Software engineering practices in IAM development

* Required

Introduction

In the context of DIAMOND's Task 6.4 (Development pipelines for open IAMs), we have created a **framework of good practices**, collected from the literature on IAM development and scientific software engineering in general. This framework aims to help WP3 modellers (and, ideally, the whole IAM community of practice) to develop models that can be open, well-tested, and well-documented.

The framework can be seen in the figure below and includes **18 practices**, grouped in five categories: model development, documentation, evaluation, versioning and collaboration, and deployment. It is noted that practices on model use and model inter-comparisons (e.g., harmonisation) will be covered in a future deliverable.

In the following pages, you will read a short definition for each of these practices; you can then provide your feedback on whether you have used any of these practices and whether you think they are relevant for the development and uptake of your model.

Thank you for your participation!

Model development	Documentation	Evaluation	Versioning and collaboration	Deployment
Requirements Engineering	Conceptual documentation	Code testing	Version control	Dependency management
Practices for data processing	Code documentation	Model validation	Issue tracking	System requirements
Programming paradigms, patterns, and standards	Data documentation	Continuous integration	Open-source development	Continuous deployment
Development lifecycle models	Tutorials and case studies			
Practices for refactoring and adding new features				

Your details

1

What is your name? *

2

Which organisation are you representing? *

3

Which model are you working on in the context of DIAMOND? *

- [ ] OMNIA
- [ ] CLEWS-EU
- [ ] GCAM-Europe
- [ ] NEMESIS-World
- [ ] OPEN-PROM
- [ ] GEMINI-E3 EU
- [ ] ENGAGE
- [ ] MAgPIE
- [ ] OpenSCM & METEOR
- [ ] openTEPES
- [ ] Other

4

Do you have anyone with software engineering background in your developing team? *

- ◯ Yes
- ◯ No
- ◯ Other

Requirements Engineering

Requirements engineering includes a variety of structured processes to elicit and manage stakeholder needs for the development of software. Typical methods that are employed for the elicitation include interviews, workshops, and surveys.

Category: Model Development practices

5

How did you define the requirements for the model development that took place until now in DIAMOND? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Proposal call and Grant Agreement of DIAMOND
- [ ] Input from stakeholders through the engagement process in WP2
- [ ] Ideas from other modelling teams within DIAMOND
- [ ] Ideas from modelling teams beyond DIAMOND (e.g., GCAM community of practice)
- [ ] Ideas from the literature
- [ ] Other

6

In your view, how relevant are requirements engineering practices to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Practices for data processing

Data processing includes various steps to make data usable such as data collection, cleaning, and transformation. There are several practices for improving data quality, including the FAIR data principles, automating data processing through scripts, and using version control to monitor changes in the processing scripts and in the data themselves.

Category: Model Development practices

7

Where do you store the input data of your model? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Offline system, e.g., your PC or an external hard drive
- [ ] Online data sharing system, e.g., OneDrive
- [ ] On the same repository as your code, e.g., on GitHub
- [ ] Zenodo
- [ ] Other

8

Do you use any of these data processing practices? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Keep a backup of both raw and processed input data.
- [ ] Use a version control system for monitoring changes in the data.
- [ ] Process the data through scripts that are included in a version control system (e.g., GitHub).
- [ ] Use data quality assurance system such as the pedigree matrix approach.
- [ ] Other

9

In your view, how relevant are data processing practices to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Programming paradigms, standards, patterns

Programming paradigms are high-level approaches to structure a computer program (e.g., object-oriented programming); design patterns are efficient ways to solve typical problems in software design (e.g., using Prototypes to clone objects); and coding standards involve coding conventions and layouts to improve code quality (e.g., PEP 8, the official style guide for writing Python code, see: https://peps.python.org/pep-0008/).

Category: Model Development practices

10

Did you use any programming paradigms, design patterns, or coding standards in the development of your model? If yes, which ones? *

e.g., object-oriented programming, PEP 8 style guide for Python, etc.

11

How relevant are programming paradigms, design patterns, or coding standards to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Development lifecycle models

In software engineering practice, there are various methods (or so-called "models") to manage the lifecycle of software development from project conception to delivery. The most typical example is the waterfall model where, after the initial requirements elicitation from the stakeholders, the development proceeds without major interactions with the stakeholders or the users until the final delivery of software. Other lifecycle models emphasise the need for more frequent user feedback such as the iterative model. A common variation of this model is the so-called agile development which is predominantly user-driven.

Category: Model Development practices

12

Did you explicitly use any specific development lifecycle model? *

e.g., waterfall mode, iterative model, agile development

13

How relevant are established lifecycle models (e.g., iterative or agile development) to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Practices for refactoring and adding new features

Refactoring is the process of re-writing existing code to improve it, e.g., in terms of efficiency or clarity. Many good practices exist such as writing regression tests and using a Test-Driven Development approach (i.e., write tests in parallel with the development of new code).

Category: Model Development practices

14

Have you explicitly used any established methodology for adding new features to or refactoring your model? *

e.g., Test-Driven Development, regression testing

15

How relevant are practices for refactoring and adding new features to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Conceptual documentation

In climate-economy model development, this documentation typically includes a description of the purpose of the model, main components (algorithms, input data, and main outputs), key assumptions, uncertainties, pertinent features (technologies or policies modelled), as well as the temporal, spatial, and sectoral resolution of the model. In the case of DIAMOND, if there is already documentation for the base version of a model (e.g., GCAM), the conceptual documentation of the new model can just include a link to the original documentation and a table of changes introduced by the new model.

