Flow metrics

Tags: Flow (https://pluralsight.knowledgeowl.com/help/search?phrase=:Flow)

A list of common metrics, their definitions and why they are important.

In this article

Who can use this?

Core	Plus
✓	✓

Code metrics

Coding days

- **What:** Any day where a developer contributed code to the project.
- **Why:** Coding Days answers the question, “How often are my team members able to create solutions in the codebase?” A simple metric that provides insight into blockers that may be limiting a team’s ability to deliver.
- **What's** normal for the Industry ()?
    - Leading Contributor - 4.1
    - Primary Contributor - 3.3
    - Occasional Contributor - 2.1

Rework ()

- **What:** Code which is deleted or rewritten shortly after being written (less than 3 weeks old)
- **Why:** This metric tells you how much and how often an individual is rewriting their own work. Large deviations from your rework baseline is an early indicator that something may be affecting the health of your development lifecycle.

Commits per coding day

- **What:** The number of commits that a developer made on days when they were able to commit. A developer receives credit for a commit on any branch when the commit is pushed to the remote git server.
- **Why:** This metric helps visualize how team members commit their work. Is it in multiple small commits or are they keeping their code locally and submitting one large commit at the end of a sprint? Teams running CI/CD, or teams that place a heavy emphasis on small-frequent solutions place a high value on this metric.
- **What's** normal for the Industry?
    - Leading Contributor - 8.4
    - Primary Contributor - 4.0
    - Occasional Contributor - 2.1

Efficiency

- **What:** The percentage of all contributed code which is productive work.
- **Why:** Efficiency is trying to answer the following question, “How much of an author’s work is not re-writing their own code?” Answering this question relates directly to the concept of rework—the amount of an author’s work that is self re-written.
- **What's** normal for the Industry?
    - Typical - 70%

Help others

- **What:** Code where a developer modifies someone else’s recent work (less than 3 weeks old)
- **Why:** This metric helps you to determine which team members are spending more time helping others, than perhaps working on their own work.

Impact ()

- **What:** Severity of edits to the codebase, as compared to repository history.
- **Why:** Impact attempts to answer the question, “Roughly how much cognitive load did the engineer carry when implementing these changes?” It can help identify team member’s patterns in taking on projects that are similar in nature and encouraging them to diversify their work.
- **What's** normal for the Industry?
    - Leading Contributor - 463
    - Primary Contributor - 214
    - Occasional Contributor - 77

Legacy refactor

- **What:** Code that updates or edits old code (older than 3 weeks).
- **Why:** This metric shows you how much time an individual or team spends paying down technical debt.

New work ()

- **What:** Brand new code that does not replace other code.
- **Why:** This metric shows you the quantity of code an individual or team spends on writing new features and products.

Productive throughput

- **What:** Any code that is not rework is productive work.
- **Why:** Productive Throughput is a simple output metric that shows you how much code that we contribute sticks around longer than three weeks.

Raw throughput (Raw LoC)

- **What:** All code contributed, regardless of work type.
- **Why:** This metric shows you how much code is written, across all work types.

Commit complexity ()

- **What:** A measure of the cognitive load associated with rectifying a bug found in a commit. The calculation includes how large the commit is, the amount of files touched, and how concentrated the edits are, etc.
- **Why:** This metric allows team leads to identify and review the most anomalous work first which, maximizes their most precious resources: time and attention.

tt100 Productive ()

- **What:** The amount of time consumed to contribute 100 lines of productive code, after rework.
- **Why:** This metric can help you identify how an individual is progressing on any given project. If they are writing lots of code (tt100 Raw), very quickly but taking a longer time to write productive code it can be a signal of a blocker.

tt100 Raw ()

- **What:** The amount of time consumed to contribute 100 lines of raw code, before rework.
- **Why:** This metric can be used with tt100 Productive to see how much time it takes an individual to write productive code in relation to raw code.

back to top

Submit metrics

Learn more about Submit metrics ().

Responsiveness

- **What:** The time it takes a Submitter to respond to a comment on their pull request with either another

comment or a code revision.

- **Why:** This metric answers the question, “Are people responding to feedback in a timely manner?“ Driving this metric down will ensure PRs are being reviewed and merged in an appropriate time frame.
- **What's** normal for the Industry?
    - Leading Contributors - 1.5 hours
    - Typical Contributors - 6.0 hours

Unreviewed PRs

- **What:** The percentage of PR’s without comments or approvals.
- **Why:** This metric shows you the number of PRs merged without being reviewed. This shows whether we are limiting risk of solutions we are providing our customers, by limiting the amount of work that does not get a second pair of eyes.
- **What's** normal for the Industry?
    - Leading Contributors - 5%
    - Typical Contributors - 20%

Comments addressed

> **Note:** Comments addressed is only available in Flow Enterprise Server.

