Applying agile principles for ICT operations management in libraries

Wouter Klapwijk

Information Technology Services, Library and Information Service, Stellenbosch University, Stellenbosch, South Africa.

E-mail address: [wklap@sun.ac.za](mailto:wklap@sun.ac.za)

Creative Commons LicenseCopyright © 2022 by Wouter Klapwijk. This work is made available under the terms of the Creative Commons Attribution 4.0 International License: <http://creativecommons.org/licenses/by/4.0>

Abstract:

*During the COVID-19 pandemic, many Information Technology (IT) practices in libraries have changed, some out of pure necessity given the new demands placed on IT resources to deal with the new reality. Most IT practices that needed change benefited from a foundation that was laid well before the pandemic to allow for such change to happen in a brief period. One such change necessary was to have IT operations teams in libraries to work in a physically separate, but organizationally coherent, fashion. To adapt to the change required the implementation of methods and tools that allowed for new forms of autonomy, responsibility, and collaboration amongst IT operations teams. By applying agile principles and methods that were implemented at Stellenbosch University (SU) before the pandemic, the SU Library and Information Service’s Information Technology Services (ITS) operations team could respond to the change in a flexible and responsive way.*

*Working and improving on what was already done, and known, about agile IT practices, the ITS team could adopt agile methods such as Kanban using the Jira project management software within a multi-disciplinary team that is responsible for IT operations management across an eco-system of technologies and service offerings. This allowed the team to take on an agile mindset to work, act and support the SU’s Library and Information Service in being responsive to service delivery and project implementation. This paper outlines specific agile methods and practices adopted by the ITS team, as well as the new agile project management methodology implemented to successfully implement vendor-driven software projects while still working to clear strategy-driven management priorities and reporting structures. More importantly, the paper shows that it is not needed to implement agile practices from scratch, because so many agile principles are already applied without practitioners actively realizing it.*

**Keywords:** Agile principles, Jira, Kanban, autonomy, efficiencies.

**Introduction**

Software project management, while incorporating all the key elements of generic project management, also must deal with the peculiar problems associated with creating software. Software project management therefore addresses the important consideration that *how* systems are implemented is as a vital consideration as that of *what* a system is to do. In response to this important consideration, the publication of the Agile Manifesto in 2001 (Beck et al., Manifesto for Agile Software Development) marks the birth of agile as a methodology to improve on the way how systems at the time were implemented using heavyweight implementation methodologies. Since then, many agile frameworks have emerged such as Scrum, Kanban, Lean and Extreme Programming (XP), some which existed well before the writing of the manifesto and only became more popular thereafter.

Rather than being defined by a set of ceremonies or specific development techniques, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement. From a more practical perspective in the work environment of the Information Technology Services’ (ITS) team of the Stellenbosch University (SU) Library and Information Service (“the library”), agile means a lesser focus on the detailed planning and specification of a product or service’s criteria; rather, it means the assignment of greater autonomy and enablement to members of the team. Openness, trust, and autonomy are emerging as the cultural currency for institutions who want to attract the best information workers and get the most out of them. Agile is a cultural value, and teams should be empowered to work how they best seem fit. This work culture has lately been advanced by the COVID-19 pandemic necessitating information workers to work in a physically separate, but organisationally coherent, fashion. In the case of hybrid library environments, it has also become one of the solutions to the challenge to be more responsive to the demands placed on Information and Communication Technology (ICT) teams to deliver digital library services.

Similarly, there are applications of agile that do not have anything to do with software development, even though the manifesto starts off by saying: “We are uncovering better ways of developing software by doing it and helping others do it.” Indeed, embracing agile outside the realm of software has caught on, and so in libraries. In the pre-Manifesto times of 1998, Lorraine Haricombe and T.J. Lusher edited “*Creating the Agile Library: A Management Guide for Librarians*”. At the time the book was meant to be a response to the rapid changes in the library environment due to the impact of technology in society and education – an environment that has been stable for decades. The book was well received, and reviewers acknowledged the need of a “Agility of mind, innovation and public entrepreneurship” to cope with continual change, but by building on the core and old principles of librarianship (Hendry). Likewise, the Library 2.0 movement focused on librarianship and the need to adapt to change and interact more with users, eventually culminating in a Manifesto for Librarians as defined by Laura Cohen (Cohen, 2006).

