

Delivering high-quality research software that lasts

Emma Hogan

RSE SW, Friday, February 20, 2026

10:05 - 10:25

Many thanks to Peter Hill (University of York) and my team (the Model Inputs And Outputs / MIAO 🐱 team) for their feedback on this talk 🙏



Instructions

Go to

www.menti.com

Enter the code

3462 8257



Or use QR code

Overview

Go to
www.menti.com

Enter the code

3462 8257



Or use QR code

-  Why maintainable code matters
-  The value of iterative, customer-focused development practices
-  Real examples from industry
-  Is there a genuine need for learning resources in this space?

Background



**SOCIETY OF RESEARCH
SOFTWARE ENGINEERING**



Research Software Maintenance Fund

Software is pervasive in modern research, used in every discipline and underpins modelling and simulation; data management, analysis and visualisation; and experiment and laboratory control. Yet it can be difficult, even for widely used software, to find resources for maintenance, refactoring and community development, because the focus of much research funding is on novelty.



? Ask the audience!

Instructions

Go to
www.menti.com

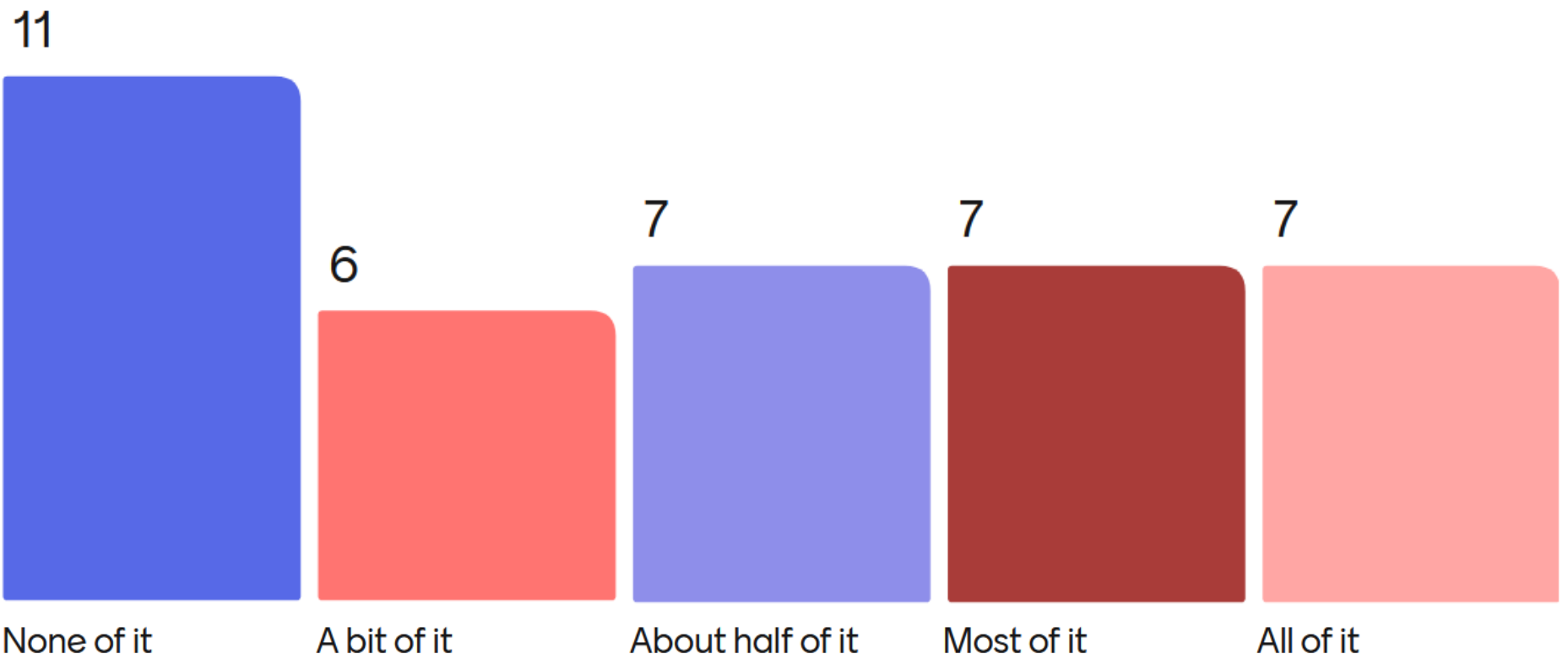
Enter the code

3462 8257

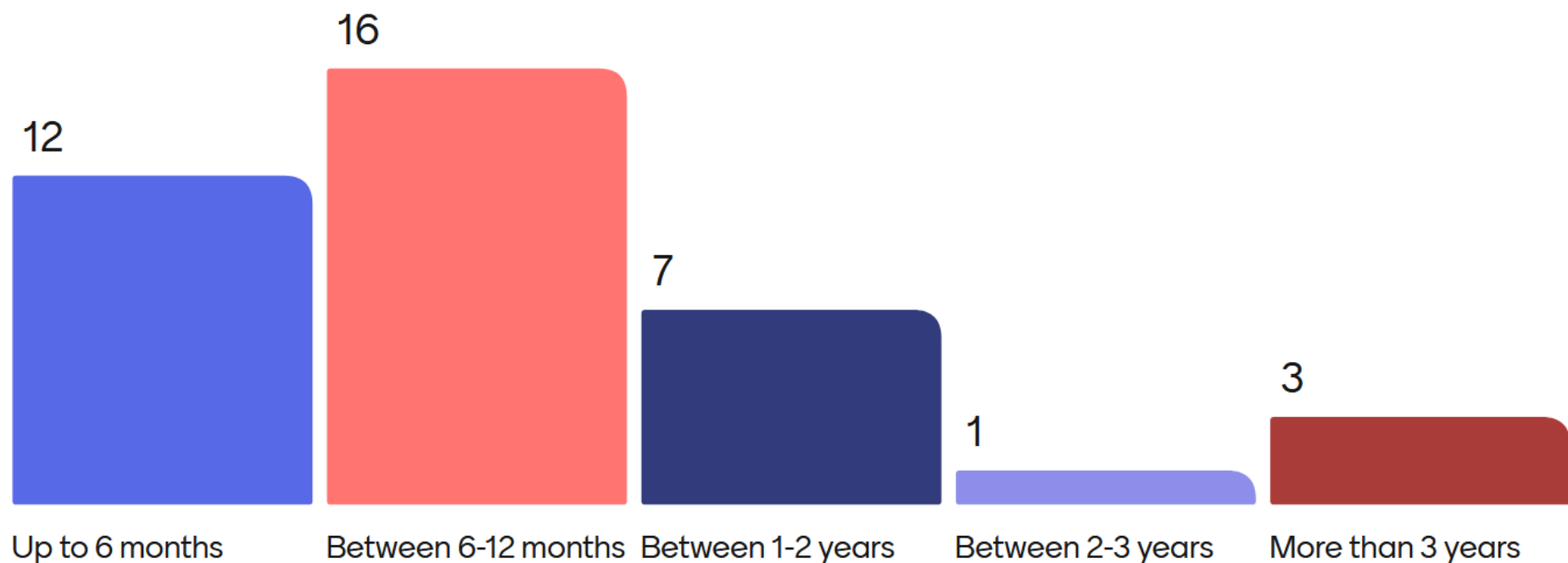


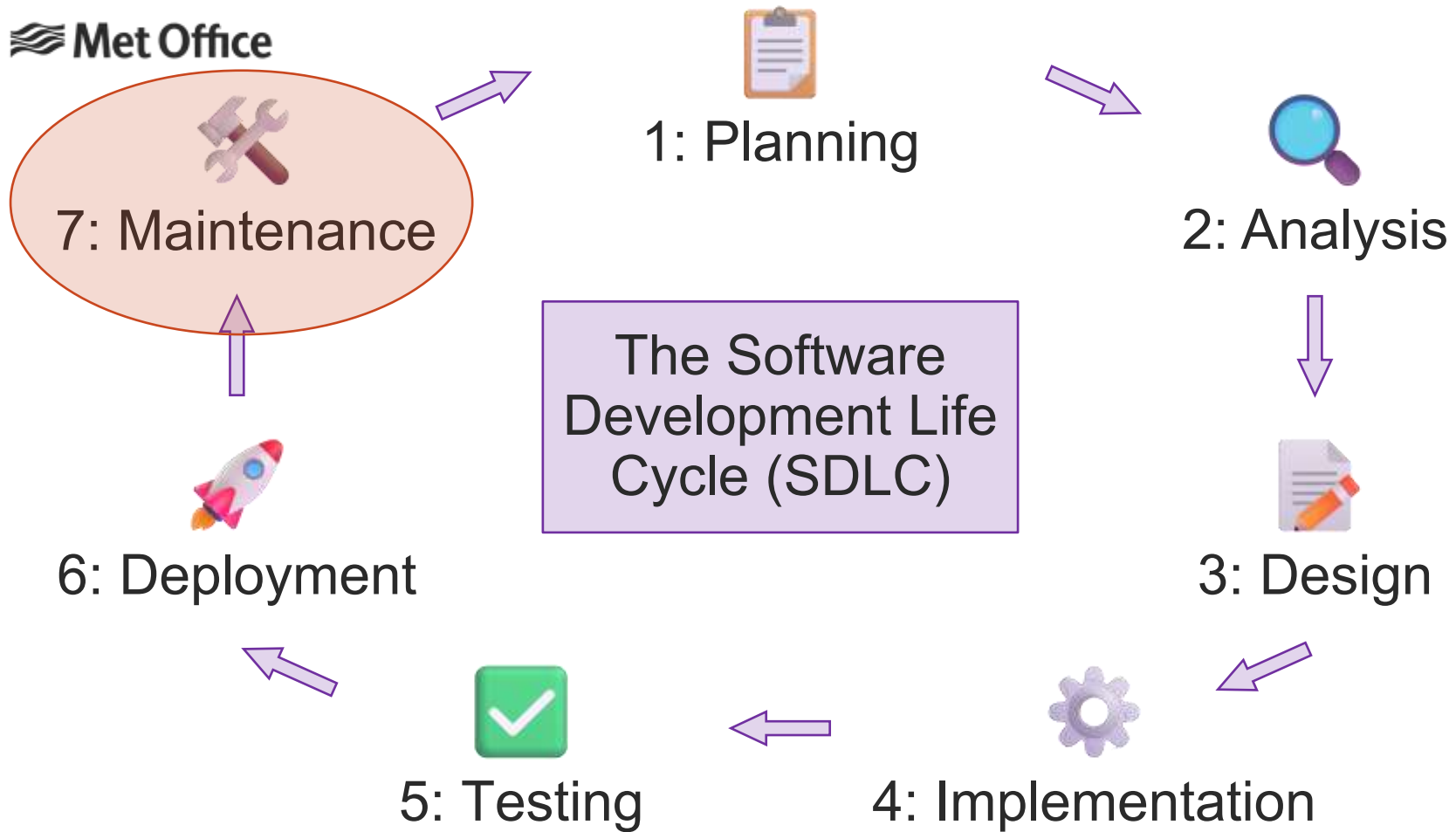
Or use QR code

How much of the research software you develop do you deliver to a customer?



Thinking about your recent projects, how long do you typically develop research software before delivering it to the customer?





