

A Focus Area Maturity Assessment Model for Research Software Projects

Deekshitha
PhD student

Netherlands eScience center, Utrecht University and University of Leiden

netherlands



Universiteit
Utrecht



Universiteit
Leiden

United in Vision, Driven by Passion: Meet Our Team



Deekshitha
Phd Candidate
Netherlands eScience center,
Utrecht University,
University of Leiden



Dr. Slinger Jansen
Associate Professor
Utrecht University



Prof. dr. Rob van Nieuwpoort
Full Professor
Leiden University



Dr. Jason Maassen
Technology Lead, Computing
Netherlands eScience center



Dr. Rena Bakhshi
Programme Manager
Netherlands eScience center



Dr. Carlos Martinez Ortiz
Scientific Community Manager
Netherlands eScience center



Dr. Simak Farshidi
Lecturer
Utrecht University



Dr. Antti Knutas
Associate Professor
LUT University, Finland

Introducing the FAIR Principles for research software

[Michelle Barker](#) , [Neil P. Chue Hong](#), [Daniel S. Katz](#), [Anna-Lena Lamprecht](#), [Carlos Martinez Ortiz](#), [Fotis Psomopoulos](#), [Jennifer Harrow](#), [Leyla Jael Castro](#), [Morane Gruenpeter](#), [Paula Andrea Martinez](#) & [Tom Honeyman](#)

[Scientific Data](#) **9**, Article number: 622 (2022) | [Cite this article](#)

17k Accesses | **52** Citations | **229** Altmetric | [Metrics](#)

Abstract



Good Practices in Research Software Development (CodeRefinery)

[About](#) [What we do](#) [Resources](#)

Best Practices for Sharing Research Software

[Home](#) / [Tools & Analytics](#) / [Best Practices for Sharing Research Software](#)

Frequently Asked Questions

Introduction

The National Institutes of Health (NIH) promotes broad dissemination of research products including NIH-funded and/or NIH-developed research software. These FAQs offer best practices for sharing research software and source code, developed under research grants in any stage of development, in a free and open format. Releasing the software source code in an “open” manner means that you permit users to use, modify, and/or redistribute the code. The [FAIR principles](#) (findable, accessible, interoperable, and reusable) also provide



< [PeerJ Computer Science](#)

Nine best practices for research software registries and repositories



[Share](#)

[Research article](#) [Computer Education](#) [Databases](#) [Digital Libraries](#)

[Daniel Garijo](#) ¹, [Hervé Ménager](#)², [Lorraine Hwang](#)³, [Ana Trisovic](#)⁴, [Michael Hucka](#)⁵, [Thomas Morrell](#)⁵, [Alice Allen](#)⁶, [Task Force on Best Practices for Software Registries](#)⁷,

E.g.: 3. Provide guidance to software contributors

Best... Practices...

1. Documentation
2. Public Repository
3. Citation
4. License
5. etc...



Many practices

Many stakeholders

How to evaluate RS project management?

Who are responsible for Research Software?

- . Research software engineers
- . **RS project managers** <- **Our target audience**
- . Researchers / Primary investigators
- . Documentation authors
- . **Funders / Policy makers** <- **Our secondary target audience**
- . Translators
- . Users
-



Maturity of a Research Software



Focus Area Maturity Model (FAMM)



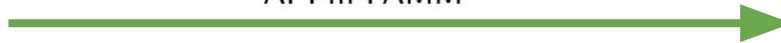
Focus Area Maturity Model (FAMM)

- ❖ **Focus area maturity model (FAMM)** is one type of **maturity model**, these are based on the concept of a number of **focus areas** that have to be developed to achieve **maturity** in a **functional domain** [1].
- ❖ The **FAMM** is used to measure how well an organization is advancing through different maturity levels in a specific area of work.



Focus Area Maturity Model (FAMM)

API-m-FAMM



Maturity Levels

Components		Maturity Levels													
		0	1	2	3	4	5	6	7	8	9	10			
1.	Lifecycle Management														
1.1.	Version Management			Implement Evolutionary API Strategy			Implement Multiple API Versions Strategy	Implement API Deprecation Protocol	Check Backwards Compatibility						
1.2.	Decoupling API & Application		Decouple API & Software Versioning			Decouple Internal & External Data Model	Decouple Internal & External Data Format	Decouple Internal & External Transport Protocol							
1.3.	Update Notification			Distribute Changelogs	Distribute Versioning Notification through Channel(s)		Extend API with Versioning Information					Announce API Versioning Roadmap			
2.	Security														
2.1.	Authentication		Implement Basic Authentication			Implement Authentication Protocol			Implement Single Sign-on						
2.2.	Authorization			Implement Access Control		Implement Token Management		Implement Standardized Authorization Protocol	Implement Authorization Scopes						
2.3.	Threat Detection & Protection		Implement Allow & Deny IP Address Lists	Implement Injection Threat Protection Policies			Implement DoS Protection		Implement Security Breach Protocol			Conduct Security Review	Implement Zero Trust Network Access		
2.4.	Encryption		Implement Transport Layer Encryption		Implement Certificate Management										

