

Swarming Success: Optimizing Efficiency in Statistical Programming

Lea Zittlau, Chrestos Concept GmbH & Co. KG, Essen, Germany

Dr. Saskia Ständler, Chrestos Concept GmbH & Co. KG, Essen, Germany

ABSTRACT

Documentation of tasks and study-related information for statistical programmers is often scattered across multiple locations, such as Excel trackers, OneNote pages, Word documents, or Teams channels. This has proven to be a time-consuming process that leads to information being overlooked as well as difficulties in communicating with team members. In today's fast-paced world, it is increasingly important to organize work efficiently - especially when working on numerous projects at the same time. Agile methodologies, which have been common practice in software development for more than 20 years, have been shown to address this issue. In this paper, we demonstrate how statistical programmers can integrate agile tools into their daily work and collaborate with each other to coordinate programming activities. We also address pertinent questions related to motivating and training of project members to use agile tools.

INTRODUCTION

As statistical programmers, we are confronted with the situation of being on the daily search for documents, not knowing exactly where to find particular information. General information about the compound and certain timelines or meeting minutes might be stored in OneNote pages, while the documentation of the development progress and validation strategy is conducted through Excel. Word documents, encompassing SAP and specifications, are uploaded to SharePoint, while Teams and Outlook are employed for the communication between team members. Of course, other applications can also be used for all these things, such as tools from Google or other providers. For the sake of simplicity, however, we will only refer to Microsoft applications in this paper. Such chaotic conditions of having multiple channels in use can lead to several issues. First, documents may be misplaced and probably not found while at the same time new information or requests may be overlooked. This not only leads to annoyance within the team but can also have an influence on timelines and submissions in general. Additionally, from a newcomer's perspective, this overwhelming amount of communication channels complicates the onboarding process and makes it challenging to ascertain and locate pertinent information. Second, the constant search for information is time-consuming which allocates less resources for proper documentation and communication. This, in turn, can lead to a deficiency in clarity behind decisions. Moreover, this lack of time becomes especially important when undertaking multiple projects concurrently, which is anyway a perpetual challenge. The need is clear: the establishment of an efficient organizational structure within a team of statistical programmers.

Efficient teams are defined by three characteristics: collaboration, communication and adaptability. They collaborate by each person taking up tasks that suit them best. All team members have established a clear and open communication helping them to respond quickly and adapt their plans when changes arise.¹ To implement these three characteristics all collaborators must be on the same page and everyone should understand that it is all about the outcome. Common hierarchies need to be broken down and there should be a clear division of labour so that tasks can be done simultaneously. Every team member should understand their role and should be empowered to make their own decisions. Agile methods are one approach to allow teams to work towards these characteristics and to be as efficient as possible.²

SCRUM AND THE SWARMING SUCCESS

The general principle of agile methods is that they divide large projects into smaller iterations to gradually get closer to the final result. One out of many models that follows the agile principle is Scrum (**Figure 1**). Here the project is divided into small repeatable iteration the so-called "sprints". These sprints can be of different lengths, typically one to four weeks long³, depending on many factors like the team size, the amount of work that needs to be done, and

the influence of parallel projects.⁴ To give a short overview, in Scrum, there are three distinct roles: the product owner, the scrum master, and the developer.⁵ In statistical programming teams, the product owner is assumed to be the statistician, the study analysts are the developers, and the lead analyst takes over the role of the Scrum master.

The first step of the scrum process is the sprint planning. In preparation for this, the product owner is responsible for organizing all product requirements into a product backlog. During the sprint planning, the team examines this backlog to determine how many of the backlog items can be completed within the sprint and by whom. This planning thereby sets the sprint goal that the team wants to accomplish. Throughout the sprint the development team holds a daily scrum to check the progress, to discuss problems and issues and whether they see the sprint goal at risk. Based on the length of the sprint itself the frequency of this exchange can be adjusted. In the sprint review meeting, the team presents the product increment that was developed during the sprint. Here, the users, like in our case statisticians or medical writing, can give feedback and can decide whether backlog items are really finished or if they need to be updated during the next sprint. Finally, a sprint retrospective is held with the Scrum Master, who functions as an independent observer, to discuss what went well and what could be improved in the next round. Then the next sprint follows.⁶

