

Structured and Unstructured Teams for Research Software Development at the Netherlands eScience Center

Carlos Martinez-Ortiz, Rena Bahkshi, Yifat Dzigan, Nicolas Renaud, Faruk Diblen, Berend Weel, Maarten van Meersbergen, Niels Drost

The Netherlands eScience Center

The Netherlands eScience Center is the Dutch national center of expertise for research software engineering [1]. Together with our academic partners, we develop open-source software and apply these tools to concrete research questions. Our projects cover the entire research landscape ranging from the complex data mining of historical events [2] to large scale computation for climate science [3]. These projects can also drastically vary in size with some large-scale projects requiring the combined effort of multiple research software engineers (RSEs) [4] to small consultancy projects where RSEs guide external team in their research and development activities [5]. Alongside support staff and managements, the Center has a pool of about 40 RSEs who work on about 50 projects in partnership with research groups across the Netherlands and beyond. To cope with the growth of Center and the complexity of its ambitious mission, teams have started appearing to facilitate and organize the execution of our projects. This white paper presents the different types of teams that are currently in place and offers our own conclusions regarding what type of team suits best the different types of projects and more importantly the people working at the Netherlands eScience Center.

eScience Team Zoology

Having a bottom-up origins, the teams that have emerged at the eScience Center have all defined their own approaches to distribute the work among the team members, support each other and collectively advance research through software development. This has allowed the Center to experiment with different formats of research software teams, ranging from a single RSE assigned to a given project, to large, structured groups of RSEs working on several projects. Regardless of the format all teams interact with external stake holders, such as group leaders and PhD students that can also take part in the software development efforts.

We do not provide here a detailed presentation of every team, and instead present an overview of the different types of teams that we are experimenting with. We also avoid giving a strict definition of what constitutes a team and focus instead on the different ways of working that may suit the needs of different projects and people.

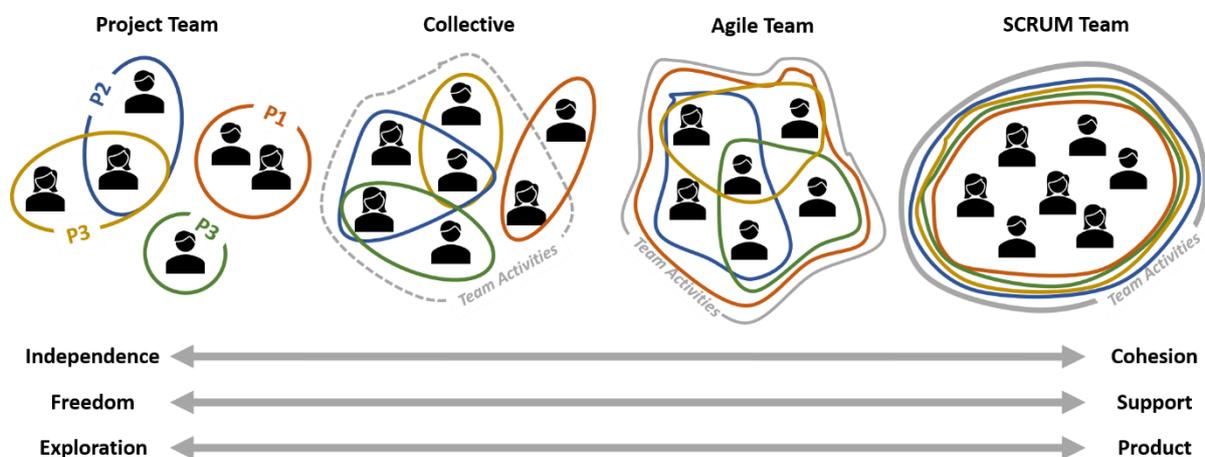
Project Teams consist of 1 or 2 engineers working on a specific project. The sole purpose of the team is the realization of that project, and the team naturally dissolves itself when the project ends. The realization of the project is usually not the only focus of the engineers, who split their time between different projects and therefore different Project Teams. The team members meet when needed to update each other and plan the development of the project. Sprint and pair programming sessions happen sporadically, the team members usually preferring working asynchronously on the project to avoid scheduling issues.

Collectives consist of a group of loosely connected engineers working on a set of similar projects. The similarity between the projects and the frequent interactions between all team members through morning stand up, facilitate cooperation through design sessions, pair programming sessions, code reviews. The execution of a given project is however done by the 1 or 2 engineers working on it with minimal contributions from the rest of the team. The loose connection between the team members allows for non-team members to contribute to projects that are executed in the team. Team wide activities, such as one week sprint or learning days are organized when opportune.

Agile Teams. A multitude of teams have adopted the Agile philosophy to organize their work and improve team dynamics. These teams vary in sizes ranging between 4 and 8 and prefer having team members commit most of their time to the work of the team. Each team works on several projects that generally have some overlap in terms of domain, technology, or both. Daily stand-ups are used to plan the work and keep the entire team updated about the progress of the projects. While collaboration is greatly reinforced by the tight team dynamic, not all team members work on the same project at the same time. Some Agile teams are working in time-constrained sprints of generally two weeks while others prefer a looser format for example based on the use of a team-wide Kanban board.

