

Revolution or Regression?

- A Comparatively Empirical Study of Two Integration Processes



Yujuan Jiang, Josh Chiang, Roy Budhai, Bram Adams

Feature Toggle

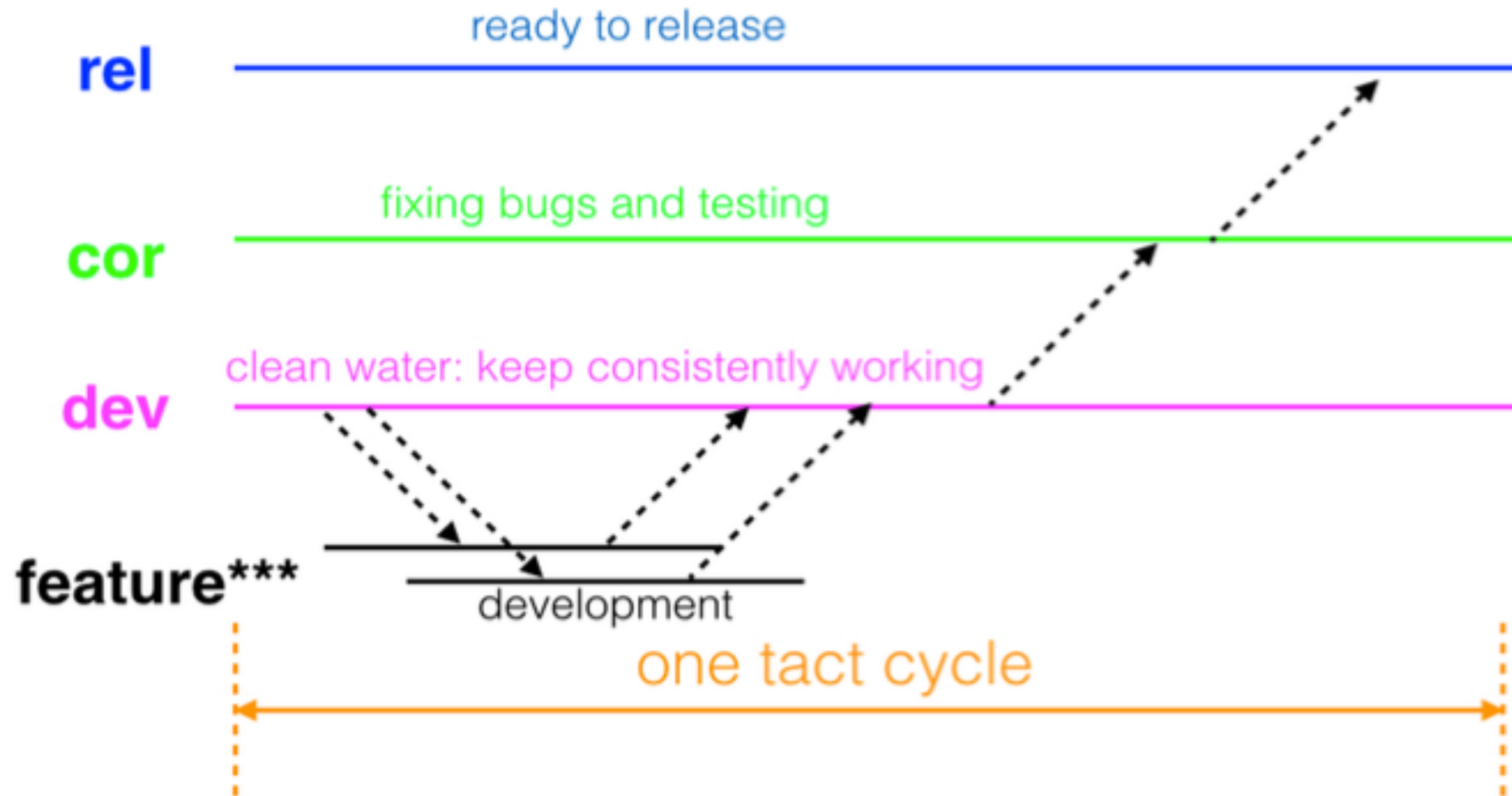


- Check in whenever you want
- Build and test on most up-to-date trunk branch
- Won't be executed in the product if the “toggle” is switched off