Today many libraries practice agile principles when implementing software projects and a few have even adopted them in managerial practices.

**Agile values and principles in organizational context at SU**

Today’s agile landscape can seem cluttered with methodologies that promise to take agile ideals and turn them into real-world realities. But today’s methodology madness is not anything new. The manifesto itself was born out of a need to find a common ground among Scrum, XP, Crystal Clear, and other frameworks. The problem, the manifesto authors agreed, was that companies were so focused on excessively planning and documenting their software development cycles that they lost sight of what really mattered, namely pleasing their customers. In the two decades since its creation, the manifesto’s four values and twelve principles have been embraced to varying degrees by countless individuals, organizations, institutions, and professions. Agile project management is deeply rooted in these principles but slightly modified to make sense in the project management, rather than software development, environment.

*SU in organizational context*

When considering how the agile principles are being applied within the organization context of SU, it would be fitting to first view the institution as an image of an organization. As an academic institution, SU can metaphorically be described in the image of an organisation as an organism (Morgan, 1998). Universities do not operate in isolation of their environments but are open systems that continually adapt themselves to their environments. A university as organisation is made up of a set of subsystems that each have their own characteristics, goals, and management styles functioning as part of an overarching global parent organisation. This open systems approach is underlined by the contingency theory that different approaches to management may be necessary to perform different tasks within the same organisation. The Lawrence and Lorsch study (Lawrence, 1967) refine the contingency approach by showing that styles of organisation may need to vary between organisational subunits because of the detailed characteristics of their sub-environments. The study also showed that the degree of required differentiation in managerial and organisational styles between departments varied according to the nature of the industry and its environment and that an appropriate degree of integration was also needed to tie the differentiated parts together again.

It therefore follows that there will be no single agreed upon approach on how agile practices are applied to software development, or organisation management tasks, at SU. Each subsystem will apply agile practices to fit their unique needs, but only if they are guided by the right principles. Then again, each of the twelve principles of the manifesto will be of lesser or greater relevance to the practices of each of the subsystems of the organisation, as is also the case at SU.

*Agile project management practices at SU*

Agile project management practices were introduced at SU through the Division of Information Technology’s (IT) software development team. The team is responsible for the project management of software development projects, and for this they need a software project management tool. To this end the Jira system from Atlassian[[1]](#footnote-1) has for the past 10 years been used for project and issue tracking. When launched by Atlassian in 2002, Jira was purely issue tracking software targeted at software developers, but later the application was adopted by non-IT organisations as a project management tool. Over time the application evolved from being a generic project management tool to also include agile project management features. Subsequently, in 2013, Atlassian announced a Jira service desk product with full service-level agreement support integrated with the agile-enabled workflows and tools. Hence, in 2018 Jira became the de facto solution for ICT service desk management at SU, and this included the service desk support for the ITS team in the library as well.

*Fulfilling ICT support services with the Jira Service Desk*

By doing service desk management with an agile-enabled tool such as Jira, the ITS team started practising agile principles even before they knew there was a name for it. In retrospect, without realising it, the team started to identify with at least two of the four values of the manifesto, and at least six of its twelve principles. The ITS team started to change its way of working by practicing agile rituals such as holding regular meetings in a stand-up format, approaching tasks assigned to the team from a Jira backlog perspective, and maintaining documentation with a minimal-specification approach in Confluence, a Jira wiki package. Working collaboratively on one service desk and software project management solution with other ICT stakeholders at SU, the team started to view itself from a cross-functional team perspective. This made the team understand with greater urgency that if only one person in the team holds a specific skill set, then that person can become a bottleneck in an ICT support or project implementation workflow. This is an important consideration for agile-based task or project teams.