Focus Areas

Practices

RSPM-MM: A Focus Area Maturity Model for Research Software Projects Management



Designing the RSPM Maturity Model

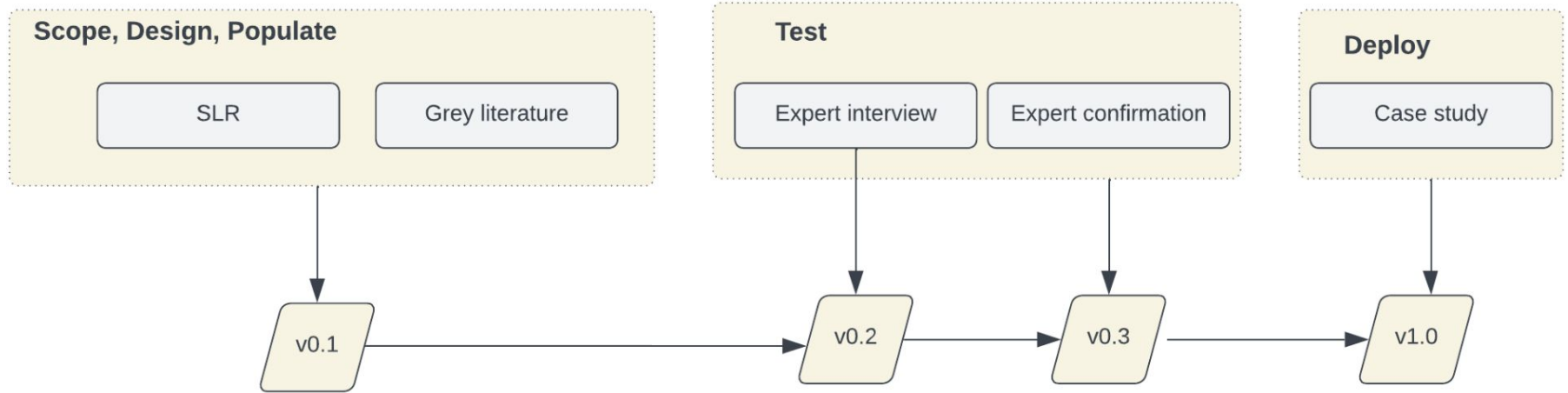


Fig 3: Design Phases of RSPM-MM

Research Software Project(s) Management- Maturity Model

Focus Area	Capability	0	1	2	3	4	5	6	7
Software project management	Requirements	Requirements are managed explicitly	Authors act on feedback	Roadmap is communicated					
	Code quality	Code reviews are executed	Security reviews are conducted	There are executable tests	Tests are executed in a public workflow	Uses common methods for security evaluation	Project stability is measured continuously	Crash reporting is used	
	Communication and Collaboration	A public repository system is used	Has a watering hole for community members						
	Issue management	An infrastructure is available for posting issues	Issues are actively worked on	There is a support team in place					
Research software management	Usage costs	Provides documentation on the cost of running the application	Considers energy consumption						
	Impact measurement	Performs infrequent measurement of impact	Performs continuous measurement	Concerned with the privacy aspects of its use	Concerned with the ethical consequences of its use				
	Sustainability	There is a software management plan in place	Obvious source of temporary funding	Viable pathways for project sustainability	Maintained by a research software center	Obvious source of continuous funding			
	Visibility	Code is citable	Project meta-data is indexed (Zenodo?)	Published in a research software directory	Continuously promoted (e.g., at conferences, Twitter, etc.)	Source code is indexed (SearchSECO, SHG)	Acknowledged by a research software center	Publicly adopted by industrial partners	
Research software community management	Partnerships	Informally acknowledges partners and funding agencies	Has an advanced partnership model						
	Community	Onboards researchers as part of the community	Imposes some community norms	Has a code of conduct					
	Developers	Documents how to join the team	Developer names are publicly available	Developers can get access to training and skill development					
	Licensing	A license is carefully selected	License policy is regularly evaluated						
Research software adoptability	Ease of use	A simple how to use is provided	A statement of purpose is provided	Online tutorials are provided					
	Documentation	A read me file is provided	Common example usage is provided	A documentation infrastructure is provided					
	Technology	Uses common non-exotic or legacy technology	Technology is easy to use in scientific workflow						
	Reproducibility	Instructions exist on how to put into research workflow	Instructions exist on how to make part of a replication package	Is part of standardized workflows	Is part of a replication package				
	Education	The project is part of an educational program	Generic educational materials have been developed						
	Deployability	The project has an SBOM	Can be deployed on a wide range of technology	Provided with standard deployment tools (docker images, etc.)	Provided as a running service				