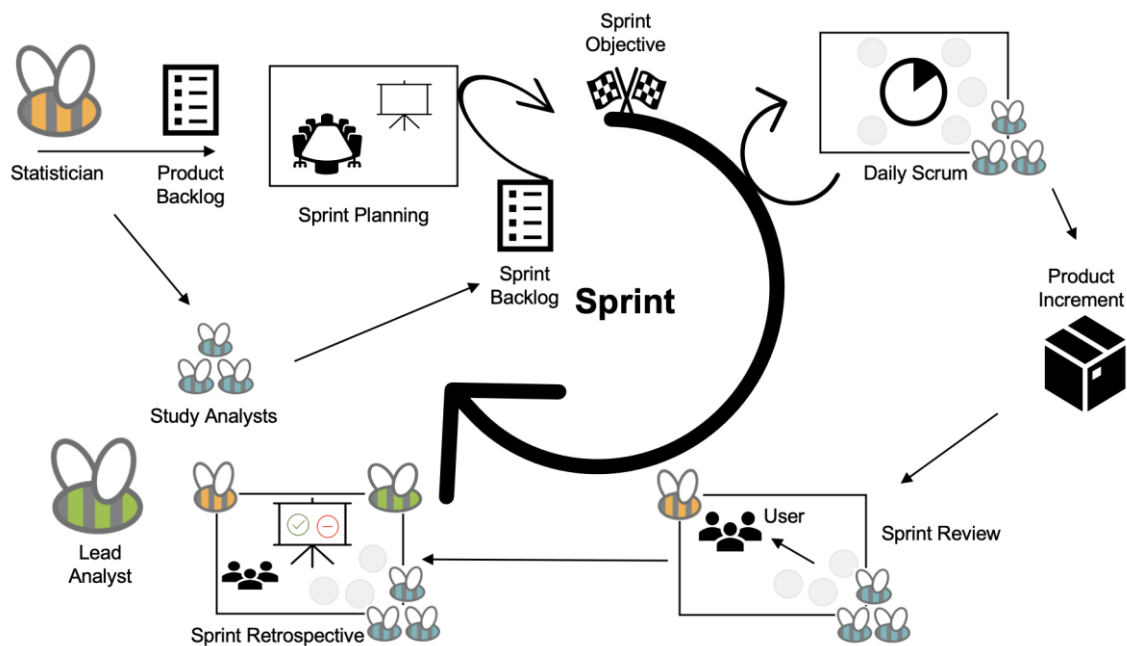


Figure 1: Schematic overview of the Scrum process.

The Scrum process is a very structured method to organize work efficiently, but due to several reasons it only partially fits into the daily work routine of statistical programmers. One example of such reasons is the unavoidable, simultaneous work on multiple projects where every project gets its own scrum process. Generally, spontaneous requests are manageable and can be considered while doing the sprint planning, even when parallel sprints are ongoing. However, a spontaneous health authority request from a finished project could jeopardize the current sprint goals. Another good example might be overly large, interdisciplinary teams. In that case, too much time is lost on things that may not even be relevant to some members, which impedes efficient working. On the other hand if teams are too small, the effort involved in all the meetings is unnecessary, as here everyday contact is already given. To overcome such issues, it is advisable to adopt and implement only those aspects of the Scrum method that are relevant for your team. Nevertheless, agile methodologies like Scrum offer a significant advantage in the form of agile tools or software like Axosoft, Taiga, Shortcut, GitLab, Scrumwise, Jira, OpenProject or AzureDevOps.⁷ This paper will focus on showing how these tools can be used in the everyday work of statistical programmers. AzureDevOps (from now on referred to as ADO) is used for demonstration purposes but any other tool for agile project management might also be suitable.

DEMONSTRATION OF AGILE TOOLS TO TRACK THE PROGRESS OF PROGRAMMING ACTIVITIES

Imagine we use an Excel worksheet to track the progress of programming activities of a fictional drug called “Beecycline” (**Figure 2**). The tracker below shows some example work items. Although the worksheet can be accessed and edited by several users at a time, this method is very static and just provides a limited overview. One won’t be informed when work items are completed, when new work items appear, or if there is a question regarding an issue which needs to be addressed. When having a complex study with many more work items and more people involved than displayed below, the tracker can get very crowded and confusing.