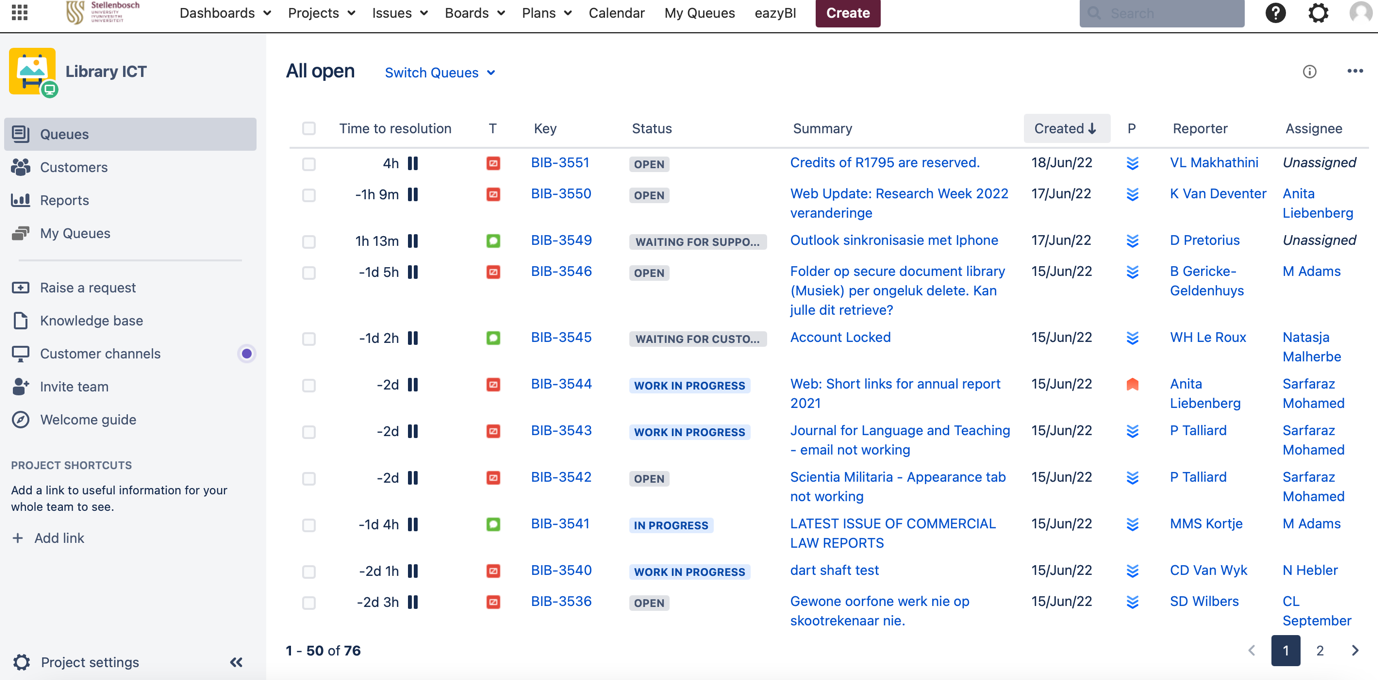


Figure 1: The Jira Service Desk system at the SU library

**Improving ICT operations management through agile principles, methods, and tools**

The ITS team was practicing agile principles out of due diligence to adhere to good ITIL[[2]](#footnote-2) (Information Technology Infrastructure Library) standards when the COVID-19 pandemic forced the library’s workforce to starting working from home in early 2020. Pivoting to an online-only format for working, collaborating, and managing, forced the team to become completely reliant on the Jira service management system to field support calls and manage tasks and projects remotely. With this came the need to improve how task assignment was managed in an online environment, prompting the ITS team leader to explore further functionality on offer in Jira. In doing so, the team discovered the versatile task management tool, Kanban.

*Using Kanban for ICT task management*

Kanban is the Japanese word for “visual signal”, an action performed through the physical passing of a card. The name Kanban comes from two Japanese words, “Kan” ([看](https://translate.google.com/#view=home&op=translate&sl=ja&tl=en&text=%E7%9C%8B)) meaning *sign*, and “Ban” ([板](https://translate.google.com/#view=home&op=translate&sl=ja&tl=en&text=%E6%9D%BF)) meaning a *board [[3]](#footnote-3)*. Often our work is invisible or intangible, so Kanban helps us to visualize our work, maximize flow (or efficiency), and limit Work in Progress (WIP) items. Kanban project management has been around since the late 1940s when it was studied by Toyota to use the rate of demand to control the rate of production of its vehicles. The car company applied it to their Lean manufacturing model, known as the Toyota production system. Kanban is about productivity and continuous improvement. As a Lean manufacturing tool, it seeks to eliminate waste and bottlenecks to keep the work flowing steadily.

