Institute and Faculty of Actuaries

IFoA Life Conference

Welcome to: **Would you like any open source with your models today?**

Presented by:

Gillian Mathias

Amit Lad

Migrating valuation model to Python with the help of AI

Insights from the Chief Actuary and the Implementation Lead

1. Why was re-platforming necessary?
2. Why choose Python?
3. What were the practical steps?
4. Reflections

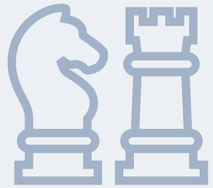
Gillian Mathias
Chief Actuary
Barnett Waddingham

Amit Lad
Principal
Barnett Waddingham

Why was re-platforming necessary?

Why was re-platforming necessary?

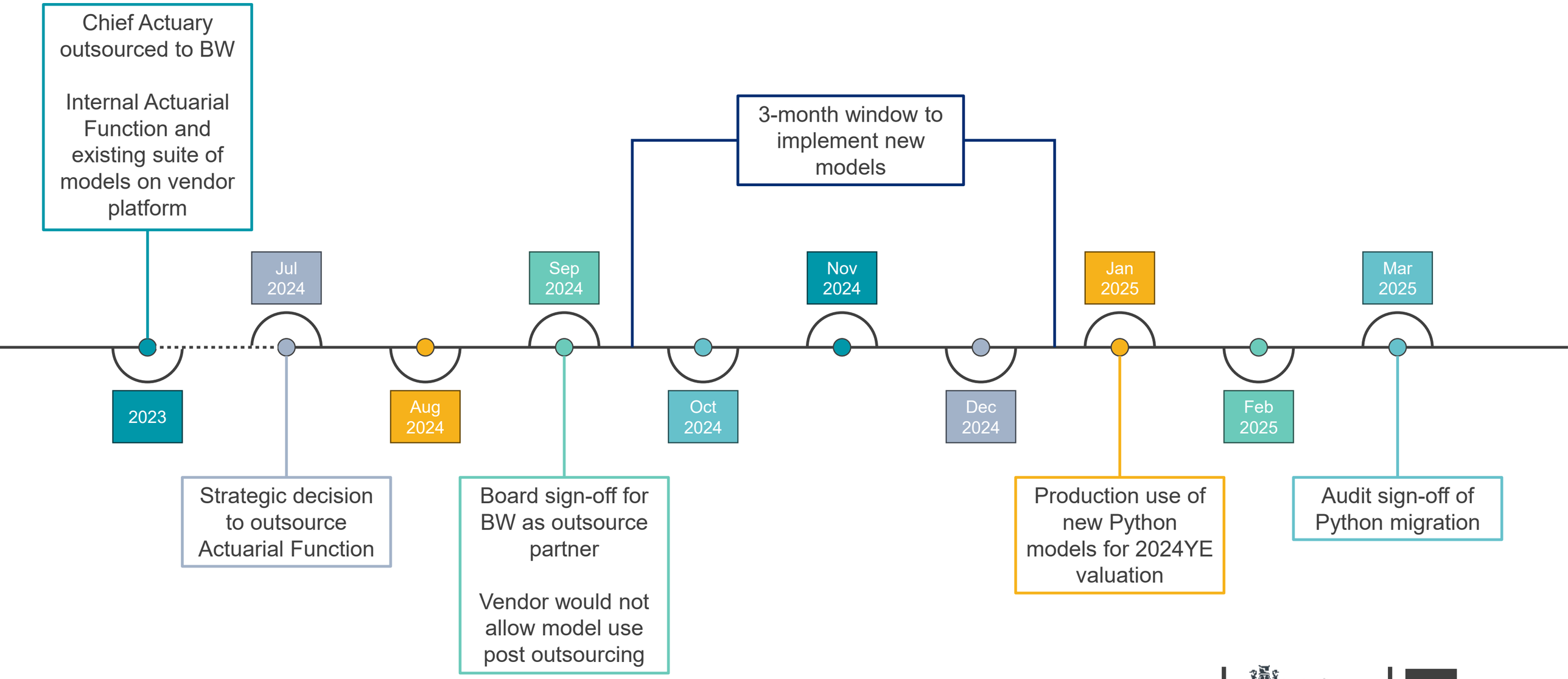
Previously in-house Actuarial Function and existing suite of models on vendor platform



