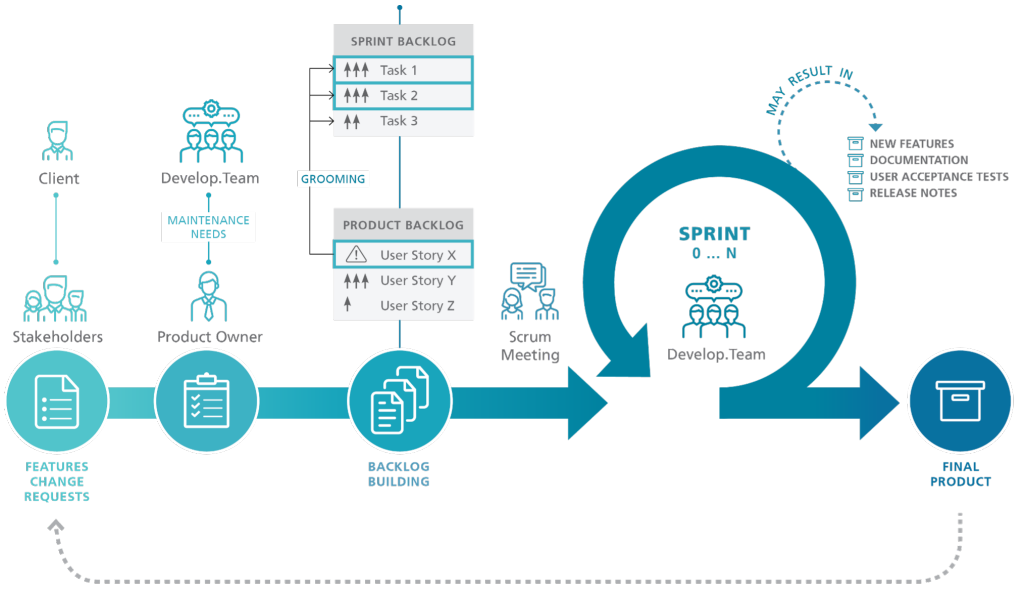
Fraunhofer PORTUGAL

FRAUNHOFER CENTER FOR ASSISTIVE INFORMATION AND COMMUNICATION SOLUTIONS – AICOS

OUR METHODOLOGY FOR AGILE SOFTWARE DEVELOPMENT

CONNECTED THINGS

Rigid software development methodologies long ago started giving way to different types of so-called agile methodologies, and it is easy to understand why. Under older methodologies, the planning is performed at the beginning of the project, followed by the development phase, without any intermediate deliveries to the Client. Often this resulted in the final product being far away from the Client's expectations. On the other hand, by using Agile Software Development Methodologies, teams are able to cope with any necessary changes, to increase Client satisfaction and match their expectations. These methodologies can be applied to a Research and Development environment, Scrum being our favourite.



1 Mobile app development workflow.

From research to industry

Our main goal is to be able to get the result of our research and development activities deployed in the industry, empowering our Clients with the ability to do new things, to achieve new goals, or to optimise their processes. For this to happen, we need to know our Clients' needs and how our technology can be used and tailored to meet them. During an initial phase, our Human Centred Design experts identify the users as well as their profiles and needs and design the system that better suits them. This knowledge is then used to build the User Stories that picture the system's functionality. The User

Stories make up the first iteration of the Product Backlog, which is the record of the tasks the Product Owner wants to hand over to our development team, ordered by priority. When the development team wants to pick a task to start working on, it chooses the top task of the Product Backlog. The Product Owner gets the role of being the Clients' representative, and therefore must know about their needs. The Product Owner's close relationship with the team is crucial for the project's success, as this is the role responsible for making sure the project is taking the direction that most fits the Client's needs.

Product Backlog Refinement

As the tasks on the Product Backlog start appearing, the development team needs to start breaking them down, as these tasks are the Product Owner's overview of the project and may not fit the granularity the team needs to start working on them. At this point, the team starts the process of Backlog Refinement - identification of all the smaller sub-tasks that have a defined goal, scope and duration, and are eligible to be assigned to a Sprint. This process comprises several iterations, as much as needed, to guarantee that there are always enough refined tasks to feed the full duration of the Sprints ahead.

On with the Sprints

A Sprint is a time-boxed period, typically lasting one or two weeks, on which the team agrees and commits to attaining a given goal. During each Sprint the team works on a collection of tasks, which make up the Sprint Backlog. By following the Product Backlog priority and the Refinement process, we develop the Sprint Backlog. As an output of a Sprint, among other artefacts, typically new features, updated documentation, or new releases are found out. It is this small development cycle that allows the change of project goals or priorities as the Product Owner sees fit, that grants so much flexibility to the Agile Software Development Methodologies. As such, this implies higher success rates and increases the number of expectations met.

To mark a set of tasks as finished, we follow a pre-defined and agreed set of steps: the Definition of Done. This concept can assume the form of a checklist, or it can be a more complex workflow, and it dictates the status of a task. Our approach for the Definition of Done takes all the testing activities into account, which includes test case generation, User Acceptance Test generation and automated testing (or manual testing, if it is not applicable).

Delivering to the Client

We deliver our artefacts to the Client mostly on two fashions: according to a pre-defined plan agreed between the team and the Client, or when the Product Owner sees significant tasks ready on the Product Backlog. At this point, the team separates new features under development from the features that are ready to be delivered. From this point on, a final testing procedure takes place. The motivation is to perform integration testing (to assess that two or more given components communicate correctly with each other), system testing (to ensure that the system works as a whole), and regression testing (to check if previously developed features work as intended, after adding new features). Upon proper assessment, all the artefacts are made available to the Client – new software features, release notes and User Acceptance Tests sheet.

Change requests and Maintenance

During the life cycle of the project, the Client may need to change the previously agreed goals, which means that the work planned so far may have to be reversed. Since the development cycles last one or two weeks, we can address this issue. Upon discussing the implications of the changes being asked and establishing the quotation (in terms of work hours and therefore cost) of the new requests, the corresponding User Stories may be added to the Product Backlog and ordered according to the Product Owner's priorities.

Changes may also occur in the environment that the software solution runs on. On this instance, the Client may not be aware of such changes as these may be too technical, so it is the development team's responsibility to trigger the fix. After the implication of such change is discussed and agreed by all the stakeholders, the Product Backlog may be updated and ordered as usual.