Agile vs Waterfall Integration

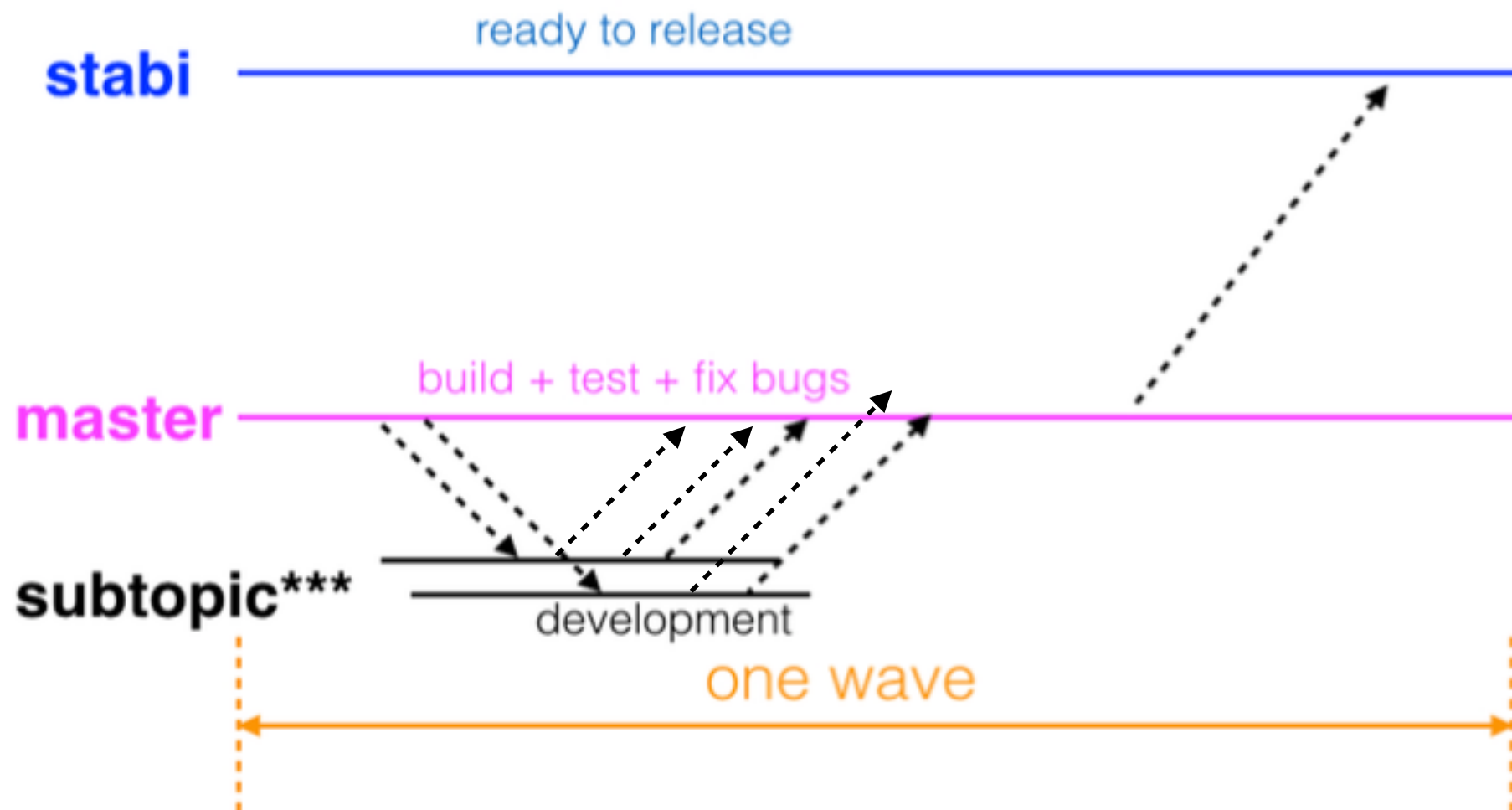


Project A: Branch-based (Waterfall) approach



Clean water development: each feature branch got merged back to dev branch only after the whole feature is completed. Only bug-fixes will be cherry-picked to cor/rel branch.

Project B: Toggle-based (Agile) approach



Each feature branch check in master branch whenever they want. Merge small, merge faster. Yet as long as the toggle of this feature is not turned “on”, the code will not be integrated into release branch.

Key Performance Indicator (KPI)

- **Integration effort:** size of merge commit
- **Productivity:** #LOC per developer, success rate
- **Quality assurance:** # of bugs

