**ASSIGNMENT COVER SHEET**

|  |  |  |
| --- | --- | --- |
| **Student’s name** | Lucas | Shaw |
| **Module name** | Software engineering and agile | |
| **Title of assignment** | Task 3 report Agile overview | |
| **Complete Word Count in my assignment** | 997 | |
| **Date submitted** | 27 June 2025 | |

All work must be submitted by the due date. If an extension of time to submit work is required, a [Mitigating Circumstances Extension Form](https://canvas.qa.com/courses/1041) must be submitted.

<https://canvas.qa.com/courses/1041>

Has an extension been approved? Yes No If yes, please give the new submission date ….…/..…./…….

|  |
| --- |
| IMPORTANT: THIS STATEMENT MUST BE READ & SIGNED  **Academic Integrity Statement**  Academic integrity and honesty are fundamental to the academic work you produce at the University of Roehampton. You are expected to complete coursework that is your own and which is referenced appropriately. The university has in place measures to detect academic dishonesty in all its forms. If you are found to be cheating or attempting to gain an unfair advantage over other students in any way, this is considered academic misconduct and you will be penalised accordingly.   ​  **I declare that the work I am submitting is my work, is properly referenced and has not been submitted elsewhere.** |

Table of Contents

[Introduction 3](#_Toc201914423)

[Kanban Framework 3](#_Toc201914424)

[Comparing Kanban, Scrum, and Lean 3](#_Toc201914425)

[Analysis and Organisational Fit 4](#_Toc201914426)

[Proposed Improvement: Automated Workflow Integration 4](#_Toc201914427)

[Improvement implementation 4](#_Toc201914428)

[Conclusion 4](#_Toc201914429)

[Appendix 6](#_Toc201914430)

[References 9](#_Toc201914431)

# Introduction

A Kanban-based Agile approach aligns with the UK Ministry of Justice (MoJ) and Government Digital Service (GDS) principles to support building quickly, testing, and iterating based on user needs (Agile delivery community, GDS, 2016). The report explains Kanban and how it integrates into MoJ workflows, alongside a comparison to Scrum and Lean principles.

# Kanban Framework

Kanban's simplicity, adaptability and focus on visualising work limits the accumulation of work in progress and maximises the pace of a workflow (dos Santos et al., 2018). Traits which suit projects that require flexibility and speedy responses to business changes. Tasks are shown on a board starting with ‘to do’, through ‘in progress’, ‘peer review’, ‘quality assessment’ and ‘ done’. With ‘blocked’ to highlight non-producible work (Appendix, figure 1). The MoJ applies this to responsive interactions with colleagues and stakeholders.

Everyone in the team can follow the events of all work, completed or otherwise. Kanban supports the “release early, release often” MoJ development philosophy. Limiting the work in progress and focusing on an efficient output with blockers resolved early, increases delivery speed.

# Comparing Kanban, Scrum, and Lean

Scrum is an Agile framework used within the MoJ, with a sprint duration of two weeks. Its focus of planning with backlog refinements, implementation of work throughout the two weeks, and reflections and analysis with turndown charts. MoJ scrum teams run sprint ceremonies consisting of stand-up sessions every day, sprint reviews and retrospectives at sprint close. This structure is ideal for fixed disciplines, close collaboration and working for deadlines.

Conversely, the rigidity of scrum can fall into issues when work is slower than expected. Outstanding work at sprint close can compound into the next sprint, which causes difficulties when planning the release of new application features etc. This can affect the analysis of metrics for reporting purposes when committing to organisational standards. Ensuring that work given within a sprint is completed can therefore be a challenge that takes time to refine (Verwijs and Russo, 2023) (Appendix, figure 2).

Lean processes focus on waste elimination and efficiency. So streamlining workflows is a salient objective. The reduction of non-essential work and fast feedback keeps the lean development cycle successful (Leffingwell, 2013) (Appenix, figure 3). For the MoJ, increasing output with a focus on fast delivery times is essential for adhering to Agile’s adaptability, whether there's a change to law or political preference. MoJ code development and User Interface (UI) design has standardised processes including a GOV Design System library (GOV.UK Design System team, 2025). So reducing non-essential work often doesn’t amount to many reductions, the majority of work is considered essential.

Continuous improvement is certainly important, reliable security and accessibility standards are two features that must be highly scrutinised, considering the sensitivity of the information and the enormous scope of the public user base (Abrahams et al., 2024).

In comparison, Kanban is least prescriptive so ideal for adaptable teams such as those in the MoJ. Scrum's time boxes and Lean’s focus on process both have their uses, however, Kanban’s visibility and adaptability make it more suited to public sector development teams which need predictability and change (Appendix, figure 4).

# Analysis and Organisational Fit

The MoJ’s guidance on Agile delivery with an emphasis on iterative user design functions well with Kanban. With transparent work, all disciplines within the development team, senior management and the end user can view the progress of the project. The team, by extension the organisation, now holds accountability which is expected within the public sector.

With Scrum, an unneeded level of complexity and rigidity would derail to dynamism that Kanban provides. The mandatory retrospectives and ceremonies would create negligible benefit to an efficient approach (Matthies & Dobrigkeit, 2019).

Lean which is effective for continuous improvement, loses the practicality and implementation, making it awkward to directly apply without a significant change of expectations.

