How to measure success in open source contribution

Ranny Haiby
Director, Open Source Group

Mar-2019
Success? What success?

• We are often required to reflect upon our investment in open source and determine whether it is successful or not
  – Triggers may be
    • decision making time – Should we add or remove resources from the effort
    • New projects – should we try to replicate our success?
    • Our group or personal KPIs
    • Those pesky questions from other groups or individuals - Is it worth our effort? Why do we investing? Where is our return?

• Success of the open source project and our organization’s success in it are two different things
  – We may be unsuccessful in a successful project, and vice versa
Quick and easy success KPI?

• Counting the amount of commits or lines-of-code seems like an easy metric
• But, it has many pitfalls. It may not be as relevant as you may think
• Side effects may include:
  – Creating an illusion of success
  – Driving developers into making low quality contributions. Preferring quantity over quality
  – Overlooking other important types of contributions
  – Arms race with other organizations and companies
  – Accounting like tricks to “cook the books”
• The easiest to measure metrics may not indicate real outcomes
Doesn’t hurt to have a goal. Or two.

- Success should be measured against goals
  - Your goals will determine your KPIs. It is easier to define the desired outcome when you have goals
  - The same action might be supportive of one goal, but obstructive to another

- How to determine your goals
  - Top down, derived from some higher level strategy
  - Bottom up – Look at what you have already been doing and try to group into buckets that will turn into goals

- Validate and update your goals from time to time
  - Items may have shifted during flight
Got my goals. How do I turn them into metrics?

• If the goals are quantitative in nature, it should be straightforward
  – If the goal is “more features”, counting the number of accepted blueprints/proposals might help
  – If the goal is “better quality”, we can count the number of bugs squashed by our teams

• If the goals are more qualitative or subjective, we have to work harder
  – What if my goal is improving my organization’s reputation?
  – How do I measure my influence in the community?
Some KPIs to consider

• How many technical proposals or blueprints that you initiated or supported got accepted by the community?
  – Pros
    • Proposals lead to features, which are the true outcome
  – Cons
    • Might encourage low quality proposals
    • We have limited control over the faith of proposals

• How many tickets/issues were resolved by our team?
  – Pros
    • Indicate we are working on what really matters to the community
  – Cons
    • Might encourage creating unnecessary issues
    • What if it was a joint effort?
More KPIs

• How many tickets/issues we created got resolved by the community
  – Pros
    • Indicates we leveraged the community to our benefit
  – Cons
    • What if we have different interests?

• How many key positions do we hold?
  – How many PTL, committer, Release manager positions are filled by people from our organization
  – Pros
    • A good indication of our clout in the community
  – Cons
    • Arms race with other organizations
    • Not putting the communities best people forward
Let’s get more creative

• How many proposals/blueprints we blocked
  – If our goal is to prevent the project going in some direction, let’s count actions that support this goal
  – Pros
    • Indicates how we are actively promoting our goals
  – Cons
    • May be hard to measure – What if multiple organizations objected?
    • Lets not make it into a habit

• Counting knowledge base contributions, security vulnerability reports
  – Pros
    • Encourage contribution in those less desirable areas
  – Cons
    • Risk of over doing it
Measuring indirect contributions and external metrics

- Counting other contributions to the community
  - Hosting and sponsoring events
  - Donating labs
  - Posting supportive publications

- Count external KPIs
  - Mentions. In the press, blogsphere, or elsewhere
  - Measure adoption of the open source project by our own organization (how?)
  - How many of your customers/partners adopted the project (But how do you determine your part in it?)
Should I stop measuring commits?

• Not necessarily

• Use it together with other KPIs but treat it as a “pulse check”. One of many vital signs

• Try to avoid the pitfalls
  – Don’t motivate developers to create unnecessary commits
  – Look at things like trends and anomalies, not absolute numbers
  – Try to filter out sporadic events (seed code contribution, some sweeping changes across the project, etc.)
Open discussion

• What is your organization measuring?
• What do you think about the metrics discussed today?
• What else should we look at?
Thank you!

Ranny Haiby