BARNETT WADDINGHAM

Part of HOWDEN

The Year Of The Snake

Shedding old views and embracing new ones

31 January 2025

Executive Summary

The Lunar New Year has ushered in the Year of the Snake. The zodiac symbol of the the snake represents an animal of wisdom, intuition but also an animal of transformation as it eventually evolves into a dragon. This seems very apt as in the actuarial world one particular serpent is definitely the centre of attention right now: Python, the open-source programming language helping to reshape insurance modelling.

Python isn't new of course, it has been around for over 30 years and is used in many industries thanks to its accessibility and versatility. But why is it now at the centre of so many actuarial modelling discussions? Could it be that Python is the long-awaited solution for balancing cost, scalability, and auditability in relation to the many models required within a company?

We will consider the 'traditional' software approaches and examine how Python could be an alternative to these. This will draw upon BW's recent experience supporting clients in implementing actuarial modelling solutions in Python which has convinced us that Python can transform actuarial technology, and that 2025 may be a turning point for its wider adoption within the industry.

Traditional Modelling Approaches

Historically, many actuaries who needed quick, cost-effective models relied on Excel. They would create yet another spreadsheet, possibly save multiple versions with varying names, add in some VBA and pivot tables, throw in some "Index" references and often hard-code parameters. This practice can easily lead to errors, make updates painful, and can make checking and auditing the models a laborious and risky task. It also provides little scope for standardised processes or automation.

On the other end of the spectrum, actuaries may adopt one of the plethora of vendor-based solutions. These can come with various licensing fee structures for different components along with libraries including layers of code, parameters and options, making them large and unwieldy. Updating and checking a single formula can take hours to verify and upgrading to a new version can take weeks of checking, slowing down the entire modelling process and increasing the risk of errors. You may also encounter older offerings that do not integrate well with new technology or changes in ownership that impact the long term business commitment to the software and its further development.

Enter Python

For many, modelling seemed to alternate between these two worlds of either the accessibility of Excel or the complexity of vendor-tools. Python does however offer an alternative path: it blends accessibility with ease of development, enterprise-grade capabilities and deep integration with a host of AI assistants. Actuaries can then build models that are:

- Structured for scalability
- Intuitive and documented with clear, maintainable shareable code
- Automated for rapid updates
- Validated, tested and documented quickly with AI
- Collaboration-friendly through version control tools like Git
- Able to integrate with many 3rd party solutions

"Python offers an alternative path: it blends accessibility with ease of development, enterprise-grade capabilities and deep integration with AI"

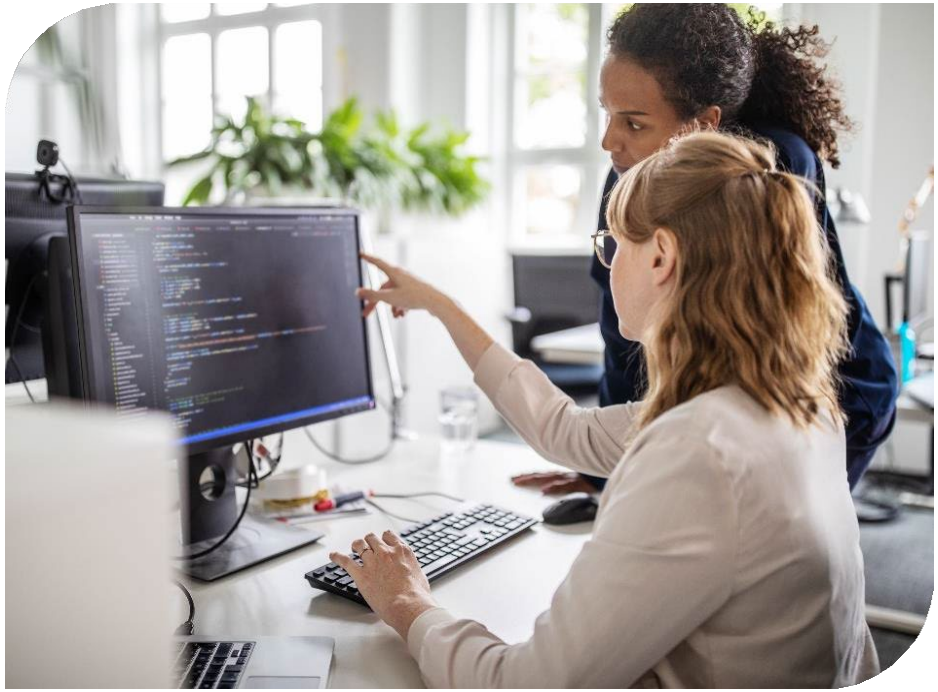
A Recent Case Study

BW recently worked with an insurer to replace their existing vendor-based actuarial valuation models with a Python-based solution. Within six months of initiating discussions, the existing vendor-based valuation models were replaced with a suite of Python models run via a graphical user interface (GUI), also developed within Python. The scope of the modelling covered a range of term assurance, whole of life and critical illness products, with different premium structures, reinsurance structures and both short and long-term reserving. The model produced reserves and cashflows for both Solvency II and UK GAAP reporting requirements simultaneously. We were able to quickly and easily create documentation for auditors to review and compare the old and new models, whilst also making improvements to the models where necessary.

