OliverWyman

Root
Insurance Co

ACTUARIES AND THE TECH MINDSET

CAS RPM Seminar
March 2022

CONFIDENTIALITY

Our clients' industries are extremely competitive, and the maintenance of confidentiality with respect to our clients' plans and data is critical. Oliver Wyman rigorously applies internal confidentiality practices to protect the confidentiality of all client information.

Similarly, our industry is very competitive. We view our approaches and insights as proprietary and therefore look to our clients to protect our interests in our proposals, presentations, methodologies, and analytical techniques. Under no circumstances should this material be shared with any third party without the prior written consent of Oliver Wyman.

MEET THE SPEAKERS

Claudine Modlin
FCAS, MAAA

Root Insurance

Claudine is the VP of Pricing, Underwriting and State Management at late stage startup, Root Insurance. She has 25+ years P&C experience and has worked as an actuary, insurance consultant and company executive.

Olivier Brown
FCAS, MAAA

Oliver Wyman

Olivier Brown is a Principal in the Montreal office of Oliver Wyman Actuarial Consulting. He specializes in developing predictive analytics and machine learning solutions for actuarial, underwriting, and claims applications.

Raphael Brissette
FCAS

Oliver Wyman

Raphael Brissette is a Consultant in the Montreal office of Oliver Wyman Actuarial Consulting. He specializes in developing software solutions for actuarial and other insurance functions. He has experience with commercial lines pricing and underwriting.

Sabrina Tan

Oliver Wyman

Sabrina Tan is a Consultant in the New York office of Oliver Wyman Actuarial Consulting. She provides P&C actuarial consulting services to a variety of insurance organizations and has experience with predictive analytics, process improvement, pricing, and reserving.

What we plan to cover...

Tech mindset

- What is it (from the trenches of an InsurTech)?
- What are some guiding principles?

How actuaries can adopt the tech mindset

- Efficiency and Scale – why move from spreadsheets to a software development approach?
- Automation and Speed – how can automation be built into the way actuaries work?
- Cutting edge Technology – what are some tools actuaries can leverage to execute these techniques?

Recap

Root
Insurance Co

1 Tech Mindset

Root
Insurance Co

A little about my journey...

- 25+ years in P&C insurance
- Started my actuarial career in traditional pricing role at Allstate
- Became an evangelist for predictive modeling in insurance (led to 17 years in consulting with WTW)
- Joined Root (late stage auto insurance start-up) as leader of Pricing, UW and State Mgt in Aug 2021

What have I learned about principles that guide InsurTechs?

- *Mindset*
- *Technology*
- *Organizational design*

Root
Insurance Co

Mindset

- Subject matter experts collaborate with software engineers and architects to support speed to market
  - Talk early, talk often
  - Expose the problem (don't jump to solutions)
  - Hire engineers who excel at architecting solutions to business problems

- Be agile
  - Have a decision framework for prioritizing work
  - Start with a ruthlessly defined MVP and iterate
    - MVPs build out your biggest assumptions
    - Define next steps for whether it fails or succeeds
    - Be clear on decision-making authority

ERRR...

CAN'T STOP. TOO BUSY!!

Root
Insurance Co
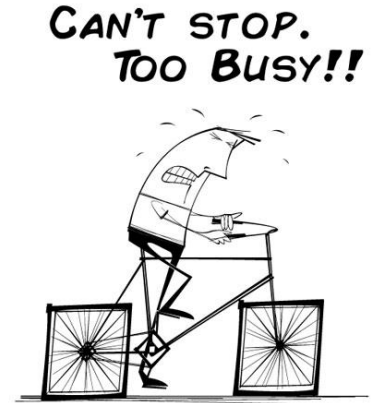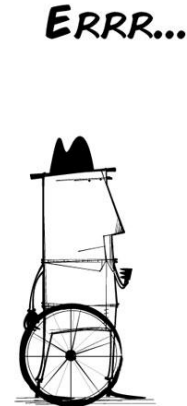
Technology

- Cloud: rent, don't own
- Invest in data architecture
  - Know your unknowns
  - Design with coding flexibility; allows you to build and pivot quickly as solutions evolve
- Regularly reduce tech debt
  - Dedicate 20% of team's efforts

