ATLASSIAN

A score for pull request complexity

Its impact on cycle time, and how we reduced it with AI

Caterina Curti
Senior Developer Advocate

Chris Williams
Senior Software Engineer

20XX — EidosMedia — 2015 — Atlassian — DevTools — Observability — Developer Advocate — TODAY

@CaterinaCurti

2010s — UNSW — Mechatronics, Computer Vision — ESCROW.COM

ATLASSIAN — 2020s — Team Lead Platform API

Server Platform — Ecosystem Platform — TODAY

Chris Williams

@chrispwill

Complex PRs murder productivity

Why is the size of a PR important?

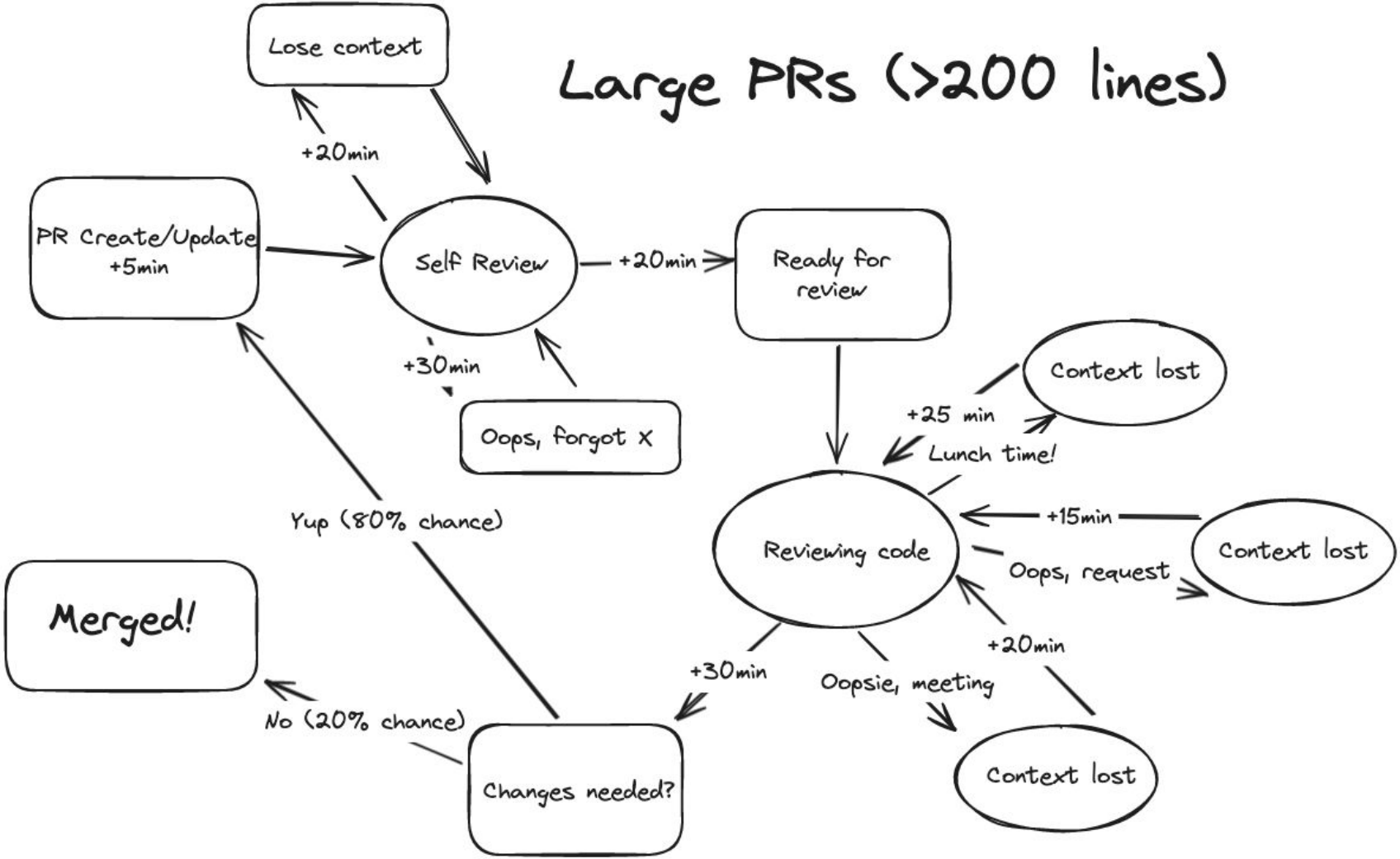
1. Reviewed more quickly
2. More thorough review
3. Easier to spot bugs
4. Less wasted work if rejected

