

Name	Files	References
Code Review Process	25	237
Chunking	1	1
Content	0	0
agreement on process	1	1
good git hygiene	6	7
allow for retrospective understanding	1	1
loading PR progress	1	1
need annotation	3	7
annotating in description	4	5
review limited	3	4
splitting in meaningful units	6	11
story-telling	5	5
lacks overview	6	8
introduce bias	2	2
need to check consistency	2	3
lowering cognitive load	6	7
narrowing scope	2	7
focus on a unit of change	2	4
focus on specific intention	1	1
managable review size	4	4
tracks requirements	2	2
Strategy	0	0
Comment-by-comment	4	13
comment follow up	3	15
comment becomes irrelevant	1	2
lacking cues to annotate the resolution	1	4
clarification with author	1	2
require follow-up	2	3
resolving comments	1	1
need to have comment reference in the code	2	10

neet to track patches	2	7
re-review specific	3	3
Commit-by-commit	11	18
File-by-file	1	1
following files modified in a patch	1	1
Issue-by-issue	1	4
avoiding conflicts between related PRs	1	1
understanding relationships between the issues	1	2
Unit based	2	3
Functional chunking	1	1
Requirements based	1	2
Individual scope	2	2
Scoping down	1	1
Tests	5	8
document what the author is doing	2	2
tests first	1	1
document intention	1	1
verify implementation and documentation	1	1
verify test quality	1	2
coverage	2	2
reliability	1	1
work well	2	2
TODO based	3	5
Completing remaining todos	3	3
Complex reviews	2	2
Issues that cannot address immediatelly	2	2
Issues that might be resolved through going through the PR	1	1
Storage for general points for the review	1	1
Context building	22	94
Assigning review strategy	1	1
provided by author	2	2

self-provided	6	7
Strategy selection	8	19
Creating review goals	1	1
Calls for action	1	1
Issues to check	2	6
Opens questions	1	2
Establishing scope	0	0
Establishing scope of understanding	6	6
Establishing state of progress of the PR	8	15
Scope of other reviewers	2	4
Evaluating context quality	7	7
Gathering relevant information sources	11	17
Improves review efficiency	1	1
Building mental model	0	0
complexity estimation	3	3
expectations	4	4
Gaining annotation	7	10
Gaining change overview	7	9
Gaining specification	15	36
ideal solution	2	2
Implementation hints	8	17
rationale	1	1
increents and improvements to the SS	2	2
comparison against evaluation criteria	2	4
speed	1	1
Interaction of others with the PR	0	0
Loading discussion content	8	12
allows alignment with opinions of others	1	1
authors' clarifications	1	2
brings up issues considered by others	3	7
identifying outdated issues	1	1

introduces information sources	4	5
potential bias	1	3
proposals of others	1	1
relavant standards and processes	2	2
reminds own activity	1	2
signifies potential unnecessary review effort	3	4
Who is involved	2	4
Clarifying who should be involved	2	2
Finding the decisions of other reviewers	4	6
Loading PR history	7	20
Prepare the environment to perform the review and testing	1	3
Decision	23	50
Checked everything in their scope	5	5
able to provide constructive feedback	1	1
check all changes again	2	2
checked everything	5	7
did a thorough review	1	1
fulfills acceptance criteria	1	1
no pending notes	2	2
reached the desired understanding	2	3
wrapped up conversation	3	3
Giving the final verdict	2	2
accept	0	0
all issues resolved	4	4
confident in the change	1	1
feature is consistent	1	1
fulfills acceptance criteria	1	1
improvements do not stop	1	1
integration and testing OK	6	8
last changes on the responsibility of the author	1	1
rationale flly clear	2	2