Today, Kanban software is a very versatile task management tool that is used in many different industries. While the core principles of the framework are timeless and applicable to almost any industry, software development teams have found particular success with the agile practice. In part, this is because software teams can begin practicing Kanban methods with little to no overhead once they understand the basic principles. Unlike implementing Kanban on a factory floor – which would involve changes to physical processes and the addition of substantial materials – the only physical things software teams need are a board and cards, and even those can be virtual. Kanban boards can be built on windows or walls, or with digital tools such as Trello and Jira. Hence, moving the task assignment boards from the traditional office space’s whiteboard format to the online environment, is what the ITS team found to be the most wanting when they started to work from home.

*Starting a Kanban board*

One of the core principles of Kanban is that it starts with what you do *now*. Kanban respects roles and responsibilities exactly how they are *today*, you simply apply the Kanban methodology on how you currently work. Its purpose is to categorize all the stages of work a work item flows through from something you have not started to something that is done. This is referred to as a workflow and each stage in the workflow has its own column. A basic Kanban board has a three-step workflow: To Do, In Progress, and Done. A work item (or card) flows through the columns on the board from the To Do stage to the Done stage.

Kanban cards should be small enough that a team can make progress on them in a reasonable amount of time, i.e., it should not take weeks to move the card forward, and it should not be so small that the card represents every task a team member is working on. Creating the right balance is the first challenge the ITS team encountered when it started to use Kanban for ICT task management. Each service request logged on the Jira service desk can represent a Kanban work item, so care had to be taken to populate the Kanban boards in such a way that it remained an effective tool for work management.

