

Research Software Engineering Team Management and Scalability?

Test Subtitle

Hartwig Anzt^{1,2}, Terry Cojean¹, Fritz Göbel¹, Thomas
Grützmacher¹, Pratik Nayak¹, Tobias Ribizel¹, and Yu-Hsiang
Tsai¹

¹Karlsruhe Institute of Technology

²Innovative Computing Laboratory, University of Tennessee,
Knoxville

Summary

This is a temporary title, please change it.

There are maybe two central ideas: - RSE team scalability in terms of engineering tasks and role attribution and sharing per person (see Tobias roles, to which we can add scientist and) - RSE team scalability in terms of governance, financing, PR and organization (single PI model doing most PR/financial aspects etc maybe doesn't scale above a certain threshold).

With possible solutions: - Creating a position similar like a scientific manager person, able to manage role attribution of people and help with proposals, etc. Issues: should be **very** flexible and keep research freedom nonetheless (it's not 100% a company). - Flexibility in team organization (multiple persons are both proposal pushers, code reviewers, developers, ...), but requires strict and coherent personnel scheduling.

TC: initial exposition

I guess yet another idea is something we discussed with Hartwig sometimes. It's related in general to the modern research team dynamics, pitfalls and potential ways to cope with them, but that's maybe more an idea for Hartwig if he wants to try it.

The base idea is that a scientific research team which has a reasonable amount of success looks a lot like a company. You need to do a lot of PR (websites, ...), raise money (proposals), manage increasingly complex budgeting and hiring

processes (with administrative barriers as well).

For now, the PI usually tries to do (most) of these, which works up to small enterprise level, like <10 people and more brings big scalability issues. Making people who supervise subprojects deal with all of this can also not scale as it makes more people (collectively) lose their time, so there would be a point where it doesn't payoff anymore.

I think a conclusion could be that either we need more adaptability in the scientific teams, but then also much better collective planning so that relevant people help push the different aspects in a timely manner (see Tobias' roles outline in the thread!), and/or we greatly need a profile which would be a mix between manager and proposal pusher, able to help push scientific proposals (e.g. build a full proposal from an outline), maybe do some form of loose project/"employee" scheduling (manager-ish tasks) as well as deal with administrative and PR aspects of the projects (website update etc)?

TR: Clarification and people's roles and responsibilities

Maybe it also makes sense to talk about roles and responsibilities, like technical administration (CI stuff), maintainer work (releases etc.), QA (tests, static analysis, reviews, ...), support (issues, feature requests, ...), architecture (library design), documentation (technical writing, examples), management (who works on what)

TC (details)

As for your roles, that is I think a great granularity for scheduling people around if there would be a form of manager, basically we are all somewhat QA, support, scientists and more, and I feel like we would greatly benefit if we were able to optimize the amount of time each person spends on QA, support, science, etc...

Otherwise, it's very easy to get lost into just one role (e.g. QA is I guess the biggest offender here) and not able to push the rest.

Another idea: you could even do equivalents of some company direction roles, like at some point a research team if it keeps growing could eventually become more like a complete lab, in which case it might makes sense to split the main leading position into equivalents of CFO, CTO and CEO for example, if the PI cannot cope with all of this alone.

References