Strategic decision to outsource Actuarial Function to Barnett Waddingham

Vendor would not allow Barnett Waddingham use of existing models

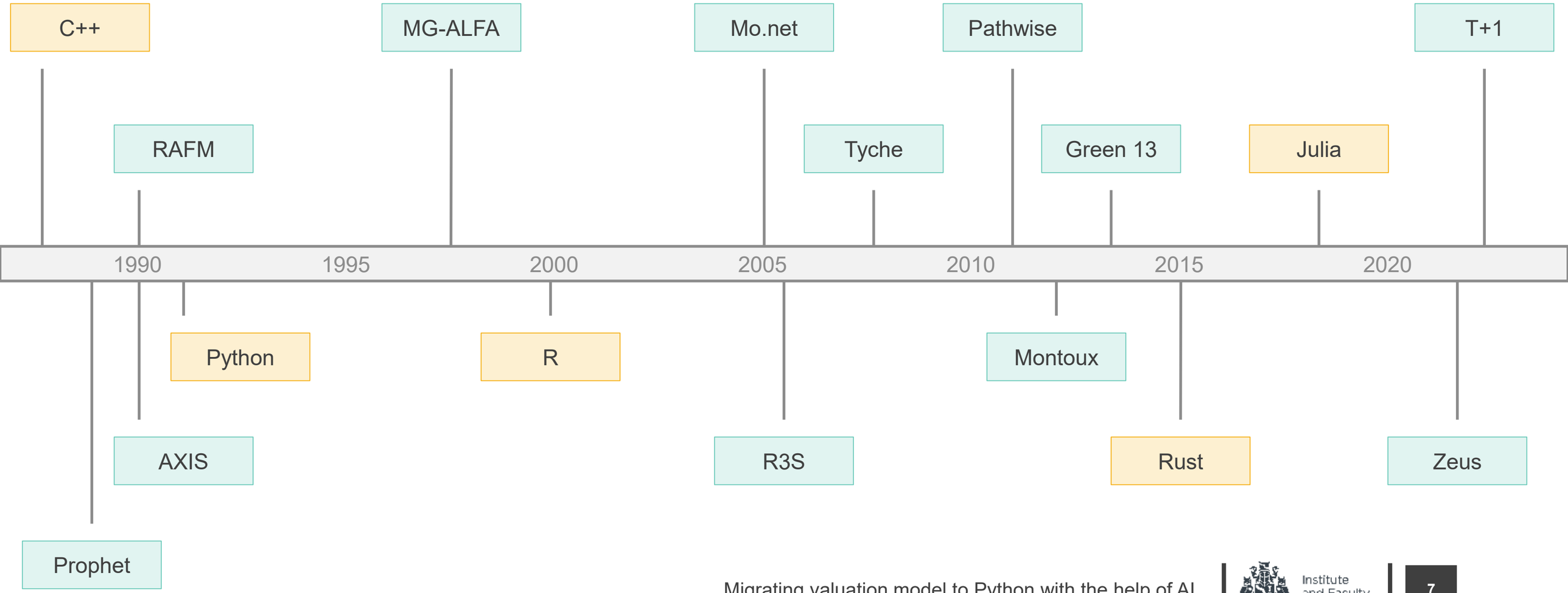
Background and timelines



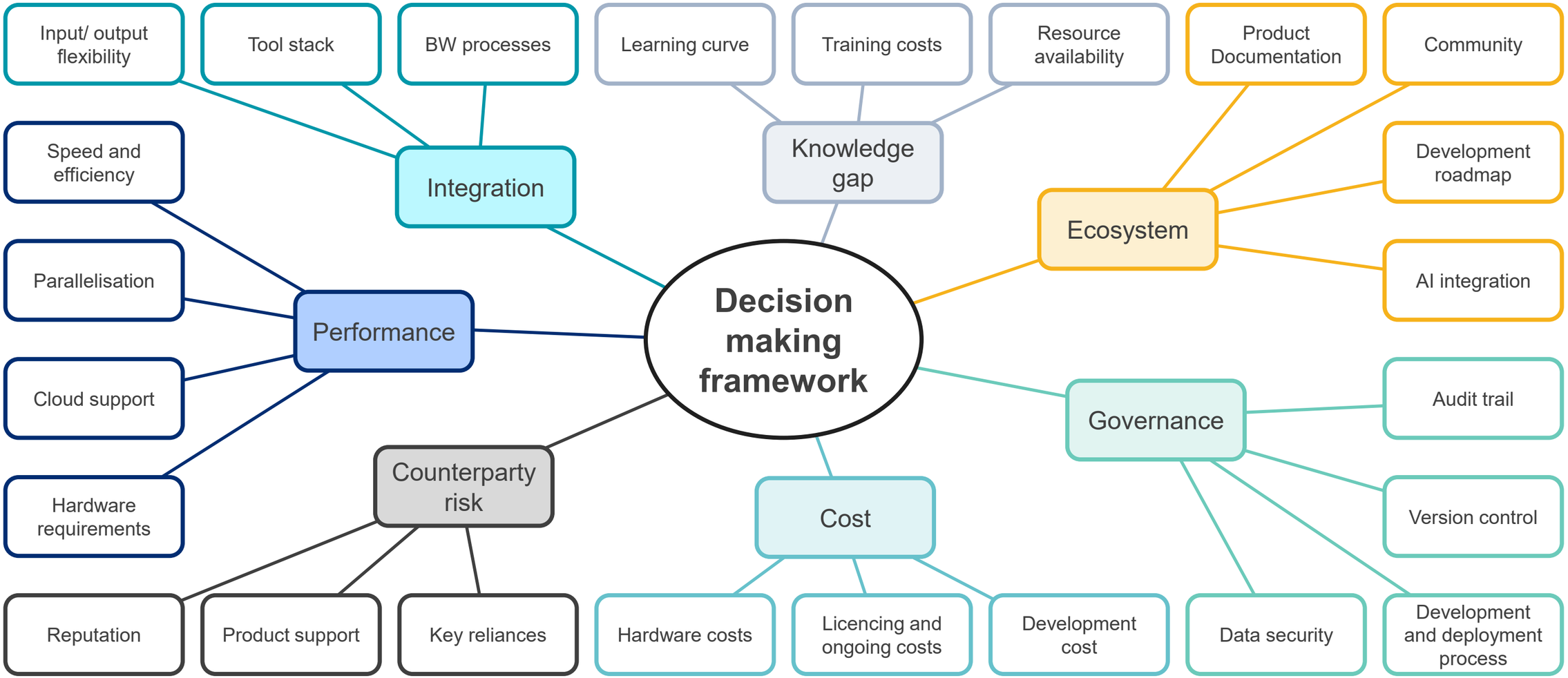
Why choose Python?

