ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEAR03

# AGILITY IN MANAGING EXPERIMENT CONTROL SOFTWARE SYSTEMS

K. V. L. Baker, F. A. Akeroyd, T. Löhnert, D. Oram, ISIS Neutron and Muon Source, Didcot, UK

*Abstract*

Most software development teams are proponents of Agile methodologies. Control system software teams, working at science facilities, are not always just devel-opers, they undertake operations work, and may also be responsible for infrastructure from computer hardware to networks.

Parts of the workflow this team interacts with may be Agile, but others may not be, and they may enforce deadlines that do not align with the typical agile implementations. There is the need to be more reactive when the facility is operating, which will impact any development work plans. Similarly, friction can occur between an Agile approach and more familiar existing long-standing risk-averse organisational approaches used on hardware projects.

Based on experiences gained during the development of IBEX, the experiment control software used at the ISIS Pulsed Neutron and Muon source, this paper will aim to explore what being Agile means, what challenges a multi-functional team can experience, and some solutions we have employed.

## WHAT DOES AGILE MEAN?

In its' truest form Agile is a way of working [1] rather than the tools to enable this way of working. What is most important to remember is that the manifesto values certain things over others, but the less valued items are still worth considering, just not at the expense of the more valued items. The Agile Manifesto values are as follows:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

It was software developers that started the use of Agile methodologies, as it is often in the fast-paced world of software that the adaptability is most important in the modern age. However, some of the tools used to support Agile methodologies originated separately from software development, and Agile Project Management is slightly different to Agile Software Development.

### *What is Agile Project Management?*

The Association for Project Management [2] describe Agile Project Management as "an iterative approach to delivering a project throughout its life cycle" [3].

The Agile Manifesto and Principles are generally applied in exactly the same way whether the project in question is software based or not, and instead of working software at each iteration, you aim for working prototypes or solutions.

### *Is Agile Always the Right Answer?*

It is certainly true that not every project is suited to using an agile methodology. Yet, there are very few project teams who would deny that continuous collaboration throughout the project from the customer or their representative, and being able to incorporate any requested, or required, changes over the course of a long term project is beneficial to producing a usable item at the end. Some of the tools which are typically seen in Agile, have other roots. A common tool used to map workflows is a board such as a Kanban board (see Fig. 1) [4].

Initially the use of Kanban was by the assembly lines at Toyota [5], but the tool is used by most Agile methodologies to track progress.

Whilst Agile isn't the answer to everything, aspects of it, and the tool sets it uses are still applicable outside of Agile projects, and vice versa. If using any of the standard toolsets used by Agile Project Management, it is worth making sure you use the one that most suits your team and environment, and to continuously evaluate the tools suitability for use as the project and team develops.

## MULTI-FUNCTIONAL TEAMS

Whilst the ideal can be to have teams focussed on just one thing, the practicalities mean that people regularly undertake a variety of tasks.

### *Types of Function*

For the purposes of this paper a function is defined a family of tasks that an individual can undertake.