Source:
https://google.github.io/eng-practices/review/developer/small-cls.html

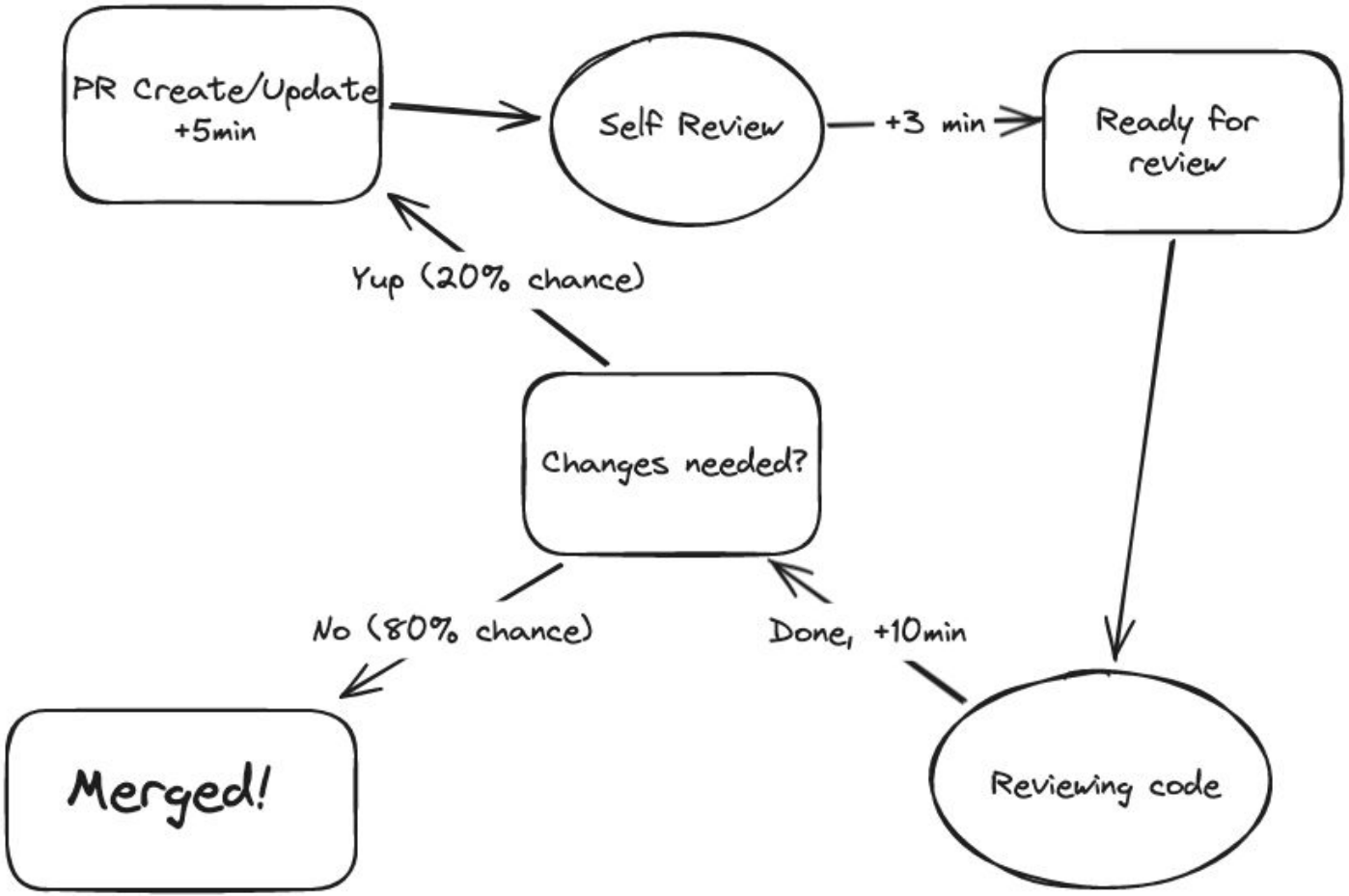
What is small?

It addresses just one thing

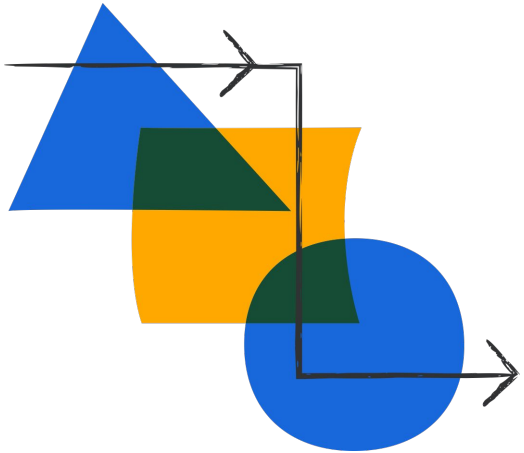
Source:
https://google.github.io/eng-practices/review/developer/small-cls.html


Large PRs (>200 lines)
Lose context
+20min
PR Create/Update
+5min
Self Review
+20min
Ready for review
+30min
Oops, forgot x
Context lost
+25 min
Lunch time!
Yup (80% chance)
Reviewing code
+15min
Context lost
Oops, request
Merged!
+30min
Oopsie, meeting
+20min
No (20% chance)
Changes needed?
Context lost

Small PRs (~50 lines)



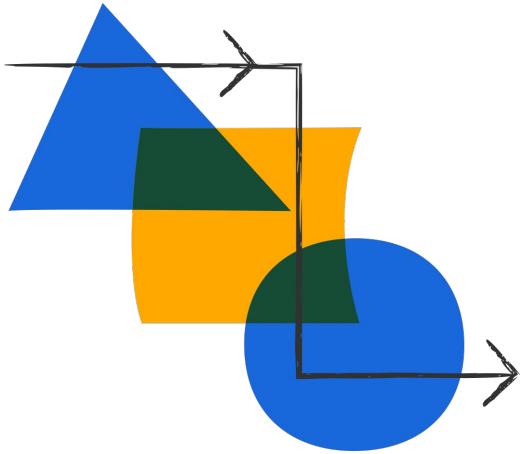
What did we find by looking at the Atlassian data?



PR complexity score

A combination of the following:

- Number of **files** changed in a PR
- Number of **comments** made in a PR
- Number of **SLOC** made in a PR

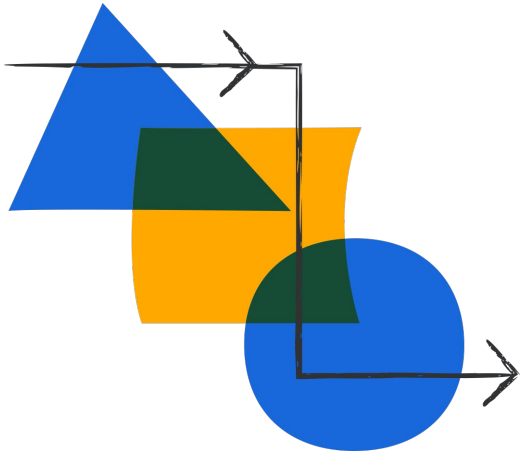


PR complexity score

A combination of the following:

- Number of **files** changed in a PR
- Number of **comments** made in a PR
- Number of **SLOC** made in a PR

Each one has an equal weight!