First Release v0.1

Focus Areas - 4

Capabilities - 18

Practices - 61

Maturity Levels - 0-7

Fig 2: RSPM-MM v0.1

The four Pillars of RSPM-MM



**Software Project
Management**

RSPM-MM



**Research Software
Management**



**Research Software
Community
Management**

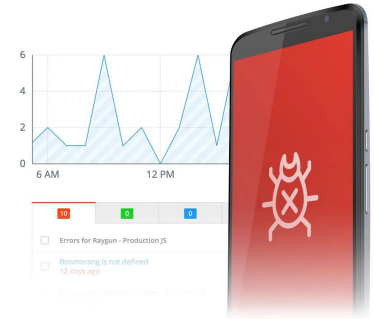
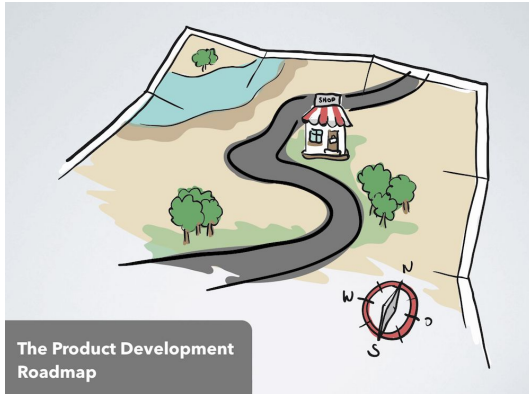


**Research
Software
Adoptability**



Focus Area - Software Project Management

Software Project Management involves the effective planning, organization, oversight of the development, and maintenance of software.



Roadmap for the research software

Communication channel

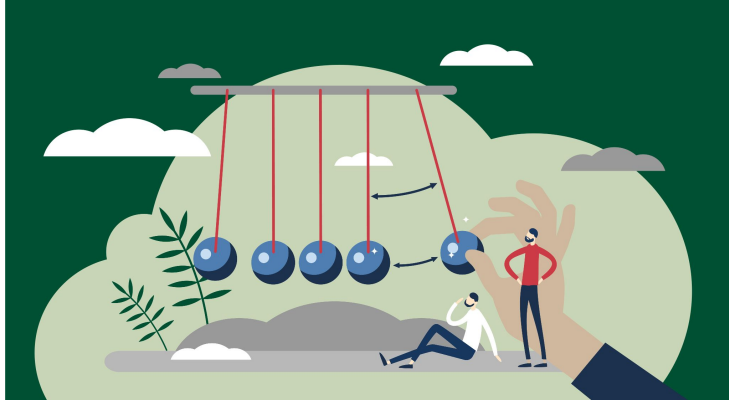
Crash Report

<https://www.product-bakery.com/blog/product-development-roadmap>

<https://softwareengineeringdaily.com/2018/07/26/crash-reporting-improving-the-customer-dev-experience/>

https://commons.wikimedia.org/wiki/File:Slack_Technologies_Logo.svg

Focus Area - Research Software Management



Impact measurement



Funding

 Readme

 AGPL-3.0 license

 Cite this repository ▼

 Activity


<https://www.wifor.com/en/impact-definition/>
<https://www.cartoonstock.com/directory/f/funding.asp>

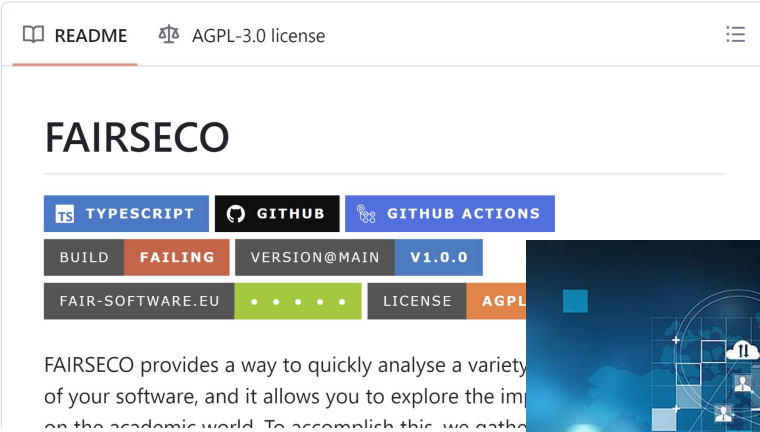
Focus Area - Research Software Community Management

Focuses on building a collaborative and supportive community for the research software



Focus Area - Research Software Adoptability

How easily and effectively research software can be adopted and utilized by users.



