Saoirse Siobhan Ebert

Software Development Tools for Large Teams

Working as a part of a large software development team requires project management tools and techniques that foster communication, organization, and collaboration. Tools for editing, maintaining and accessing source code, continuous integration, software testing, and workflow management need to be considered and utilized in a way that makes sense for the project at hand.

At the core of a successful team is a commitment to a software development methodology, such as Agile. In Agile, there is less focus on structure in favor of iterative processes which test the software being continuously developed by breaking tasks down into sprints[[1]](#endnote-1) which may be ineffective for large teams. Therefore, for Agile methodology to be successful, a large company or any other large group of individuals will need to be separated into multiple teams[[2]](#endnote-2). As a general guideline, the ideal team would be small enough such that two pizzas would be enough to feed them for lunch[[3]](#endnote-3). Other SDLC models can be used such as Lean, Waterfall, Iterative, Spiral and DevOps. Another example is Feature Driven Development, which is based on Agile, is strictly used for larger teams and allows them to work on multiple tasks simultaneously by breaking tasks down into smaller tasks.

For a project or group of projects to be successful, there needs to be defined roles. The Project Manager is needed to make key decisions based on the scope of business requirements. However, a software development team using Agile will also assign Scrum masters if they use Scrum framework as a project management tool. In some cases, the responsibilities of Scrum master and Project Manager overlap. In addition, other roles will be defined such as Product Owner and Development team. UX/UI designers, Software architects and developers, Software testing engineers, Test automation engineers, and DevOps engineers are just some of the roles that can be included[[4]](#endnote-4). Gantt charts can show the progress of the Sprints in which a Scrum team completes a set amount of work[[5]](#endnote-5). However, some project management philosophies do not employ the Scrum methodology and instead use Kanban boards, which visualizes tasks rather than provides a set structure to the team[[6]](#endnote-6). In this case a pull system is utilized to control the flow of work by replacing the completed tasks with new ones and therefore there are no “works-in-progress” which is continual in Scrums (i.e. there are no “sprints”).

There are a handful of different collaboration tools which can aid teams plan and communicate effectively during the lifecycle of a software development project. In smaller teams, Wrike, Jira, Asana and Trello are among a vast array of project management platforms which manages the productivity by tracking time and assigning tasks[[7]](#endnote-7). They may include templates to make Gantt Charts and Scrum and Kanban boards. They also integrate well into other third-party applications such as Google Drive, Dropbox, and Gmail[[8]](#endnote-8).

In addition, large teams may also utilize online code editors, such as Github Codespaces and AWS Cloud9, which allows teams to collaborate across multiple devices. This also is helpful as teams increasingly become remote and tend to expand work being across several time zones because of offshoring or nearshoring of projects. Version control tools are also important as teams become larger so that as source code is edited it can be updated to be accessed by the software development team through a repository. Git is often used over SVN because a local copy is made for each user and therefore if the main repository is compromised it can be restored from any of the user’s local copies[[9]](#endnote-9).

Another consideration is how to test the software being developed. With any large team, bugs obviously needed to be addressed as soon as possible. Automation is a proactive strategy utilized to alert the developers what needs to be fixed. A Quality Assurance team should be formed to execute this purpose. Another form of automation relates to Continuous integration and continuous delivery. The goal of these pipeline tools is to allow developers to build and package applications more consistently and accurately[[10]](#endnote-10). They are often integrated into project management systems like the ones described above.

For a large software development team, or group of teams, to be successful, it is important to consider the tools needed to continuously deliver quality software products that executes the vision of the project and satisfies the needs of the stakeholders. Designating the team roles, deciding upon a workflow management system and the tools needed to execute that methodology are all considerations to take when developing and testing software.

1. Nikolaieva, Aliona. “8 Best Software Development Methodologies.” *Uptech Blog*,Uptech, https://www.uptech.team/blog/software-development-methodologies.  [↑](#endnote-ref-1)
2. Emmons, Patrick. “How to Divide Your Scrum Team.” *DragonSpears*, DragonSpears, 1 Sept. 2021, https://www.dragonspears.com/blog/dividing-scrum-teams.  [↑](#endnote-ref-2)
3. “Top Agile Pitfalls: Why Agile Doesn't Work for Large Projects.” *Top Agile Pitfalls: Why Agile Doesn't Work For Large Projects*, https://www.sigconsult.com/blog/2019/12/why-agile-fails-in-large-enterprises-and-how-your-team-can-avoid-the-same-fate?source=google.com.  [↑](#endnote-ref-3)
4. “Software Development Team Structure: Roles & Responsibilities.” *ITRex*, 15 Nov. 2022, https://itrexgroup.com/blog/software-development-team-structure/.  [↑](#endnote-ref-4)
5. Ramos, Diana. “Using Gantt Charts for Agile.” *Smartsheet*, 27 July 2021, https://www.smartsheet.com/content/agile-gantt.  [↑](#endnote-ref-5)
6. “What Is Kanban Methodology: Introduction to Kanban Framework.” *What Is Kanban Methodology | Introduction to Kanban Framework*, Kissflow, 24 Aug. 2022, https://kissflow.com/project/agile/kanban-methodology/.  [↑](#endnote-ref-6)
7. “Kanban vs. Scrum: What's the Difference?” *Coursera*, 10 Aug. 2022, https://www.coursera.org/articles/kanban-vs-scrum.  [↑](#endnote-ref-7)
8. Payne, James, et al. “Best Project Management Tools for Developers.” *Developer.com*, 27 Oct. 2022, https://www.developer.com/project-management/best-project-management-tools-developers/.  [↑](#endnote-ref-8)
9. Stickman, Nathaniel. “Git Vs SVN: Pros and Cons of Each Version Control System.” *Linode Guides & Tutorials*, Linode, 13 May 2022, https://www.linode.com/docs/guides/svn-vs-git/.  [↑](#endnote-ref-9)
10. “Continuous Integration.” *Scaled Agile Framework*, 22 Dec. 2021, https://www.scaledagileframework.com/continuous-integration/.  [↑](#endnote-ref-10)