	A	B	C	D	E	F	G
1	Program name	Programmer	Programming status	Val. Level	Validator	Issues	Validation status
2	Datasets						
3	d_adsl	Lea	done	Double programming	Saskia		done
4	d_adae	Lea	in development	Double programming	Saskia		
5	d_adlb	Lea	done	Double programming	Saskia	baseline flag not right assigned, format for planned treatment is incorrect	in validation
6	Table & Figures						
8	t_demographics	Saskia	done	Review	Lea	race is missing as category, missing should be last category for all	in validation
9	t_abnormal_lab						
10	Listings						
12	t_adverse_events	Saskia	in development		Lea		

Figure 2: Development and validation tracker of the project on the fictional drug “Beecycline”.

Frequent checking of such crowded documents might be time-consuming. Furthermore, it may involve a constant back and forth via Teams or Outlook to confirm any document updates or discuss questions with the responsible colleagues. When discussions about certain items are finished, one needs to return to the document and do proper documentation. Sometimes during this transfer, information may get lost or not documented at all because time is running out. This is where the big advantage of ADO or any other agile tool comes into play.

In ADO there are several options to visualize the projects’ workload either in queries, on a dashboard, or in a Kanban board which offers a good project overview. There is no strict rule what needs to be used; one can simply choose based on taste or what’s best in a particular situation. First of all, queries visualize work items in a tree-like shape with parent-child assignments (see **Figure 3** below). For tables, figures and listings the structure is reminiscent of the specifications table of contents. One can see the type of work item, the title of the item, to whom it is assigned for now, the status and one can even display the sprint iteration. A big plus about these dynamic queries is that one can see all content immediately and even determine how far the whole process is by looking at the states of each item.

Speaking about work items, these are dependent on the kind of setup (how to do the setup will be shown later) one chooses, and what is meaningful within the respective workaround. For demonstration purposes we have chosen the following ones: **Epics** are a sort of higher-level items that we use for classification of sections like programming of datasets or the section of the TLFs report. **Product Backlog Items** are child items of these epics and give an overview about what needs to be done per section. In our use case, those backlog items are given the respective program names that are being worked on. Since all developed programs have their own backlog items it is possible to create so-called **bug items** as childs to its backlog item clarifying issues during the validation process. Lastly, there are **task items** which cover additional to-dos and general questions to the project.

Queries > My Queries > Study1 Run query +

Results Editor

ID	Work Item Type	Title	Assigned To	State
4	Epic	👑 Datasets		In Development
7	Product Backlog Item	📄 d_adsl		Done
8	Product Backlog Item	📄 d_adae	Lea Zittlau	In Development
11	Product Backlog Item	📄 d_adlb	Saskia Ständler	In Validation
18	Bug	🚩 Baseline flag not right assigned	Lea Zittlau	In Development
19	Bug	🚩 Format for planned treatment is incorrect	Lea Zittlau	To Do
5	Epic	👑 Tables & Figures		In Development
9	Product Backlog Item	📄 t_demographics	Lea Zittlau	In Validation
20	Bug	🚩 Race is missing as category	Saskia Ständler	To Do
21	Bug	🚩 Missing should be last category for all	Saskia Ständler	To Do
10	Product Backlog Item	📄 t_abnormal_lab		To Do
6	Epic	👑 Listings		In Development
12	Product Backlog Item	📄 l_adverse_events	Saskia Ständler	In Development
17	Task	📋 Clarify timelines for TLF submission	Lea Zittlau	To Do

Figure 3: ADO query interface.

Another option for workload visualization is the ADO dashboard where one has the option to choose an “Assigned to me” Widget (Figure 4, left side). This widget is one of the advantages of the dashboard, as it allows users to directly see which work items are assigned to them. Furthermore, one can add a markdown widget (Figure 4, top right) in which all study related information such as timelines or links to the SAP, TLF specifications and TLF drafts can be stored. This is especially helpful for new members as they can easily onboard into the study. Additionally, one can also add a “Sprint Overview” widget which gives a visual overview about the current sprint goal and its process (Figure 4, bottom right). Here, it is shown how many days are remaining for the sprint and how many work items remain unfinished.