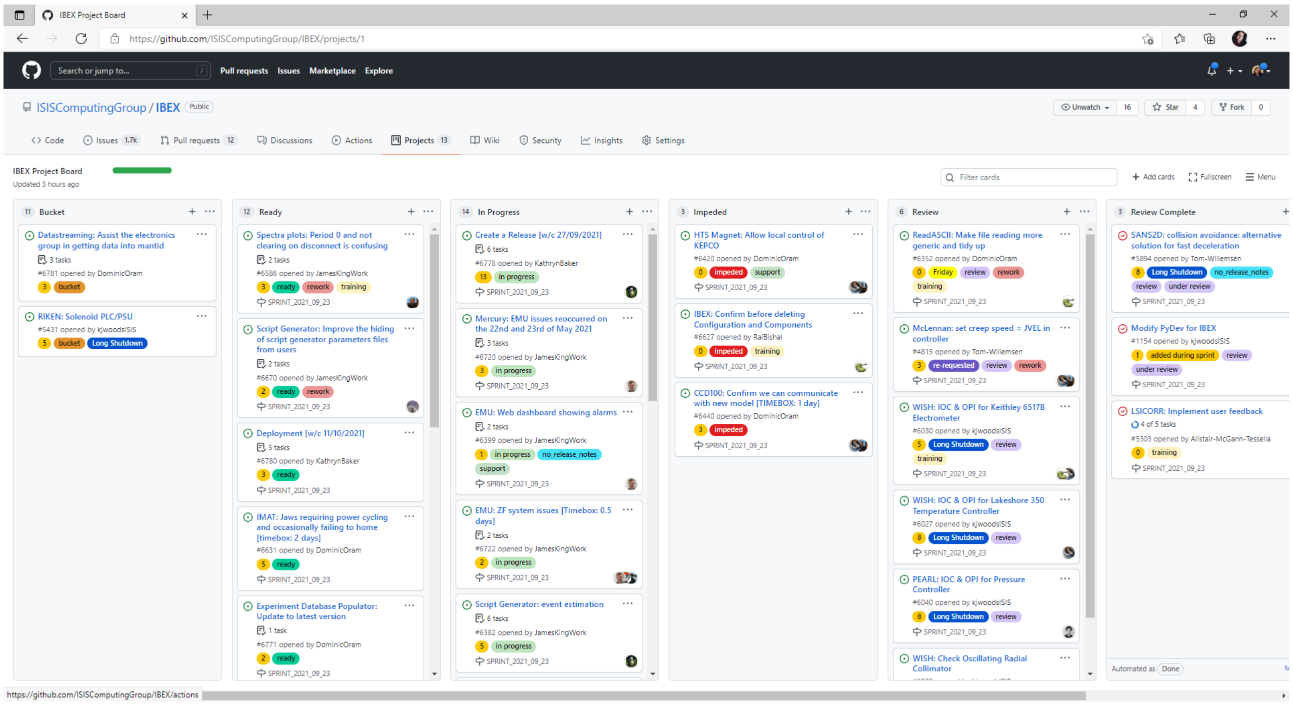
**Development (Dev)** The obvious task undertaken by most software developers is development. This mainly covers the introduction of new features to a code base, for example, adding in code to allow a user to change the colour of the interface they are using.

Development covers the full software stack, from the user interface to the lowest levels the team can support, which may even be circuitry. For non-software teams this could be electronic systems, or mechanical ones.

**Systems (Sys)** The focus for systems tasks is usually the hardware and fundamental aspects of the environment the control system is running in, e.g. computers, network switches, and operating systems. It also covers patching and updating the elements mentioned.

**Operations (Ops)** Operations tasks are responses to requests for support on problems that need to be solved in a short time frame to keep systems and software running correctly.

It also covers some of the later parts of the process, such as fixing code. For software teams this is hearing about bugs and dealing with them. It is more likely to be failures and faults for those who are not focussed on software, and is the nature of operations for the Systems function. As such


ISISComputingGroup / IBEX Public

IBEX Project Board
Updated 3 hours ago

11 Bucket
- Datastreaming: Assist the electronics group in getting data into mantid — 3 tasks — #6781 opened by DominicOram — 3 bucket
- RIKEN: Solenoid PLC/PSU — #5431 opened by kjwoodsISIS — 5 bucket Long Shutdown

12 Ready
- Spectra plots: Period 0 and not clearing on disconnect is confusing — 2 tasks — #6586 opened by JamesKingWork — 3 ready rework training — SPRINT_2021_09_23
- Script Generator: Improve the hiding of script generator parameters files from users — 2 tasks — #6670 opened by JamesKingWork — 2 ready rework — SPRINT_2021_09_23
- Deployment [w/c 11/10/2021] — 5 tasks — #6780 opened by KathrynBaker — 3 ready — SPRINT_2021_09_23
- IMAT: Jaws requiring power cycling and occasionally failing to home [timebox: 2 days] — #6631 opened by DominicOram — 5 ready — SPRINT_2021_09_23
- Experiment Database Populator: Update to latest version — 1 task — #6771 opened by DominicOram — 2 ready — SPRINT_2021_09_23

14 In Progress
- Create a Release [w/c 27/09/2021] — 6 tasks — #6778 opened by KathrynBaker — 13 in progress — SPRINT_2021_09_23
- Mercury: EMU issues reoccurred on the 22nd and 23rd of May 2021 — 3 tasks — #6720 opened by JamesKingWork — 3 in progress — SPRINT_2021_09_23
- EMU: Web dashboard showing alarms — 2 tasks — #6399 opened by JamesKingWork — 1 in progress no_release_notes support — SPRINT_2021_09_23
- EMU: ZF system issues [Timebox: 0.5 days] — 2 tasks — #6722 opened by JamesKingWork — 2 in progress — SPRINT_2021_09_23
- Script Generator: event estimation — 6 tasks — #6382 opened by JamesKingWork — 5 in progress — SPRINT_2021_09_23

