26th International Congress on Modelling and Simulation, Adelaide, SA, Australia, 30 Nov–4 Dec 2025
mssanz.org.au/modsim2025

The struggles of software engineering in a science-based context

Dean Holzworth [a], **Neil Huth** [a], **Val Snow** [b], **Hamish Brown** [b], **Rogerio Cichota** [b], **Andrew Paroz** [a], **Julian Rich** [a]

[a] *CSIRO Agriculture and Food, Forest Hill, Queensland, Australia,* [b]*Bioeconomy Science Institute, Christchurch, New Zealand,*
Email: dean.holzworth@csiro.au

Abstract: In the 35 years since the first APSIM (Holzworth *et al*. 2018) software was developed, many things have changed in the software, how developers interact with it, and in external expectations. These changes have accompanied evolution in the usage from a handful of internal users with a strong shared focus to an international user base with highly diverse expectations. We propose that adopting ideas and elements of a software process will strengthen the usage and development of science-based model development.

Keywords: *APSIM, Software process, Agricultural modelling*

1. INTRODUCTION

The initial design and implementation of APSIM began in the early 1990s with the first publication a few years later by McCown (1995). Over time, the governance, software and processes have changed drastically from a closed-source, pay-per-licence to an open licence for public-good research use, encouraging contributions from the broader community. This pivot, which happened with the formation of the APSIM Initiative in 2007, significantly strengthened the user and developer community, which made it an attractive option for users, universities, businesses and research organisations to engage with the Initiative.

The vision for the Initiative is that "APSIM will be widely used and trusted to enhance innovation in the development of new or improved agricultural systems that meet the global demands for food and fibre security and deliver multiple indirect benefits to the environment and society". Many metrics can be used (publications, GitHub contributions, count of user downloads, commercial partner usage, longevity etc) to demonstrate that this vision has been achieved. The APSIM Initiative currently funds 1.0 FTE of software development, 0.8 FTE of a project officer to help coordinate everyone, and it also funds periodical science developments, all of which shows financial viability. APSIM has survived for 35 years, successfully adapting to the changes caused by mobile communications and the internet, which many other software products have not. APSIM and the Initiative that governs it, can be considered a success. Like all long-term efforts, it has not all been smooth sailing. This extended abstract and presentation makes some observations and suggestions for reflection.

2. SOFTWARE PROCESS

Process is crucial. Without process there will be chaos and inefficiencies that lead to poorly written, unstable software that users do not want to use. This causes multiple unofficial versions of the code and developers will 'step on the toes' of one another. The process for APSIM consists of design, peer-review, small iterations of code, testing, automation, documentation, and refactoring. Each of these are applied throughout the development of any feature so it is not a linear process. The collaborative process takes place within Microsoft's GitHub platform in a public repository (https://github.com/APSIMInitiative/ApsimX). Peer-review of all software changes is mandatory. Small iterations are strongly encouraged but are not always possible. Tests (unit tests, integration, validation and sensibility) are mandatory and automatically executed to ensure there are no surprises through unexpected model behaviour. Refactoring (simplifying) the code is a continuous process so when new features are added or defects fixed, the code around it is refactored. While the process works well, there is room for improvement around collaborative design before a feature is implemented, the subsequent documentation once it has been incorporated and testing of the user interface.

3. SCIENCE PROCESS

Since 2007, one of the roles of the Initiative's Reference Panel is to "oversee, approve and manage all APSIM development activities". New models to APSIM or major science changes are peer-reviewed at a high level once they are ready to be incorporated into APSIM. This involves a presentation by the authors of the improvement, with suggestions made for improvement. The process is similar to a manuscript review at a journal.

The Initiative has tried many times to coordinate and prioritise APSIM science developments across member organisations but is unable to set the development priorities of member organisations. Instead, externally funded projects, outside the control of the Initiative, drive most of the APSIM development.

As a group of core users and developers, we would like to see a deeper engagement by the Initiative on science modifications to APSIM. In other settings, science quality and collaboration are often demonstrated through journal papers, and we need the same level of peer-review and rigour in science (and software) contributions to APSIM. Aspects of the software process described above could work for coordinating science changes. GitHub can be used to share designs and ideas of a proposed science change before development commences. This would strengthen collaboration and lead to better designed science and potentially additional test data sets. Status updates could be supplied and questions asked during the new development which will increase transparency. Peer-review can be conducted through GitHub, providing suggestions during the development and testing phases. Currently the Initiative peer-reviews major science changes to APSIM with many smaller changes receiving limited science peer-review. This should also change. One of the challenges with APSIM, is that it is science software. While most scientists have a strong opinion on what APSIM should do, few can implement their ideas in the code base and usher them through the version control system into release. As a result, work is often started to produce results on a software branch for a publication, but then left to stagnate, as integrating the changes proves too long or difficult to complete.

A key reason APSIM software has a robust process is due to direct funding from the Initiative and significant internal funding from key member organisations. To implement a similar process for science contributions it will require funding and a commitment from member organisations to supply in-kind support (noting that in-kind support already dominates the support levels from the members). The project-based way that science is conducted, with commitments and deadlines that are often very tight, will make this challenging. We need to extend the software process thinking to science process thinking which will require increased clarity on what good science looks like. The financial and labour engine room is in projects, not the Initiative. Funding bodies will not fund software, but they will fund science. Project milestones do cause the urgency that undermines process, so the process will need to be stronger to resist this problem. Or perhaps, by making the process clearer, projects will better respond with better planning and the need for timeliness.

There has always been a tension between complexity (so the model can do lots of things) and simplicity (so the code can be kept clean with structural uncertainties minimised). Over time, we have learnt that the simpler approaches lead to better outcomes. There are many examples in APSIM where increasing complexity, brought about by the evolutionary nature of development, has led to hard-to-understand science and code where the user is not exactly sure what they are running. This is often brought about by the lack of upfront design and a lack of refactoring (simplifying) the code.

4. CONCLUSIONS

There have been several disruptors during the history of APSIM that have led to key pivot points. Moving to open-source, distributed development and rewriting APSIM to a more modern platform are three examples (Holzworth 2018). Cloud computing and Artificial Intelligence (AI) are major disruptors that cannot be ignored. They are being used at various points in the software process (coding, peer-review of modifications) and have the potential to contribute to science model development. It is a time-consuming process to gather / clean / understand datasets, calibrate models, design parameter sensitivity analyses. AI may have an important role to play to help create better science contributions to APSIM more quickly.

The one thing that has and will always be constant is the requirement for competent users running models. No matter how good the models are, they will give invalid results if poorly configured. Fixing a poorly configured simulation can have as much or more bearing on model performance than changing the model's parameters.

While there are several observations and suggestions for improvement in this manuscript, the Initiative is "doing a lot right". That is evidenced by its longevity, size of the APSIM community, number of publications etc.

REFERENCES

Holzworth DP, Huth NI, Fainges J, Brown H, Zurcher E, Cichota R, Verrall S, Herrmann NI, Zheng B and Snow V (2018) APSIM Next Generation: Overcoming challenges in modernising a farming systems model. Environmental Modelling & Software, Volume 103, pp 43-51, https://doi.org/10.1016/j.envsoft.2018.02.002.

McCown RL, Hammer GL, Hargreaves JNG, Holzworth D and Huth NI (1995) APSIM - An Agricultural Production System Simulation-Model For Operational-Research, Mathematics And Computers In Simulation, Volume 39, Issue 3-4, pp 225-231, https://doi.org/10.1016/0378-4754(95)00063-2.