Types of Technical Debt

Root
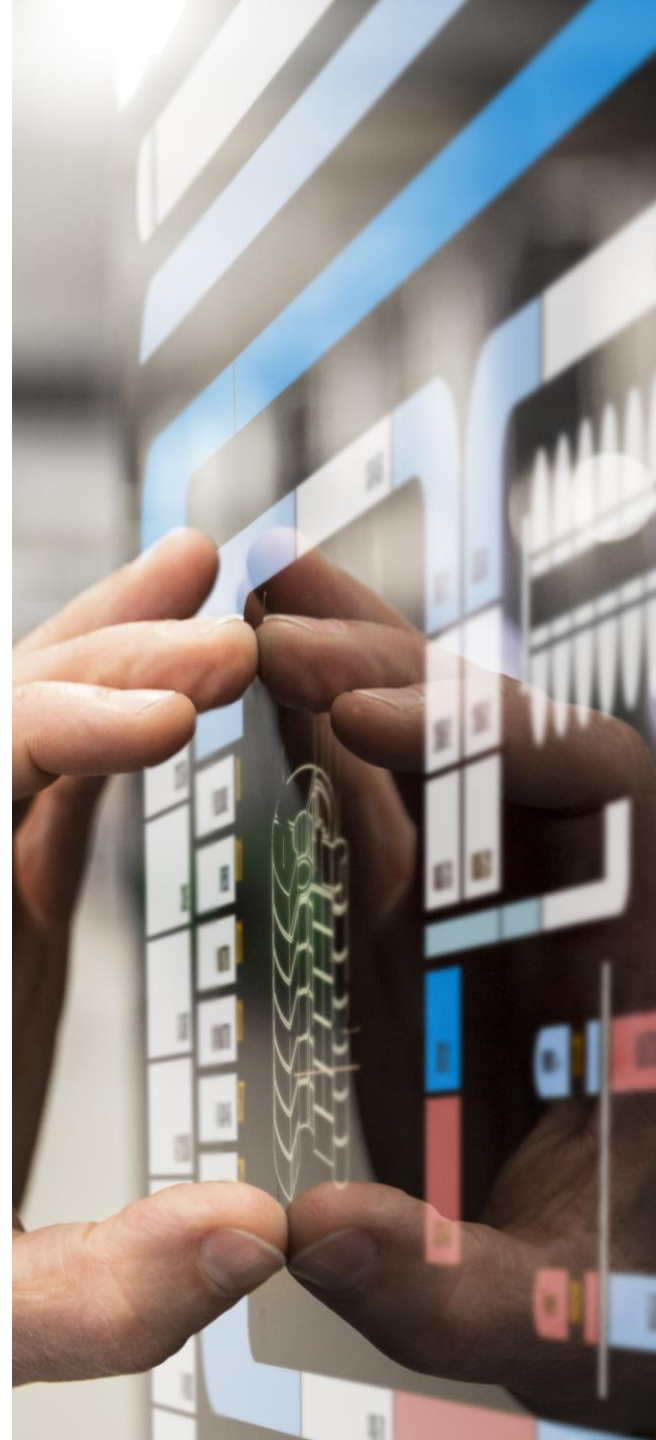Insurance Co

Organizational design

- Typically a business unit engages with IT professionals with assistance from business analysts (BAs) and a Program Management Office (PMO)
  - Often involves BAs writing comprehensive requirements that may or may not make sense to IT
  - Can result in re-work

- Insuretechs have a Product team that
  - Takes the business strategy and breaks it into implementable features in a roadmap
  - Develops cost/benefit projections to inform priorities
  - Measures its success through Objectives and Key Results (OKRs)
  - Pros: data-driven and results-oriented
  - Cons: can muddy role clarity with business teams if not defined well
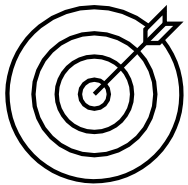