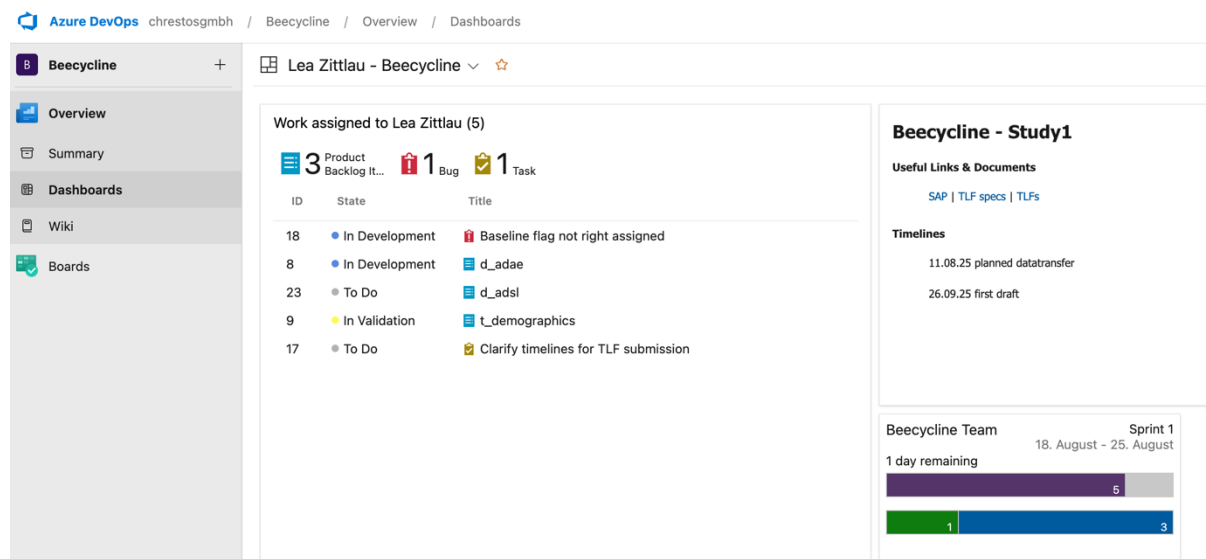


Figure 4: Display of ADO dashboard with its “Assigned to me”, “Markdown” and “Sprint Overview” Widgets.

For those seeking an alternative visualization of their work items, instead of looking into the dashboard or the query, they can also have a look into the Kanban board (Figure 5). It can be found under boards in the left-hand bar of the project page. The Kanban board favors the display of all backlog items in their respective states together with their child items so one can see quickly what still needs to be done. This is particularly interesting in sprint meetings as one can easily move items via drag and drop. The Kanban board also features another advantage as there is the option to highlight items from different studies by color code to get a better overview when working on several

studies simultaneously. In the figure below, one study is displayed in pink and the other in blue. Further, one can also apply filters like sprint iteration, work item status, or who the item is assigned to.

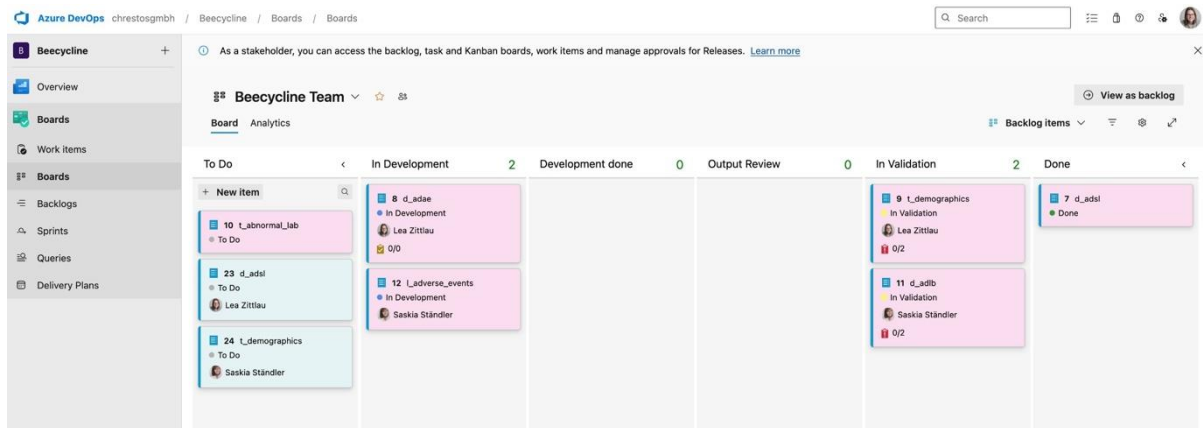


Figure 5: Interface of ADO Kanban Board with backlog items highlighted by color code.