3 Impeded
- HTS Magnet: Allow local control of KEPCO — #6420 opened by DominicOram — 0 impeded support — SPRINT_2021_09_23
- IBEX: Confirm before deleting Configuration and Components — #6627 opened by RaiBishal — 0 impeded training — SPRINT_2021_09_23
- CCD100: Confirm we can communicate with new model [TIMEBOX: 1 day] — #6440 opened by DominicOram — 3 impeded — SPRINT_2021_09_23

6 Review
- ReadASCII: Make file reading more generic and tidy up — #6352 opened by DominicOram — 0 Friday review rework training — SPRINT_2021_09_23
- McLennan: set creep speed = JVEL in controller — #4815 opened by Tom-Willemsen — 3 re-requested review rework — SPRINT_2021_09_23
- WISH: IOC & OPI for Keithley 6517B Electrometer — #6030 opened by kjwoodsISIS — 5 Long Shutdown review training — SPRINT_2021_09_23
- WISH: IOC & OPI for Lakeshore 350 Temperature Controller — #6027 opened by kjwoodsISIS — 8 Long Shutdown review — SPRINT_2021_09_23
- PEARL: IOC & OPI for Pressure Controller — #6040 opened by kjwoodsISIS — 8 Long Shutdown review — SPRINT_2021_09_23
- WISH: Check Oscillating Radial Collimator

3 Review Complete
- SANS2D: collision avoidance: alternative solution for fast deceleration — #5894 opened by Tom-Willemsen — 8 Long Shutdown no_release_notes review under review — SPRINT_2021_09_23
- Modify PyDev for IBEX — #1154 opened by kjwoodsISIS — 1 added during sprint review under review — SPRINT_2021_09_23
- LSICORR: Implement user feedback — 4 of 5 tasks — #5303 opened by Alistair-McGann-Tessella — 0 training — SPRINT_2021_09_23

Automated as Done

https://github.com/ISISComputingGroup/IBEX/actions

Figure 1: An example of a Kanban board, the one in use on the IBEX project at the start of October 2021.

operations teams can need an understanding of what both the systems and development work entails, although not always to the same depth.

**Project Management, Admin** Whilst it can be the case that project management and admin tasks are undertaken by specialists in those fields. It isn't always the case, and it can fall to the same team that is designing the software to control an experiment to manage the project and to keep the admin tasks related to that system up to date.

This can also be stakeholder management, ensuring that those communication and collaboration channels are kept open for dealing with our scientists.

### *What the Basic Challenges are for Multi-Functional Teams*

If you are able to split the work between team members with defined functions that do not overlap, then whilst the team might be multi-functional, the only challenges occur when those within a function are not available, and that can be the case for any specialist aspect of the control system.

For a team where each member can be involved in multiple functions, the challenges are often quite different. Not least of which is context switching. If a team member is undertaking some programming work, and is interrupted by a phone call reporting a failed network connection, there is a moment where they have to switch between thinking in a programming language, to how to go about troubleshooting what the issue with the network might be. This kind of switching between tasks has a cost [6], which can have a significant impact on the productivity of a team.

Similar to the context switching, if people are working on multiple projects then interruptions will have an effect on productivity too [7]. The issue can come on how you differentiate between development work and operational work, and just how many projects the team/individuals are working on.

It is also rare that a team member can be specialist enough in all those areas to be able to provide the same level of support, which can be frustrating for those seeking help of an operations function.

### *Where Agile can Help*

Due to the central precept of embracing change Agile methodologies allow for systems to change frequently, which will be of benefit to the ever changing needs of a science facility.

If you have individuals with specialist skills then it can be easier to manage their input across multiple projects.

### *Where Agile Hinders or is Counterintuitive*

The thing we have found causing the greatest friction for us relates to how things are defined as complete. Our iterative approach to development provides a Minimum Viable Product (MVP) and then adds to the product, which often leads to difficulty in defining when a product is in a finished state, effecting team morale.

With a facility such as ISIS, where the customer representatives (the scientists who take care of the instrument) work extended hours during the times that the accelerator is running it can be hard to find their time, or for them to prioritise the collaboration aspects needed for Agile working. Especially when combined with the iteration process.