Readme



Technology



Designing the RSPM Maturity Model - Test

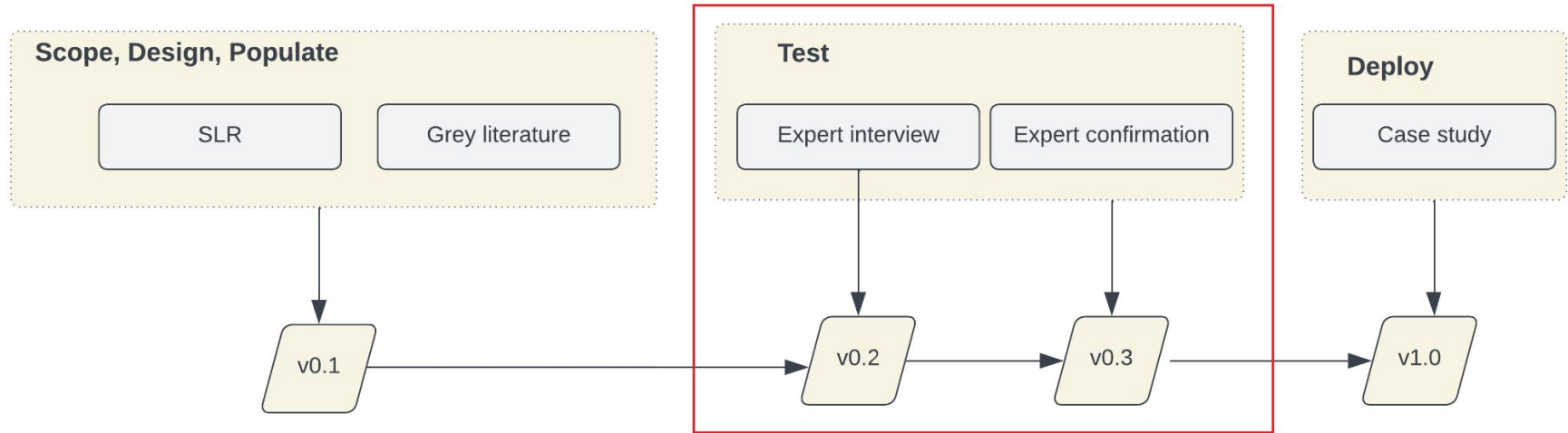
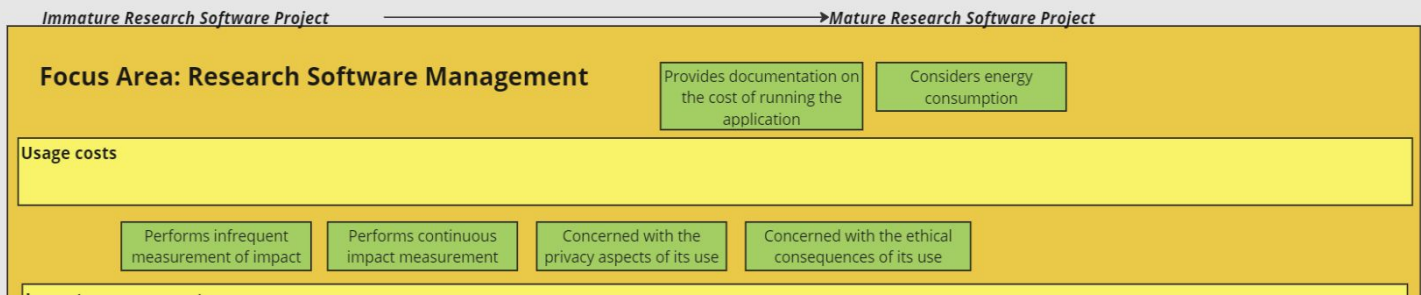
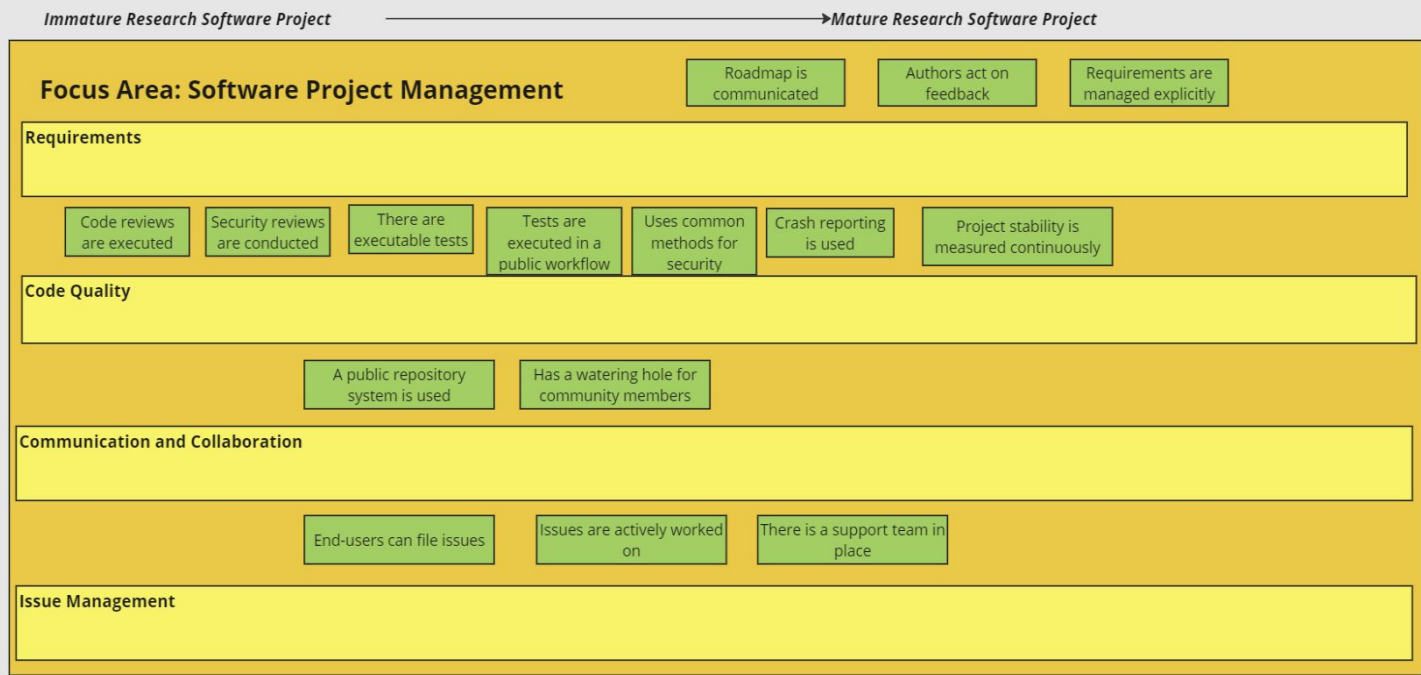