satisfied with all changes	4	4
comment	0	0
in progress	2	3
draft	1	2
in waiting for another factors	1	1
Information missing to fully evaluate	1	1
self-assigned review	1	1
consideration of verdicts of others	2	2
request changes	0	0
conflicting acceptance criteria	0	0
feedback needs to be implemented before continuing	2	2
issues to fix	8	9
confident changes are needed	1	1
stopper issue	1	1
rationale incomplete	2	2
writing review message	2	2
considering added value	1	1
contextualizing feedback	7	8
expectations for acceptance	2	2
high level questions and concerns	3	3
motivating	7	7
skipping review message	1	1
stating their review scopee	2	2
documenting review results	1	1
Difficulty-based	0	0
Core based	0	0
Core-based	5	6
identifying the core	1	1
addresses the crucial part of the specification	1	1
base element that is then used throughout the review	1	2
identify what part of the system is changing	1	1

known file that tends to be the base part of the system	1	1
main logic	3	3
tests if complex	1	1
the biggest change in the PR	2	3
more difficult to review	1	1
Expects to spend most time and energy	1	1
understands the implementayion of the core parts	1	1
Data and execution flow based	4	5
architecture understanding	2	2
complexity build-up	1	1
core initiated	2	2
Easy first	0	0
Easy to review	11	23
code quality independent	0	0
files dependent on expected future changes	1	1
unnecessary removed files	2	2
visual front-end changes	1	2
not author written code	0	0
auto-generated files	3	4
binary files	1	1
boiler plate	1	2
not production critical	0	0
documentation	1	1
tests	1	1
small changes	1	1
renames	2	4
small logic	1	1
strings	1	1
upgrades	1	1
usage of the main implemented element	1	1
white space changes	1	1

structurally constant changes	1	2
declarative changes	2	4
configurations	1	1
schemas	1	1
established patterns in the code base	1	1
experience	4	6
Hard to review	10	19
complex changes	1	1
chained events	1	1
Logic	9	10
potential for substantial issues	2	2
core	2	3
Main PR purpose implementation	2	2
Large changes	2	2
Requires effort and energy	1	2
Need to read line by line	1	1
Needs to stop to analyze	3	4
Discussion management	6	12
at the beginning	0	0
checking their own past activity	1	1
creating todos for the review	1	2
faster reviewing	1	1
generating ideas on how the issues could/should be resolved	1	1
knowledge of open issues	1	1
loading context	2	2
next steps of the PR	1	1
sets what is needed to focus on	1	1
shows progress on the issues of the PR	2	3
shows whether review is needed	1	2
Untitled	0	0
at the end	0	0

fresh state of mind	2	3
not setting expectations and biases	2	3
overview of open issues	1	1
wrap-up of conversations	1	1
avoid duplication of comments	3	3
check for related comments	1	1
checks whether their opinion did not change	1	1
deleting redundant comments	2	2
not addressed conversations	2	2
resolving tentative comments	1	2
role of discussion	0	0
adding missing perspectives	2	2
finding alignment	2	2
identifying issues they have opinions about	2	2
learning what they missed	2	3
shows what others did and think	2	3
Linear	2	2
Content	0	0
Managable review size	0	0
More time efficient than jumping	1	1
Poor git hygiene	1	1
Strategy	0	0
One-view	5	6
Checking impact across code base	1	1
Checking implementation nearly not needed	3	4
Might be connected to more complex code	2	2
Nothing to see	1	1
Scanning	8	11
Checking if something was missed	1	1
Detecting areas to get back to	1	1
Estimate change complexity	1	1



Identifying core of the change	1	1
Overview of changed files	4	4
Overview of execution flow	1	1
Overview of what was done and catch next steps	1	2
Selecting review starting point	1	1
Top-to-bottom	16	20
Consistency	1	3
No need to ensure validity of understanding	1	1
Not complex change	5	5
Merging	2	2
Micro strategies	3	3
Comprehending	0	0
Comparing	4	6
Distant context	3	4
Element comprehension	1	2
Hypothesis confirmation	4	6
Implementation details	3	4
Implications comprehension	1	1
Delayed comprehension closure	1	1
Retrospective commenting	6	6
Revisiting	4	5
Locating	0	0
Change locating	3	5
Comment location	2	5
Usage locating	6	9
Value element location	3	3
Pair review	2	8
Complex change	1	1
Difficult to build the mental model	1	1
Mentoring	1	1
Need for author's clarification	0	0

