Observation: More typed input on matrix in this session versus oral input and conversation compared to the morning CS contributions.

What questions are the same?

Getting contributors--both groups need a population of contributors.

What are the skill levels of the contributors? How do we train the participants we have?

Goals:

Getting Skilled (remove?) Contributors

- How do you attract people?
- What kinds of contributors are needed during different phases of the project.
- Marketing efficiency (click-through rate); are there other metrics that are more appropriate
    - Measuring the skill--how do you measure the skill level of participants?
    - Individual perceptions of skill vs. project managers perception of skills
        - Assumptions of skill based on community involvement and contribution. How does that apply to CS?
        - People metrics vs. project metrics
            - Whole new object with a new variable when you begin to look into people metrics
            - Not sure there's enough time to evaluate at the people level vs. project level. (higher level may be more appropriate given the time).
- What are the needs of projects based on where it is in development.
- When do you need new contributors? Healthy level of churn needed to prevent stagnation.

Train, Socialize, Onboarding of Participants

- How do we measure skill level?
- How do you know when your training is effective?
- Contributor questions may match to different stages of the project
    - Every step of the process is going to have specific error messages so can be measurable.
    - CS has really struggled with assessing participant activity and skill level.
        - Method papers on how to measure and study participant activity assessments.
    - There is overlap, just need to look at the right way to look at activity metrics.

Borrow Skilled People from Other Projects/Arenas

- Larger universe of projects how do you onboard participants without cannibalizing other projects.
    - Are you enticing enough to attract waning participation.

- What are the best practices and do they compete with one another.
- Forking happens in OSS and CS
  - Siphoning not a good thing
  - Forking is NOT sharing of best practices, but is the feeding back in; just because you have a shared origin does not mean that the projects will differentiate or remain active
    - It's the final check for project governance.
    - Other motivations to fork--your fork is the group's; personal satisfaction, personal credit.
- How much overlap/redundancy is there?
  - Reuse of protocols, code, standards
  - Similar projects; healthy level of churn
- Movement between projects
  - Social network analysis

Community Handoff/Retirement

Have we lost track of the issues...conversation has moved into more of a people-problem rather than the technical.

Best practices/standards/etc in CS vs. OS

- How do science teams integrate participant contributions into work.
- Easy to get lost in CS; micro-contributions from OS = dark matter of open source that do not flow upstream.

Is it okay to have some projects fail to promote project health. Which type of health and sustainability are we focusing on? Is what the project produces useful for the platform?

Some subsystems are more about doing computer science, more biological, etc.--it's sexier than other forms of OS.

Who are the metrics for? People want different health metrics based on perspective!