BW longlist of options

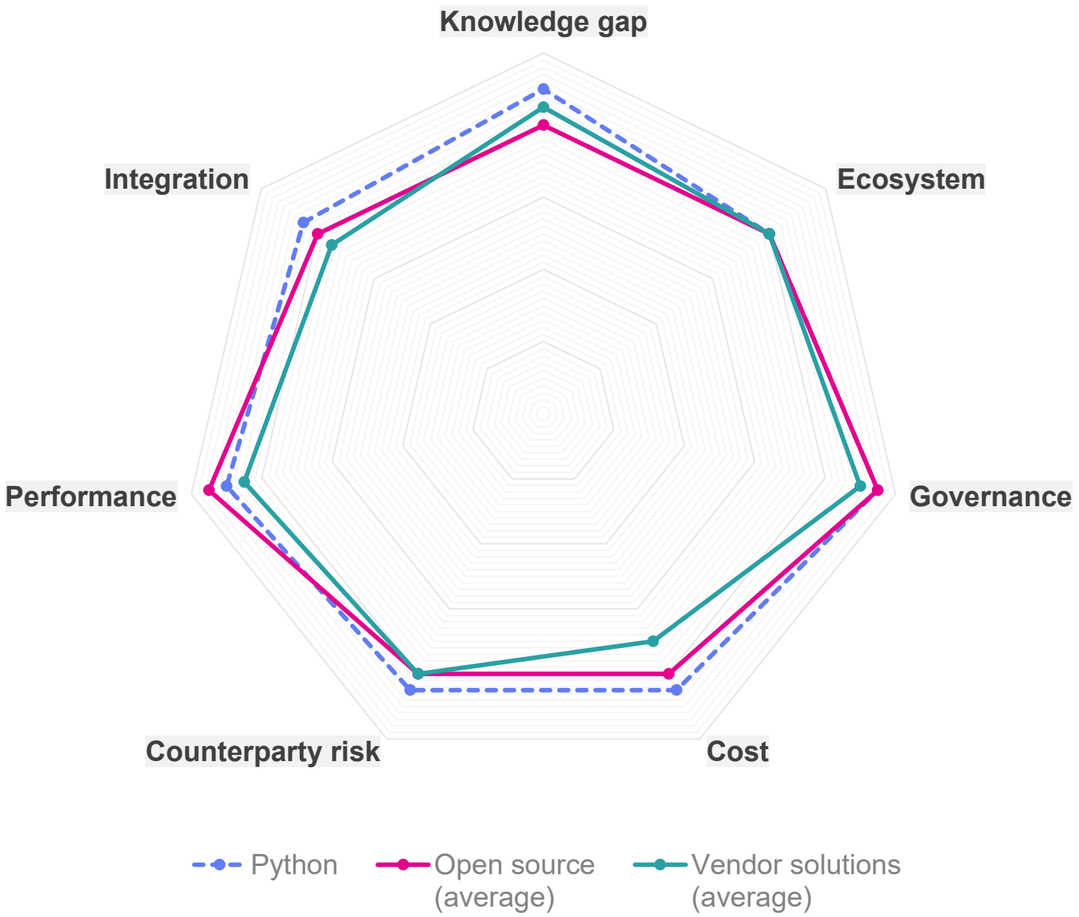
Ordered by approximate first release



All platforms capable of modelling life actuarial features



Assessment results



Python strengths:

- Incredibly flexible general purpose programming language
- Already used within the business – skills and expertise available
- No ongoing licence cost
- Hardware infrastructure already exists
- Integrates with other BW systems and databases, and AI/ML work
- AI integration to speed up development
- Possible to transfer ownership to client, and for client to run and operate models

Key risks identified:

- Limited external actuarial support for issues
- Technical debt in a new platform

What were the practical steps?

What needed modelling?

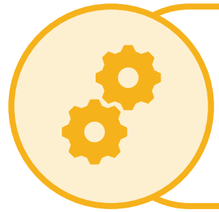
48,000 policies over 12 products with 23YE SII BEL of £14.4m

Products

- Life and Critical Illness
- Long and Short-term products
- Term Assurance and Whole of Life

Features

- Lapses
- Premium structures
- Currencies
- Commissions
- Expenses
- Reinsurance
- (lots more…)



Capabilities

- UK GAAP (including zeroisation capabilities)
- SII BEL, SCR stresses and Risk Margin calculations
- Projections and Scenarios for ORSA

Model build process

Area	Task	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14
Project setup															
Planning															
Building															
Testing															
Completion															

3-month delivery window

(Planned longer, but contingency used up before project began!)

3 key parts to model build process.

Often running in parallel to ensure delivery

Model build process - overview

Area	Task	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14
Project setup	Engagement terms signed														
	DevOps/ git														
	Python environment														
	Visual studio code														
Planning	Specify design requirements														
	Chief Actuary sign-off on design requirements														
	Specify model specification														
	Chief Actuary sign-off on specification														
	Specify model documentation structure														
	Specify testing framework														
	Specify testing tolerances														
	Chief Actuary sign-off on test plan														
	Auditor sign-off on test plan														
Building	Create modular, extendable, modelling framework														
	Create model front-end														
	Model outputs - debugging and for model testing														
	Model outputs - aggregate outputs for production and for model testing														
	Calculation engine for products prioritising materiality and complexity														
	Model documentation														
	Model user-guide														
Testing	Unit testing of code														
	Per policy cashflow reconciliation for sample policies														
	Product level reconciliations for each product														
	Reconciliation of previous results for SII and UK GAAP (base and sensitivities)														
	Testing documentation														
Completion	December audit committee (incl. Chief Actuary sign-off)														

