

# THE MAX IV WAY OF AGILE PROJECT MANAGEMENT FOR THE CONTROL SYSTEM

V. H. Hardion, M. Lindberg, D. P. Spruce  
MAX IV Laboratory, Lund University, Lund, Sweden

## Abstract

Projects management of synchrotron is both complicated and complex. Building scientific facilities are resource consuming although largely made out of standard and well known components. The industrial approach of project management resolves this complication by requiring analysis and planning to facilitate the execution of tasks. The complexity comes by all the research making unique the accelerators, the beamlines and its usage. Known unknown requires experiments which evolve continuously causing the development path to be naturally iterative. Agile project management has come a long way since its definition in 2001. Nowadays this method is ubiquitous in the software development industry following different implementation like Scrum or XP and started to evolve at a bigger scale (i.e Scaled Agile) applied within an entire organization. The versatility of the Agile method has been applied to a Scientific technical development program such as the MAX IV Laboratory control system. This article describes the experience of 7 years of Agile project management and the use of Lean Management principles to develop and maintain the control system.

## MAX IV WAY

MAX IV Laboratory is the first 4th generation synchrotron which is based on an innovative multibend achromat design carrying high expectations in terms of brilliance, stability and coherence of the X-Ray beam. This new laboratory inaugurated in 2016 is based in Lund, Sweden and has been built upon the foundations of Max-Lab, a facility which had operated 3 previous generations of accelerators until 2015. Max-Lab began in the 1980's with the first accelerator, MAX I, constructed by the team members themselves [1]. The build-up was handled by a very small staff and on a very limited budget. In spite of that the MAX I facility could be taken into operation in 1985. The small staff also implied that all personnel had to take a large responsibility [2].

The Control and IT Support group (KITS) were the first to introduce Agile methodology at the MAX IV Laboratory. Its main responsibility is to provide Software, Electronics and IT to build and develop the MAX IV Laboratory.

## COMPLEX AND COMPLICATED

Synchrotrons are complex systems in the sense that they are made up of many interacting components, from the different subsystems with their corresponding experts, to the varied flora of scientific experiments. The interactions between the components and the constantly changing scientific requirements result in many unknowns. Getting each separate part to work well is a complicated problem which can

be solved, but the general complexity of the facility can only be managed by a willingness to try, learn and adapt.

## AGILE

Agile [3] is a movement that describes an iterative approach to project management and development, with a focus on evolving requirements and self-organizing teams. This methodology is based on 4 principles which define the "The Agile Manifesto" Fig. 1.

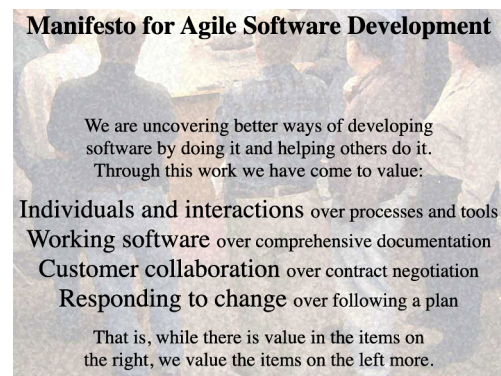


Figure 1: The 4 principles of Agile.

Several implementations of the methodology are based on Agile such as SCRUM or XP. SCRUM is a framework that describes a set of rules and methods that enable a team to collaborate on complex problems in an iterative and agile way. Scrum is more focused on the project management while eXtreme Programming (XP) emphasises Agile through the techniques of development. Most of them are known outside of this framework such as Continuous Integration [4], Pair programming, Unit Testing and Code Review. These are often cherry-picked in different methodologies than Agile.

## Lean Management

Lean management [5] in an organization aims to deliver better value to the customer by systematic, iterative and continuous improvement of the workflow and processes involved. Kanban is a framework that allows management and visibility of the workflow making it possible to adapt and collaborate in a lean and agile way.