Fig 3: Design Phases of RSPM-MM

Research Software Project - Focus Area Maturity Model



RSPM-MM
v0.1a

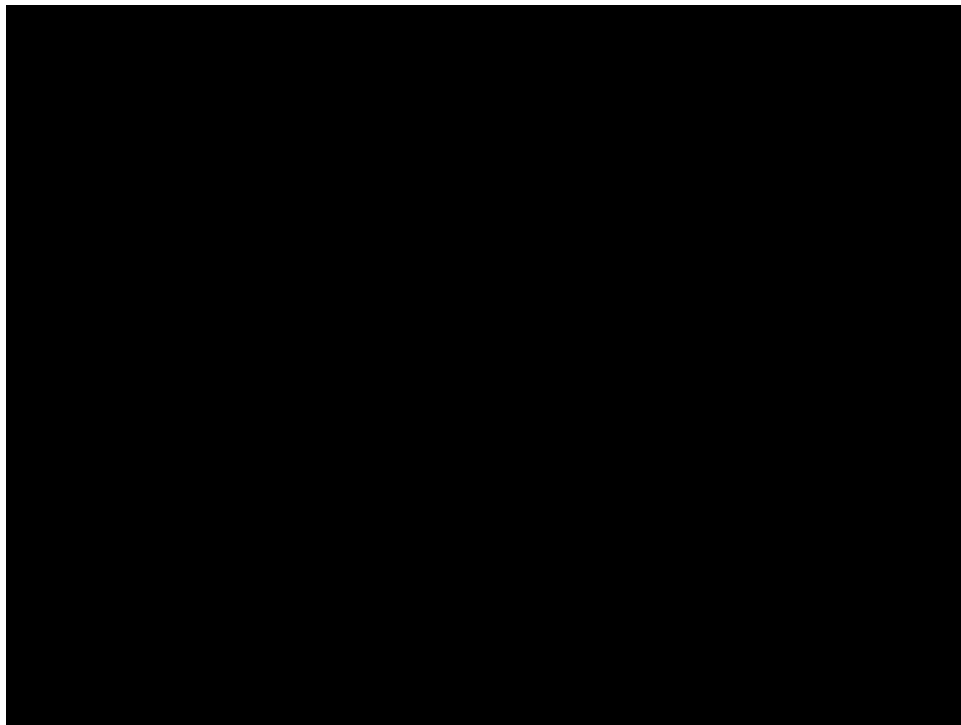
Focus Area- 4

Capability- 18

Practices- 61

Maturity levels -
0-7

How experts ranked practices in a miro dashboard?