Poor PR quality	1	1
Poor PR structure	1	1
Substantial issues	1	1
Need for a lot of negative feedback	1	3
Private environment	1	1
Pre-context	19	52
Lowers need for context-building	3	5
Broadens knowledge base	0	0
Interference with other PRs	1	1
Already reviewed tests	1	2
Touching PRs	0	0
deducting changes to this or touching PRs	1	1
identifying unnecessary parts of the PR	1	1
know which affecting changes were merged already	1	2
knows implications for rebase	1	1
potential to simplify other PRs	1	1
Previous attempts at the PR	0	0
Topic activation	0	0
Expecting the PR	3	4
Hearing from other reviewers about it	2	3
Pre-builds the mental model	1	1
Alternatives available	1	1
Expected	1	1
Implementation details	1	1
Anticipation of issues to comment on	1	1
First look	1	1
pre-agreed implementation	1	1
pre-alignment	1	1
agreement on direction of the PR	5	6
efficiency boost	2	2
potential bias	1	2

questions answered prior to review	1	1
sets expectations	4	6
shared understanding of the implementation	1	1
re-review	4	4
issues already addressed by the author	1	1
previous comments set the expectations	1	1
review as mental model rebuilding and updating	1	1
steps taken to address issues	1	1
Specification modelling	0	0
Feature specification	3	4
Cross-team alignment	2	2
Interaction with the issue and PR	0	0
Issue definition and discussion	4	5
Open decisions	1	1
Sets the scope	0	0
Dynamicity of the code base	1	1
Status of the PR and related documents	0	0
documentation not ready	1	1
Team approval before maintainer review	1	1
Urgency	0	0
Customer requests	1	2
Move fast	1	1
Pressure to merge	3	3
Preparing the infrastructure	0	0
Changes pulled	0	0
Commit squash	1	1
Strategy affected by complexity	1	2
Testing	5	11
CICD approval	3	3
interpreting fails	1	3
re-running	1	1

Hands-on testing	0	0
Behaviour observation	0	0
Build testing	0	0
Interface testing	1	1
Local environment testing	3	5
Comprehension scope	24	151
Comprehension completeness	0	0
Comprehension beyond the PR	0	0
contextualizing with other PRs	1	2
Focused review	0	0
Expertise-based reviewing	2	2
Component	1	1
Design	2	2
Language	2	2
Product	1	1
Responsibility	4	9
Higher level	2	2
Higher logic correct rather than reaching full understanding	3	3
higher level understanding	2	4
Rationale and direction	1	1
Focused on pre-agreed direction of implementation	1	1
reasons for change	1	1
Requirements	1	1
Technical details less relevant to check	1	1
Switching hats for different reviews	2	2
Topical review	0	0
Code quality	3	5
correctness of implementation	3	3
issues and defects	1	1
crucial parts of the repository health	1	1
formatting	1	1

legacy code	0	0
Logic	1	1
Naming	2	2
not breaking	2	2
readability, understandability and maintainability	2	2
reviewing for maintainability and future comprehension	1	1
reliability	1	1
risk of creating undesirable consequences	1	1
usability	1	2
well documented	3	3
Design	1	2
Specific issues to check	1	1
review checklist	1	1
Full	3	4
added, removed, changed	2	2
Do a thorough review	1	1
task completeness	1	1
Fulfilling acceptance criteria	1	1
Partial review	0	0
Decomposing complex changes	1	2
narrowing scope within the review process by chunking	1	1
need for step back	1	1
Splitting review in phases	1	2
reviewability in isolation	1	1
Giving more feedback brings little added value	2	2
Feedback needs to be implemented before continuing reviews	1	1
Given feedback will substantially change the shape of the PR	2	2
Substantial issues to fix	1	1
Narrowing through code review iterations	1	1
Addressing of comments	2	2
re-review	1	1