Research Software Development Principles

Helping your team



FAIR

Reusable by as many users as possible



Secure

Respect data privacy and assume attacks



Maintainable

Easy to adapt and to correct faults

Helping your peers



Reproducible

Enable trust in research



Recognition

Reward all roles and develop the next generation



Inclusive

Accessible and supportive of a broad community



Responsible

Build to reduce impact on our environment



Open & Global

Transcend national and discipline boundaries



Humanist

Unbiased, ethical and in support of humanity

Helping the world

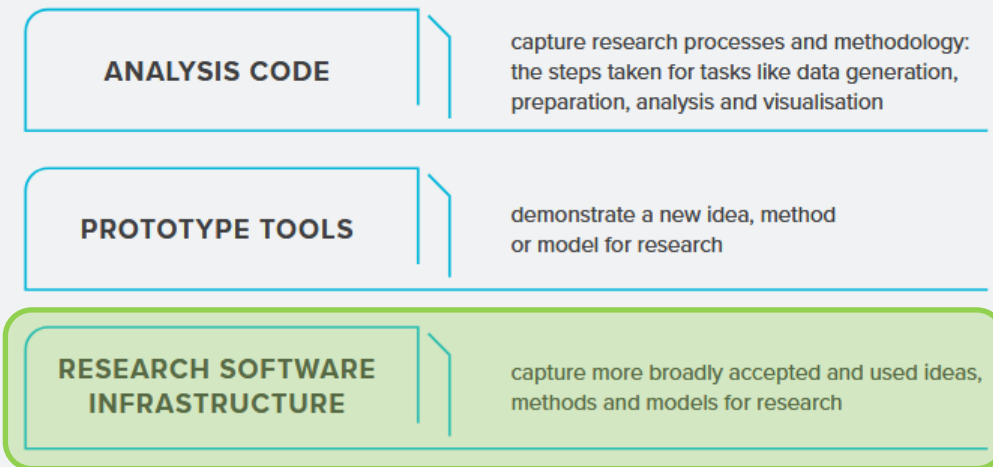
Three-tier model

- Published by the Australian Research Data Commons (ARDC)
- Good practice in the production of these types of research software mean quite different things

Long-lived
research
software

WHAT IS RESEARCH SOFTWARE?

Software can come in many forms, including scripts, code, notebooks, computational workflows, libraries, modules, frameworks, utilities and applications. Here we focus instead on what they get built for.



There is a relationship between these 3 kinds of software – software at the top of the table builds on the existence and stability of the software lower down. Conversely the existence and availability of software at the top builds the case for the value of the software built upon.

? Ask the audience!

Instructions

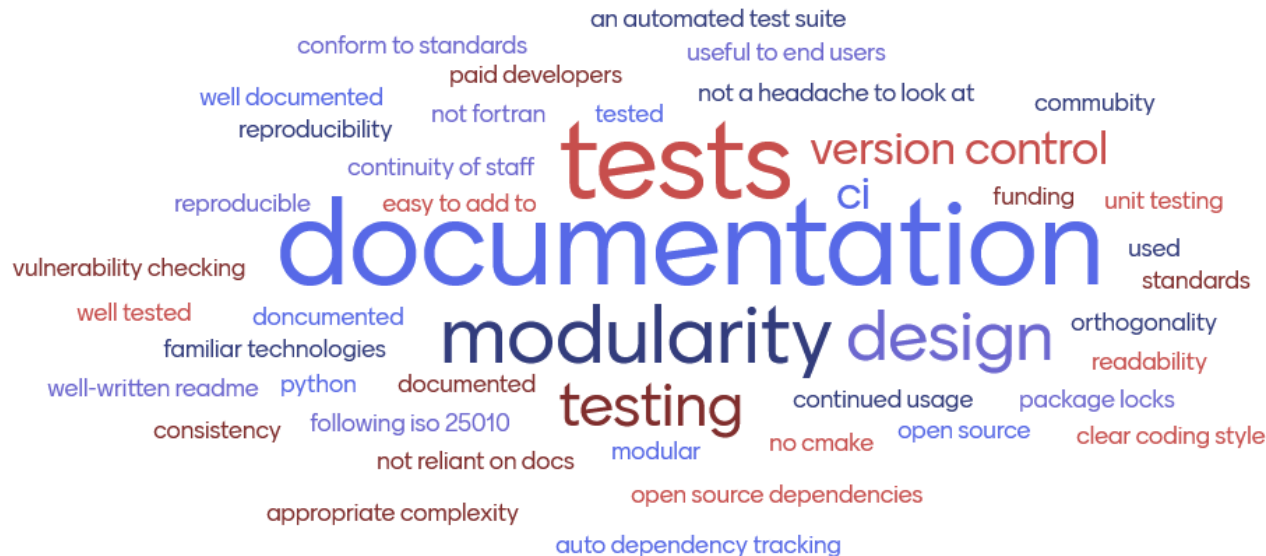
Go to
www.menti.com

Enter the code
3462 8257



Or use QR code

What characteristics do you think make research software maintainable?