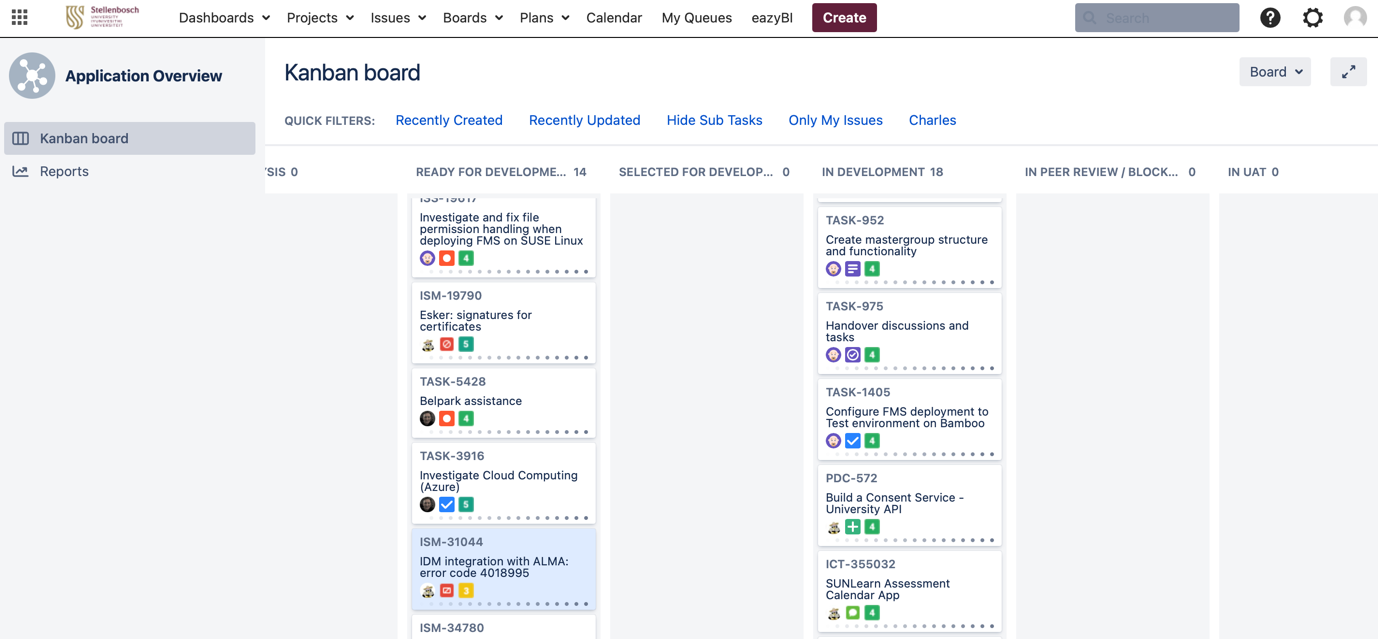


Figure 2: A Kanban board for library software integration projects at SU

*Benefits of using a Kanban board*

Once the ITS team started building a Kanban board and filling it up with cards it started to experience one of the key benefits of Kanban: seeing cards that bunch up revealing a bottleneck in a workflow. As already mentioned, this allowed the ITS team to get a sense of what size of cards are needed to move forward with in a timely manner. These efficiencies are called flow, and Kanban is designed to help teams flow work better from a “backlog” until it is done.

With Kanban the ITS team is now actively, and in real-time, measuring cycle times of work done, easing the process for the ITS team leader to measure how effectively and productively the team members are meeting their Key Performance Areas (KPA) in their SU Work Agreements. The Kanban methodology relies upon full transparency of work and real-time communication of capacity; therefore, the Kanban board should be seen as the sole source of truth for a team’s work. A further benefit of Kanban is to actively manage the WIP limits of team members simply because it can so easily be visually measured to help avoid too much multitasking which kills the efficiency of team members. Before using Kanban, the ITS team leader would assign tasks to team members while team members were still stuck on previous assignments, not always actively realising where team members needed assistance to finalize a task. This has helped the ITS team leader to become a better manager at an operational level, and it has helped the team to become more self-organised and self-managed at fulfilling ICT support tasks, since Kanban helps each team member understand why and how tasks are assigned to them.

Although Kanban seems like something simple that you have been applying in your work environment for some time already – and just not calling it Kanban – it is worthwhile to be cognizant of the fact that explicitly thinking about the Kanban principles when applying it, changes the way you approach and use it. Putting a name to a workflow practice, such as Kanban, can be the first step in tackling it. Until you can name something you do not really know what to do about it. This is what Kanban does, once you name it you can start to identify with working more easily with its methods, and to its strengths.

**Applying agile principles and methods to project management**

Project managers use Kanban boards to manage work items, WIP limits, recurring tasks and in agile slang, the “product backlog”. Project managers can oversee the process while team members can easily manage their tasks with their personal Kanban boards too. But some projects are not structured to agile principles and methods and are conceived and planned according to the waterfall methodology.

Generic project management at SU is formally done according to the Project Management Body of Knowledge (PMBOK), a set of standard terminology and guidelines for project management overseen by the Project Management Institute’s (PMI) committee on standards (Project Management Institute, 2008). PMBOK is heavily associated with traditional, plan-driven project management which is based on a Defined Process Control Model. SU leans on most of the processes described in PMBOK to implement institutional projects, even when implementing vendor-based software systems and/or services.

The constraints in the choice of project approach one can use to manage a project is sometimes enforced externally onto a project team or team leader. Such projects are usually strategy-driven, with projected outcomes set at certain intervals (stages) and predetermined by a project sponsor or external vendor. Amid the COVID-19 pandemic and lockdown the ITS team was assigned such a project for implementation. The project’s funding was purely strategy-driven and financed from the SU’s Strategic Fund, necessitating regular feedback and progress reports at set intervals to authorize the release of further stage funding. This necessitated a waterfall methodology for project execution, but within a team that had now become accustomed to working with agile tools and methodologies. The PMBOK however is not sufficiently aligned with agile principles, hence the ITS team turned its attention to the PRINCE2Agile methodology.

*Implementing library systems software with the PRINCE2Agile methodology*

The ITS team leader was sufficiently knowledgeable in PRINCE2 (Projects IN Controlled Environments, version 2) from a purely academic, not practitioner, perspective to feel comfortable to lean on some of the methodology’s methods to tailor the project’s implementation path according to an agile context. PRINCE2 [[4]](#footnote-4) focuses on managing resources and risks by dividing projects into smaller steps, defining clear roles and responsibilities, and using seven processes to manage the project lifecycle. By organising a project into logical steps, PRINCE2 demands a project management framework that has an organised and controlled project plan before starting, and one that maintains its organisation through the project lifecycle.