Video clip shows how one of the expert rank the practices



RSPM-MM v0.2

Focus Area- 4

Capability- 19

Practices- 79

Maturity Levels- 0-9

Focus Area	Capability	0	1	2	3	4	5	6	7	8	9	10
Software project management	Requirements	Requirements are managed explicitly	Authors act on feedback	Roadmap is communicated								
	Code quality	Code reviews are executed	There are executable tests	Tests are executed in a public workflow	Project stability is measured continuously	Crash reporting is used	Code coverage targets are defined	Code standard	Workflows can be coordinated over different machines	Continuous integration		
	Security	An industry standard for security is followed (e.g., OWASP SAMM, BSIMM, etc.)	Security reviews are conducted									
	Communication and Collaboration	A public repository system is used	Public communication platform (e.g., email list, Slack, etc.)	News letter	Community website							
	Issue management	End users file issues in an issue tracker	Issues are actively worked on	There is a support team in place	Front page chat support	Release management						
Research Software Management	Impact measurement	Performs infrequent measurement of impact	Performs continuous impact measurement	Concerned with the privacy aspects of its use	Concerned with the ethical consequences of its use	The project defines a clear audience	The project evaluates whether the audience's goals are met	The project regularly explores new audiences				
	Sustainability	There is a software management plan in place	Obvious source of temporary funding	Viable pathways for project sustainability	Maintained by a research software center	Obvious source of continuous funding	End of life policy defined					
	Visibility	Code is citable	Project meta-data is indexed (Zenodo?)	Published in a research software directory	Continuously promoted (e.g., at conferences, Twitter, etc.)	Source code is indexed (SearchSECO, SHG)	Acknowledged by a research software center	Publicly adopted by industrial partners				
	Usage costs	Provides documentation on the cost of running the application	Considers energy consumption									
Research software community management	Partnerships	Informally acknowledges partners and funding agencies	Has an advanced partnership model									
	Community	Onboards researchers as part of the community	Imposes some community norms	Has a code of conduct	Organizing community events							
	Developers	Documents how to join the team	Developer names are publicly available	Developers can get access to training and skill development	Response time on pull request							
	Licensing	A license is carefully selected	License policy is regularly evaluated	License is selected								
Research software adoptability	Ease of use	A simple how to use is provided	A statement of purpose is provided	Online tutorials are provided	Low barrier to community/ users docs							
	Documentation	A read me file is provided	Common example usage is provided	Documentation as repository								
	Technology	Uses common non-exotic or legacy technology	Technology is easy to use in scientific workflow	Technology relevance regularly evaluated								
	Reproducibility	Instructions exist on how to put into research workflow	Instructions exist on how to make part of a replication package	Is part of standardized workflows	Is part of a replication package	Easy to integrate into replication package						
	Education	The project is part of an educational program	Generic educational materials have been developed	Training events in person								
	Deployability	The project has an SBOM	Can be deployed on a wide range of technology	Provided with standard deployment tools (docker images, etc.)								

How to evaluate the RSPM-MM?

- **Operational feasibility:** Do you think this model helps organizations to improve their research software production practices?
- **Ease of use:** How easily can organizations adapt this model to improve their practices? Do you find this model easy to understand?
- **Effectiveness:** Do you think this model is effective in achieving the goal?
- **Target Audience:** Who do you think are the main users and consumers of the RSP-FAMM?