The definition of maintainability

ISO 25000

ISO 25000 STANDARDS ▾

NEWS

PRODUCT EVALUATION ▾

CERTIFIED PRODUCTS

Maintainability

This characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.

ISO 25000

ISO 25000 STANDARDS ▾

NEWS

PRODUCT EVALUATION ▾

CERTIFIED PRODUCTS

SOFTWARE PRODUCT QUALITY

FUNCTIONAL SUITABILITY	PERFORMANCE EFFICIENCY	COMPATIBILITY	INTERACTION CAPABILITY	RELIABILITY	SECURITY	MAINTAINABILITY	FLEXIBILITY	SAFETY
---------------------------	---------------------------	---------------	---------------------------	-------------	----------	-----------------	-------------	--------



ESMValTool

Earth System Model Evaluation Tool

[HOME](#)

[ESMVALTOOL ▾](#)

[COMMUNITY ▾](#)

[DOWNLOAD](#)

[DOCUMENTATION ↗](#)

[TUTORIAL ↗](#)



ESMValTool

A community diagnostic and performance metrics tool for
evaluation and analysis of Earth system models

ESMValTool: <https://esmvaltool.org/>

10 years ESMValTool!

Posted on December 18, 2025

Version 1.0 of the ESMValTool was released on 18 December 2015!

Watch the 10 years [ESMValTool anniversary video](#) on YouTube!





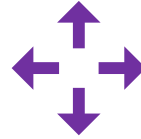
Characteristics of maintainable code



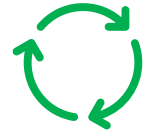
Documentation



Consistent



Extendable



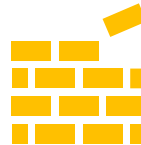
Reusable



Testable



Simple



Modular



Readable

? Ask the audience!

Instructions

Go to

www.menti.com

Enter the code

3462 8257



Or use QR code

Which characteristic of maintainable code do you prioritise?



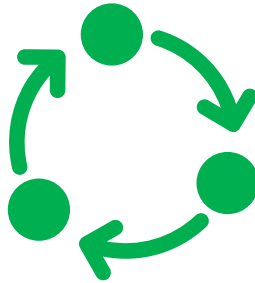
Overview

-  Why maintainable code matters
-  The value of iterative, customer-focused development practices
-  Real examples from industry
-  Is there a genuine need for learning resources in this space?

What is Agile?



A set of values and principles



The ability to respond to change



Focuses on the people doing the work

The following 12 Principles are based on the [Agile Manifesto](https://agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/).

1 Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.	7 Working software is the primary measure of progress.
2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9 Continuous attention to technical excellence and good design enhances agility.
4 Business people and developers must work together daily throughout the project.	10 Simplicity—the art of maximizing the amount of work not done—is essential.
5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11 The best architectures, requirements, and designs emerge from self-organizing teams.
6 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The following 12 Principles are based on the [Agile Manifesto](#).

1

Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.

7

Working software is the primary measure of progress.

2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's

8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a

1

Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.

5

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

11

The best architectures, requirements, and designs emerge from self-organizing teams.

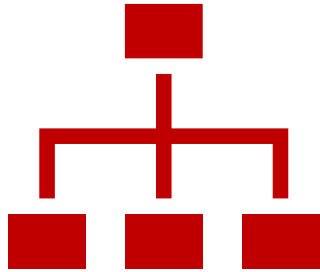
6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

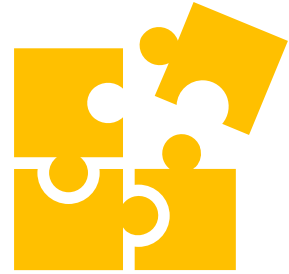
What is Scrum?



