

Table A: Details of “Why” expression categories

Category	Subcategory	Explanation	Statistic	Example
Describe Issue	Describe error scenario	Describe where and how an error occurs directly	50(19.8%)	“As-is it throws unchecked exceptions on unexpected charsets. This is a problem because it can cause a misbehaving webserver to crash the client.”
	Introduce issue report	Describe issues mainly by citing errors/defects or warnings from quality assurance tools	11(4.4%)	“remove warnings found by errorprone. [...] Call-Test.java:2056: warning: [UnnecessaryParentheses] Unnecessary use of grouping parentheses [...]”
	Describe shortcoming	Highlight the shortcomings or weaknesses in the current implementation, which is the motivation to make this commit	9(3.6%)	“I’m unhappy with java.io: No timeouts [...] Features like mark/reset and available() are clumsy [...]”
Illustrate Requirement	Usage need	Describe specific needs or requirements of users in software development	11(4.4%)	“Error-prone only works on pre-12 at the moment and we need this configuration to apply for all JDKs”
	Out of date	Indicate the obsolescence of some features or code	24(9.5%)	“Remove outdated key. The ‘spring.metrics.export.redis.aggregate-key-pattern’ is no longer defined but was still referenced in the documentation.”
	Runtime or development environment change	Indicate an adaptation to the current code development or runtime environment	11(4.4%)	“API has changed, fixing the example”
Describe Objective	To fix defects	Make explicit that the purpose of a change is to resolve a defect	9(3.6%)	“Fix concurrent problem of zookeeper configcenter, wait to start until cache being fully populated”
	To make improvements	Describe the improvement of an author’s code implementation, e.g. functional improvement or nonfunctional goals directly	13(5.2%)	“AndroidLog: Added [...] methods for easier subclassing.”
	Conventions and standards	Describe or refer to any conventions or standards that are the basis for a change, thus proving the sound rationale of the change	15(6.0%)	“it is common to add tests to the same package as the class under test”
Imply Necessity	Relation to prior commits	Explain the relationship between the current commit and any commits that were already merged into the repository	9(3.6%)	“the code was changed by commit <other url> but unfortunately the test was not part of the commit”
	Relation to an implemented feature	Relate to an existing feature, and this relationship provides the context of the new commit	9(3.6%)	“a short step on the road to HTTP body format agnostic support.”
	Improvements and benefits	Describe the need for a change by explaining the improvements and benefits that the change will bring indirectly	39(15.5%)	“Use custom exception type [...]. Since we omit the stack trace, this more clearly indicates the source being from Retrofit’s mock behavior”
Missing Why	Test cases	Involve adding test cases to the repository	4(1.6%)	“tests for canceling async requests.”
	Typographic fixes	Involve the correction of typographic errors	7(2.8%)	“fix typo a->an”
	Text file changes	Involve changes made only in text files	13(5.2%)	Changes in special function files, such as “ChangeLog”, “README”
	Annotation changes	Refer to the motivation to modify the content of annotations	5(2.0%)	“add docs about null responses”
	Code refactoring	Involve refactoring and formatting of code	15(6.0%)	“Polish pom.xml. Apply consistent formatting, drop JDK 8 support and cleanup repo [...]”
	Version management	Include changes that involve version management, such as the updating of version numbers	5(2.0%)	“prepare version 2.8.1”

Table B: Details of “What” expression categories

Category	Subcategory	Explanation	Statistic	Example
Summarize Code Object Change	Characteristics of changes	Highlight the characteristics of the current code change and comparing them with other implementation methods to summarize the code changes	13(5.2%)	“attempt at a 3rd I/O interface [...] inspired by InputStream and OutputStream, but using growing buffers instead of byte arrays as the core data container”
	Object of change	Summarize the changes from the point of view of the code objects	143(56.8%)	“remove creation of ‘fat’ jar...”
	Change list	Indicate changes of several code objects, involving one or more source files	6(2.4%)	“this commit removes the following deprecated properties: * ‘server.connection-timeout’ * ‘server.use-forward-headers’ [...]”
	Contrast before and after	Contrast the state of code objects before and after changes	16(6.4%)	“rename HeldCertificate.Builder.issuedBy() to signedBy()”
Describe Implementation Principle	-	Represent commit messages describing technical principles underpinning the changes	6(2.4%)	“SslContextBuilder was using InetAddress.getByName(null) [...] On Android, null returns IPv6 loopback, which has the name ‘ip6-localhost’”
Illustrate Function	-	Summarize and explain code changes from a functional perspective	65(25.8%)	“Rename preferred-mapper property so its clear it only applies to fJSON”
Missing What	-	Refer to commit messages that lack any specification of what was changed	19(7.5%)	“fix typo”