For a multifunctional team however, Agile methodologies can increase context switching, as the next most important thing to do might be related to something completely different to what you were just working on, but or the developer can find themselves jumping from one programming language to another.

## HOW OUR TEAM FUNCTIONS, AND HOW WE ARE AGILE

### *What Type of Team is the IBEX Team?*

Within the realm of supporting control software on instruments at a facility such as ISIS [8], we cover all functions within the team. Some members of the team focus more on the Systems side, others on development, and still others will undertake the management tasks.

All members of the team take on an operations role, especially when the ISIS accelerator is running, as that is when the instruments are most active. As the available beam time is limited, we work to support the instruments to ensure that the time can be used to the best possible advantage.

The whole team is also encouraged to take on responsibilities for different parts of the system as their skill sets grow and improve, so that they act as a point of contact to a beamline.

The IBEX team could be described as a SysDevOps Team, and we cover the management tasks as well.

### *How the IBEX Team Employs Agile Methodologies*

At present we use a modified Scrum approach.

Scrum [9] is one of many Agile frameworks. It provides a heuristic way of tracking the work that has been done, and aids in predicting what can be done. It is focused on the team being self-organising. A Scrum team should consist of three main roles:

- A Scrum Master [10] who looks after the Scrum processes and focuses on keeping the work flowing easily by removing impediments to the work
- A Product Owner [11] who ensures that the product being produced is what is actually useful and who makes sure the priorities are suitable understood
- The Developer role [12] as well as actually developing the product being produced are heavily involved in the planning and defining the end point of the work.

Scrum teams work in timebox [13], which for Scrum are referred to as Sprints. A Sprint is a predefined period of time inside which you do as much work as you can and then consider what has been achieved. Knowing what you have achieved in the past allows you to predict what you can achieve next. These predictions help manage expectations as to what will be provided when, with the feedback of what has been achieved before.

Each Sprint starts with a planning meeting, where the work to be produced is agreed. How much and which items are influenced by the availability of the developers, and the items that the product owner has identified as most important to achieve. The requirements for each package of work should also be finalised here.

Each day in the Sprint the team, including the Scrum Master and Product Owner, meet to discuss what is being worked on and what issues may have been found. Identifying those issues, or impediments if a Developer is waiting on information from a third party, in good time allows the others in the team to help as appropriate.

At the end of each Sprint the work produced is demonstrated to as many people as are interested, users and Scrum team members.

The team also puts time aside after the demonstration to discuss in a retrospective the Sprint, what went well, what went badly, what could be done differently. Over time those retrospectives will shape the way the team works potentially to improve the workflows and interactions based on the current team members and the wider work environment of the team.

### *How We have Adapted Scrum to Work for us*

When we implemented our Scrum system we found one big barrier to following the standard framework, it was hard to identify a single Product Owner who could give us the time needed to take on that role. Instead, the developer team took on an element of that, with individual developers talking to groups of users and refining requirements for what those users were interested in developing. Whether that was a specific ISIS Instrument being converted from our older control system to our new one, or functionality like the script generator [14] which runs across many of the instruments. The priorities are often set in a less detailed way by a Project Board which is concerned about the running of the project, and the Science Advisory Group which is concerned with looking for things that impact and improve multiple instruments or scientific disciplines. This disconnect can lead to tension within the planning meetings as there isn't a single voice dictating the priority, and often leads to many of the tasks not being achieved in the best possible order. It also increases the amount of time needed to manage the expectations of our users.

Similarly, it was hard to gather any of our users to demonstrations at the end of each 20 day Sprint, certainly when ISIS was running their time was not available to us. As such, our demonstrations are made just to the development team each Sprint. When we deploy the latest version of IBEX to instruments we talk to the team of scientists who run instruments in appropriate batches to demonstrate what has changed since we last spoke to them, which can be many Sprints worth of work.

Due to the inclusion of operational tasks as well as development ones our stand up migrated from just an update, to a check on services and systems to ensure we got to errors before they occurred as well as the update.

Those operational tasks also mean that we have implemented a pre-planning meeting, to undertake an initial view of the work to be done in the upcoming Sprint, so that we are not spending the whole day in a planning meeting. This also benefits the focus and well-being of the team.

### *Where Scrum hasn't been the Answer for us*

The rigid timeboxes of Scrum, which are usually the same length, have often been unhelpful for the team. Attempting to undertake the change from one Sprint to the next during heavy operational times, some of which can be predicted, added a layer of stress to the process.