PRINCE2Agile on the other hand allow practitioners to combine the strong governance principles of PRINCE2 with the adaptability and flexibility of the agile environments and methodologies like Kanban. The adaptability of agile is already embedded in PRINCE2 and the framework needs only to be tailored to agile methodologies. The only requirement is to wrap around the agile processes the control mechanisms that PRINCE2 require, while the performance targets of time and cost remain fixed. This suits vendor-driven software implementation projects very well since the cost and time targets are usually pre-set and remain fixed.

Explained in purely general terms, with PRINCE2Agile there are two levels of development: the macro processes and the micro processes. The macro processes are closely related to the waterfall process model where we need to have some dates when we know that major activities will be finished so that we know when we will need to bring in staff to work on subsequent activities. Within these macro processes there will be micro process activities which might involve iterative working. The packaging of micro processes within larger macro processes means that it is possible for agile projects to exist within a more traditional stage-gate project environment, which has formal milestones where the business case for the project is reviewed.

Not being practitioner experts in this framework the ITS team lend itself to only setup its high-level project plan according to the seven PRINCE2 macro processes, and within each macro process continued to execute the micro processes according to agile methods. For the high-level project plan the team continued to use the well-known Gantt chart, and within each micro process it continued to manage the tasks with a Kanban project board. An advantage when using Kanban within a project management context such as this one, is that with Kanban there is no need for fixed-length iterations which you find with the software project management methodology Scrum (Schwaber, 2009). As of late, Gantt chart tools are often referred to as roadmap tools, and Jira includes a roadmap tool which allows a team to maintain a Gantt chart that creates plans around Jira issues assigned to a team. The roadmap tool helped the ITS team to maintain a coherent project strategy despite the iterative nature of the micro processes involved in the project.