SCRUM Teams. Building up from the Agile philosophy, some of our teams have adopted, more or less strictly, the SCRUM methodology to organize their work. Following this approach, SCRUM teams work in 2-weeks sprints during which all team members work on a single project. The sprints start with an extensive planning session and ends with a sprint review and a retrospective. Daily stand ups are used to continuously update and fine tune the execution of the project. The release of a new feature or product is usually made at the end of each sprint. Team members have clearly identified roles, such as product owner, scrum master, etc ... providing clarity and allowing to distribute the responsibilities among the team.

Not all teams fit exactly in the classification presented here, many teams being somewhat in between two types. It is also possible for any team to change its modus operandi either permanently or temporarily: for example, several Project Teams teaming up as a SCRUM for the execution of a single sprint, or the members of an Agile team deciding to work independently of each other on personal projects for a week.



Overview of the diverse types of teams used at the Netherlands eScience Center ranging from unstructured project teams to structured SCRUM teams. All types have pros and cons and suit better diverse types of projects and people.

What Works When for Whom

Our experience has confirmed that there is no one format which works in every situation. Instead, each format has its own advantages and disadvantages, and different formats are suitable for different project needs and different personality traits.

Teams & Projects. Due to the different funding instruments we use, our projects vary in scope, duration, and hourly budget. Small explorative projects seem to fit better in unstructured teams such as Project Teams or Collectives. This format allows for ample time to clearly identify what the end goal of the project should be and explore different strategies and approaches. These explorative tasks are better performed by 1 or 2 engineers as they require a significant burn in period. These unstructured formats however do not facilitate the development of final products and often focus on prototyping and/or specific improvement of an existing software.

Large projects fit naturally better in structured teams such as Agile or SCRUM teams. The large contributions necessary for these projects require the high degree of synchronization that a tight team dynamic can provide. Team members are continuously aware of each other activities allowing to quickly alleviate bottlenecks and to ensure that their respective contributions are aligned. Smaller projects with a high degree of similarity can also benefit from being clustered together within such a team. When the overlap between these projects is sufficient, team members can easily contribute to several projects emulating a large project with smaller components. In addition, projects that are more product oriented fit also very well in structured teams. There, the tight collaboration between team members improves code design and maintainability. These structured teams generally require more planning and therefore an additional project management effort.

Teams & People. Another aspect that should not be underestimated, is that different formats fit better with the personal preferences of different individuals: a format that is inspiring and productive for some, may be completely unnatural and counterproductive for others. Attention to personal preferences, continuous reflection on the working format, and flexibility are some of the key ingredients for structuring a good team.

Unstructured teams such as Project Teams offer a lot of independence to the team members. Each member can plan their work independently and decide in which direction the project should go. This offers a lot of freedom to the research engineers that can rapidly develop prototypes and quickly explore and test different ideas. However, these unstructured teams provide little cohesion between the team members that may feel isolated in their work. In addition, members of unstructured teams find little support in their teammates as they are not deeply involved in each other work.

Structured teams provide a very cohesive working environment and give a feeling of belonging to the team members. This allows team members to truly support each other and to distribute responsibilities among all the team members. However, the structure of the team limits the freedom and independence of each team member as they are fully committed to the work of the team. This can in the end decrease the sense of ownership of the team members.

Making the best of both worlds

All the different team structures briefly presented above have pros and cons and can provide the perfect working environment or a never-ending hell for different people. Frequent and open discussions between members as well as with their line manager are crucial to identify personal preferences and find the best team for everyone. We should therefore not seek to unify the inner working of every team but let each team find that for itself.

Having these diverse types of teams working alongside has even been a great asset for the eScience Center. As an example, the Integrated Omics project [6] started as a very research oriented academic project for the exploration of machine learning techniques for understanding the interactions between microbes and human cells. A Project Team constituted of a single engineer was set up to work on the project. During the exploration phase the Project Team experimented with word2vec [7] a method originating from natural language processing. The success of the prototype prompted a consolidation effort of the initial code that was carried out by a SCRUM team. This SCRUM team significantly improved the code design and maintainability of the initial prototype enabling the adoption of the methods by a large community of bioinformaticians [8, 9]. This success would not have been possible without the collaboration between structured and unstructured teams within the same organization.

References

- [1] <https://www.esciencecenter.nl/>
- [2] <https://www.esciencecenter.nl/projects/evidence/>
- [3] <https://www.esciencecenter.nl/projects/stochastic-multiscale-climate-models/>
- [4] <https://www.esciencecenter.nl/projects/ewatercycle-ii/>
- [5] <https://www.esciencecenter.nl/news/the-magic-of-machine-learning-2021-call-winners-announced-2/>
- [6] <https://www.esciencecenter.nl/projects/integrated-omics-analysis-for-small-molecule-mediated-host-microbiome-interactions/>
- [7] <https://en.wikipedia.org/wiki/Word2vec>
- [8] <https://github.com/iomega/spec2vec>
- [9] <https://github.com/matchms/matchms>