When it comes to project organization and communication, ADO and other agile tools offers several options for structuring the team effectively. We already mentioned the dashboard markdown widget in which timelines, working locations, all team members and even all necessary links to documents can be summarized. This is not only favorable in case of onboarding processes but also for long-term team members as this reduces the search for information to a minimum. Via the assignments, the developer and validator will always be informed what exact tasks they need to work on as this can be automatically visualized in the aforementioned options. Moreover, one of ADOs many benefits is that you can directly communicate with team members within all created work items, which not only includes other programmers but also statisticians and medical writers by linking those in the discussion feed (as in **Figure 6**). One could argue that this is simply another communication channel but the advantage of structured and direct documentation of all decisions cannot be denied. Another counterargument could be that, similar to the worksheet tracker, one won't know if there is something that requires your attention or comment. However, this is not true. As long as the respective person is linked to the questions (as the statistician below), the person gets notified via email with a direct link to the discussion.

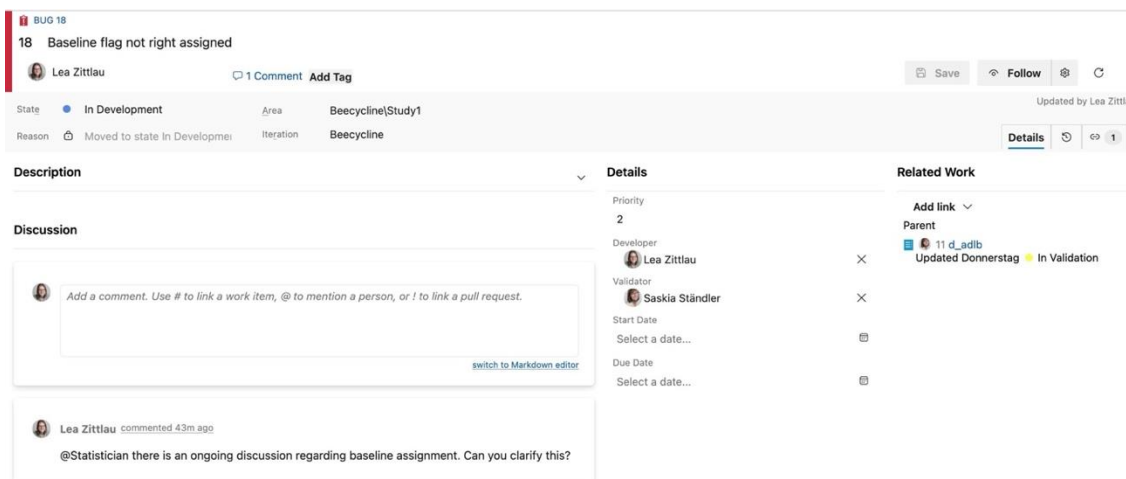


Figure 6: Bug Item with a started discussion feed in which the statistician is linked to a question.

HOW TO SETUP AZURE DEVOPS (ADO)

As we have now pointed out a few advantages of ADO and agile tools in general, let's dive into the initial setup and how to generate work items and set up sprint iterations. ADO provides several options to start a basic, agile, scrum or CMMI process to design your tool (**Figure 7**). To access all these options, one can click on the process button

under the settings menu. For the purpose of this paper, we have created a customized process called “Statistical Programming” which is based on the scrum process by clicking on the three dots highlighted below.

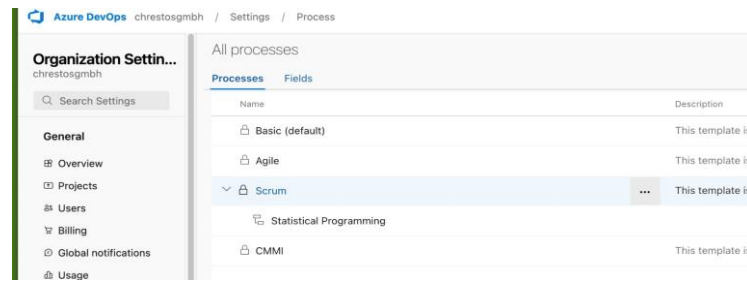


Figure 7: Overview of available process templates in ADO.

Next, by selecting the customized process, in our case “statistical programming”, one will be forwarded to another page showing all work item types that are preset for the respective process that was chosen (**Figure 8**). Out of this pool it is possible to pick items that should be integrated and adapt them, disable those not needed, and even create new ones.