PR complexity score

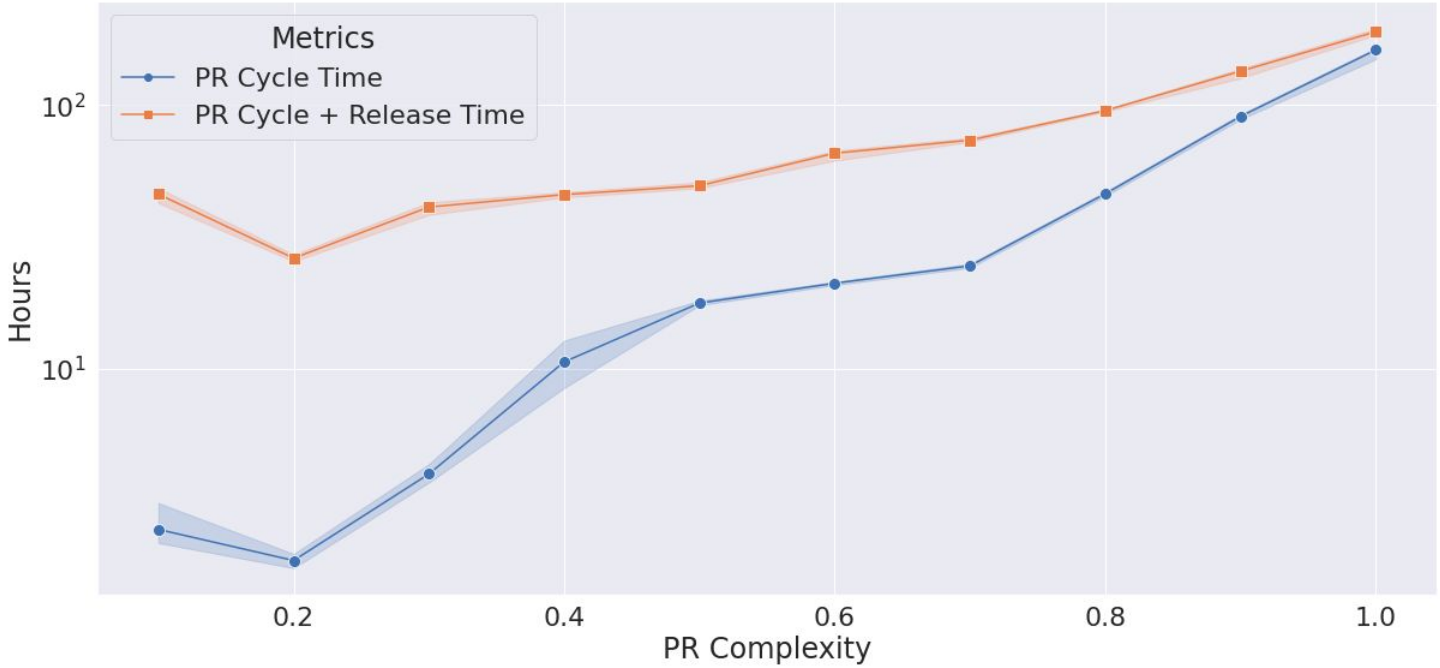
A combination of the following:

- Number of **files** changed in a PR
- Number of **comments** made in a PR
- Number of **SLOC** made in a PR

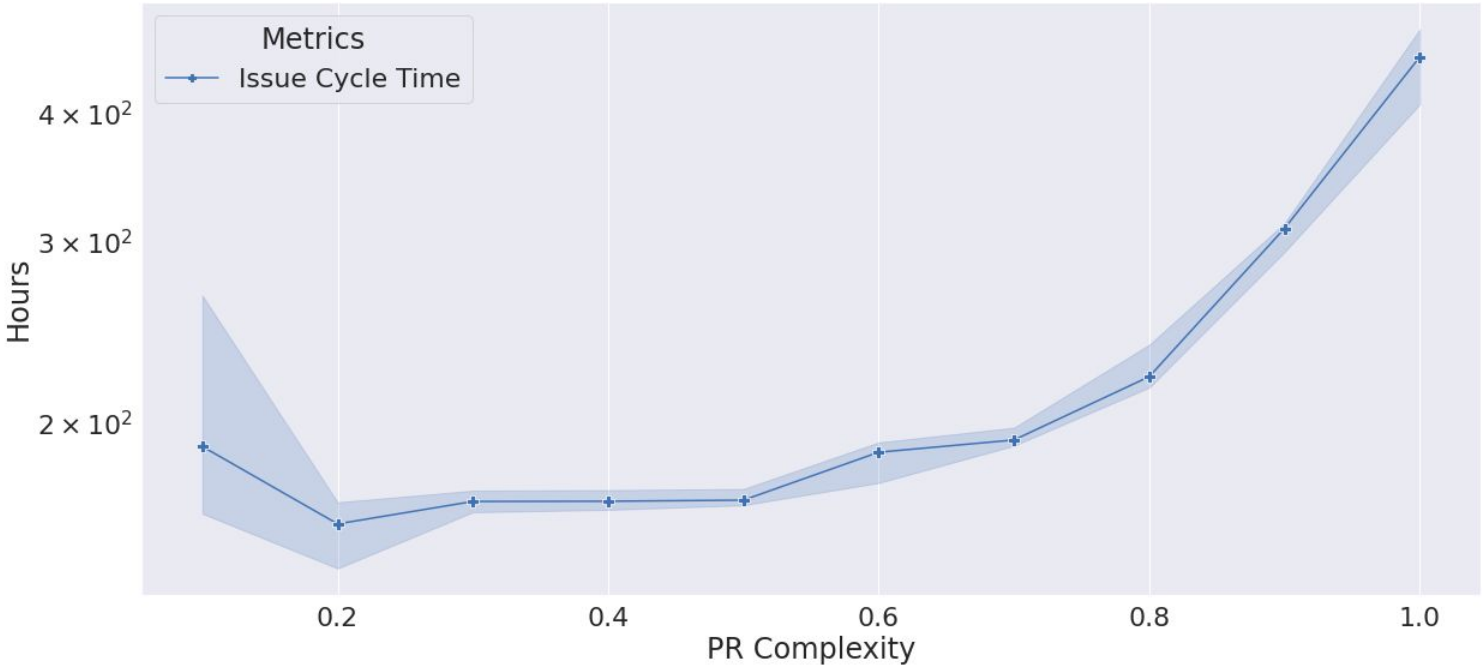
Each one has an equal weight!