increment from previous state of the SS or PR	1	1
irrelevant code and comments	1	2
Need to involve the author in person	1	1
need to sync with the author to continue	1	1
Shallow review	0	0
Basic understanding, but makes sense	1	1
Confidence that it will work	1	1
Substituted by testing	4	5
Substituted by automation and tooling	4	8
Trust in the system's quality	1	1
Consistency in the code base	1	1
Incomplete understanding	1	1
Smells of issues	1	1
Factors	0	0
Author	0	0
Closeness	0	0
low knowledge of the author requires more specific feedback	1	1
Experience	0	0
Junior	0	0
Learning	0	0
learning needs	1	1
Learning of the author	1	4
novice get a lot of feedback for learning	2	3
submissions by juniors or volunteers	0	0
need more attention	1	1
Need to shape their style of code	0	0
Senior	0	0
project lead as the responsible understander	3	3
Need to keep the author motivated	1	1
Code change	0	0
Amount of available constructive feedback	3	3

Complexity	0	0
Ease of reviewing	2	2
similarity with other components	1	3
Simple change requires less thorough inspection	5	5
can be understood in entirety	2	2
Does not require much understanding	3	4
Number of involved components	1	1
Criticality	3	4
Criticality of the system	2	2
Deleted without affecting the new functionality	1	2
just visual stuff	1	1
performance	1	1
production code	1	2
tests verification more relaxed	1	1
security sensitive	3	3
User-affecting	3	4
Stage of the PR	0	0
Draft	1	4
Far from done	1	4
Expectation of next review round	4	7
Can get feedback from the author	0	0
The code will change	4	10
merge not possible atm	1	1
First look	1	1
In progress	0	0
PR moving in a correct direction	1	1
PR progressing somewhere	1	1
Need to merge	1	1
need to progress	1	1
non-blocking issues	3	4
Time pressure	0	0

Urgency	3	4
Cummulative reviewing capability	3	3
Added value in ones review	4	6
Necessity of additional reviewer	2	2
Second pair of eyes	2	2
encouragement to build the needed knowledge	1	1
Availability of other reviewers	1	1
Complementing expertise	4	7
trust in others expertise	5	11
Possibility to ask the author	2	4
nudging the author to reach full understanding	2	3
Project	0	0
Criticallity	1	1
Dependency of others on the component	1	1
Established practices	1	1
Testing culture	1	1
Thoroughness culture	1	1
Reviewer	0	0
Length of the review queue	1	1
Limited cognitive resources	0	0
cognitive resources	1	2
limited working memory	1	2
Fatigue	2	2
Topic burnout	1	1
Personality	0	0
Conscienciousness	1	1
Project tenure	1	1
Time limitations	3	5
time limitation	2	2
More effective time usage	3	4
Information Sources	21	77



Code base	2	2
Documentation	2	2
IDE	6	10
Tests	3	3
Code Review Application	5	10
Issue tracking	11	18
Local software	1	2
Specialized dev tools	1	2
Browser developer tools	0	0
Prototyping Design tool	1	3
Terminal	5	9
Meetings	9	14
1-on-1 meetings	2	2
Online sources	0	0
Blogposts	2	2
ChatGPT	1	1
Grammarly integration	1	2
Language documentation	1	1
Search engines	3	5
PR	0	0
CICD	8	12
Code Diff	1	1
Code Context	3	6
Docstrings	3	6
File path	1	1
Left diff	4	9
File tree	6	12
Labels	0	0
Draft	1	2
PR commits	8	14
PR Conversation	13	24

PR Description	18	22
PR Patches	3	4
Review size	6	7
LOC changed in a file	1	1
Review complexity	2	2
Verdicts	4	4
Related PRs and commits	9	11
Additional review	2	3
TODO notes	5	19
Knowledge base	20	117
Domain knowledge	2	2
Git hygiene	1	1
Programming plans	6	11
Coding alternatives	1	1
Programming standards	4	12
Formatting	2	2
The 'correct' implementation	4	4
Experience	2	2
Area of expertise	2	3
Libraries	0	0
Own problem-solving strategies	1	1
Performing code reviews	1	1
Project history	2	3
Usage of tools	1	1
Wiith the code base	2	2
Interpersonal	0	0
Colleague's coding and working style	2	2
Colleague's expertise	4	6
Communication style of others	4	5
Growth needs of the colleagues	4	5
Mental model of the code base	1	1