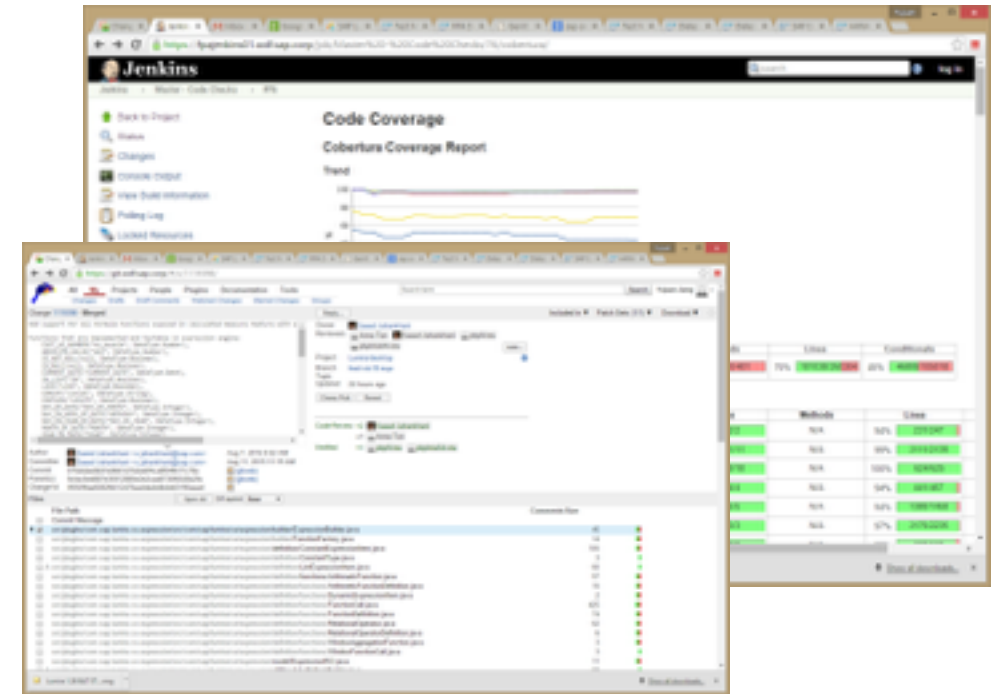
Key
Performance
Indicator



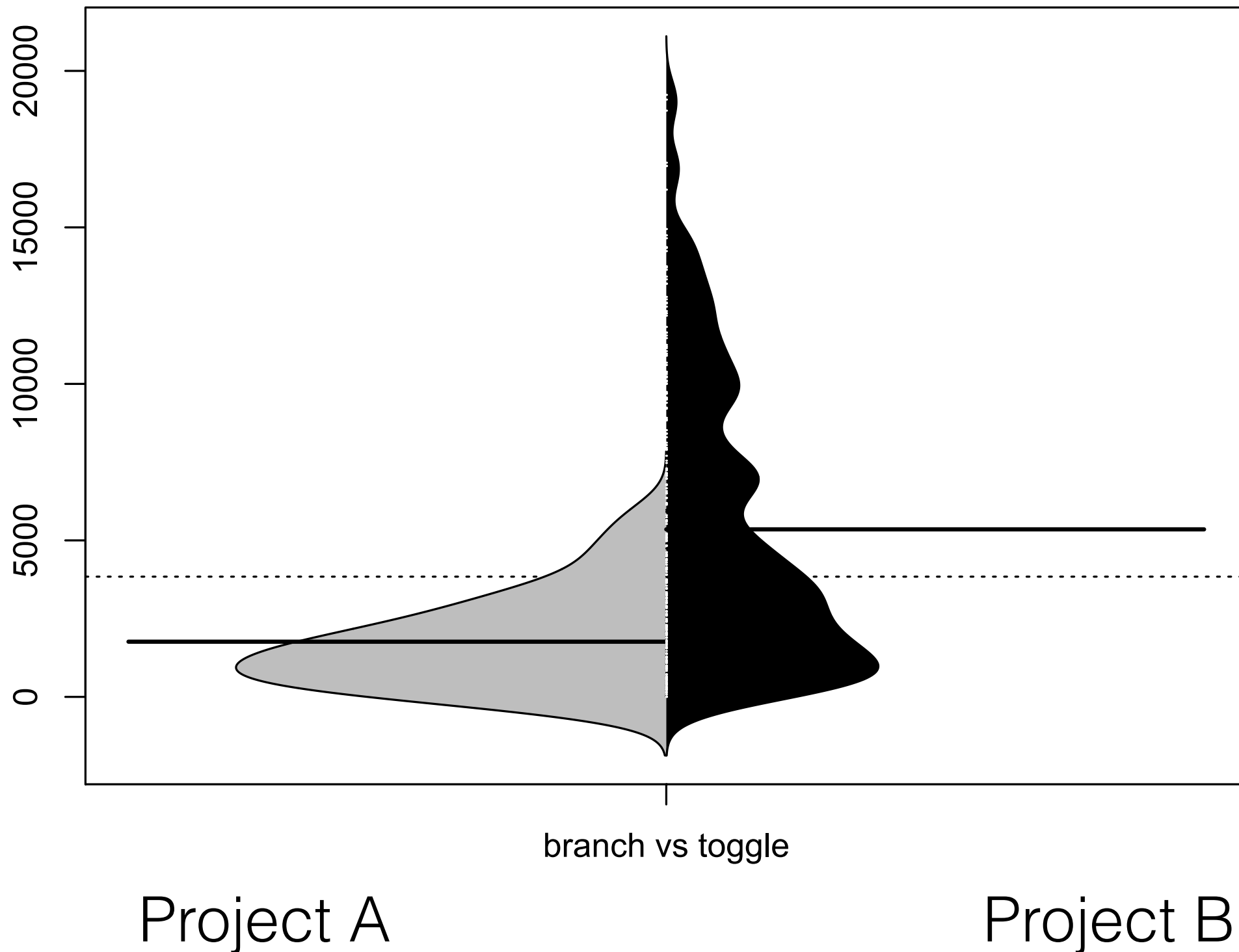
Fetching Data

- From Jenkins (build logs)
- From Gerrit (git logs)

