Coding Book

Arjan Dekker & Jesse Maarleveld

March 2023

1 Preface: Definitions & Terminology

1.1 Basic Definitions

The basic definitions used for Existence, Executive, and Property issues are based on the following paper by Kruchten:

Kruchten, Philippe. 'An Ontology of Architectural Design Decisions in Software Intensive Systems'. 2nd Groningen Workshop on Software Variability, 2004.

For all issue labelling, the basic definitions outlined in that paper should be kept in mind. The rules in this coding book mostly define and substantiate how we interpreted these definitions and how issues should be assigned to them.

1.2 Significant Enough Changes/Significant Effort etc

Often, if an issue is architectural or not depends on the effort involved. What we mean by this, is that something is hard to change. For instance, the addition of a new component is architectural if making large changes later is difficult. Hence, the key question we ask if necessary is "Would it be difficult to change the implementation?".

The following issues were considered existence:

- An addition of a new partition strategy to Cassandra: CASSANDRA-8866
- Blacklisting of ill-performing nodes in Hadoop: HDFS-289
- Encryption of SSTables in Cassandra: CASSANDRA-9633
- Refactoring to expose an interface: HADOOP-15038
- CASSANDRA-5283
- YARN-4619
- CASSANDRA-4011
- CASSANDRA-14213

The following issues were considered architectural, but not existence:

- Changes to memory handling in HDFS:
- Moving some functionality to a separate thread: HDFS-16016
- Throttling in HDFS: HDFS-9723
- Optimisation of small repair streams (seems major, but not enough conclusive evidence to warrant existence): CASSANDRA-13290
- Better load distribution among threats: CASSANDRA-4292
- Umbrella for OS-level optimizations: HADOOP-7714
- CASSANDRA-12104
- /CASSANDRA-13291

2 Guidelines for Certain Issue Types

- User requests should be evaluated based on the types of decisions discussed in the request.

3 Architectural

ARCH-1 Use-cases/requirements are architectural (HADOOP-9659)

ARCH-2 Umbrella tasks that contain many links to subtasks are architectural. Depending on the focus of the task, it can be existence, property and/or executive (HADOOP-15977 , HADOOP-1771777)

4 Existence

EXI-1 Large behavioural changes to commands/operations are existence (HADOOP-15845 , YARN-613)

EXI-2 The introduction of tactics is existence if the implementation requires the addition/modification/removal of components or behaviour between components (HADOOP-18458)

EXI-3 The addition of major new functionality is existence, even if the implementation might be relatively straightforward (CASSANDRA-17059)

EXI-4 Refactoring may be existence if done on a scale where many places/components are affected (CASSANDRA-8609 , HDFS-6315)

EXI-5 Property issues which motivate the addition/removal/modification of existing components or interactions are often also existence, given that the changes are sufficiently large (CASSANDRA-8609 , YARN-8673)

EXI-6 New features with significant implementation challenges/non-trivial implementation/large implementation effort are often existence (CASSANDRA-1339)

EXI-7 Major restructuring/sub-project creation is existence (CASSANDRA-1228)

EXI-8 The implementation of large utility tools is existence (HDFS-8968)

EXI-9 Parallelism is often existence, unless the implementation is trivial (CASSANDRA-2901)

EXI-10 Changes to the interaction between components is existence (HDFS-7607 , CASSANDRA-8345 , CASSANDRA-6752 , CASSANDRA-4761)

EXI-11 Changes spanning multiple components are existence (CASSANDRA-9633)

EXI-12 Vague or small change descriptions are still existence, if components and/or protocols are mentioned (MAPREDUCE-5189 , HDFS-2181 , YARN-3409)

EXI-13 Large non-trivial code changes (in patch) are existence (YARN-6620 , YARN-3998)

EXI-14 Issues without patch if changes to components and/or protocols are described (HDFS-6658)

NON-EXI-1 Trivial code movement is not architectural (YARN-2107), unless a proper architectural reason is specified (HADOOP-9649), or the code being moved is large enough (HDFS-12259).