A lightweight
framework



An iterative, incremental
approach



A Team, Events,
Artifacts & Commitments

? Ask the audience!

Instructions

Go to
www.menti.com

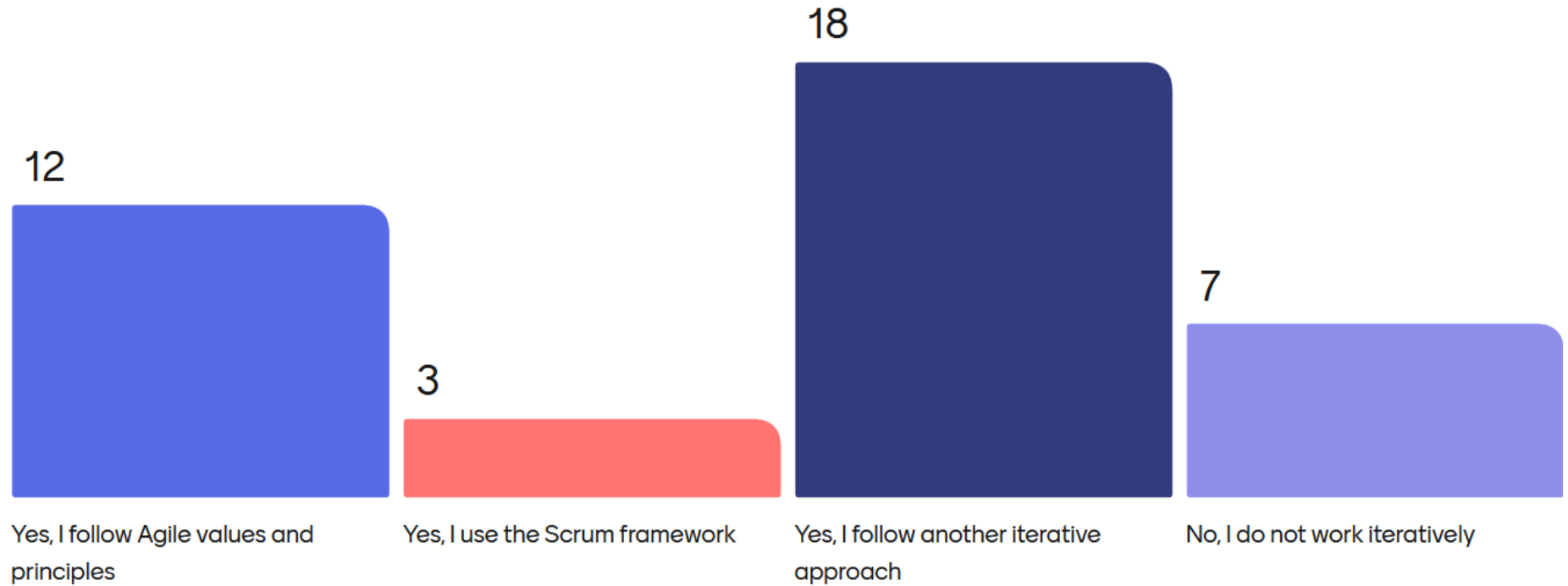
Enter the code

3462 8257



Or use QR code

Do you work iteratively when developing research software?



[Home](#) | [News and blogs hub](#) | [Digital and Professional Skills for Industry: Understanding Employers and Graduates Experiences](#)



Bookmark this page

Author(s)



[B. Thomas](#) | Policy Researcher

Posted on 21 March 2025

Estimated read time: 2 min

Share on blog/article:



Digital and Professional Skills for Industry: Understanding Employers and Graduates Experiences



Project management identified as a skill gap

Project management was identified as a skill gap by several employers who noted that graduates often lacked experience with software lifecycle management processes such as Agile. One employer with insight into higher education pointed out that whilst Agile was taught on computer science degrees, it was rarely done effectively. Agile techniques like daily standups were developed for industry settings, requiring a level of immersion that, it was noted, was difficult to replicate in an academic environment when students only worked on a project once a week. It is 'very difficult to teach that stuff' in a way that mirrors real-world application (E2, work coach, MNC4). A recent PhD graduate described his struggle with the demands of his current job that required planning, prioritising tasks, and creating timelines, a skill set he admitted he had not really needed in his PhD and was still developing (G5).