Code authors and modifiers	1	1
Coding patterns	4	4
Coding standards	3	3
Team standards	2	3
Implementation of code elements	5	6
Overview of changes happening to the codebase	4	5
author's activity on the PR	1	1
Knowledge of issues	2	2
Knowledge of related meetings and their outcomes	1	1
Previous review	0	0
Previous comments	2	3
Previous quality	1	1
Previous review on related changes	1	2
System architecture	2	2
Team processes	2	3
Culture	1	1
Process maturity and shortcomings	1	1
State of project and PR	3	3
Usage	0	0
Efficient and fast	2	2
Availability of responses and suggestions	9	14
Bigger context in memory lowers need for context switching	3	3
No need to understand related artifacts in the review	1	1
Creates expectations	1	1
Code quality	2	3
Needed reviewers	1	1
Preliminary PR mental model	0	0
Implementation	1	1
PR Goals	1	2
Standards to follow	0	0
Detect level of improvement	1	1

Mental model of the code base does not need to be build	1	2
Surprising, novel changes take longer	1	1
Usual information sources	1	1
Efficiency knowledge	0	0
Common issues in the code base	2	2
Issues encountered	1	2
Obvious defects	2	2
Typical issues introduced by a colleague	1	1
Typical issues related to specific solutions	3	3
Identifying issues	1	1
Acknowledging elegant solutions	1	2
Deviation from plans and standards	6	12
Implementation relevant knowledge of programming patterns	2	4
PR conflict with code base model	3	4
Architecture conflict	1	1
Deviation from codebase standards	2	2
consistency	3	4
Incorrect usage of code elements	1	1
Inferring undesirable behaviour of the PR	4	4
Extending mental model of the code base	1	1
Scoping the review	3	6
Mental Model	13	67
Activities	0	0
Commenting	21	71
Identifying defects	3	4
Tracking review progress	1	1
Mental list	0	0
TODO notes	0	0
Viewed files	0	0
Models	6	8
MMs	0	0

actual	14	42
Annotation	18	94
Hourglass model	0	0
Bottom-up	0	0
Makes sense	4	5
parallel to top-down	1	1
What is the implementation trying to achieve	2	3
Focus	2	2
Top-down	0	0
comment to fix	1	1
PR description to overall implementation	3	4
purpose of elements	1	2
through commit messages	2	2
through docstrings	2	2
through expected usage	1	1
through naming	2	2
through the file tree	1	1
vision for the SS	1	1
changing through review iterations	1	1
Implementation	14	52
the deepest level of understanding	1	1
just the technical thing	1	1
What changed	5	6
What is the code doing	4	5
Why is it as it is	1	1
benefits of an implementation approach]	1	1
correct place of implementation	1	1
Reasoning	2	3
Specifications	15	45
forming expectations for evaluation	0	0
checklist of tasks	1	1

devidable review units	2	2
Expectations of implementation	2	3
expected GUI form	2	3
expected program behaviour	1	1
expected usage bu the user	1	2
problem understanding	0	0
problem being solved	7	11
reason for change	6	7
relationships between problems being solved	1	1
scope of cases addressed	2	2
supported scenarios	1	1
What is the PR author trying to achieve	3	3
the high level	1	1
Collective	1	1
expected	13	30
adherence considered correct	6	11
adherence to expected MM	2	2
adherence to SE practice	1	1
expected changes to the PR based on comments	1	2
just pass if expected	1	2
knowledge driven	1	1
expectations from code author	0	0
code quality	1	1
lower need to check in detail	1	1
expectations on what was implemented	1	1
pre-existing MM	13	32
efficient	3	3
building guesses in lack of documentation	1	1
efficient (prealignment)	1	1
expected implementation	1	1
expected implementation based on specification	2	2