Why choose Python?

When selecting a platform, the project team reviewed Excel, vendor solutions, open-source languages, and enterprise tools like C# but ultimately rejected them for this project for the following reasons:

- Excel is inexpensive and familiar but struggles with large-scale processing, lacks robust version control, auditing or production features.
- Vendor solutions come with multiple licensable options, some charging access to cloud infrastructure, others offering hosted only solutions. With all of them you are beholden to the vendors, which can and does change, with regard to their timing of development roadmaps and contractual terms potentially restricting access and ownership.
- C# is a solid enterprise language, especially if you have a skilled in-house development team and models built will run quickly if that is what is needed. We required a more quickly accessible and transparent language that a range of users could use.
- C++ and C offer speed too but involve elements such as manual memory management, which introduces security risks and the need for more specialist resource requirements.



Other open-source languages, such as R, Julia, and Rust, each have some strengths in other areas, but all lack either a mature ecosystem, accessible structure for the style of calculations that are needed for actuarial cashflow models or a depth of resource availability. Python however, stood out due to its ease of use, with our resident experts able to train up members of the team quickly, and with a wealth of packages available means it makes it faster to prototype and implement. Also its automatic memory management means it's coding in C or C++ is less risky.

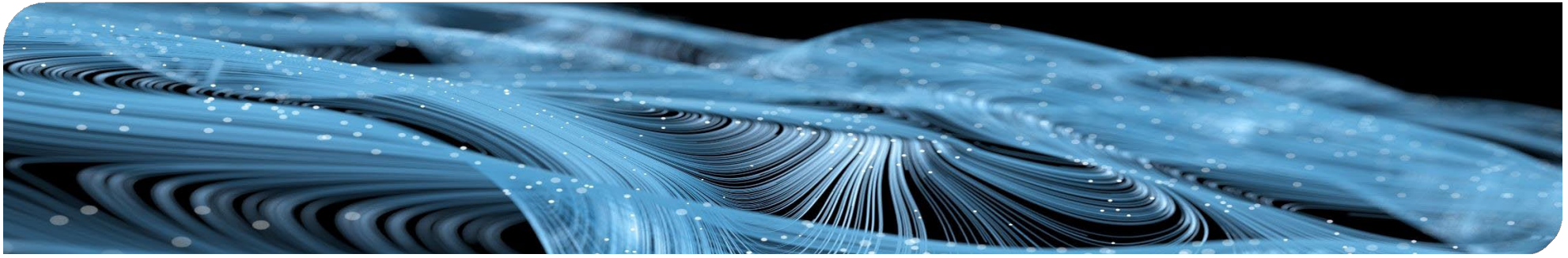
One other positive for Python is the recent advancement of integration with AI-based coding assistants, which help write, test and document code. This is a big advantage over proprietary or less common languages as the AI assistants aren't integrated as well or don't have the depth of material in which to reference and learn from. This further enhances its accessible nature and highlights the speed with which new technologies are easily integrated.

Finally, an open-source solution like Python can significantly save the client long-term costs. The insurer not only saves over the lifetime of the models by avoiding vendor license fees, they also reduce the possibility of requiring another vendor implementation in the future should the situation change.

Implementation and Outcomes

BW worked with the client to deliver 13 fully reconciled, tested and documented valuation models. A user-friendly bespoke interface was also created that separated development from production, reducing potential errors and saving time due to exposing only the options that needed exposing at runtime. The models used the pre-existing data formats and produced replicas of the outputs from the previous models, making it easy to slot into the current valuation processes. Model output was also improved to produce both high-level summaries and detailed cashflow projections much more easily. Critically, the AI tools helped verify coding approaches and generate the documentation meaning the process was accessible, intuitive and efficient for all users. This was all completed within 3 months of getting access to the original data and results, through to submission and agreement for use by the Audit Committee.

Now the client owns all of their models, they are more accessible, will spend less time setting up and running them due to refined runtime processes, and will recoup the implementation costs within a few years due to the removal of ongoing software licenses.



The Future of Modelling

So, is Python the modelling solution we have been waiting for….a flexible, stable low-cost, accessible solution? We believe it definitely is! It has always offered an alternative to Excel and vendor-based systems in the past and has enabled users to start small and evolve quickly to create robust, secure processes and systems. However, the depth of expertise and trust around the language was lacking in the actuarial world. The advent of AI tools that are easily integrated into the development process are no longer just hypotheses or conjecture, in our experience they are now actual tangible tools of great practical benefit in implementing 'real life' actuarial models, meaning those uncertainties are easily removed. Code can now be created and verified almost instantly, automated testing can be implemented, and documentation explaining the model in natural language can be created in minutes.

So, let's embrace the small yellow-and-blue Python, as its namesake would its' prey in the natural world, as part of the future of actuarial modelling.

Welcome to 2025, the Year of Python in the actuarial world!

Contact (Amit.Lad@Barnett-Waddingham.co.uk or Gillian.Mathias@Barnett-Waddingham.co.uk) to arrange a demonstration and discussion of our experience and insights in more detail, as well and how we can assist you in embracing Python.

www.barnett-waddingham.co.uk