Designing the RSPM Maturity Model - Deploy

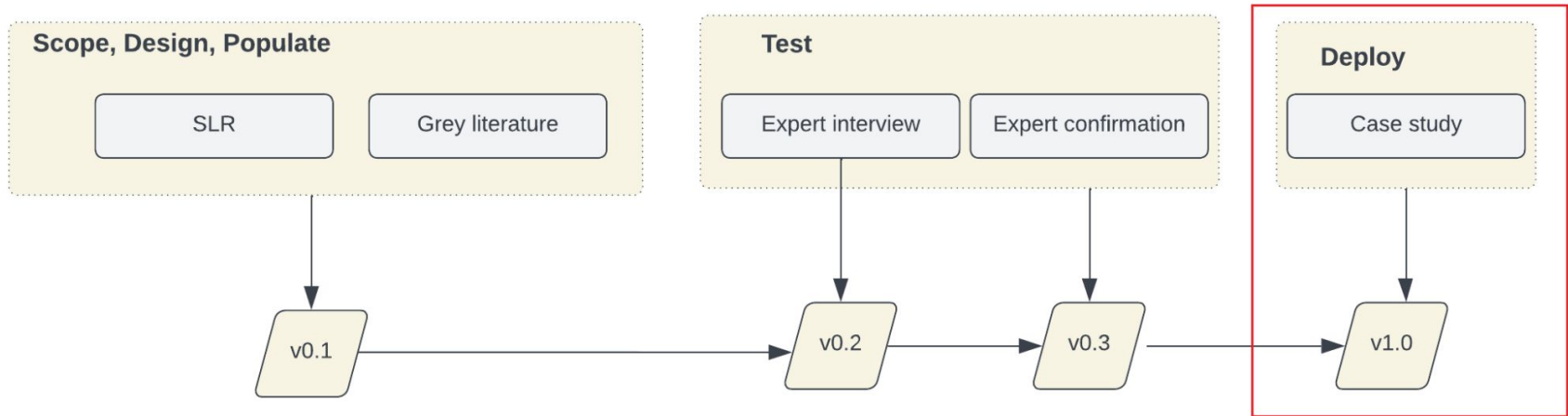


Fig 3: Design Phases of RSPM-MM

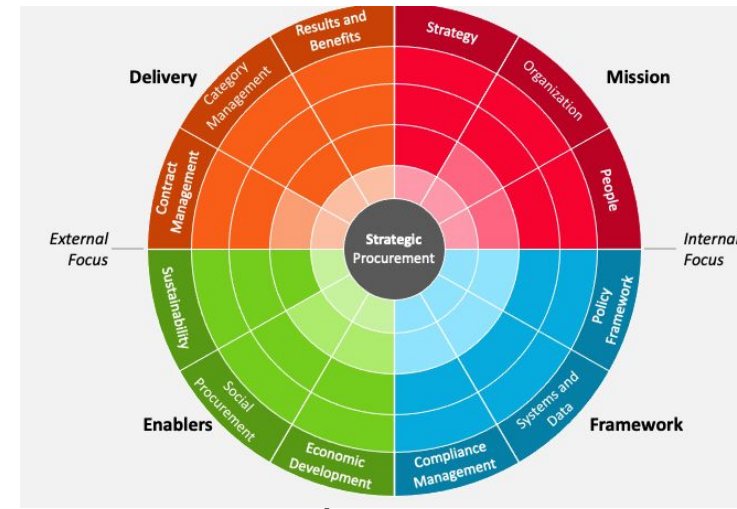
Evaluation table

Focus Area	Capability	0	1	2	3	4	5	6	7
Software project management	Requirements	yes	yes	no					
	Code quality	yes	no	no	no	no	no	no	
	Communication and Collaboration	yes	yes						
	Issue management	yes	yes	no					
Research software management	Usage costs	no	no						
	Impact measurement	no	no	no	no				
	Sustainability	no	no	no	yes	no			
	Visibility	yes	yes	yes	yes	yes	yes	yes	
Research software community management	Partnerships	yes	no						
	Community	yes	yes	yes					
	Developers	yes	yes	no					
	Licensing	yes	no						
Research software adoptability	Ease of use	yes	yes	yes					
	Documentation	yes	yes	yes					
	Technology	yes	yes						
	Reproducibility	yes	yes	yes	yes				
	Education	no	no						
	Deployability	no	yes	yes	no				

Evaluation table for GGIR

Future Work

- We have evaluated over 50 projects with our model and will soon contact their RSEs to validate our evaluation
- Do you want us to evaluate your project? Send us your link to d.deekshitha@uu.nl
- Beautify for marketing purposes?