Chief Actuary sign-off critical

Area	Task	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14
Project setup	Engagement terms signed														
	DevOps/ git														
	Python environment														
	Visual studio code														
Planning	Specify design requirements														
	Chief Actuary sign-off on design requirements														
	Specify model specification														
	Chief Actuary sign-off on specification														
	Specify model documentation structure														
	Specify testing framework														
	Specify testing tolerances														
	Chief Actuary sign-off on test plan														
	Auditor sign-off on test plan														
Building	Create modular, extendable, modelling framework														
	Create model front-end														
	Model outputs - debugging and for model testing														
	Model outputs - aggregate outputs for production and for model testing														
	Calculation engine for products prioritising materiality and complexity														
	Model documentation														
	Model user-guide														
Testing	Unit testing of code														
	Per policy cashflow reconciliation for sample policies														
	Product level reconciliations for each product														
	Reconciliation of previous results for SII and UK GAAP (base and sensitivities)														
	Testing documentation														
Completion	December audit committee (incl. Chief Actuary sign-off)														

Upfront discussion and buy-in from auditors

Area	Task	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14
Project setup	Engagement terms signed														
	DevOps/ git														
	Python environment														
	Visual studio code														
Planning	Specify design requirements														
	Chief Actuary sign-off on design requirements														
	Specify model specification														
	Chief Actuary sign-off on specification														
	Specify model documentation structure														
	Specify testing framework														
	Specify testing tolerances														
	Chief Actuary sign-off on test plan														
	Auditor sign-off on test plan														
Building	Create modular, extendable, modelling framework														
	Create model front-end														
	Model outputs - debugging and for model testing														
	Model outputs - aggregate outputs for production and for model testing														
	Calculation engine for products prioritising materiality and complexity														
	Model documentation														
	Model user-guide														
Testing	Unit testing of code														
	Per policy cashflow reconciliation for sample policies														
	Product level reconciliations for each product														
	Reconciliation of previous results for SII and UK GAAP (base and sensitivities)														
	Testing documentation														
Completion	December audit committee (incl. Chief Actuary sign-off)														

Modelling framework first, calculation engine second

Area	Task	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14
Project setup	Engagement terms signed														
	DevOps/ git														
	Python environment														
	Visual studio code														
Planning	Specify design requirements														
	Chief Actuary sign-off on design requirements														
	Specify model specification														
	Chief Actuary sign-off on specification														
	Specify model documentation structure														
	Specify testing framework														
	Specify testing tolerances														
	Chief Actuary sign-off on test plan														
	Auditor sign-off on test plan														
Building	**Create modular, extendable, modelling framework**														
	Create model front-end														
	Model outputs - debugging and for model testing														
	Model outputs - aggregate outputs for production and for model testing														
	Calculation engine for products prioritising materiality and complexity														
	Model documentation														
	Model user-guide														
Testing	Unit testing of code														
	Per policy cashflow reconciliation for sample policies														
	Product level reconciliations for each product														
	Reconciliation of previous results for SII and UK GAAP (base and sensitivities)														
	Testing documentation														
Completion	December audit committee (incl. Chief Actuary sign-off)														