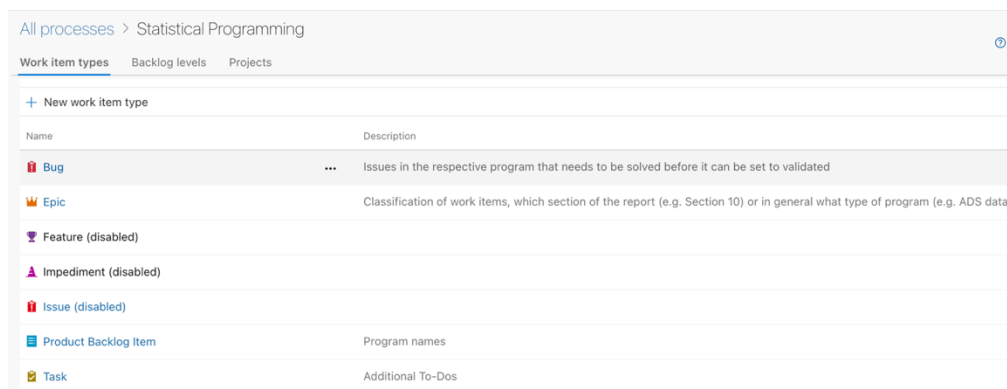


Figure 8: Overview of work item types in the “statistical programming” process.

Furthermore, each of these items can be customized for their indicated states (**Figure 9**). Our design reflects the following: The state “To Do” implies that nobody has taken care of the item for now. If someone picks up the item and starts with the development, the state should be changed to “In Development”. In case the development is finished, the state will be changed to “Development done” and another team member can pick up the work item for “Output Review” or “In Validation”. When the work item is finally validated, it can be set to “Done”. If a work item is not part of the analysis anymore, it can be changed to the state “Removed”, instead of being deleted, to maintain documentation.

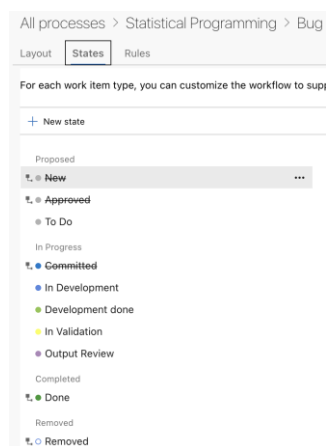


Figure 9: States that can be assigned to items.

Besides the status of work items, one can also change the general layout of each work item like epics or backlog items. As an example, an adapted layout of a product backlog item is shown below (**Figure 10**). For our purposes we wanted to have some more programming-related information on the overview page. Hence, under **Details** we have added “Level of Validation”, “Developer”, “Validator”, “Item Start Date” and “Item Due Date”. Under rules that can be specified for each work item in the settings area, we have classified that the assignment should automatically be changed to the Developer or Validator whenever a work item state is changed to “In Development” or “In Validation”. However, one can also add more information in the details area. Under **Related work**, a prerequisite is the option to link the work item to other work items. In the given example, the product backlog item “d_adlb” has been linked to the parent epic “Datasets” and to two child bugs that have been identified during validation. As good practice, we would suggest making use of these links as this helps a lot with navigating across work items. Furthermore, another good practice would be to give some general information in the **Description** area, for example the table title for designated TLF programs. An additional recommendation would be to put the links for ADaM and TLF specifications in the respective epic descriptions as one can navigate to them from any kind of child item. This is helpful in not getting confused if there are several studies to work on at the same time. However, the area description also shows to which study the current work item is assigned. Lastly, as already mentioned, we strongly advocate discussing all questions in the **Discussion** feeds of the respective items.