## PRINCIPLES APPLIED AT MAX IV

The following section explains the organisation of the Control Group in terms of project management and development techniques seen through the narrow prism of Agile. Each of the principles applied are related to the way the old facility Max-Lab was organised and how these principles

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

have been followed to scale up the laboratory to the current MAX IV.

### Individual and Interaction

"Individual and Interaction, processes and tools" is one of the first principles of Agile applied 7 years ago, at the start of the project to build a new facility. Communication is really important, especially when the staff reaches more than 250 people. At this time only 80 people were working at Max-Lab and 4 persons were actively working on the control system [6]. The interaction with the other groups of the facility did not need any formality nor meetings, just corridor chat was enough. Although it was already important to establish a system which could scale.

Our inspiration is based on the structure of SCRUM in which the Product Owner (P.O), the person representing the users of the product to develop, communicates with the Scrum Master who represents the development team, see Fig. 2. They usually communicate at high frequency in order to steer the development in relation to the iterative requirements like in a closed loop system. In Scrum the Product Owner is considered part of the team.

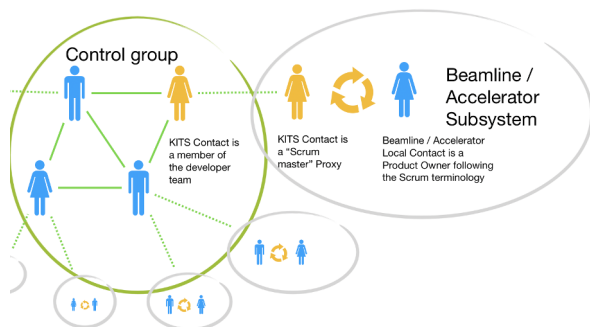


Figure 2: The organisation of KITS regarding project management.

Using SCRUM for building and developing a synchrotron is different than using it for an industrial project. The expertise of many different subsystems and domains i.e physics, vacuum, magnets, safety and security, radio frequency and diagnostics is necessary. Each of these subsystems represents an entire project and a channel of communication for the control system which integrates them all. So, in our organisation, each control system developer is responsible for the follow up of one or more subsystems. This is a classical organisation to assign a Control System resource for each subsystem group. But the responsibility of this role, the so-called "KITS Contact", is focused on the understanding of the subsystem, raising the requirements and demonstrating the progress of the development together with the local contact. In this way the role is equivalent to a Scrum Master.

The fundamental difference with the classical approach is that the KITS Contact is not necessarily the actual developer. Another developer will pick up the task of development in order to share the work load, share the general knowledge and share the good practices. Some initial effort has to be

made to reach a critical mass as this system works against the natural way of working. Additionally one representative, the aforementioned Product Owner (PO), has been defined in each of the subsystems, in order to represent the users and to resolve the priority, one of the most important responsibility of the PO. At a higher level a similar organisation has been established for the entire laboratory.

Although this system was a bit overkill for 4 persons at the beginning of the MAX IV project, it was still possible to apply early on thanks to its very lean implementation without affecting the available resources for the development. The main advantage proved to be in the scalability of this system for a group of 30 persons and for the different phases of the MAX IV development i.e first accelerator then beamlines. Also starting from a small group early on avoided having to face a certain resistance of change, a common issue in a larger group.

In KITS the "Individual and Interaction" principle brings a high level of communication which allows to quickly share common values, knowledge, cross training and resolve conflict as soon as possible, achieving a good level of trust. These points have opened the possibility for KITS to apply good practices:

- Pair programming. Developers are encouraged to work together on one task. In KITS it represents 50% of the development.
- Code Review. Each development is checked by another teammate in order to ensure the current quality standard. It's an excellent distributed way to increase the awareness. In KITS the code review is mandatory before deploying in operation.
- Task Force. All the team members are engaged to work together on one task for a short period of time. When an emergency occurs, when there is a risk of delay on a high priority or for very tedious, highly parallelisable and repetitive tasks. The notion of emergency is shared.
- Retrospective. Regularly the whole team looks back on the way of working or the technical debt and take decisions on how to improve it.