Therefore, Kanban closely aligns with the MoJ GDS recommendations on open work with documentation and iterating often based on feedback. Kanban would be the optimal fit for the MoJ’s software development teams.

# Proposed Improvement: Automated Workflow Integration

An aspect to improve the Kanban approach is automated workflow integrations, in particular Continuous Integration (CI) pipelines through an already existing GitHub repository (GitHub Docs, n.d). The GDS Agile Service Manual promotes this idea already. Catching bugs early and supporting rapid iteration (Agile delivery community, GDS, 2016). The CI tool GitHub Actions can listen for a status update on a piece of work. When shifting from ‘peer review’ to ‘quality assessment’ an automated integration test suite (such as Capybara) can run to verify the functionality (Appendix, figure 5).

This alignment with MoJ workflows that obsess with security and compliance, is pivotal. The CI pipeline would require valid code conventions and linting standards (Rubocop), security (Snyk) and functionality (RSpec) before allowing new code into the production codebase. Within a team, this reduces personal preference interfering with convention and reduces accidental error. For example, a new feature being developed within the MoJ would integrate with Jira then update the ticket status with the test outcome, and provide an immediate notification (Atlassian, Continuous Delivery, n.d).

# Improvement implementation

The implementation of CI in a Kanban project is easy to configure. First, implement the GitHub Actions pipeline tool. Then set up the pipeline triggers to start when a commit is pushed to the remote GitHub repository, which initiates the tests. Within the Jira project environment, link the GitHub repository to view build statues from the Kanban board (Atlassian Support, 2025).

# Conclusion

The use of Kanban gives the adaptable and user-focused approach that is essential when working within the MoJ software development environment. The flexibility, transparency and spirit of the team's development culture make it an ideal Agile methodology to implement. Scrum and Lean both offer good features, yet they alone aren’t sufficient when developing in an environment with these requirements. The CI improvement offers a great addition by allowing the team to build quickly with more resilience and confidence. The software becomes reliable, understandable and better for when needs change.

# Appendix

A diagram of a question and answer

Description automatically generated

Figure 1 Kanban board steps

A diagram of a product

Description automatically generated

Figure 2 Scrum workflow and ceremony



Figure 3 lean workflow

A diagram of a agile project

Description automatically generated with medium confidence

Figure 4 Process analysis, Kanban, Lean, Scrum

A diagram of a flowchart

Description automatically generated

Figure 5 Continuous integration flow

A diagram of a software project

Description automatically generated

Figure 6 Jira GitHub workflow integration

# References

Agile delivery community (Government Digital Service), 2016. *Creating an agile working environment*. GOV.UK. Updated 23 May 2016. Available at: <https://www.gov.uk/service-manual/agile-delivery/create-agile-working-environment> [Accessed 3 June 2025].

dos Santos, P.S.M., Beltrão, A.C., de Souza, B.P. et al., 2018. On the benefits and challenges of using kanban in software engineering: a structured synthesis study. *Journal of Software Engineering Research and Development*

Christiaan Verwijs and Daniel Russo. 2023. A Theory of Scrum Team Effectiveness. ACM Trans. Softw. Eng. Methodol. 32, 3, Article 74 (May 2023), 51 pages. <https://doi.org/10.1145/3571849>

Leffingwell, D., 2013. *The lean gap: A review of lean approaches to large-scale software development*. *Journal of Systems and Software*, 86, pp.1–X. (Article S0164121213001477). Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0164121213001477> [Accessed 7 June 2025].

GOV.UK Design System team, 2025. *GOV.UK Design System*. GOV.UK. Available at: <https://design-system.service.gov.uk/> [Accessed 7 June 2025].

Abrahams, T.O., Farayola, O., Amoo, O.O. et al., 2024. Continuous improvement in information security: a review of lessons from superannuation cybersecurity uplift programs. *International Journal of Science and Research Archive*, 11(01), [Accessed 20 June 2025].

Matthies, C. & Dobrigkeit, F., 2019. *Towards empirically validated remedies for Scrum retrospective headaches*. arXiv Preprint. Available at: <https://arxiv.org/abs/1910.08763> [Accessed 12 June 2025].

GitHub Docs, n.d. *About continuous integration with GitHub Actions*. Available at: <https://docs.github.com/en/actions/about-github-actions/about-continuous-integration-with-github-actions> [Accessed 21 June 2025].

RuboCop team (Rubocop.org), n.d. *RuboCop – A Ruby static code analyzer and formatter*. Available at: <https://docs.rubocop.org/rubocop/index.html> [Accessed 21 June 2025].

Snyk Limited, n.d. *About continuous integration with Snyk Actions*. Available at: <https://docs.snyk.io/actions/about-github-actions/about-continuous-integration-with-github-actions> [Accessed 11 June 2025].

RSpec Core Team, n.d. *RSpec – Behaviour Driven Development for Ruby*. Available at: https://rspec.info/documentation/ [Accessed 27 June 2025].

Atlassian, n.d. *Continuous Delivery*. Available at: <https://www.atlassian.com/continuous-delivery> [Accessed 19 June 2025].

Atlassian Support, 2025. *Link GitHub workflows and deployments to Jira work items*. Atlassian Support. Published 3 weeks ago. Available at: <https://support.atlassian.com/jira-cloud-administration/docs/link-github-workflows-and-deployments-to-jira-issues> [Accessed 8 June 2025].