- **What:** The percentage of comments to which a submitter responds.
- **Why:** This metric helps answer the question, “Are people acknowledging feedback from their teammates?” Use this metric to prompt and encourage healthy discussion in reviews.
- **What's** normal for the Industry?
    - Leading Contributors - 45%
    - Typical Contributors - 30%

Receptiveness

> **Note:** Receptiveness is only available in Flow Enterprise Server.

- **What:** The percentage of comments the submitter accepts as denoted by code revisions.
- **Why:** This answers the question “Are people incorporating feedback from their teammates?” This metric looks at whether the PR submitter is taking people’s feedback and incorporating that into their code.
- **What's** normal for the Industry?

- Typical Range - 10%-20%

back to top

Review metrics ()

Reaction time

- **What:** The time it takes for the reviewer(s) to review a pull request or respond to a comment
- **Why:** This metric answers the question, “Are reviewers responding to comments in a timely manner?”. In practice, the goal is to drive this metric down. You generally want people to be responding to each other in a timely manner, working together to find the right solution, and getting it to production.
- **What's** normal for the Industry?
  - Leading Contributors - 6.0 hours
  - Typical Contributors - 18.0 hours

Involvement

> **Note:** Involvement is only available in Flow Enterprise Server.

- **What:** The percentage of pull requests that the reviewer(s) participated in.
- **Why:** Involvement aims to show you how many pull requests are reviewed and by whom. This is very context dependent and answers the question, “Are some people more involved in reviews than others?”.
- **What's** normal for the Industry?
  - Leading Contributors - 95%
  - Typical Contributors - 80%

Influence

> **Note:** Influence is only available in Flow Enterprise Server.

- **What:** The ratio of follow-on commits made after the reviewer(s) commented.
- **Why:** This metric shows how often someone updates code based on reviewer comments. It gives you insight into your review process and helps you identify trends in how often submitters respond to comments with code revisions and additions.
- **What's** normal for the Industry?
  - Typical Range - 30%-40%

- Typical Range - 20%-40%

Review coverage

> **Note:** Review coverage is only available in Flow Enterprise Server.

- **What:** The percentage of hunks commented on by the reviewer(s).
- **Why:** This metric shows you how much of the code in the PR was actually reviewed by team members versus comments on PR itself. It answers the question, “How much of each PR has been reviewed?”.

back to top

Team collaboration metrics ()

Time to merge (https://help.pluralsight.com/help/time-to-merge)

- **What:** The time it takes from when pull requests are opened to when they are merged.
- **Why:** This metric gives you visibility into how long on average it takes your team to merge a PR.

Time to first comment (https://help.pluralsight.com/help/time-to-first-comment)

- **What:** The time between when a Pull Request is opened and the time the first reviewer comments.
- **Why:** This metric answers the question, “On average, how long does it take for a reviewer to comment on a PR?” The lower this number, the less wait-states your team encounters and the faster you can move work through code review.

Follow-on commits

- **What:** The number of code revisions added to a pull request after it is opened for review.
- **Why:** Knowing the number of follow-on commits that are added to an open PR gives you visibility into the strength of your code review process. If you are seeing a trend of lots of follow-on commits being added there may be a need for more planning and testing.

PR activity level

- **What:** A measure of how active a Pull Request is on a scale of Low, Modest, Normal, Elevated, High, as calculated by the comment count, word length, and recency of the comment.
- **Why:** This metric will help you gauge how much chatter is happening around a PR without having to read the actual comments on the PR. Identify outliers, long-running, low activity PRs or PRs with lots of activity, and nudge them forward in the review process.

Raw activity

- **What:** A raw count of comments and follow-on commits associated with a PR. As a Pull Request ages, it's

- **What:** A raw count of comments and follow-on commits associated with a PR. As a Pull Request ages, it's raw activity does not change and will always stay the same. Pull Requests are ordered by raw count of comments + follow-on commits in an increasing or decreasing order.
- **Why:** Allows you to find the most historically active PR's regardless of their age.

Ticket activity level

- **What:** A measure of how active a ticket is on a scale of Low, Modest, Normal, Elevated, High, as calculated by the comment count, word length, and recency of the comment.
- **Why:** Similar to PR Activity, Ticket Activity visualized where people's attention is going in the review process. Identify outliers to keep features and projects on track for delivery, and review comments/discussions for tone, content and material scope creep.

back to top

Knowledge sharing metrics ()

Sharing Index

- **What:** Measures how broadly information is being shared amongst a team by looking at who's reviewing who's PRs.
- **Why:** The Sharing Index helps us understand whether we are increasing the number of people that participate in code review, or whether we are consolidating the group of people that do code review for our organization. It can be helpful in identifying knowledge silos and or individuals that are "alone on an island" when it comes to code review.

Number of PRs reviewed

- **What:** Total number of PRs that were reviewed.
- **Why:** Gives you visibility into the total number of PRs that were reviewed.

Number of users reviewed

- **What:** Total number of submitter Users that were reviewed
- **Why:** Understand whether the code review is being distributed to many different team members.

back to top