NON-EXI-2 Small changes to internal interfaces are not existence (CASSANDRA-6248)

NON-EXI-3 Very small interface additions (e.g. “hookds”) are not existence (CASSANDRA-5545)

NON-EXI-4 Very vague change descriptions not mentioning components and/or protocls are not existence (HADOOP-1986)

5 Executive

NB: Protocols, tools, hardware and interfaces can be considered a technology

EXEC-1 Issue deals with external technologies (e.g. support for technology Y, CASSANDRA-1193) are executive, unless the changes are minor (CASSANDRA-11519). However, in case much effort is needed to analyse the effects of such changes is large, the issue can also be considered executive (CASSANDRA-3031)

EXEC-2 Dependency additions/upgrades/removals are executive (HADOOP-17947 , CASSANDRA-13291)

EXEC-3 Creation of utility tools for development (e.g. benchmarking) is executive (HDFS-8968 , HADOOP-12725)

EXEC-4 External code contributions (e.g. merging existing projects) is executive (HADOOP-2878 , YARN-2670)

EXEC-5 Requests or proposals from companies or other users to support technologies is executive (CASSANDRA-11703 , HADOOP-9484)

NON-EXEC-1 Code to work around some flaw in a dependency, external tool etc. is not executive (HADOOP-17597)

NON-EXEC-2 User requests are not executive (HADOOP-17268)

6 Property

PROP-1 Quality attribute enhancements are often property, if applied on a high-enough level (e.g. parallelism, HTTPS connection re-use), and provided that the improvement is one of the main goals of the issue (not mentioned off-handedly) (TAJO-1970). Note that improving quality attributes is sometimes implicitly described by words such as refactoring.

PROP-2 Discussions (as in, discussion without implementation) on quality attributes (e.g. security) are property if the change being is major enough (e.g. implementation would be considered existence), even if in the end no implementation is done or proposed. Investigations also fall under this category. (CASSANDRA-7129 , CASSANDRA-7045)

PROP-3 Refactoring might be property if done on a system-level (i.e. implication in many places), and/or expressed in terms of quality attributes (CASSANDRA-8609 , HDFS-2353)

PROP-4 The implementation of tools/utilities meant to measure (and later improve) quality attributes is property (HDFS-8968)

PROP-5 Parallelism is in general property if done for performance, even if the implementation is simple (CASSANDRA-2901)

PROP-6 Umbrella issues focusing on the improvement of quality attributes are property, even if no clear decisions are made in the issue itself. (MAPREDUCE-563 , YARN-2745)

PROP-7 Mitigation of security vulnerabilities is property (CASSANDRA-15121 , TAJO-1214), as well as changing security settings (HADOOP-16779)

PROP-8 Reporting a problem w.r.t. a quality attribute is property (HADOOP-55)

PROP-9 Usability issues are generally property (HDFS-2849)

NON-PROP-1 A single interface or API being X (where X is a quality attribute) is not immediately property (CASSANDRA-754)

NON-PROP-2 Backwards-compatibility related issues are generally not property (CASSANDRA-10990)

NON-PROP-3 Quality attribute improvement or discussion is not property if the change is trivial (e.g. configuration change) (CASSANDRA-14678)

NON-PROP-4 Indirect improvements ("this change would enable the development of faster algorithms") are not property (TAJO-196)

7 Non-Architectural

NON-ARCH-1 Not any of the above

NON-ARCH-2 Small changes are non-architectural

NON-ARCH-3 Configuration changes are not architectural, even if this may lead to the ability to use more/different technologies (HADOOP-14417)

NON-ARCH-4 The addition of small utility tools which do not provide or require novel functionality, is not architectural (SOLR-11179)

NON-ARCH-5 License/legal use clarification of source code is not architectural (CLOUDSTACK-161)

NON-ARCH-6 Small changes in/additions of error handling are not architectural, even though it may lead to new behaviour in case the error is encountered (SOLR-3505 , CASSANDRA-18042)

NON-ARCH-7 Small utility additions/new features need not be architectural, even though they introduce new parts of a public interface (SOLR-10485 , SOLR-11338)

NON-ARCH-8 Tests/QA are non architectural CLOUDSTACK-1000)

NON-ARCH-9 Formalities/release guidelines are not architectural (JSPWIKI-559)

NON-ARCH-10 (Detailed) implementation issues without design discussion are not architectural (HDFS-5616)

NON-ARCH-11 Coding standard issues (e.g. “not compliant with SQL syntax”) are not architectural (TAJO-1970)

NON-ARCH-12 Minor behavioural clarifications are not architectural (CASSANDRA-9131)

NON-ARCH-13 Umbrella issues without their own decisions or implementation details are non-architectural, even if the large effort being coordinated requires substantial architectural changes.

NON-ARCH-14 Small code refactorings are non-architectural.

NON-ARCH-15 Small bugs/bug reports are non-architectural, given that the solution is a relatively straightforward code change.

NON-ARCH-16 “TODO Lists” are non-architectural (HADOOP-5064)