Skills you would like to gain or improve



Agile Methods for RSEs

A formula for scrum-driven software
development in research?

Ann Gledson, Adrian Harwood & Scott Archer-Nicholls
Research IT, IT Services, The University of Manchester

RSE Con 24 | Newcastle | September 2024



Agile Methods for RSEs: A formula for scrum-driven software development in research?



Society of Research Software Engineering
492 subscribers

Subscribe



YouTube: RSECon24: https://youtu.be/qKNhwTsozT4?si=7S_GB4c6zFRD1d8w

Customer engagement & high-quality software



Product Owner



Sprint Review / demo



Definition of Done

Maintainability

Overview

-  Why maintainable code matters
-  The value of iterative, customer-focused development practices
-  Real examples from industry
-  Is there a genuine need for learning resources in this space?



Real examples from industry

Project	Scrum Team (Part Time or Full Time)	Scrum Events	Scrum Artifacts	Scrum Commitments
ESMValTool: Climate Model Evaluation Workflow (CMEW)	Product Owner: 1 PT Scrum Master: 1 PT Dev team: 5 / 3 FT	✓ Sprint ✓ Planning ✓ Stand-up ✓ Demo ✓ Retro ✓ Refinement	✓ Product backlog ✓ Sprint backlog ✓ Increment	✓ Product goal ✓ Sprint goal ✓ Definition of Done
Science Git Migration Project: Git and GitHub documentation	Product Owner: 1 PT Scrum Master: 1 PT Dev Team: 8 PT	✓ Sprint ? Planning ✓ Stand-up ? Demo ✓ Retro ✓ Refinement	✓ Product backlog ✓ Sprint backlog ✓ Increment	✓ Product goal ✗ Sprint goal ✓ Definition of Done

? Ask the audience!