Whilst not specific to Scrum, Agile software development tends to employ a code and refactor mentality. You code the bare minimum to achieve what is required, and as you add new functionality you refactor the code to keep it looking tidy. The issue is the larger your code base the harder this can be, and as the development of IBEX has proceeded the harder it can be to implement a change for certain instruments without a large refactoring process which reduces the throughput of the team, and the team starts to lose any of the gains made with the other Agile processes compared to a more traditional project management framework.

Typically, Science focused organisations prefer those more traditional methods, as the code and refactor method is not easy to perceive, especially if you are not involved directly in the team and the work being done.

### *Where Scrum has Worked Well for us*

The use of a single backlog, and individual developers picking up the next ticket in the list has ensured that the skills and knowledge of the system has been shared well across the whole team has been beneficial for the operations side of our work.

The regular consideration of whether our way of working is appropriate has been beneficial as well. This has allowed our processes to grow and adapt to support the ISIS instruments in the best way.

### *What the IBEX Team are Currently Trying*

In keeping with those continuous changes as highlighted from the retrospectives, the team has been looking at other ways of working.

Because of the nature of deploying and dealing with an accelerator that runs in cycles of supplying beam to the instruments we are also experimenting with variable length sprints. Varying the duration of the sprint means that we can avoid the sprint end/start meetings falling when we anticipate that operational requirements will be high.

At present we are trying a different way of filling our list of tasks to undertake in a Sprint. Previously we would look at all the items that have to be worked on, which could mean that important items that didn't have a forceful enough advocate in the planning meeting would be missed. Instead, we are assigning certain members of the team to a "theme" which is more in keeping with eXtreme Programming (XP) [15], which is a different Agile framework.

XP is aimed at solving some of the issues found in large and brittle codebases that have been adjusted to fulfil ever-changing requirements. Adding some of the other XP practices as a team may improve the IBEX codebase, especially the even greater emphasis on testing and the encouragement to have the programmer consider the actions of the user.

### *What We might Try Next*

Given that the prioritisation is a recognised stalling point, and whilst the team is small the number of products supported are numerous, some of the concepts employed by SAFe [16], which is the Scaled Agile Framework, might benefit the management of the workload. For example, having a "Program Backlog" for a longer term timebox, 3 months or longer, which agrees the work to be undertaken by the team from the business perspective. The team can then plan Sprints with a Scrum flavour, or XP, or more traditional methods, to achieve those agreed items.

## CONCLUSION

Whatever framework is used, Agile or non-Agile, the one tool that does seem to live up to the hype is the Kanban board. As mentioned previously it came from the assembly lines at Toyota, and the ability to track a task or item though an interface that is visible to all can lead to a sense of accomplishment. Whichever aspect of our work is being considered, whether it is an operational task or a new feature, being able to see whether it is waiting to be started, is being worked on, or is complete at a glance is one of the most useful actions we started using.

Different Agile frameworks, such as Scrum, XP, or SAFe, all have benefits and issues. Whilst pure Scrum would not suit the kind of rhythm we have at ISIS in relation to when user cycles start, it can be modified. Other frameworks may offer tools or ways of thinking to overcome the issues we have. However, Agile is a methodology, and the core concepts and where to place value are independent of any framework, and worth bearing in mind when undertaking any work.

## REFERENCES

[1] https://agilemanifesto.org/

[2] https://www.apm.org.uk/

[3] https://www.apm.org.uk/resources/find-a-resource/agile-project-management/

[4] https://dictionary.cambridge.org/dictionary/english/Kanban

[5] https://mag.toyota.co.uk/kanban-toyota-production-system/

[6] https://www.apa.org/research/action/multitask

[7] https://www.researchgate.net/publication/317989659_Impact_of_task_switching_and_work_interruptions_on_software_development_processes

[8] https://www.isis.stfc.ac.uk/Pages/home.aspx

[9] https://www.scrum.org/

[10] https://www.scrum.org/resources/what-is-a-scrum-master

[11] https://www.scrum.org/resources/what-is-a-product-owner

[12] https://www.scrum.org/resources/what-is-a-scrum-developer

[13] https://www.agilealliance.org/glossary/timebox/

[14] J. C. King *et al.*, “The IBEX Script Generator”, presented at ICALEPCS’21, Shanghai, China, Oct. 2021, paper TUPV049, this conference.

[15] http://www.extremeprogramming.org/

[16] https://www.scaledagileframework.com/