Even though Agile brings the control group several advantages we encounter different barriers related to the domain:

- It's hard to find a Product Owner who wants to or can represent a group of users and understand this role.
- The Product Owner needs to devote time to be involved in the development. They often have other responsibilities. Having little time for the P.O. to stay engaged means that often the team has to work without feedback.
- The standard company organisation has a hard time to understand the Agile project management. Even if it's proven to be efficient, gaining trust is hard.
- Having more than one Product Owner i.e beamline projects, a good mediator is a must-have to resolve priority.
- The knowledge sharing between scientist and engineer takes time.

- The time used for communication vs development is hard to manage, especially for a KITS Contact with many systems to follow up.

### Working Software

Working software (applies to hardware as well) over comprehensive documentation has been the way the small Max-Lab laboratory worked before the start of the MAX IV Accelerators project e.g solutions achieved through discussion and experimentation. In the transition to a bigger infrastructure this natural behaviour suffered from the formalism introduced by a more structured organisation. At the genesis of the Control System group an attempt at writing specification documents for each subsystem was a half success. On one hand it provided a good enough overview of the technique and the plans for each component of MAX IV. On the other hand these plans were not detailed enough, e.g no model/documentation of the hardware controller. The plan for the Control System could not be established clearly despite the time and effort spent chasing the requirements. One reason was also the lack of resources the subsystem groups could provide to us, they were also actively working on their system with the few resources allocated.

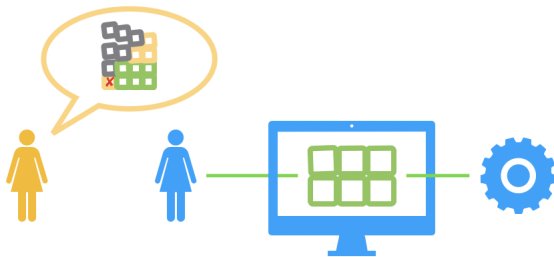


Figure 3: Building the software by iteration with the goal to provide the minimal viable product first.

Instead of pushing for the details of the specification we decided to start developing the new control system. We had the benefit of an operational laboratory: Max-Lab. The alternative would have been to use a simulation but this approach has other drawbacks. Furthermore the management supported the idea to try the new technologies for MAX IV on different systems at Max-Lab.

This successful strategy allowed to always deliver the minimal viable product on time, see Fig. 3. Of course the resulting control system is not perfectly polished but it's rather very pragmatic (or experimental, which fits the facility). Although a winning concept it can end up in a total mess of incoherent solutions in the long term. Missing from the old Max-Lab way was the concept of refactoring introduced by Agile XP. The refactoring consists of applying good engineering practice just after the solution has been demonstrated to the user. It can mean to standardize a solution, to make a clean design, to make the solution "deploy-able" etc. It means also to reconsider the standard in case of rejection of the demonstrated solution.

Overall the "Working Software" principle lead us to establish good practices:

- Minimum viable product. At any point of the project there is a working product in operation avoiding deadline rush.
- Integration. Reuse the software already developed from other facilities as it solves many requests of the users. More focus on the software integration instead of reinventing the wheel. Note that the side effect is that it gives us more time to develop new products.
- Faster feedback. The User eXperience (UX) has usually a very high focus as the development follows the user's feedback.
- The refactoring is efficient as it occurs only on validated features.
- Problem solving. The major issues are often solved first to prove the feasibility of the project.
- Less waste. At the end of the project most of the software or interventions brings a value to the system.

Challenges we have encountered following this principle:

- Establishing a contract with the user built on trust rather than a detailed list of predefined requirements.
- Winning the understanding of the customer for the iterative process.
- Avoid working on architecture first and not refactoring afterward, especially true for the developer with little Agile experience.