Instructions

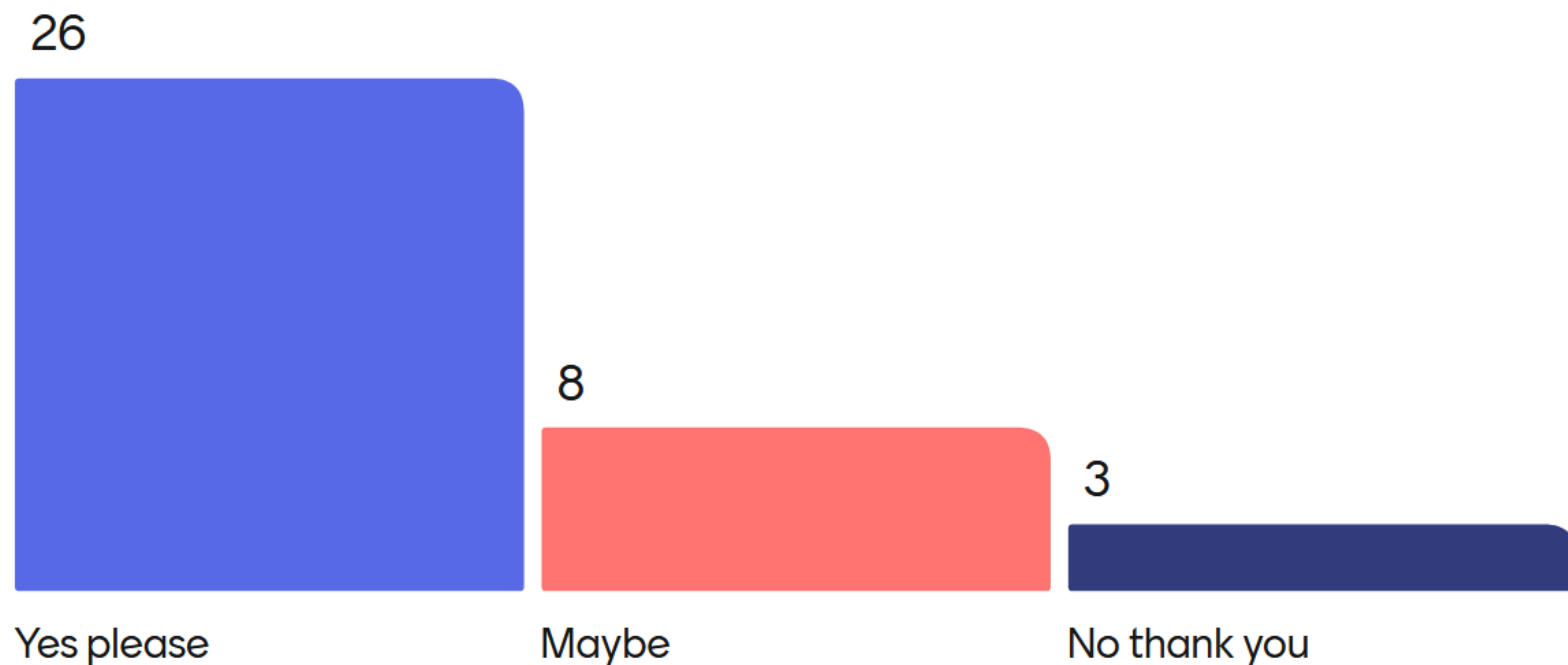
Go to
www.menti.com

Enter the code
3462 8257



Or use QR code

Would you like to learn more about the topics discussed today?



What learning resources (topics, delivery methods, etc.) would be most useful to you when developing research software?

✦ 7 groups found



Written Guides

6 responses

"Guides", "- Written documentation - Practical tuto...



Project Management

5 responses

"Project management tools", "Delivery Methods", "...



Code Examples

5 responses

"Working examples", "Others' examples", "Example...



Team Collaboration

4 responses

"Asynchronous working for international teams.", "...



What learning resources (topics, delivery methods, etc.) would be most useful to you when developing research software?

✦ 7 groups found



Documentation

3 responses

"Documentation for all related software", "Videos", ...



Workshops

3 responses

"Workshops with short practical sessions of simul...



Community Support

2 responses

"Community", "Handling customer engagement"





Written Guides

6 responses



Guides



- Written documentation -
Practical tutorials



A framework guide to follow



Guides on how to help others
engage in Scrum effectively



Online resources we could use for
in-house training.



Best practices (in all facets)





Project Management

5 responses



Project management tools



Delivery Methods



sprint planning techniques



Clearly defined requirements



Identify the main problems around quality research software needs to address and then consider approaches that target these, rather than taking Agile from industry wholesale.





Code Examples

5 responses



Working examples



Others' examples



Examples of extendable vs non-extendable code



Blog posts by other RSEs



"Turing Way"-like





Team Collaboration

4 responses



Asynchronous working for international teams.



How to evaluate team effectiveness.



Multi disciplinary resources to understand what we are working on or the science we are working with



Examples of working with different sized teams, including single RSEs in a research group of non-RSEs





Documentation

3 responses



Documentation for all related
software



Videos



Software reproducibility





Workshops

3 responses



Workshops with short practical sessions of simulated projects



User research methods



cultural change - self paced learning is difficult, the best way to learn these things is from more skilled peers. There's no funding incentive to provide this, however





Community Support

2 responses



Community



Handling customer engagement





Summary

-  Maintainable code matters:
 -  our ability to continuously deliver value to our customers
 -  depends on how effectively and efficiently we can modify our code
-  Iterative, customer-focused development practices are valuable:
 -  engaging with customers throughout the development process
 -  leads to robust code that customers can maintain independently
-  Is there a genuine need for learning resources in this space?