Each one is a percentile value!

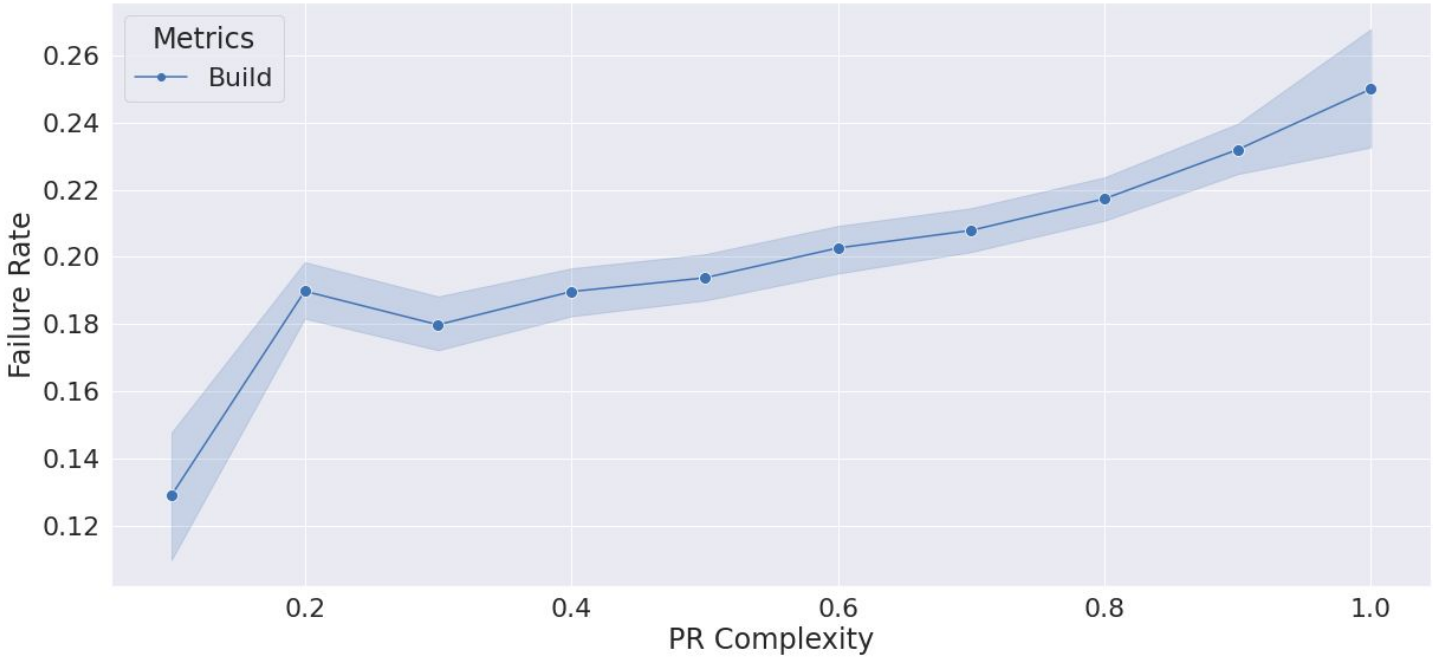
PR Cycle Time & Release Time



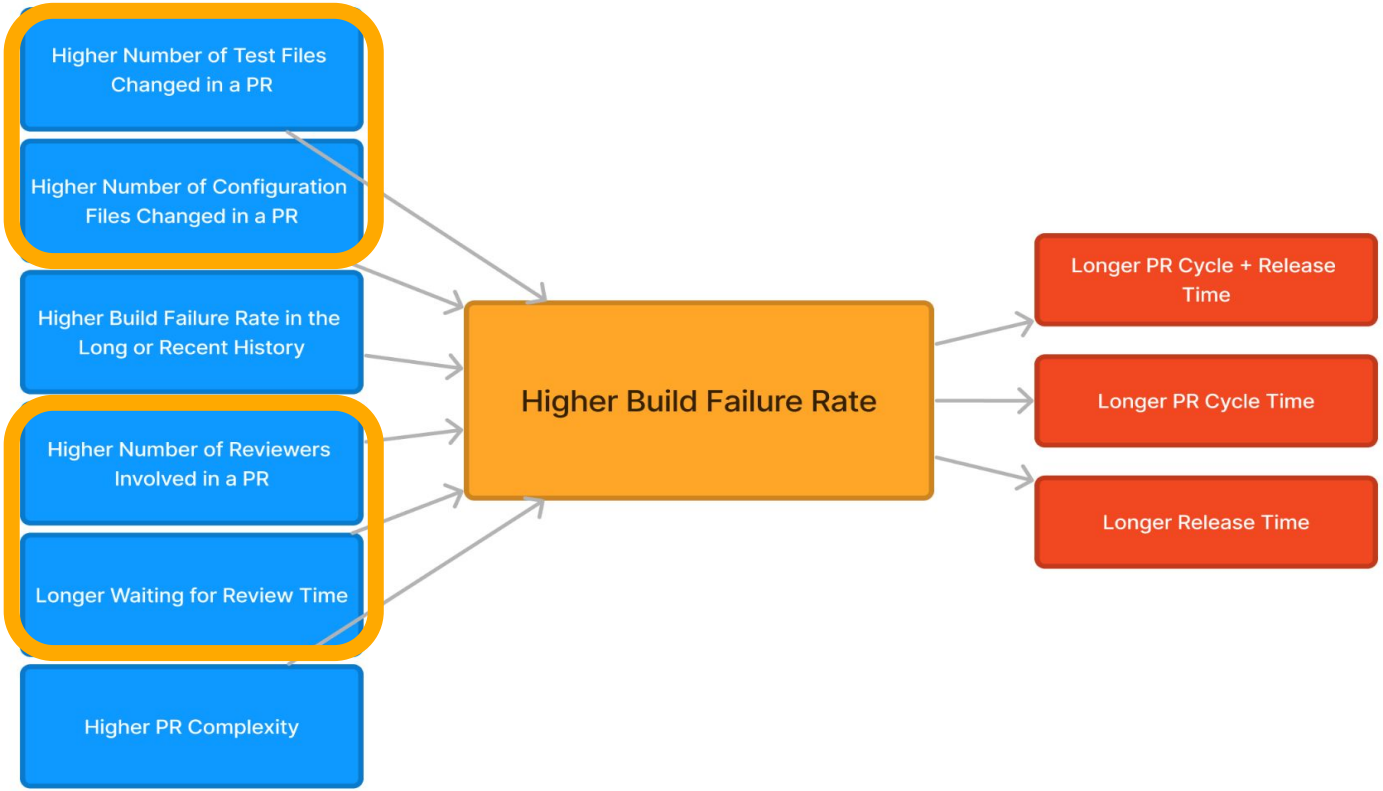
Issue Cycle Time



Build Failure Rate

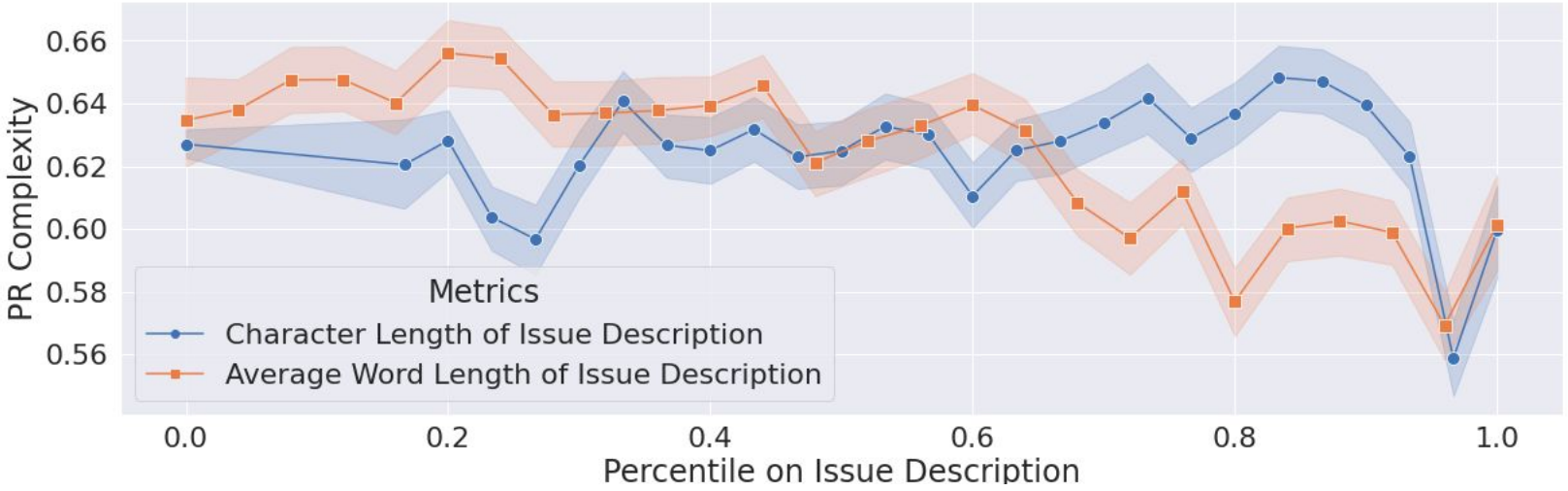


What contributes to a high Build Failure Rate?