Root Insurance Co
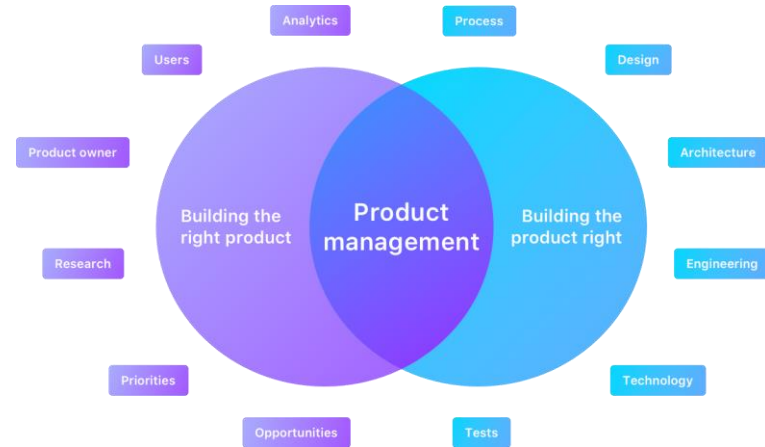
2

HOW ACTUARIES CAN ADOPT THE TECH MINDSET
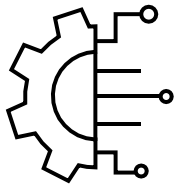
OliverWyman

HOW CAN ACTUARIES FACILITATE SPEED TO MARKET?

**Actuaries can add tremendous value
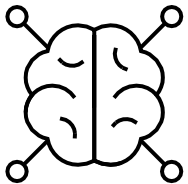to insurance organizations**

Data Driven Thinking
Actuaries understand the importance
of using data to drive decision making

Technical Expertise
Actuaries have a high degree of technical
skill and can leverage sophisticated tools

Knowledge of Insurance Business
Actuaries have deep knowledge
of insurance data, products,
and operations

**Yet many actuarial teams struggle with completing
basic actuarial tasks**

- Actuaries still perform **routine**, repetitive tasks
  such as running reports and preparing data

- There is significant **duplication of effort**
  in most actuarial processes

- Actuaries have **limited involvement** in other areas
  such as underwriting, claims, marketing, and finance

- Actuaries are working at a **fraction of their potential**
  and spend most of their time working on **tasks they
  don't enjoy**

ACTUARIAL TEAMS NEED TO ADOPT SOFTWARE DEVELOPMENT APPROACHES

OliverWyman

Leading organizations in all industries have adopted software development approaches to run their businesses. Actuarial teams have not adopted these approaches

	Key Opportunities	Current Approach	Software Development Approach
01	Efficiency and Scale	Spreadsheet based work products are • Difficult to reuse • Not scalable • Require significant duplication of effort • Hard to verify for errors	Software based work products • Are modular and reusable • Eliminate duplication of effort • Have built in error verification • Enable organizations to gain lasting efficiencies and economies of scale
02	Automation and Speed	• Automation is built on an as-needed basis using scripts (usually in VBA) • Automation scripts often break and produce errors because the underlying spreadsheets or processes change • Most scripts are "black-boxes"	• Leads to better structuring of data and processes • Enables automation since it is built in the way actuaries work • Eliminates manual process errors • Allows to run processes quickly
03	Cutting Edge Technology	• Modeling is done primarily using traditional GLM modeling tools • Access to cutting edge tools is possible, but difficult	Enables easy access to cutting edge tools • Machine learning tools (e.g. PyTorch) • Computation optimization tools (e.g. Numba, Dask) • User Interface tools (e.g. Dash, Django)

2A

EFFICIENCY AND SCALE

OliverWyman

SPREADSHEET BASED WORK PRODUCTS LEAD TO SIGNIFICANT DUPLICATION OF EFFORT

Most actuarial teams work primarily in spreadsheet environments
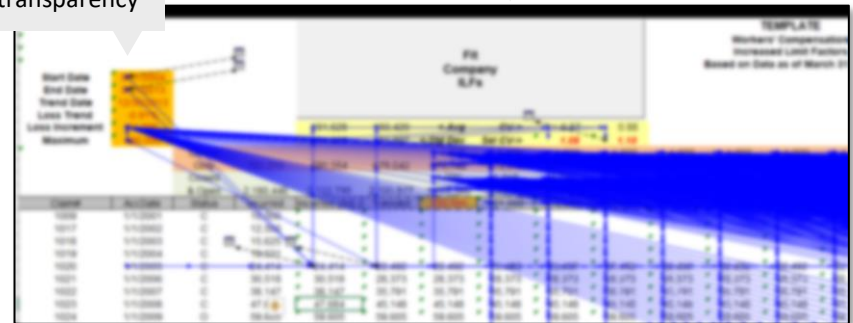
- Spreadsheet environments promote development of **inefficient tools and processes**

- Spreadsheet designs typically create significant **duplication of effort** when analyses need to be updated

- As a result, actuarial resources are perpetually tied up in **repetitive tasks** to update analyses