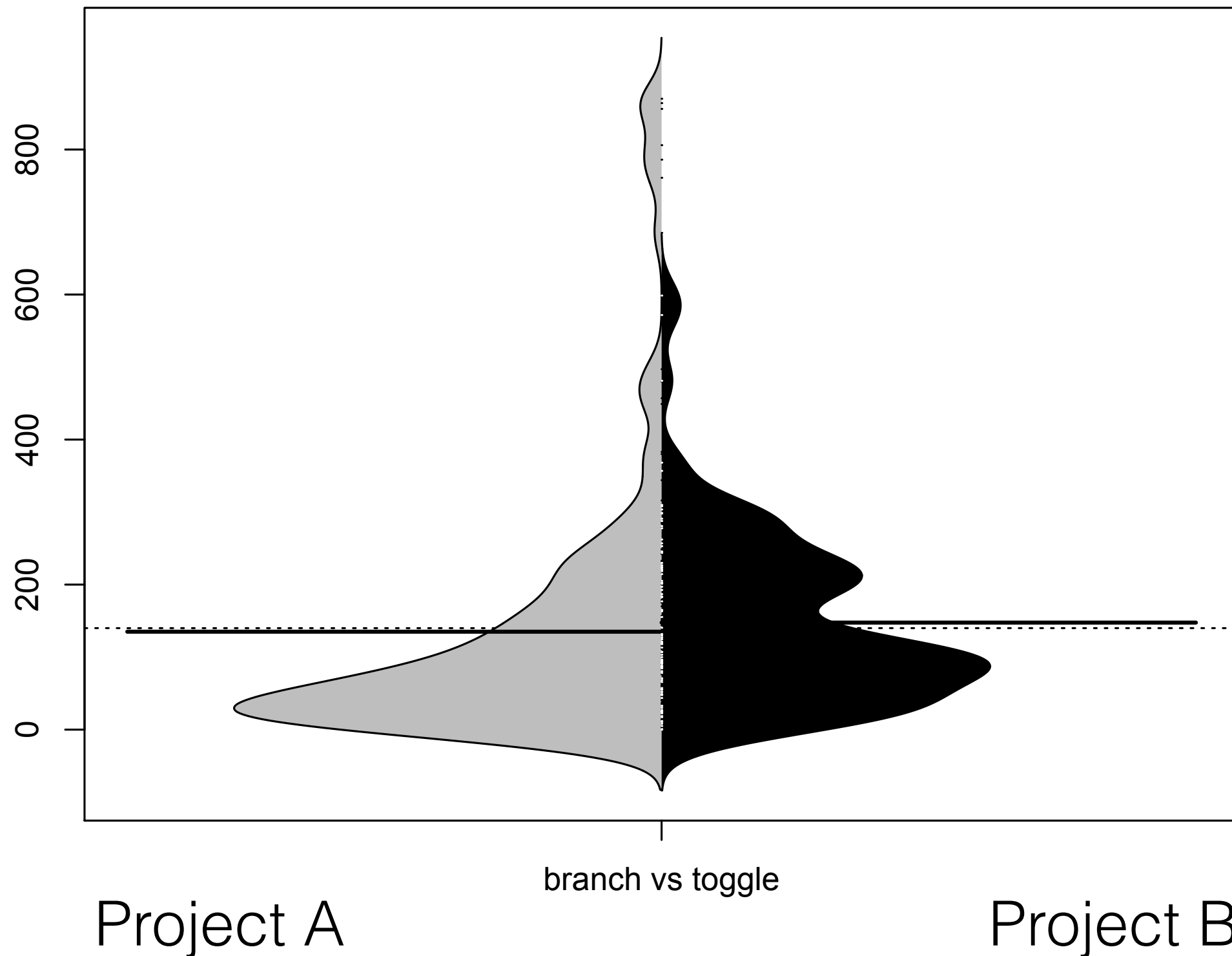
* JSON + Python + Perl + .javascript



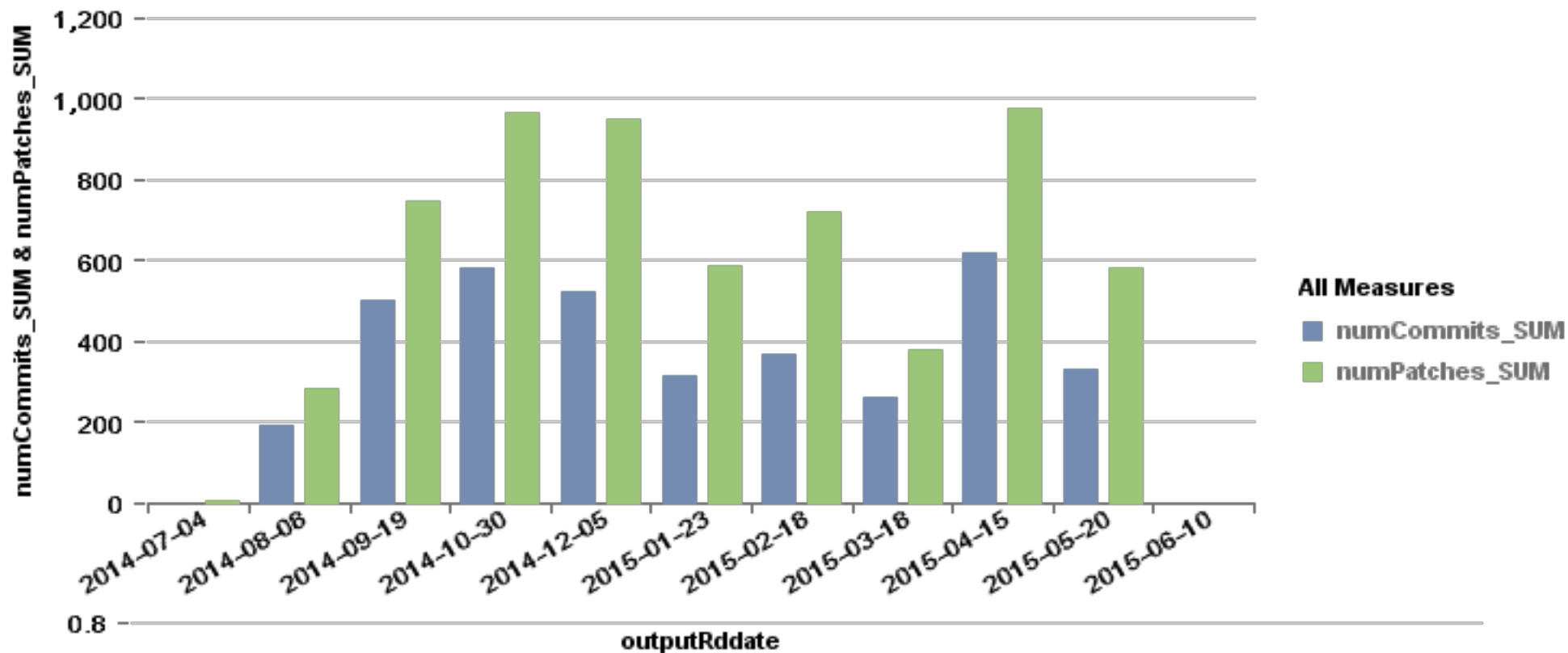