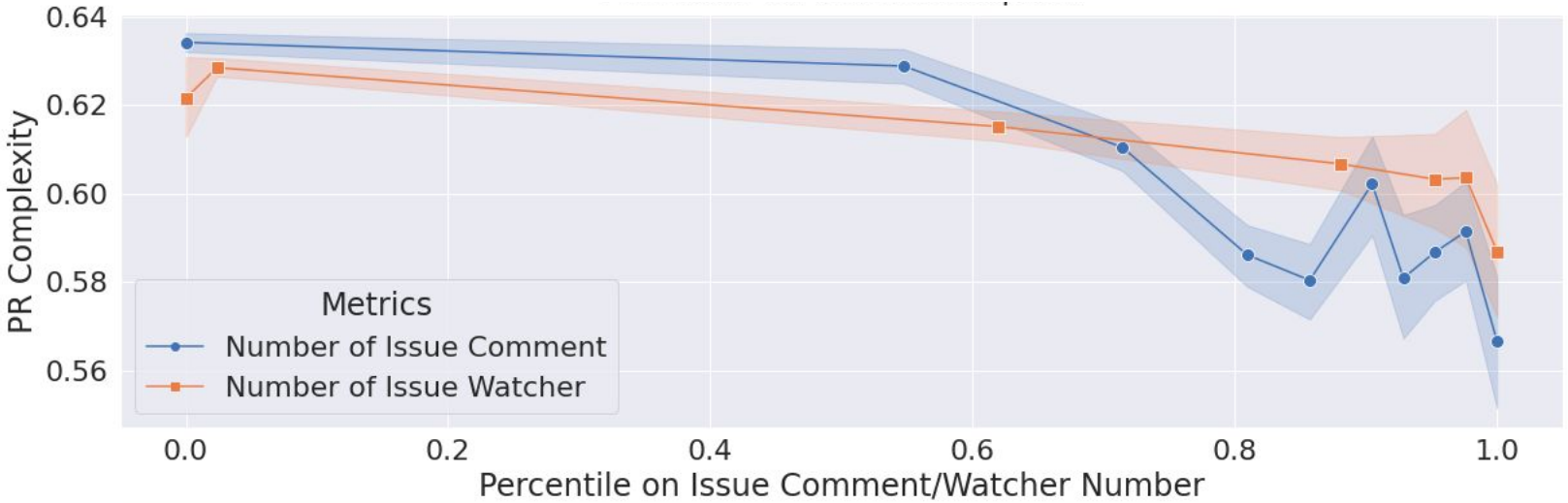
PR Complexity Driver

Size of Issue Description



PR Complexity Driver

Issue Watcher and Comment



Consequences of PR complexity

01

PR Cycle Time & Release Time

Complex PRs take longer to review, which leads to longer PR cycle times and delays in release time