Manual data entry is typically needed in multiple locations within a workbook

Macros, used to automate certain calculations, often break when the underlying spreadsheet or data changes

Actuarial analyses contain hundreds of formulas which are linked in arbitrary ways making them hard to verify and creates lack of transparency

Spreadsheets are created in an "ad-hoc" manner, designs vary significantly and are not necessarily intuitive

TRANSITIONING TO A SOFTWARE DEVELOPMENT ENVIRONMENT ENABLES ACTUARIAL TEAMS TO GAIN LASTING EFFICIENCIES AND ECONOMIES OF SCALE

OliverWyman

Advantages

- Using software development practices allows actuaries to
  - Produce **modular and reusable** work products
  - Increase **ability to collaborate**
  - Build in **automated error verification** within their work products
- This eliminates duplication of effort, reduces errors, and **saves time** in the long run
- These practices drive integration with a software developing organization by blending **actuarial** and **continuous development**

Examples of Software Development Practices
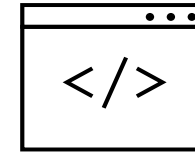
Object Oriented Programming
Enables reusability
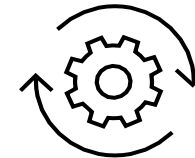
Unit Testing
Reduces errors and saves time

Rigorous Version Control
Improves ability to collaborate

OBJECT ORIENTED PROGRAMMING ENABLES REUSABILITY VIA MODULAR DESIGN AND ABSTRACTION

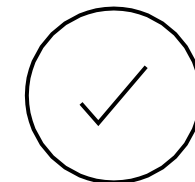*In a tech-focused organization, reusability can facilitate increased collaboration and shared tooling between teams.*

Modular Design

- Modular design means breaking down systems and processes into **small, specialized, modules**

- Modules are built and tested independently, which means they can **work individually** and be **reused** as part of multiple processes or systems

Abstraction

- **Abstraction** reduces complexity of use by hiding unnecessary details from the user

- This is what allows software modules to be used **without in-depth knowledge** of how they were built

Example

- **Python packages** are an example of software built with modular design and abstraction

- Software developer around the world have access to **hundreds of thousands of software packages** to accelerate their work

- For instance, a clustering algorithm (k-nearest neighbors) can be implemented in four lines of code, without knowing anything about the details of the algorithm

```python
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(predictors_train, target_train)
knn.predict(predictors_test)
```

UNIT TESTING REDUCES ERRORS AND SAVES TIME

What is unit testing?

- A set of tests is created for **each software module**

- The tests ensure each software module **works as intended**

- **All tests run** when a change is made to one part of the software system

Why do we need unit testing?

- Contributors can update a program and make sure they are not breaking it

- Helps new developers understand how the program works

- Quickly find broken sections of a program during the development phase

Note: Modular design is required to implement unit testing

Example

- A set of scripts used for re-rating need to be updated

01   📄 📄 … 📄   **Re-Rating Scripts**
Change is made to one of the scripts

02   ⊗   Entire process fails

03   🔍   Debugging takes many hours because the source of the error is hard to find

- Similar issues arise every time a change is needed

- Unit testing prevents these issues by
  – Reducing the number of potential errors
  – Quickly identifying the precise location of the error

VERSION CONTROL TOOLS SUCH AS GIT IMPROVE ABILITY TO COLLABORATE

Collaborating with large teams in spreadsheet environments creates issues…
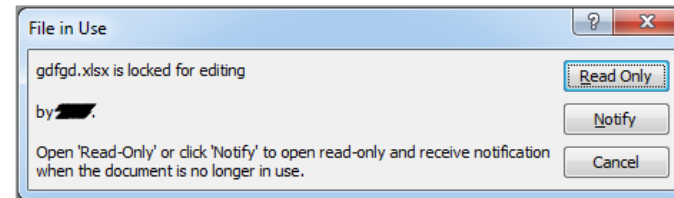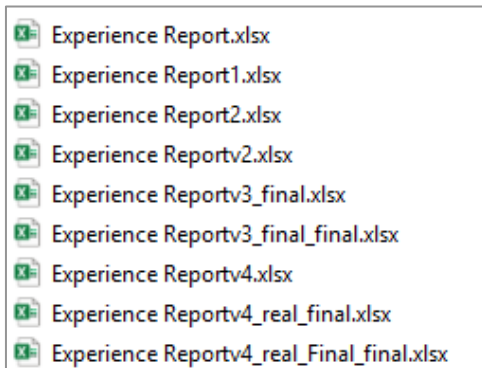
Git, a version control tool for software developers, solves these issues