based on naming	1	1
increment = specification	1	1
incrementalk construction	3	6
pre-alignment	1	1
from pre-alignment	2	2
re-review	3	4
relevance of comments and changes	2	3
writing issues	2	2
supported by information sources	1	1
external artifact	3	3
Building PR MM	1	1
ideal	14	59
different levels of formation	0	0
acceptable alternative	2	4
adherence to optimums is appreciated	1	1
improvements beyond the PR	1	1
looking for an ideal solution	2	2
prefered solutions	3	3
suboptimal actual PR	5	9
unspecific MM	2	2
missing specific proposition	3	4
knowledge driven	1	1
adherence to standards	4	5
experience driven	1	1
personal preferences	1	1
representation of the correct or improved version of the current PR	5	10
better alternative available	4	5
complementing unfinished scenarios	0	0
detailed imagination of implementation	1	1
multiple parallel options	1	1
should be like...	4	7

usage scenario	4	4
of others	3	3
Usage	0	0
Building the MM	0	0
Establish distribution of the change	2	4
Complexity of change	2	2
Find the core	3	3
Identify increment	3	6
What part of the system, is changed	1	1
Starting review with a pre-existing model of the PR	2	2
units of meaning	1	2
Utilizing information sources	0	0
constructed through issues	3	4
constructed through review comments	2	2
constructed through testing	3	4
Cross-referencing with documentation	2	2
Utilizing the PR	0	0
constructed through comparison to the removed part	1	1
constructed through execution sequence building	4	6
constructed through observing behaviour of the system	3	4
constructed through reproducing the bug	1	1
constructed through tests	3	3
Interpreted through tests	1	1
Identify issues Looking for information Write comments	1	1
Action	0	0
get information	1	3
comment	1	1
ask	2	2
IDE	3	4
navigate code	1	1
report	0	0



hints an issue	0	0
propose solution	1	1
reports an issue	2	2
Compatibility of MMs	0	0
Compatibility with the MM of the code base	4	4
affected by other changes	1	1
conflict	1	2
consistency	2	3
inconsistency	2	2
separate issues for later	1	2
desirable improvement	2	2
duplication	3	4
incorrect element usage	1	1
missing elements	1	1
re-usable elements	1	1
Conflict with actual PR MM	0	0
conflict with expected MM	3	5
Conflict with expectations	1	2
Implementation against alignment	2	3
conflict with the ideal or prototypical MM	2	4
consistency within the Mm of the PR	2	2
Previous version of the PR	1	1
External artifact	2	2
Discrepancy scope	10	19
acceptable solution	1	1
Exchange of functionality vs acceptable code quality	2	2
Adherence to code quality standards	2	2
consistent within code base	4	5
could be better	3	5
extracting out of the pr	2	3
too much effort to fix	1	1

nits	1	1
poor PRs do not reach the tem lead	1	1
stopper issues	1	1
Faulty comments, false positives	1	1
Logic precedes functional correctness	1	1
MM construction	0	0
Annotation	0	0
Annotation coverage	2	2
Additional features to expected implementation	1	1
annotation completeness	1	1
missing implementation of a specification	1	1
Not harmful extras	1	1
unadressed specifications	1	1
Undesirable behaviour	1	1
Unexpected parts	1	3
Mismatch of specification and implementation layer	1	1
Unnecessary parts	5	10
Identify unnecessary piece of code	1	1
Annotation unclear	4	7
Ambiguous usage of code elements	1	1
Faulty annotation	1	1
Discrepancy in annotation	3	3
Inconsistency in behaviour	2	3
Identified in testing	2	3
Location of annotation	2	3
annotating comments to their resolve	3	6
un-addressed comments	1	1
Building MM not possible	3	3
For reviewer	1	1
For user	1	2
when cannot form, IDE	1	1

Implementation	0	0
duplication of code or issues	1	3
Implementation errors	2	2
Risk of failure	2	2
Sequence of execution faulty	2	4
Implementation reasoning	2	4
Implementation unclear	4	5
Specification	0	0
Incomplete specification	8	9
Not aligned specification	1	1
Meaning of change missing	1	2
Update specification	3	3
Learning process	1	1
Completing the MM	0	0
Actively using author to complete	1	2
enriching by comments from others	1	1
Updasting mental models	2	2
of the PR and codebase	1	1
Specific vs fuzzy	14	39