02

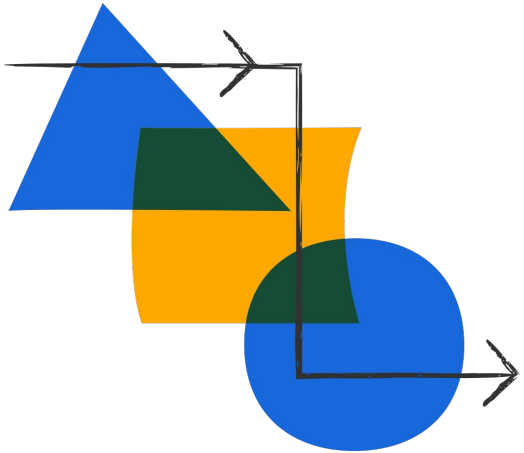
Issue Cycle Time

Issues take longer to be fully completed

03

Build Failure Rate

Merging into the main branch is more likely to fail



PR complexity score

A combination of the following:

- Number of **files** changed in a PR
- Number of **comments** made in a PR
- Number of **SLOC** made in a PR

The score is calculated

- Using equal weights
- Each value is the percentile representation

Additional drivers of PR complexity

01

Size of Issue Description

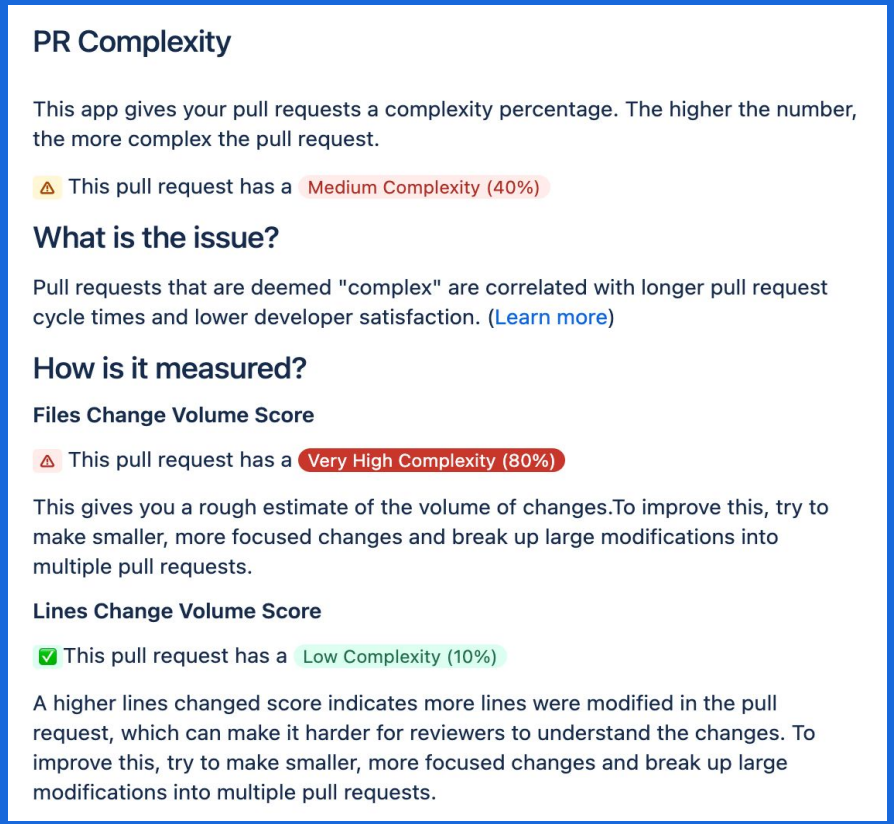
Changes that are not well planned result in more complex PRs

02

Issue Watcher and Comment

A high number of early contributions to an issue leads to less complex PRs

How did we help developers keeping their PRs less complex ?

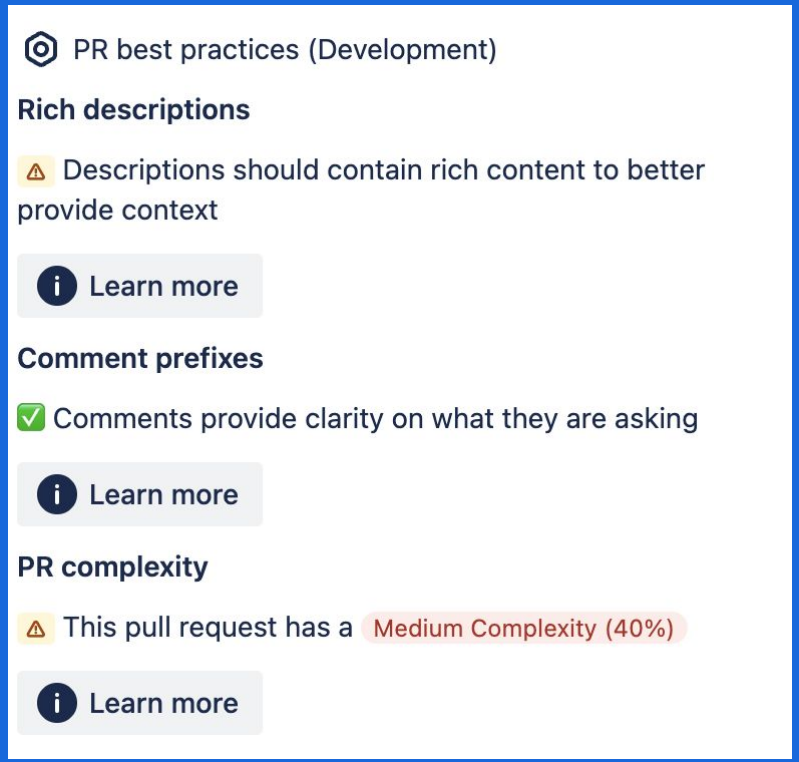


With a Forge app!