Git

- Allows multiple contributors to work simultaneously on the same code

- Tracks all updates made by contributors

- Can revert to any prior version

- Identifies conflicts automatically when contributors overwrite the same piece of code
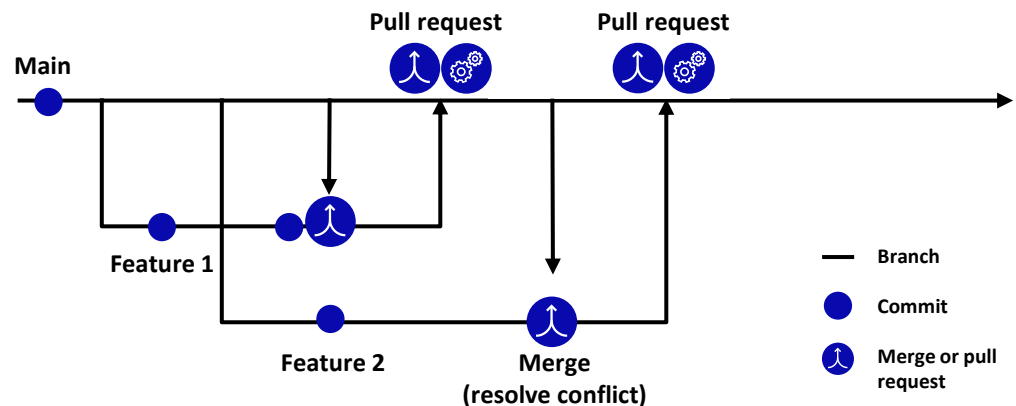
2B

AUTOMATION AND SPEED

OliverWyman
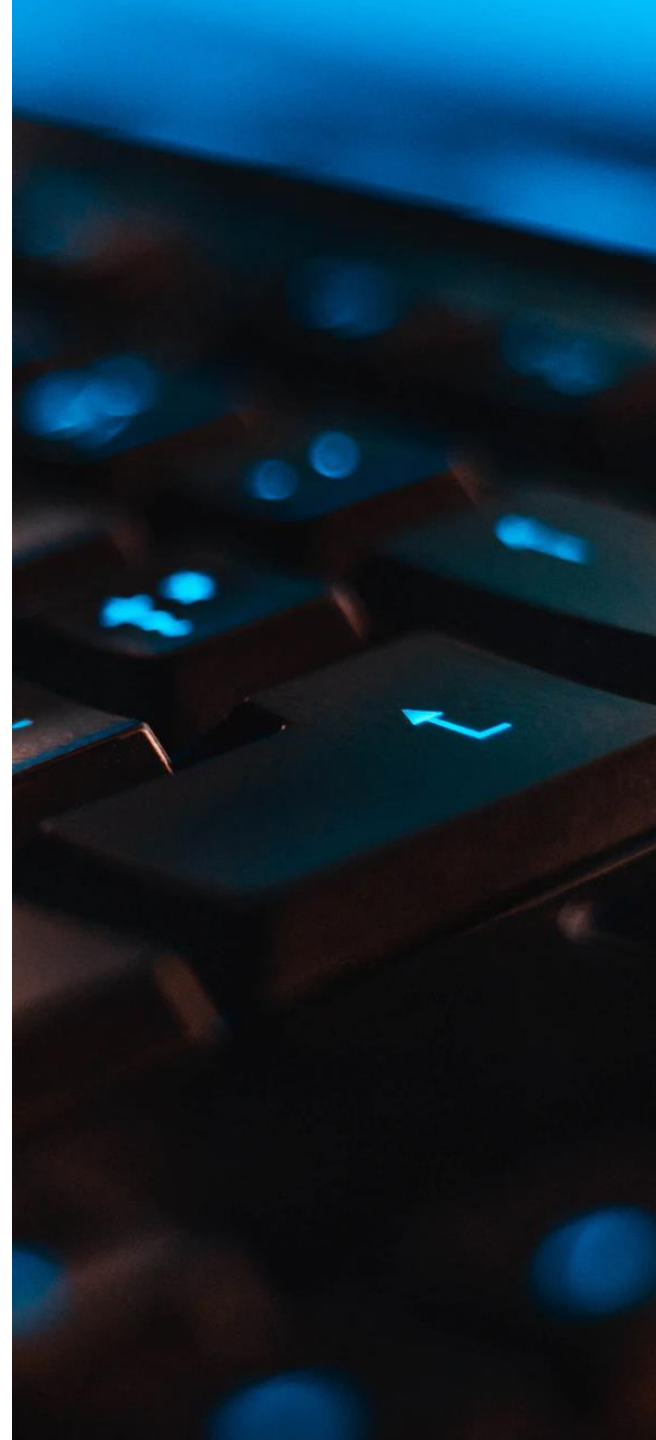
DUPLICATION OF EFFORT IS COMMON IN MOST ACTUARIAL PROCESSES