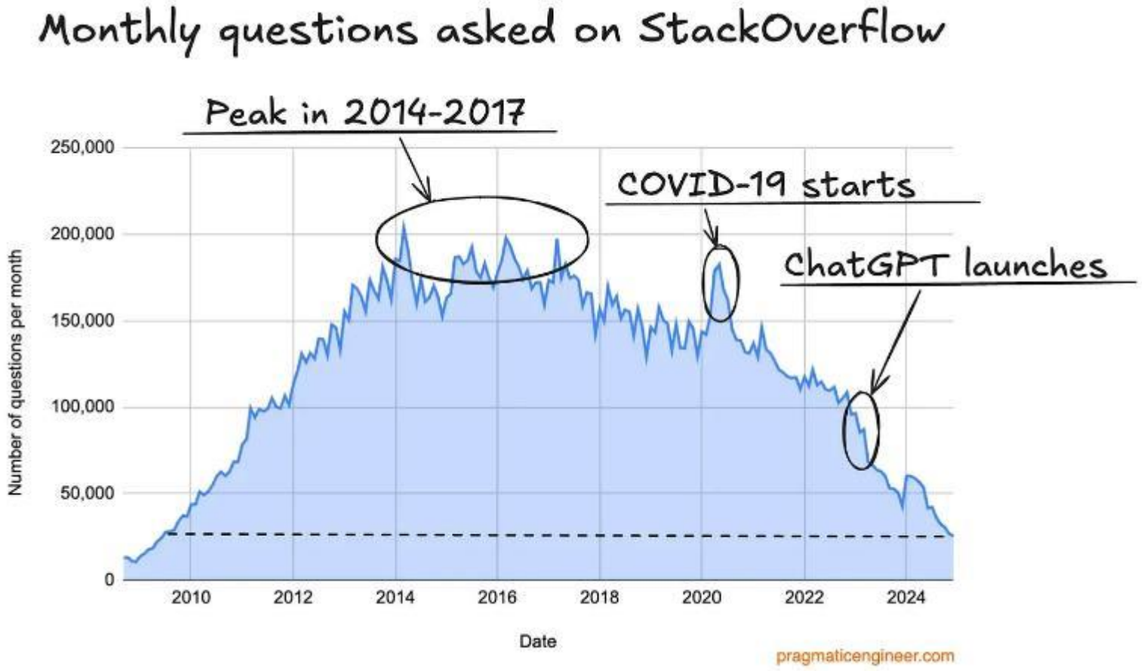
Documentation and testing carried out in parallel

Area	Task	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14
Project setup	Engagement terms signed														
	DevOps/ git														
	Python environment														
	Visual studio code														
Planning	Specify design requirements														
	Chief Actuary sign-off on design requirements														
	Specify model specification														
	Chief Actuary sign-off on specification														
	Specify model documentation structure														
	Specify testing framework														
	Specify testing tolerances														
	Chief Actuary sign-off on test plan														
	Auditor sign-off on test plan														
Building	Create modular, extendable, modelling framework														
	Create model front-end														
	Model outputs - debugging and for model testing														
	Model outputs - aggregate outputs for production and for model testing														
	Calculation engine for products prioritising materiality and complexity														
	Model documentation														
	Model user-guide														
Testing	**Unit testing of code**														
	Per policy cashflow reconciliation for sample policies														
	Product level reconciliations for each product														
	Reconciliation of previous results for SII and UK GAAP (base and sensitivities)														
	Testing documentation														
Completion	December audit committee (incl. Chief Actuary sign-off)														

Using AI to meet 3-month development window

… better than using StackOverflow…?

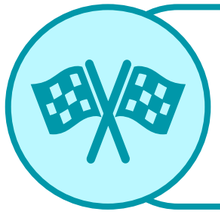
- Coding is proven use case for LLMs.
- Integrated into our development environment. AI had access to code-base, but not data.
- Key use cases:
  - Coding patterns (vibe+ coding)
  - Debugging
  - Automated documentation
  - Writing unit tests
- AI outputs needed testing and review.



Source: https://blog.pragmaticengineer.com/stack-overflow-is-almost-dead/

Reflections

Reflections



Overall

- Delivered a fully tested model to agreed timelines
- No audit queries
- Successfully used for 2024YE valuation
- Have now migrated 5 other firms' valuation models to Python

Success factors

- Up-front buy-in from Chief Actuary and auditors
- Dedicated Python developers in implementation team
- Set out modelling framework before building individual products

Benefits

- New run times 10%-95% of old run times
- Improved model governance and run audit trails

Next steps

- Address technical debt
- SCR projections for business planning

Q&A

Institute and Faculty of Actuaries

Thank you

For more information, please contact:

gillian.mathias@barnett-waddingham.co.uk

amit.lad@barnett-waddingham.co.uk