Integration Effort



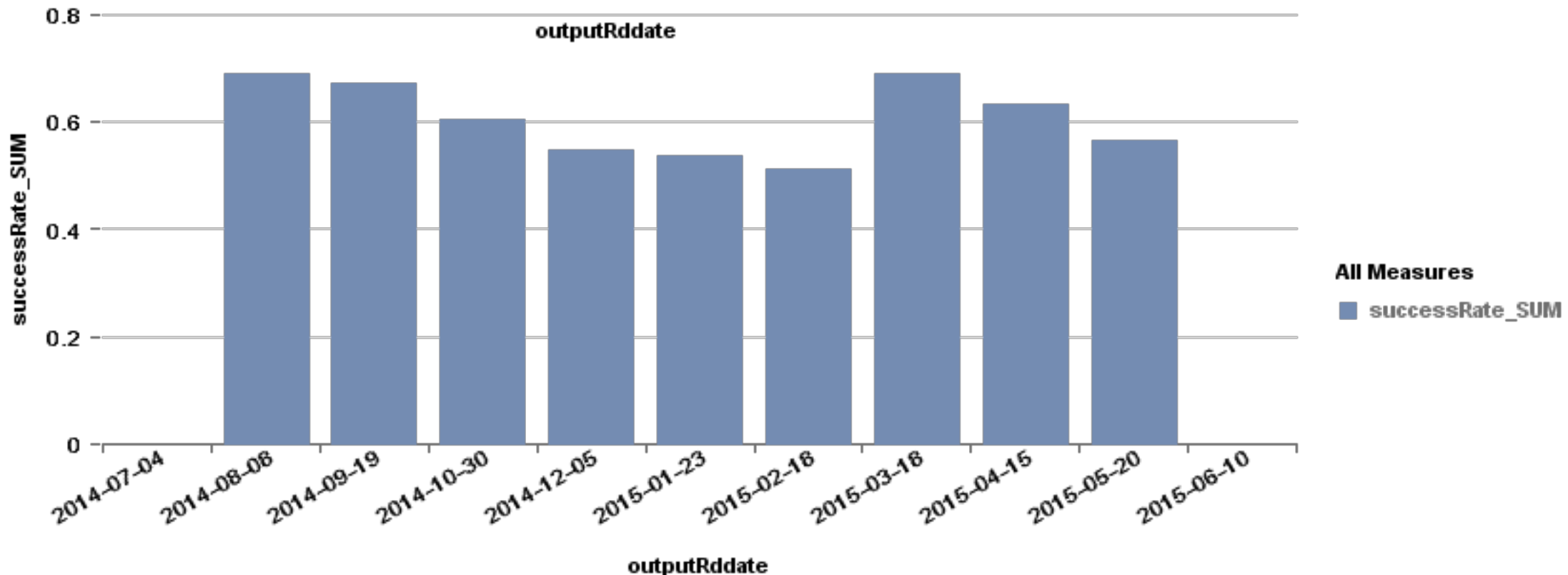
Productivity (#LOC)