OliverWyman

Rate Level Indication Process Example – Current Approach

Indication Process Start	Extract policy data	Extract claim data	Validate Data	Enter Data in Indication Template
Analyze Large Losses	Select Premium Trend	Calculate On-Level Premiums	Enter Rate Changes in Indication Template	Pull Rate Change Info
Adjust for Large Losses	Pull Catastrophe Loss Info	Analyze Catastrophe Losses	Enter Catastrophe Loadings	Create Triangles
Review Analysis	Calculate Preliminary Indications	Select Loss Trends	Develop Losses	Analyze and Select LDFs

Task Requires Manual Input   Task is Fully Automated

SOFTWARE DEVELOPMENT PRACTICES LEAD TO BETTER STRUCTURING OF DATA AND PROCESSES, WHICH REDUCES DUPLICATION OF EFFORT

Rate Level Indication Process Example


```
run_indications()
    load_indication_data()
        extract_policy_data()
        extract_claim_data()
        validate_data()
        populate_template()
    update_indication_premium()
        pull_rate_change_history()
        calculate_on_level_premium()
        select_premium_trend()
    update_indication_losses()
        adjust_large_losses()
            pull_large_loss_data()
            select_large_loss_cap()
            select_large_loss_loading()
        adjust_catastrophe_losses()
            pull_catastrophe_data()
            select_catastrophe_loading()
        develop_losses()
            create_triangles()
            select_ldfs()
            apply_ldfs()
        trend_losses()
            select_loss_trends()
            apply_loss_trends()
    calculate_indications()
```

Indication Process Start

Enter Parameters and Initiate Run

Run Indication Tasks

Review Analysis

Machine learning can be used to automate assumption selection

Manual steps are significantly reduced and focused on analysis rather than repetitive tasks

Task Requires Manual Input   Task is Fully Automated
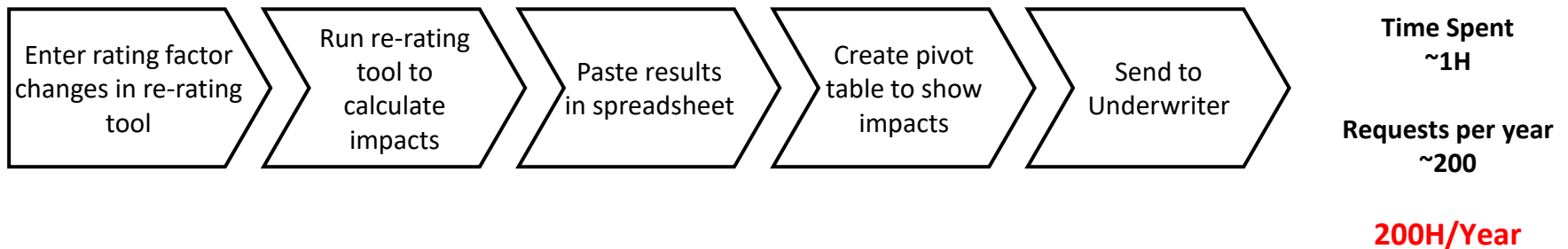
Software Development Approach

- Indication tasks are broken down into software modules using Object Oriented Programming principles

- Unit testing ensures each module is working as intended

- Versions are tracked in Git, allowing large teams to collaborate on the same code base

- Modules are modified and improved individually while still maintaining overall process functionality

