• Safety Issues (check)