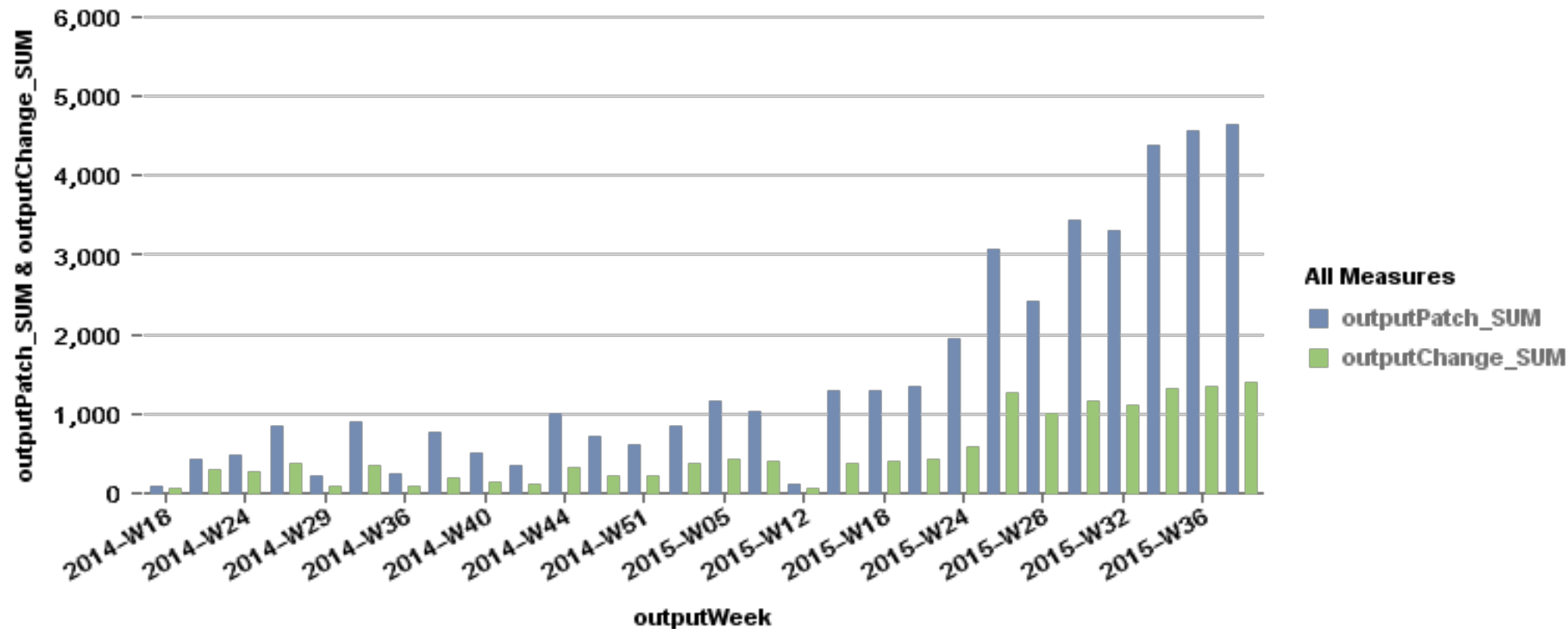
Productivity (Success rate)