- Process data is stored in structured datasets and can be retrieved easily
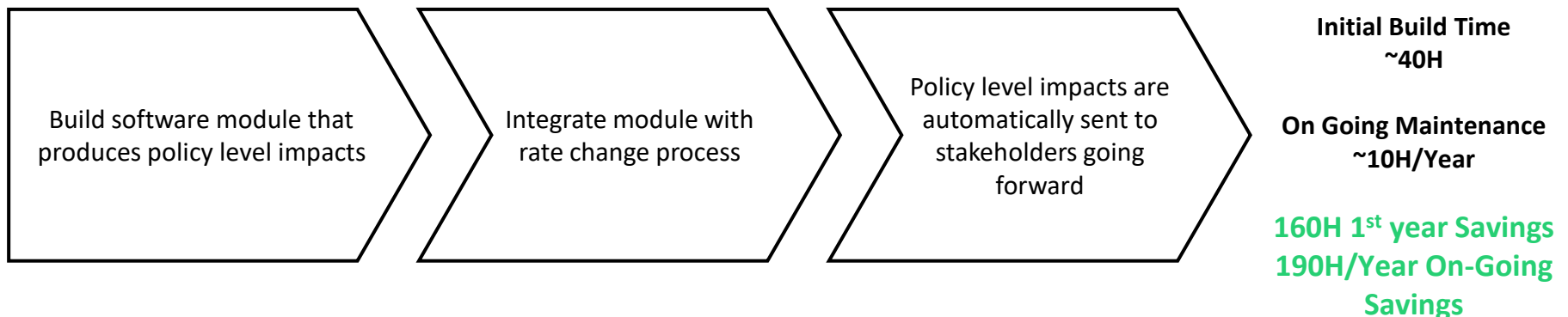
IN A SOFTWARE DEVELOPMENT ENVIRONMENT, AUTOMATION IS BUILT IN THE WAY ACTUARIES WORK

OliverWyman

Illustrative Example: Underwriter asks actuary to pull policy level impact of rating factor changes in Ohio
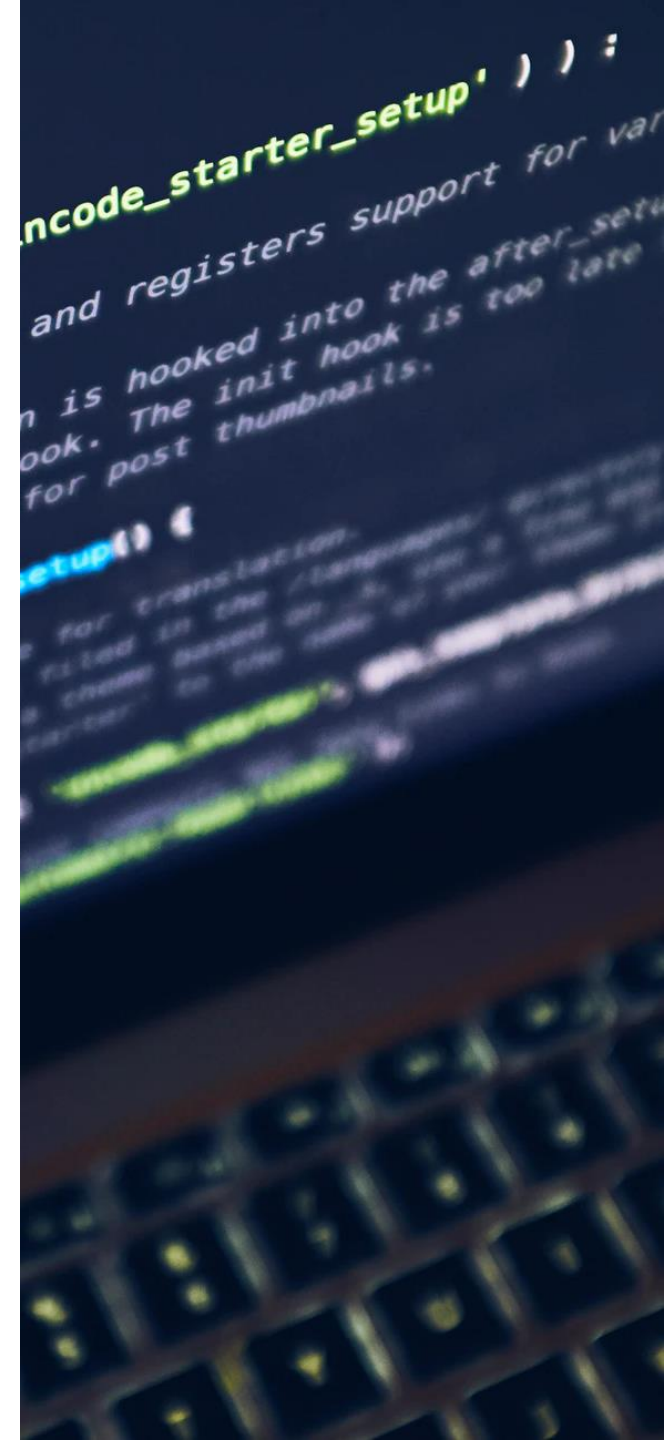
Typical Approach