### Customer Collaboration

Customer collaboration was a natural way of product development in the small laboratory. A small independent group of people collaborated around an idea or a need submitted by this same group. There was little formalism and everyone helped out which made the work quite efficient. But the result was not homogeneous at the laboratory level. Each group had different solutions or local standards. And working on one system and not laboratory-wise meant that it was impossible to offload the resource where needed. The downside was that the bigger picture did not come into play and the overall effort was not well balanced. Building a larger project like MAX IV has required more structure. And the conventional approach is to separate into groups by function as mentioned before. Even if the old way can still work right after a change to a functional organisation, simply the fact of the growing structure implies changes (Conway's law). Split responsibilities introduce a contract of "Who does What". Not long after arises a relationship of Requester and Support\* which implies a formalisation of the requests. Time management becomes critical as the support groups are a shared resource. And without customer collaboration it can end up in a blame spiral with more and more formalism.

Part of this Agile principle "Customer Collaboration" is what KITS calls "User Autonomy". At the first glance collaboration and autonomy may be seen as antonyms. But this idea developed and promoted very early at MAX IV is quite simple and based on previous experience. The word "User"

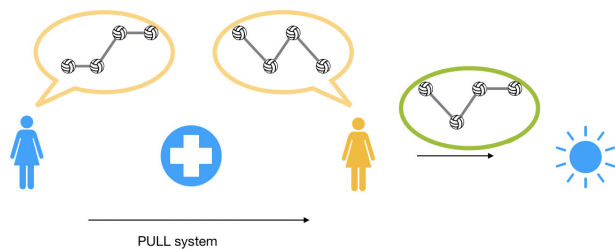


Figure 4: Exchange of scientific and engineering knowledge is important to develop the right system.

refers to the people who will use the product of the development e.g a scientist, technical staff or an experimenter. And in a synchrotron the developers have the chance to be close to the user, contrary to many industries. Taking the analogy further the customer and user roles are often the same person, meaning good conditions for collaboration. The word "Autonomy" refers to the capabilities of the user to interact with the system. By experience the synchrotron staff and users have different skills and competence regarding IT, i.e some can program a software, others may know exactly how a user interface should look, and some are only interested in the final result. Most of the scientists have used a programming tool during their studies. On the other hand the developer needs to understand the language of the user in order to provide the best product. Collaboration is the key to achieve this goal. What the developer provides in support is paid back with the feedback/knowledge of the User, see Fig. 4. Our software follows this principle as predicted by Conway's law. A user can develop their own Tango device or Sardana [7] controller, their own Taurus user interface, their own python program, etc.

Apart from the User Autonomy, customer collaboration means also for the Control group to follow good practices:

- All development should be steered by the user's needs. In Lean Management this system is call "Pull".
- Work on possible solutions. The advantage of a close collaboration is to get immediate feedback from the user if the development goes in the right direction. Everything is possible with IT but it's better to concentrate the effort to provide something simple rather than creating a complex solution that will fit any corner case. Behind the specification of the product there are many assumptions that the user can clarify.

The main challenges we still encounter:

- Time for own development. There is a contract with the main stakeholder to reserve a part of our time, around 25%, to work on our own infrastructure (continuous improvement) or projects for which the goal is to "sell" to a P.O such as WebJive [8] or the synoptic [9] Additionally 10% can be used for innovation.
- Long term conflict with short term. The main critical point for the developer is to sense the risk of a particular

project while negotiating the tasks. Sharing the same vision is a real challenge.

- Keep the involvement of the stakeholder. The concepts of Agile do not work if the stakeholders are not involved. The iterative feedback is necessary since there is no detailed specification to check the progress against. For some projects KITS spends a large fraction of time to run after people.
- Difficult integration in contract oriented project management. There is always the temptation to provide a detailed time plan if an Agile project is included in a higher level project managed according to waterfall principles. Our management understand for now the benefits of delivering a functional product on time.
- Adapting the levels of User Autonomy. People may have their own opinion and can end up building their own solution, with its own pros and cons.