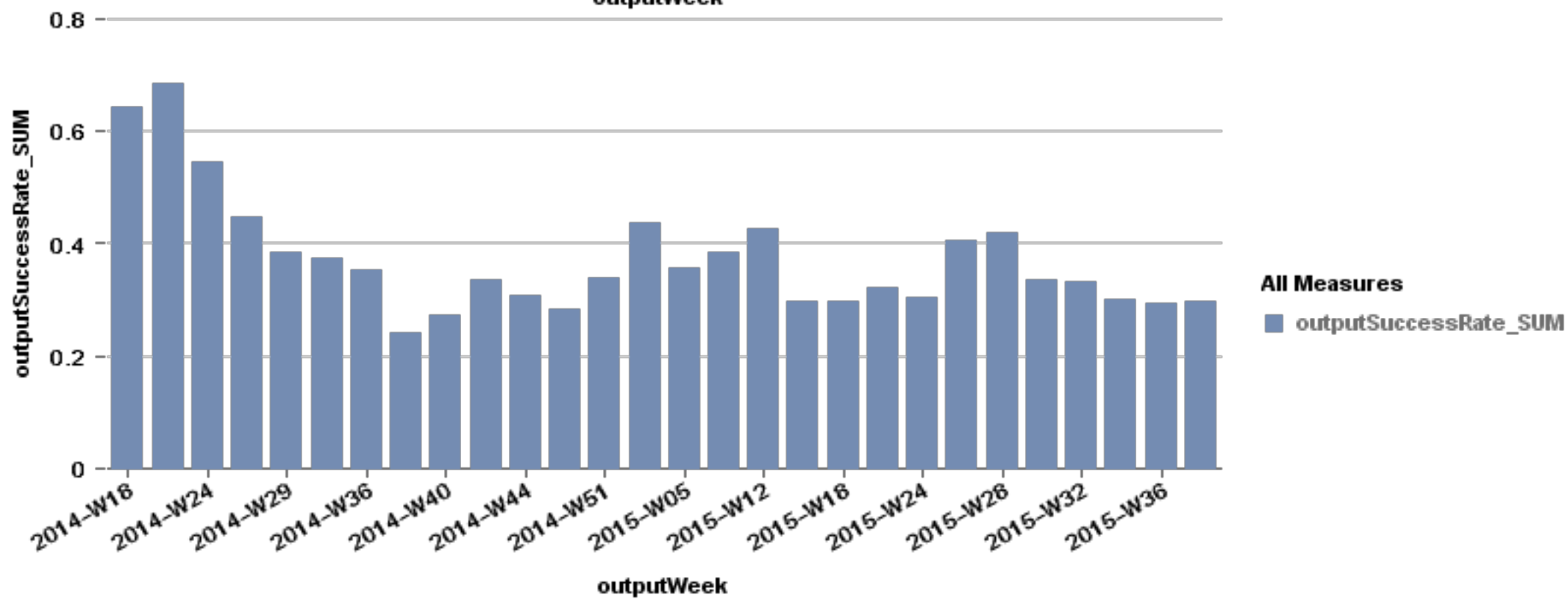
Project A



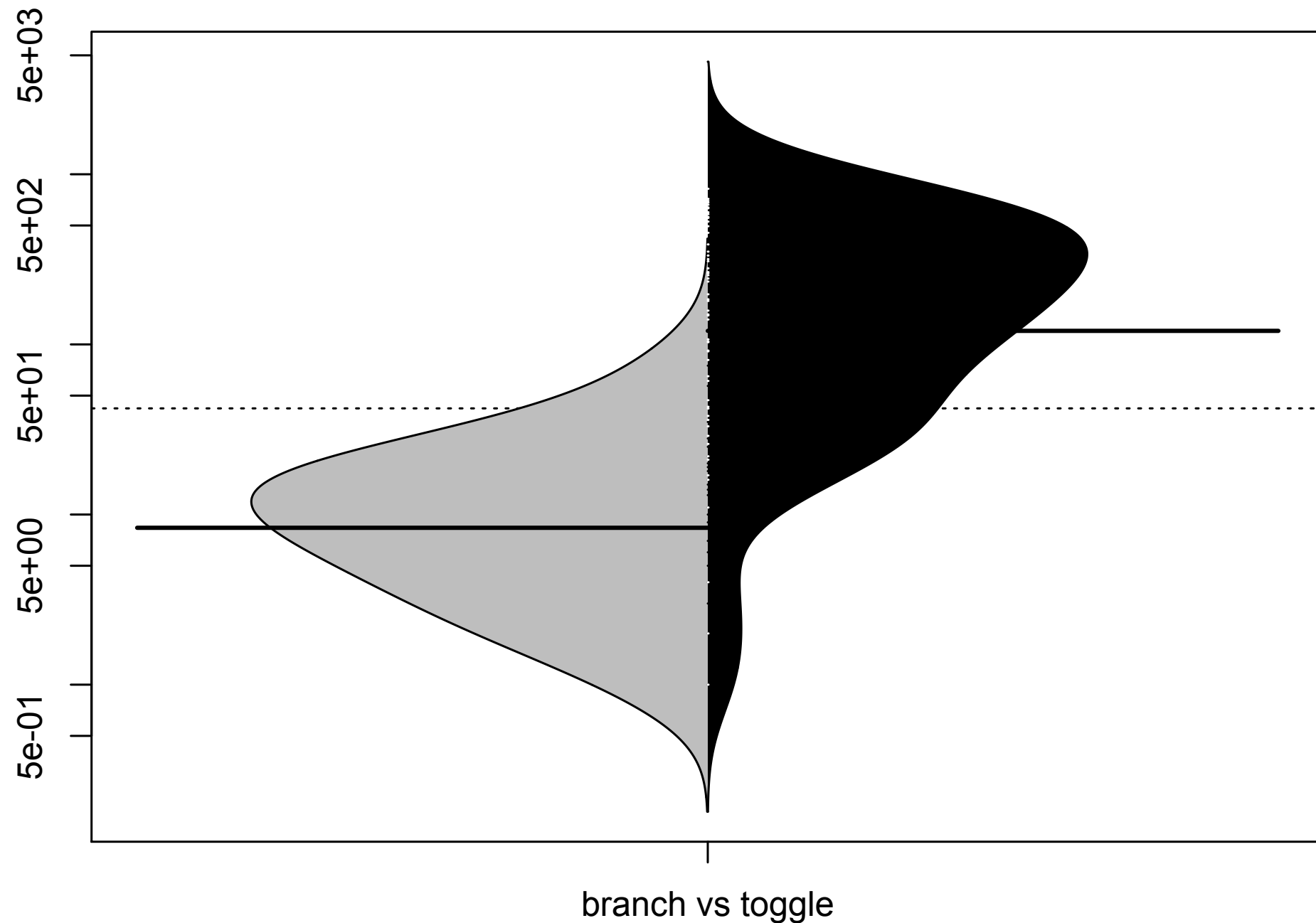
Productivity (Success rate)