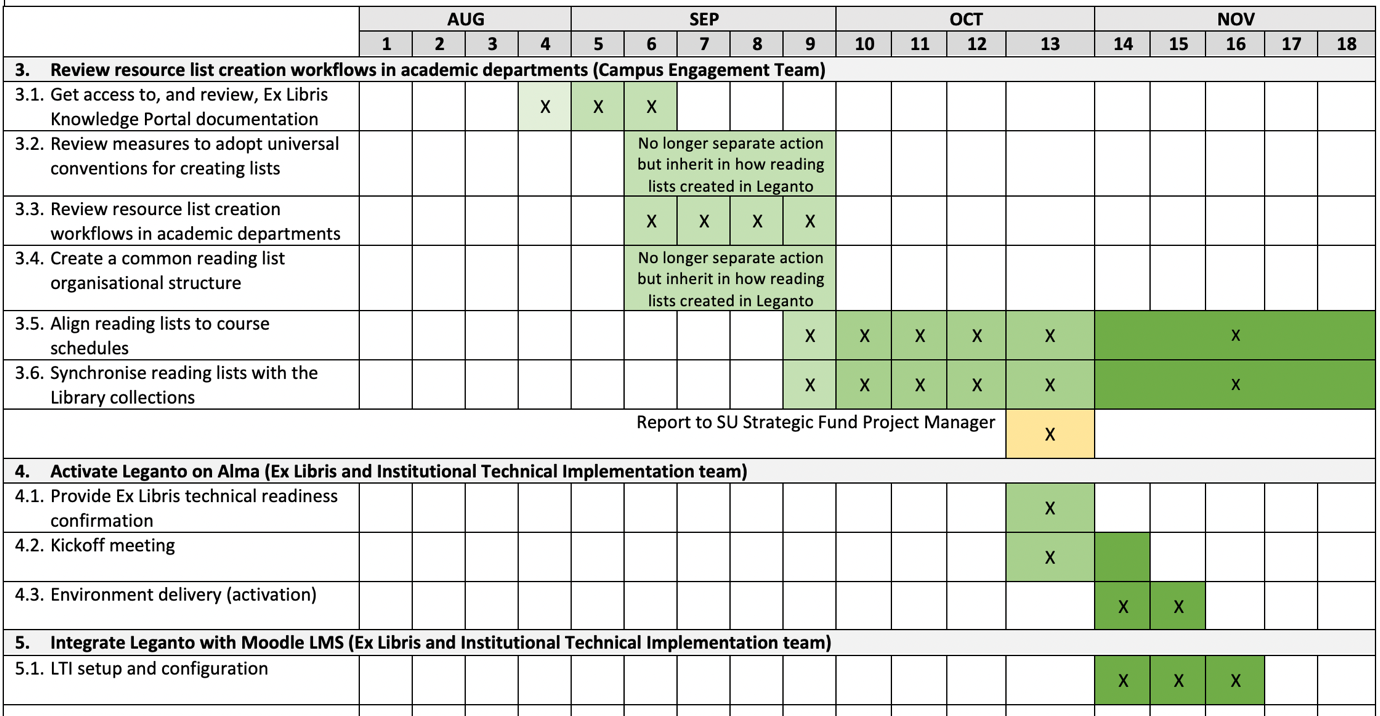


Figure 3: A Gantt chart for a library software implementation project

Transitioning from the ITS team’s previous GUI-based helpdesk management software, Heat, to a web-based and agile-supported helpdesk system such as Jira, allowed the ITS team to become familiar with agile methodologies, methods, and tools. It should be noted however that teams should not limit themselves to the agile tools and methodologies that are made available by their institutions, since many agile services are available that can be used independently, of which monday.com immediately comes to mind. As a method, Kanban remains a darling among service-oriented teams like IT or human resources. If you are a queue-oriented team like IT, then Kanban provides a solid foundation for your agile practices.

**Summary**

Although little about fulfilling ICT operations relates directly to the manifesto’s four values, did the ITS team over time manage to adopt an agile mindset to work with greater autonomy, trust, and flexibility in an environment influenced by a global pandemic that called for it. Without adopting an agile mindset, the greatest risk for any team is that it continues to work as before, and that no change will happen in its workflows and operations.

Today, many agile teams combine practices from a few different frameworks, spiced up with practices unique to the team. Some teams adopt agile rituals while others create a new agile practice, of which the Agile Marketing Manifesto is a good example. The original manifesto did not prescribe two-week iterations or an ideal team size. It simply laid out a set of core values that put people first. The way a team lives those values today – whether it uses a specific methodology by the book or blend it with elements of another, such as Kanban – is entirely up to the team and the organisational context in which it functions.

References

Beck et al. (2001). Manifesto for agile software development. Retrieved from <http://agilemanifesto.org/>

Cohen, Laura (2006). A Librarian’s 2.0 Manifesto. Retrieved from <https://web.archive.org/web/20070209003635/http://liblogs.albany.edu/library20/2006/11/a_librarians_20_manifesto.html>

Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International Digital Library Perspectives*, *27*(1), 18–22. <https://doi.org/10.1108/10650751111106528>

Hendry, J. D. (2000). Creating the Agile Library: A Management Guide for Librarians. *Library Review, Vol. 49 Iss: 1*, pp. 40 – 48

Kanban Tool. Retrieved from <https://kanbantool.com/kanban-guide/kanban-history>

Lawrence, P., and Lorsch, J., "Differentiation and Integration in Complex Organizations" *Administrative Science Quarterly 12*, (1967), 1-30.

Morgan, G. (1998). *Images of Organization*. Berrett-Koehler Publishers. https://books.google.co.za/books?id= a\_W8jXt4\_eMC

Project Management Institute. (2008). A guide to the project management body of knowledge (PMBOK® Guide) (4th ed.). Newtown Square, PA: Project Management Institute.

Schwaber, K. (2009), detailed information on Scrum methods available at: <http://www.scrum.org>

1. Atlassian website: Retrieved from <https://www.atlassian.com> [↑](#footnote-ref-1)
2. Axelos. ITIL 4: the framework for the management of IT-enabled services. Retrieved from <https://www.axelos.com/certifications/itil-service-management> [↑](#footnote-ref-2)
3. Toyota Production System. Retrieved from <https://kanbanzone.com/resources/lean/toyota-production-system/> [↑](#footnote-ref-3)
4. Axelos. PRINCE2 Project Management Certifications. Retrieved from <https://www.axelos.com/certifications/propath/prince2-project-management> [↑](#footnote-ref-4)