Enter rating factor changes in re-rating tool	Run re-rating tool to calculate impacts	Paste results in spreadsheet	Create pivot table to show impacts	Send to Underwriter

Time Spent ~1H

Requests per year ~200

200H/Year

Software Development Approach

Build software module that produces policy level impacts	Integrate module with rate change process	Policy level impacts are automatically sent to stakeholders going forward

Initial Build Time ~40H

On Going Maintenance ~10H/Year

160H 1st year Savings
190H/Year On-Going Savings

2C

CUTTING EDGE TECHNOLOGY

OliverWyman

CUTTING EDGE TOOLS ARE BUILT USING SOFTWARE, FOR SOFTWARE DEVELOPERS

Programming languages like Python have ecosystems of sophisticated tools that can be easily leveraged by software developers

Python Ecosystem of Tools

Machine Learning

Computation Optimization

User Interface and Visualization

CUTTING EDGE TOOLS ARE EASY TO INTEGRATE IN A SOFTWARE ENVIRONMENT

- The latest software tools are **not designed to be used in spreadsheets**
- However, they are **easy to use in a software environment**

01

```
C:\Users>pip install scikit-learn
```

02

```python
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(predictors_train, target_train)
knn.predict(predictors_test)
```
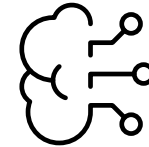
3

RECAP

OliverWyman

WHY SHOULD ACTUARIAL TEAMS ADOPT A TECH MINDSET?

1

Speed to Market and Product Iteration
- Actuaries need to adopt these approaches to facilitate speed to market
- Integration of actuarial and software/product teams drives rapid product iteration
- Advocates for actuaries' presence in the digital space
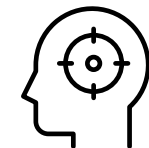
2

Process Improvement
- Gain efficiency and economies of scale
- Automate and expedite processes
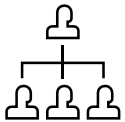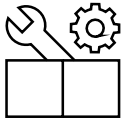- Get easy access to cutting edge tools

3

Innovation
- Young actuaries want to learn and apply these transferable skills
- Use of spreadsheets for non-trivial tasks is becoming obsolete
- Helps with talent acquisition and retention

HOW TO GET THERE?

01   Free up capacity

02   Create teams with combined actuarial and software development expertise

03   Target processes with significant improvement opportunities

04   Rework targeted processes using software development approaches

QUESTIONS

QUALIFICATIONS, ASSUMPTIONS, AND LIMITING CONDITIONS

This report is for the exclusive use of the Oliver Wyman client named herein. This report is not intended for general circulation or publication, nor is it to be reproduced, quoted, or distributed for any purpose without the prior written permission of Oliver Wyman. There are no third-party beneficiaries with respect to this report, and Oliver Wyman does not accept any liability to any third party.

Information furnished by others, upon which all or portions of this report are based, is believed to be reliable but has not been independently verified, unless otherwise expressly indicated. Public information and industry and statistical data are from sources we deem to be reliable; however, we make no representation as to the accuracy or completeness of such information. The findings contained in this report may contain predictions based on current data and historical trends. Any such predictions are subject to inherent risks and uncertainties. Oliver Wyman accepts no responsibility for actual results or future events.

The opinions expressed in this report are valid only for the purpose stated herein and as of the date of this report. No obligation is assumed to revise this report to reflect changes, events, or conditions, which occur subsequent to the date hereof.

All decisions in connection with the implementation or use of advice or recommendations contained in this report are the sole responsibility of the client. This report does not represent investment advice nor does it provide an opinion regarding the fairness of any transaction to any and all parties. In addition, this report does not represent legal, medical, accounting, safety, or other specialized advice. For any such advice, Oliver Wyman recommends seeking and obtaining advice from a qualified professional.