### Responding to Change

Changes can happen rapidly within a human-scaled laboratory which was the case for Max-Lab ; mainly due to the close interaction between people but also due to the minimal formalism between the requester and the developer. Even in this case the overall efficiency was limited by the resource and by the trade off on quality. This system can be chaotic but the balance was found as the people shared the same vision because they have a common sense of these constraints e.g priority between repairing a broken monochromator and reaching the maximum performance of the detector. A more structured organisation is a prerequisite for developing larger facilities. Since then, the responsibilities are split and specifications are established prior to development. Depending on the classification of the problem (see Cynefin framework [10]) there is a risk regarding the delegation of development that has to be shared by all stakeholders. In most cases the aim of the developer is to work on stable requirements and consequently to avoid any changes during the development.

This begs the question of trust between the requester and the developer. The more complex the development the more divergent are the ideas of the stakeholders, especially when they already experienced the same type of development i.e a scientist who did the same development at Max-Lab. It can be challenging to keep the trust and not enter a blame system, especially if there is high competition for the shared resources.

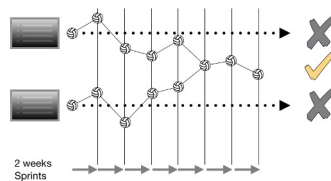


Figure 5: Changes during projects can ends up in a win-win situation.

The Agile way to face the challenge of keeping trust with the stakeholder is to define the completion of a project by making changes possible in a controlled way. Initial specifications are defined at the start of a project in order to give a vision of the goal. The development is made to respond to changes at a certain pace as this vision becomes more and more clear during the project. In the synchrotron domain changes can be due to new requirements e.g the availability of a new performant detector or another experimental method in order to adapt to state-of-art research instrumentation. Changes can also have been discovered by the development team who are adjusting their understanding of the requirements or have underestimated the complexity. Changes have to follow an experimental and incremental process i.e a Plan, Do, Check, Act [11]. At the end the initial scope does not have to be strictly implemented but the end result should give a better expectation, see Fig. 5, for the same initial resource budget<sup>1</sup>. Frequent demonstration of the development allows for the user to assess the viability of the project.

The Control group has adopted part of the Scrum methodology from the start of the MAX IV project. The different aspects were introduced step by step but also adapted to the situation. The first and not the least important feature implemented was to define a minimal period of straight development, avoiding the impact of changes and unplanned tasks. This so called "Sprint" period allows the group to focus efficiently on the development and corresponds to an acceptable delay of response by the user. At MAX IV 2 weeks is now the standard considering that at any time the development can be stopped to respond to urgent cases that block user operation. The development process takes in account the early feedback of the user either by going to the control room or by making a demonstration in the office. In case the feedback is positive the functionality is delivered, meaning made accessible to the user. But it can be negative and the development is then rejected. From this point of time the developer or the user learn how to adapt from the initial plan (often referred to as "Fail Fast" in the Agile mindset).

The resulting good practices:

- Just In Time. The Control team really try to deliver the product on time without developing too far in advance. The user can start to use the product and give immediate feedback. Few developments are wasted like unnecessary abstraction, mock up or unused features.
- Same pace for everyone. A stand up meeting is held every morning. This is the right moment to sync with each other or ask for help. Having the same "Sprint" makes everyone aware of the delivery time, meaning the integration usually goes smooth.

The main challenges are:

- Understand the Scope Trade. The PO appreciates the possibility to change but may still think that the devel-

opment will be done as an addition to the scope. The trade off system is not always accepted. But this is a way to stay on time.