To present the complexity score directly in the content of the pull requests and suggests improvements to reduce it.

What is Forge?

Forge is the Atlassian Developer Platform. It enables developers to extend the Atlassian products and to automate and customize your workflows!



PR best practices app

A Forge app from the Atlassian platform engineering team which focuses on improving the developer experience of our developers.

The PR complexity score has been added to this app.

MyRepo

Source
Commits
Branches
Pull requests
Pipelines
Deployments
Jira issues
Security
Downloads
Repository settings

PR Best Practices: adding error handling

code-improvements → main OPEN

Unapprove Merge

Overview Files changed 1 Commits 1

THURSDAY 25 JANUARY 2024

Caterina Curti edited the description 2024-01-25

WEDNESDAY 24 JANUARY 2024

Caterina Curti commented on index.jsx 2024-01-24

hello-world-app / src / index.jsx

```
@@ -13,5 +13,7 @@
13 13  import api, { route, routeFromAbsolute } from "@forge/api";
14 14
   15 +
   16 +
15 17  /**
16 18   * Some Bitbucket REST API groups can receive calls from Forge apps via the requestBitbucket
17 19   * API. Endpoints from these API groups all require Forge app scopes.
```

Caterina Curti 2024-01-24
[Nit] unnecessary extra space
Reply · Resolve · Edit · Delete · Like · Create task

forge-bitbucket-assign-pr-reviewer commented 2024-01-24

forge-bitbucket-assign-pr-reviewer 2024-01-24
The PR has been automatically assigned to @CC
Reply · Delete · Like · Create task

forge-bitbucket-assign-pr-reviewer added 1 reviewer 2024-01-24

Get visibility into your code with code insights. Set up a pipe or an integration to start viewing the reports. Learn more

Set up a pipe

PR best practices (Development)

Rich descriptions

⚠ Descriptions should contain rich content to better provide context

Learn more

Comment prefixes

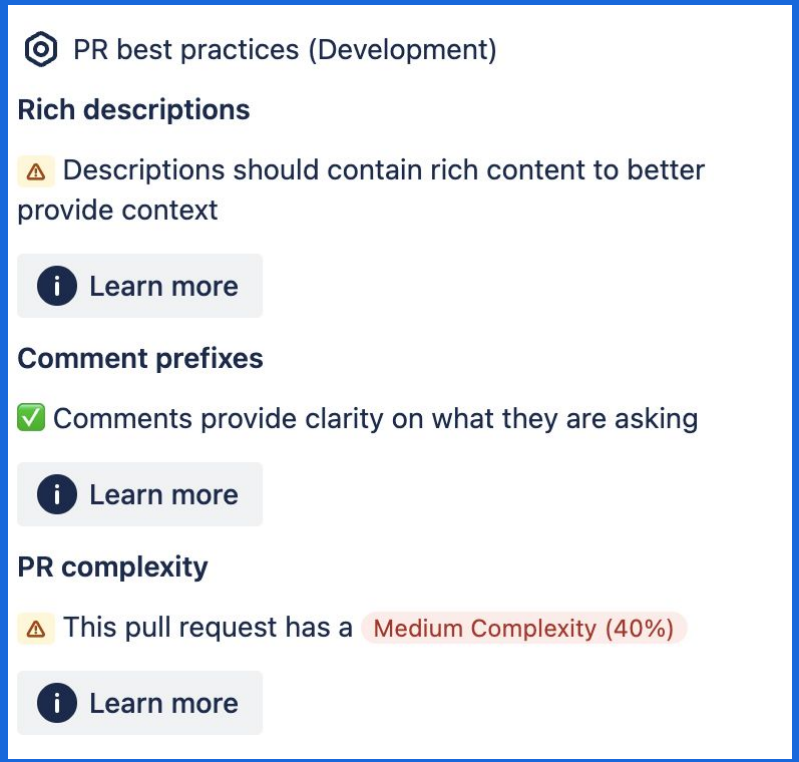
✅ Comments provide clarity on what they are asking

Learn more

PR complexity

✅ This pull request has a Low Complexity (15%)

Learn more



PR complexity score

The PR complexity score has been added to the best practices app to present the score in the UI of every PR.

PR complexity score

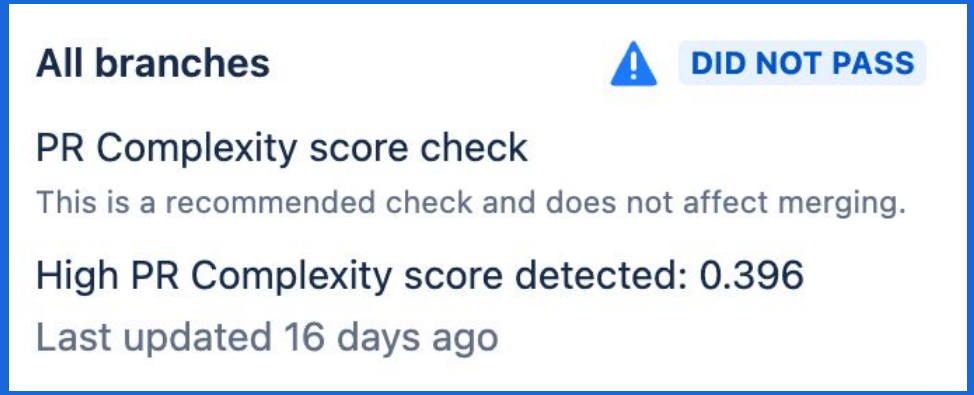
Each criteria is shown in the “Learn more” dialog