Focus Area	Capability	0	1	2	3	4	5	6	7
Software project management	Requirements	Requirements are managed externally	Authors act on feedback	Roadmap is communicated					
	Code quality	Code reviews are executed	Security reviews are conducted	There are executable tests	Tests are executed in a public workflow	Tests are executed in a public workflow	Tests are executed in a public workflow	Tests are executed in a public workflow	Tests are executed in a public workflow
	Communication and Collaboration	A public repository system is used	A voting system is available for community members						
	Issue management	Issues are actively tracked		There is a support team in place					
Research software management	Usage costs	Provides documentation on the cost of running the application	Considers energy consumption						
	Impact measurement	Performs intelligent measurement of impact	Performs continuous impact measurement	Concerned with the privacy aspects of its use	Concerned with the ethical consequences of its use	Concerned with the ethical consequences of its use	Concerned with the ethical consequences of its use	Concerned with the ethical consequences of its use	Concerned with the ethical consequences of its use
	Sustainability	There is a software management plan in place	Obtains source of temporary funding	Obtains source of temporary funding	Obtains source of temporary funding	Obtains source of temporary funding	Obtains source of temporary funding	Obtains source of temporary funding	Obtains source of temporary funding
	Visibility	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)	Project materials are visible (Zenodo)
Research software community management	Partnerships	Internally acknowledges partners and funding agencies	Has an advanced partnership model						
	Community	Obtains researchers on part of the community	Engages some community members						
	Developers	Documents how to join the team	Developer names are publicly available	There is a code of conduct	Developers can get access to training and self-development	Developers can get access to training and self-development	Developers can get access to training and self-development	Developers can get access to training and self-development	Developers can get access to training and self-development
	Leadership	A license is publicly available	License policy is regularly updated						
Technology	State of use	A simple how-to is provided	A statement of purpose is provided	Online tutorials are provided					
	Documentation	A need for the file is provided	Common example usage is provided	A documentation infrastructure is provided					

Conclusion

- We deliver the Research Software Projects Management Maturity Model, for project managers, funders, and policy makers.
- The model allows for *incremental improvement, benchmarking, standardized reporting*, and overall better research software project management.
- We'd love feedback on the model, want to think along? Plan a call with us!
slinger.jansen@uu.nl
- Use the model yourself! Contact us for the latest version!



What are your thoughts
on mature Research
Software?

Does **maturity** have a
strong relationship with
impact?

d.deekshitha@uu.nl

Thank you !

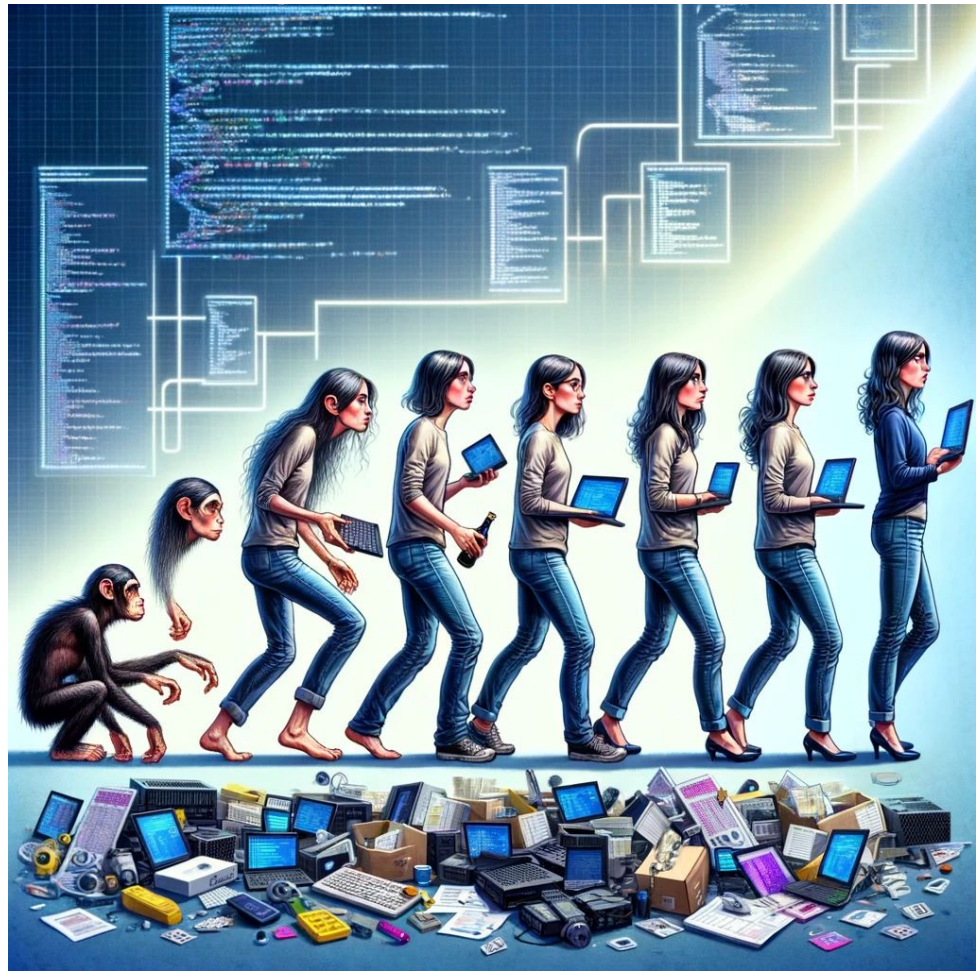


Image created by OpenAI's DALL-E.