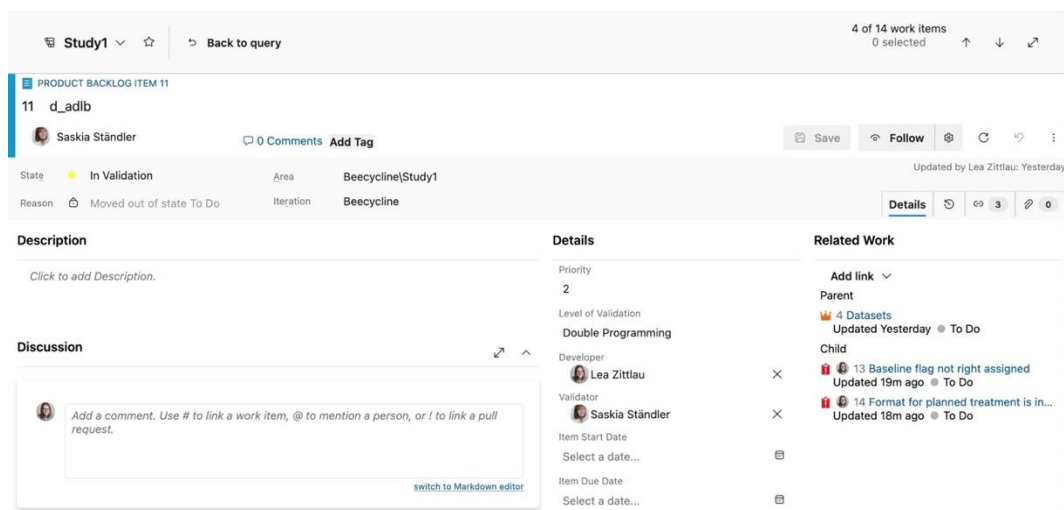


Figure 10: Overview page of a product backlog item.

Another key element that can be used in ADO is the sprint iteration. Setting up the sprints can be accessed via navigating to “Boards” and the sub-point “Sprints” on the left toolbar of the landing page. Here, the task board is automatically loaded and looks very similar to the Kanban board. Above this taskboard in the left corner, one can find a dropdown menu to filter for any sprint that is created. However, there is also the option to create new sprints. Once clicking on “new sprint” a pop-up window (**Figure 11**) is opened, and title and iteration dates can be added.

New Sprint

Name

Start

End

Location

Selecting existing iteration

Figure 11: Pop-up window to create new sprint iteration “Sprint 2”.

As soon as all planned sprint iterations are created, they can be used, for example, in a sprint meeting. Here, the backlog overview can be opened, and one can assign all work items via drag and drop to one of the sprints. In the figure below “Sprint 1” has been set up, which lasts for one week (**Figure 12**). Two backlog items and two bugs have been assigned to this sprint. As one of the bugs is in state “Done”, this appears as a green task on the dashboard (**Figure 4**).

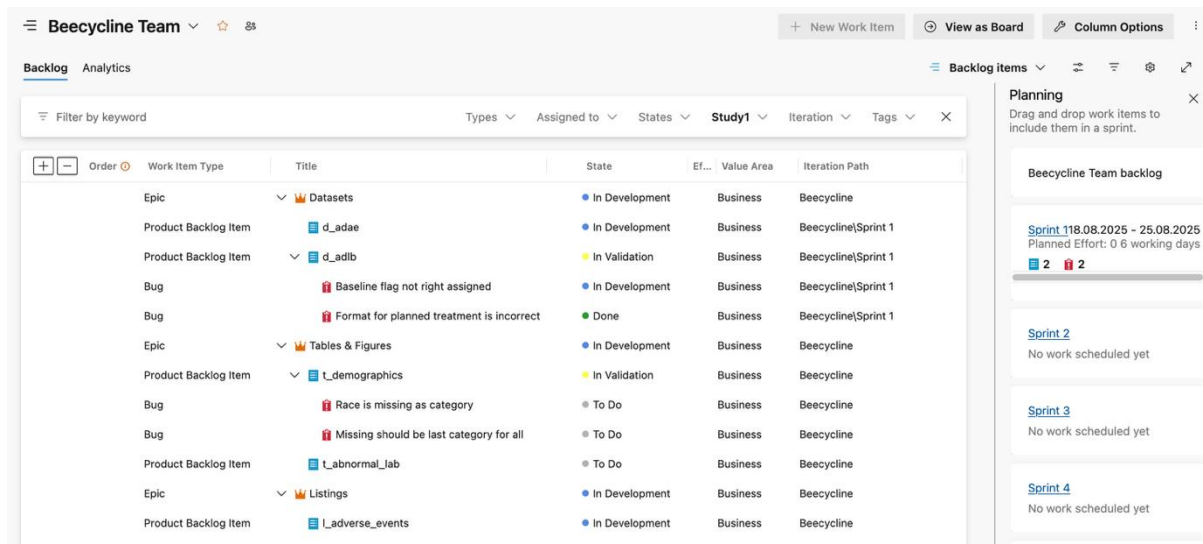


Figure 12: Overview of backlog items with planned sprint 1 shown on the right side.