- Sprint planning validation. The KITS Contact has to understand if the user will be ready to test the new feature to avoid planning development too far ahead.
- 3 months release ceremony. It's difficult to align every PO for a 3 months release ceremony (SCRUM) as the projects have different pace. This would be fantastic to join the effort for common features.
- Notion of Readiness is tricky to maintain. In order to deliver on time the idea is to work ahead of time on the most risky developments such as detector which takes 3 to 6 months of work. Once past the risk, the development can be in pause until we can make sure to deliver on time.
- Necessity to be proactive. A substantial amount of time is devoted to poll the people in order to identify the risk, the trade off, understand the value. It can be more or less difficult depending on the overall organisation.

## CONCLUSION

The 4 principles of Agile has been successfully applied at MAX IV Laboratory to develop the Control System of the entire facility. We demonstrated that Agile can be adapted to a different environment. But it's important to understand the philosophy behind Agile before cherry-picking some features. In our case it helped to choose the most efficient features adapted to the situation.

It's important to understand that the support of the management is a prerequisite. But in order to be successfully applied the Agile concept has to be recognised company wide.

Finally Agile is only one perspective of our way of working, lean is another. In this article Kanban was briefly mentioned since our organisation is also inspired by the Lean Management. Effectively the responsibility of the Control Groups is also involved in the operational maintenance of the facility. Mixing development and operation in the same tasks backlog means that the organisation has to be adapted. In this sense our different implementation of Scrum gets closer to what Scrumban [12] is supposed to be.

## ACKNOWLEDGEMENT

First of all the authors would like to thanks the KITS team members, without their engaged involvement the Agile experience would not have survived this long. We are also grateful that Magnus Sjöström, Yves Cerenius, Franz Hennies and now the members of the CPO have accepted the role of Product Owners. Also the experience would not continue without the support of the management of MAX IV Laboratory, especially Ian McNulty and Marjolein Thunnissen. Finally a last thanks to Gwenaëlle Abeille from Synchrotron Soleil for the passionate discussion about Agile.

<sup>1</sup> The main constraint of a project is often the budget and the deadline. In the MAX IV case, exception apart, the maximum budget allocated for in-house development is represented by the maximum time a support group can allocate to one project.

## REFERENCES

- [1] History – MAX IV, <https://www.maxiv.lu.se>, Retrieved 27 May 2017.
- [2] Nils Martensson Mikael Eriksson, *The saga of MAX IV, the first multi-bend achromat synchrotron light source*, <https://doi.org/10.1016/j.nima.2018.03.018>, Nuclear Instruments and Methods in Physics Research, Volume 907, 1 November 2018, Pages 97-104.
- [3] Agile Alliance, <https://www.agilealliance.org/>
- [4] V. Hardion *et al.*, "Assessing Software Quality at Each Step of its Lifecycle to Enhance Reliability of Control Systems", *Proc. ICALEPCS'11*, Grenoble, France, paper THBHMUST02, 2011.
- [5] Jeffrey K. Liker, *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*, January 2004, McGraw-Hill Education, ISBN 978-0071392310.
- [6] J. Lidon-Simon *et al.*, "Status of the MAX IV Laboratory Control System", *Proc. ICALEPCS'13*, San Francisco, USA, paper MOPPC109, 2013.
- [7] T.M. Coutinho *et al.*, "Sardana: The Software for Building SCADAS in Scientific Environments", *Proc. ICALEPCS'11*, Grenoble, France, paper WEPMSO23, <http://sardanascada.org/>
- [8] WebJive, <https://gitlab.com/MaxIV/webjive>
- [9] J. Forsberg *et al.*, "A Graphical Tool for Viewing and Interacting with a Control System", *Proc. ICALEPCS'15*, Melbourne, Australia, paper WEM309, 2015.
- [10] The Cynefin framework, [https://en.wikipedia.org/wiki/Cynefin\\_framework](https://en.wikipedia.org/wiki/Cynefin_framework)
- [11] The Plan, Do, Check and Act, <https://en.wikipedia.org/wiki/PDCA>
- [12] Ladas, Corey, *Scrumban: Essays on Kanban Systems for Lean Software Development*, January 2009, Modus Cooperandi Press, ISBN 978-0578002149