-bitbucket-pr-best-practices > src > TS prComplexity.ts > calculatePRComplexity

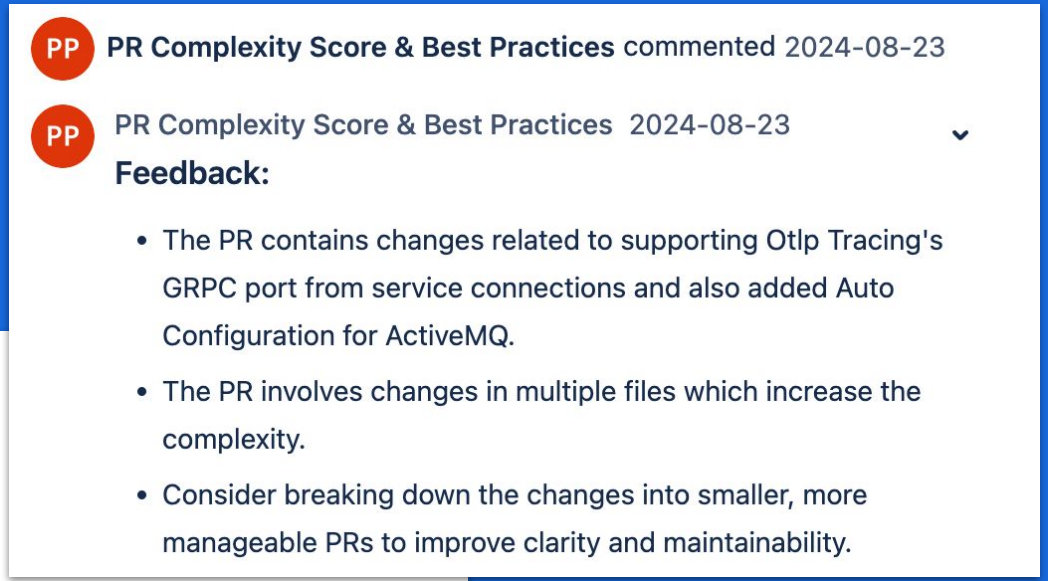
```
const addAverageChangesPerCommitScore = (numCommits: number, numLinesChanged: number) => {
  const averageChangesPerCommit = numLinesChanged / numCommits;
  complexity.averageChangesPerCommitScore = getScore(averageChangesPerCommitScores, averageChangesPerCommit);
  //console.log(`@@@ averageChangesPerCommitScore value: ${complexity.averageChangesPerCommitScore} for ${averageChangesPerCommit} averageChangesPerCommit`);
  addScore(complexity.averageChangesPerCommitScore);
};

const addAverageFilesChangedPerCommitScores = (numCommits: number, numFilesChanged: number) => {
  const filesChangedPerCommit = numFilesChanged / numCommits;
  complexity.averageFilesChangedPerCommitScores = getScore(filesChangedPerCommitScores, filesChangedPerCommit);
  //console.log(`@@@ commitRatioScore value: ${complexity.averageFilesChangedPerCommitScores} for ${filesChangedPerCommit} commitRatio`);
  addScore(complexity.averageFilesChangedPerCommitScores);
};

export const calculatePRComplexity = async (prData: any, prContext?: PRContext) => {
  numSignalsBeingUsed = 0;
  accumulatedScore = 0;
  complexity = {
    averageChangesPerCommitScore: 0,
    averageChangesPerFileScore: 0,
    averageFilesChangedPerCommitScores: 0,
    commentsScore: 0,
    filesChangedScore: 0,
    linesChangedScore: 0,
    totalScore: 0
  }
  const [diffStat, commits] = await Promise.all([fetchPrDiffStat(prContext), fetchPrCommitList(prContext)]);

  const numFilesChanged = diffStat.values.length;
  const numLinesChanged = getNumLinesChanged(diffStat);
  const numCommits = commits.values.length;

  addFilesChangedScore(numFilesChanged);
  addLinesChangedScore(numLinesChanged);
  addCommentsScore(prData);

  complexity.totalScore = (accumulatedScore) / numSignalsBeingUsed;
  console.log(`@@@ complexity value: ${stringify(complexity)}`);
  return complexity;
};
```

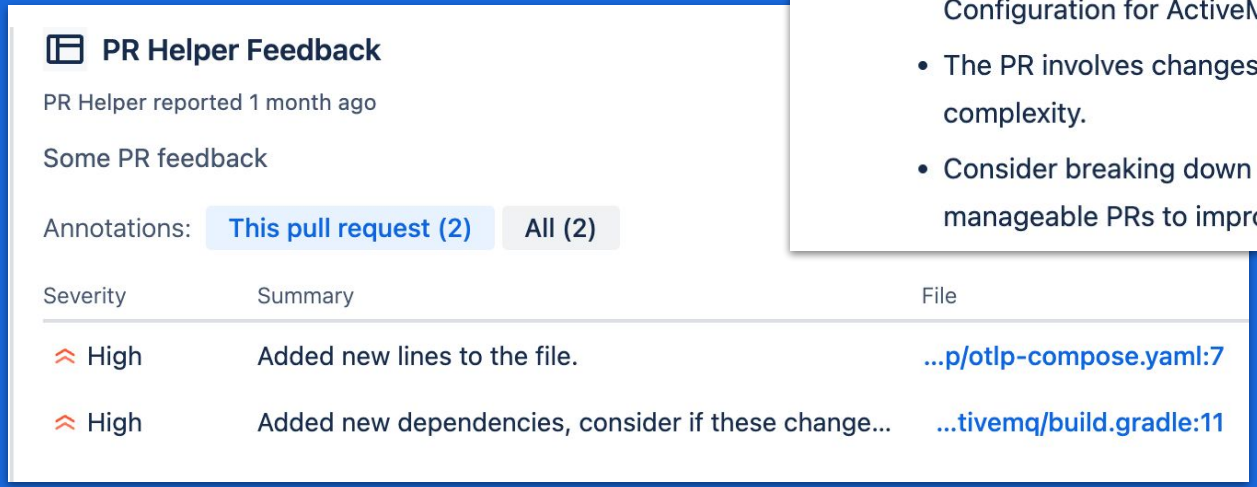
Custom Merge Check

The “PR complexity score check” can be added to inform about the PR complexity at merge time and even prevent a merge.



How did we use AI?





Bring the PR complexity score where developers are working and help them lower it. Thanks to an app!

AMA

Code and Forge resources:
go.atlassian.com/dpe-resources

Let's connect on LinkedIn
@CaterinaCurti @ChrispWill

Presentation end