TRAINING AND MOTIVATION OF TEAM MEMBERS

Now that we have discussed the advantages and implementation of agile tools, we would like to conclude by addressing a major pitfall in their implementation – the people. It is important to remember that some team members may not want to change their way of working and feel threatened or disrupted by such approaches. Here, the key to motivating and training other team members is to have a clear vision about the changes that are going to be implemented. This makes it necessary to answer questions about the why, the what and the how. Without an open, transparent and clear communication about all these questions, failure will be inevitable. Further, it should be made apparent on both sides that the implementation of new working procedures is a long-term process. This helps (new) team members get comfortable with the change, while also preventing frustration for leaders if the change doesn't happen right away.

Other important aspects about the aforementioned communication are the frequency and the development of a feedback culture. Imagine we are given the order to use agile tools, but how to use them is not properly explained and the solution which is built does not make much sense from a programmer's perspective. The majority of people probably won't invest any resources for using it in their everyday work and the whole process fails. The change leader needs to make sure that there is a frequent communication about it, for example via newsletters or blog entries which give updates on the process or examples about what can be done. Furthermore, by implementing a proper feedback culture, new team members who are getting onboarded into this process have the opportunity to help optimize this new procedure and consequently get more attached to it.

Another point about the training of other team members is that resources need to be invested. One can't expect people to understand and apply everything out of the blue, especially without having sufficient help. The change leader needs to have solutions for appropriate assistance, for example in the form of workshops or 1:1 mentoring and needs to plan resources accordingly. Of course, the decisive factors for planning are not only the time frame in which the process is to be completed, but also how many people need to be trained and how they feel about such changes. Particularly, the latter one is a cornerstone of planning for all of this. One needs to be aware that people have different engagement levels such as the so-called “early adopters” or the “laggards”.⁸ Early adopters, for example, do not need to be explicitly motivated to accept and apply changes but others, such as the laggards, might have concerns as it completely disrupts their way of working. Especially for those people, it is indispensable to be clear about the changes in the background and what can be expected from the change. Moreover, in such cases, the whole change process may be called into question. Here, it is especially important to be clear about the vision and stay committed as a leader.

CONCLUSION

Agile methods like scrum provide a way for teams to work as efficiently as possible which is increasingly important in today's fast-paced world. As statistical programmers, we often face the issue of unstructured work, and agile tools can help to address this issue. Compared to a worksheet which is used for tracking programming tasks, such tools provide a more dynamic framework and better visualization, especially for complex projects. Furthermore, linking documents to the respective issues prevents time-consuming searches as everything can be stored in one place. Using assignment options and discussion feeds can allow effective communication and eliminate the need for additional documentation. In conclusion, agile tools can help statistical programmers to structure their work and make their team more efficient.

REFERENCES

1. Scholtz, Mimi. "Agile Teams and Honey Bees: Building a Hive of Collaboration". <https://www.linkedin.com/pulse/agile-teams-honey-bees-building-hive-collaboration-mimi-scholtz/>. Accessed 25 August 2025.
2. TMG Consultants: "Erfolgreich organisiert seit über 25 Millionen Jahren: Was können wir von den Bienen lernen?". <https://www.tmg.com/blog/erfolgreich-organisiert-seit-ueber-25-millionen-jahren.> Accessed 25 August 2025.
3. Gurnov, Artem: "What Is a Sprint in Agile?". <https://www.wrike.com/project-management-guide/faq/what-is-a-sprint-in-agile/>. Accessed 26 August 2025.
4. Personio: "Agile Methoden: Erklärung und Überblick". <https://www.personio.de/hr-lexikon/agile-methoden/>. Accessed 26 August 2025.
5. ScrumAlliance: "The Scrum Team Roles and Accountabilities". <https://resources.scrumalliance.org/Article/scrum-team>. Accessed 26 August 2025.
6. Schneider, Sebastian: "Scrum einfach erklärt". <https://www.youtube.com/watch?v=lbz9STVjtzl>. Accessed 25 August 2025.
7. Miguel, Paulo Gardini: "25 Best Agile Software Development Tools Reviewed in 2025". <https://thectoclub.com/tools/best-agile-software-development-tools/>. Accessed 26 August 2025.
8. Cronkite, Wes: "Rethinking the Change Adoption Curve". https://www.asaecenter.org/resources/articles/an_plus/2017/september/rethinking-the-change-adoption-curve. Accessed 26 August 2025.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Lea Zittlau, Dr. Saskia Ständler

Company: Chrestos Concept GmbH & Co. KG

Address: Girardetstraße 1-5, Essen, Germany, 45131

Email: lea.zittlau@chrestos.de, saskia.staendler@chrestos.de

Website: <https://chrestos.de/>

Brand and product names are trademarks of their respective companies.