Project B



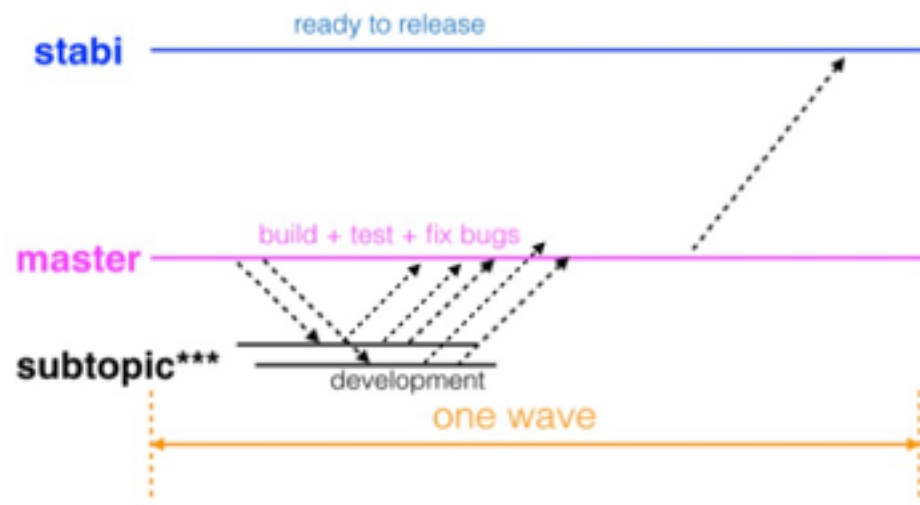
Quality Assurance



Project A

Project B

Project B: Toggle-based (Agile) approach



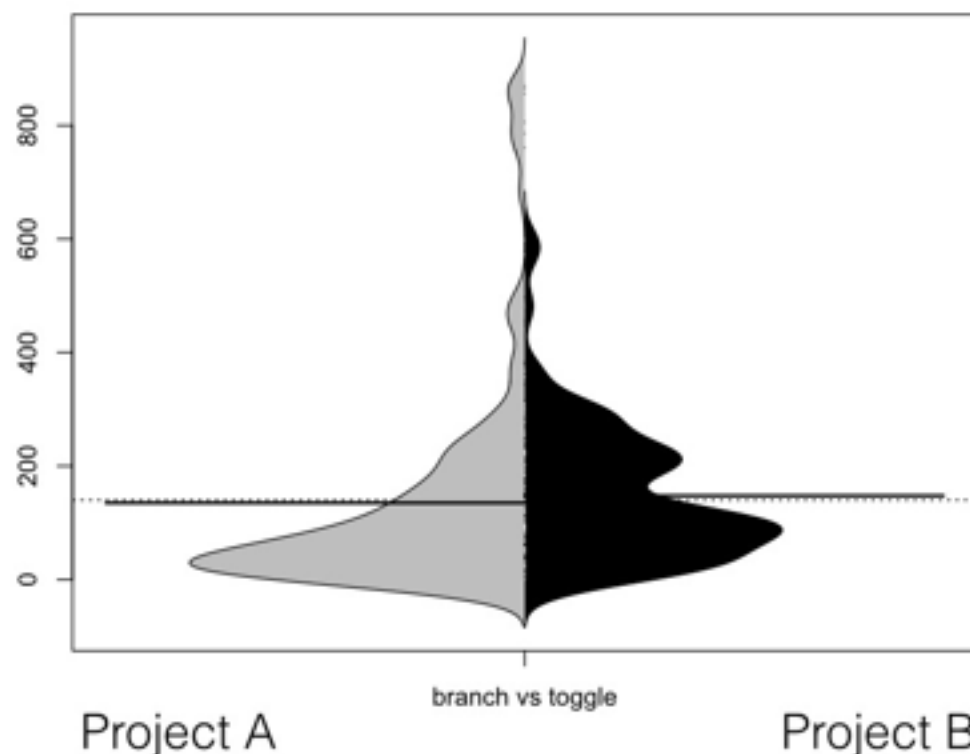
Each feature branch check in master branch whenever they want. Merge small, merge faster. Yet as long as the toggle of this feature is not turned "on", the code will not be integrated into release branch.

Key Performance Indicator (KPI)

- **Integration effort:** size of merge commit
- **Productivity:** #LOC per developer, success rate
- **Quality assurance:** # of bugs



Productivity (#LOC)



Productivity (Success rate)