Category: Documentation practices

16

Do you have any further suggestions on what to include in the conceptual documentation of our models apart from the aspects listed above? *

17

How relevant are practices for conceptual documentation to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Code documentation

Code documentation often takes the form of a short description at the beginning of a function, summarising the purpose of the function as well as its inputs and outputs (e.g., “docstrings” in Python). In-code comments can be used to automatically generate documentation, ensuring that the code and its documentation stay in sync.

Category: Documentation practices

18

How did you document your model code till now? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Comments within the code
- [ ] README files
- [ ] Online documentation platform or wiki page (e.g., Read the Docs)
- [ ] Word document
- [ ] Other

19

How relevant are practices for code documentation to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Data documentation

Documentation of all data inputs and outputs, ideally through metadata standards such as Dublin Core (https://www.dublin-core.org/specifications/dublin-core/) or at least JSON or YAML files.

Category: Documentation practices

20

How did you document your input data till now? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Metadata standards such as Dublin Core
- [ ] TXT, JSON, or YAML files
- [ ] Comments on the dataset (e.g., on the top of CSV files)
- [ ] External README files
- [ ] Other

21

How relevant are practices for data documentation to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Tutorials and case studies

Resources that can help users to actually run the model, e.g., installation guides and manuals, QuickStart tutorials, hands-on examples of common applications, or case studies that are showcasing full model use.

Category: Documentation practices

22

Do you plan to develop a tutorial as part of the first version release (July 2025)? *

- ◯ Yes
- ◯ No, it will developed for the final release of the model (end of DIAMOND project)
- ◯ No current plans, it will be created after the duration of DIAMOND
- ◯ Other

23

How relevant are practices for creating tutorials and case studies to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Code testing

Common practices include unit testing (testing individual functions), regression testing (testing whether existing code runs after adding a new feature), integration testing (end-to-end testing of the whole model), and acceptability testing (testing the model in a real-world context, e.g., using common data inputs).

Category: Evaluation practices

24

Did you use any of these types of code testing? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Unit testing (testing individual functions)
- [ ] Regression testing (testing whether existing code runs normally after adding a new feature)
- [ ] Integration testing (end-to-end testing of the whole model)
- [ ] Acceptability testing (test the model with real-world data)
- [ ] Other

25

How relevant are practices for code testing to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Model validation

Evaluate how well the model can match its conceptual description and real-world observations. Multiple methods exist for IAMs such as the IPCC AR6 vetting process, historical simulations, comparing with the results of other models, and analysing the sensitivity of different model inputs.

Category: Evaluation practices

26

Which of the following methods did you use to evaluate your model results? *

Most methods are from Wilson et al., 2021. Select all that apply and feel free to add more details in the "Other" field.

- [ ] Historical simulations
- [ ] Near-term observations
- [ ] Stylised facts
- [ ] Model inter-comparisons
- [ ] Sensitivity analysis
- [ ] IPCC AR6 Vetting process
- [ ] Other

27

How relevant are practices for model evaluation to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Continuous integration

Ensure that every change to the model code can be integrated seamlessly to the existing code repository without breaking any previous functionalities. This process is mainly based on automated code testing that ensures that no error is introduced when code is changed.

Category: Evaluation practices

28

Do you have any automated testing system in place? *

e.g., GitHub workflows

29

How relevant are practices for continuous integration to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Version control

Using a version control system such as Git is one of the most common practices for collaborative model development. A version control system ensures that all changes by different developers can be correctly integrated to the model code and also provides a way to restore software code to a previous version if any problem occurs.

Category: Versioning and collaboration practices

30

Do you use any version control system? *

- ◯ Yes, we are using a manual or offline system (e.g., only Git)
- ◯ Yes, and we upload the code in an online platform such as GitHub or GitLab
- ◯ Not yet
- ◯ Other

31

How relevant are practices for version control to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Issue tracking

Issue tracking systems can further support collaborative development by documenting to-dos in the model along with features that can be implemented in the future. GitHub, GitLab and other similar platforms provide solid issue tracking.

Category: Versioning and collaboration practices

32

Do you use any issue tracking system? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Manual issue tracking (e.g., on paper, emails, etc.)
- [ ] Offline tracking system (e.g., to-do apps on Windows or Mac)
- [ ] GitHub, GitLab or other similar platforms for code sharing
- [ ] Trello, Asana, and other online to-do apps
- [ ] Other

33

How relevant are practices for issue tracking to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Open source development

A collection of practices that ensure that the developed software can be freely distributed while minimising discrimination and availability restrictions (https://opensource.org/osd). Prominent practices include using open-source licenses and publicly sharing model code through platforms such as GitHub.

Category: Versioning and collaboration practices

34

Which open-source license will you use for your model? *

e.g., MIT, Educational Community License, etc.

35

Are your input data open access? If yes, what is their license? If not, are there any alternative open access datasets that can be used? *

36

How relevant are open-source practices to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Dependency management

Manage third-party dependencies such as libraries and databases. Good practices include keeping dependencies explicit (e.g., requirements.txt file), using an application for dependency management (e.g., Poetry in Python), and, in the case of IAMs, clearly mentioning model interconnections in the documentation.

Category: Deployment practices

37

What dependencies does your model currently have? If available, you can also provide a link to a requirements.txt file or a page that you document the dependencies. *

38

How are you documenting your dependencies? *

Select all that apply and feel free to add more details in the "Other" field.

- [ ] Automatically, through an application (e.g., Poetry in Python)
- [ ] Requirements.txt file
- [ ] README or Word-like document
- [ ] Other

39

How relevant are practices for dependency management to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ○ Very relevant
- ○ Relevant
- ○ Moderately relevant
- ○ Slightly relevant
- ○ Not relevant

System requirements management

System requirements such as computational power, memory, or disk space need to be clearly documented and, like the third-party requirements, can be managed by specialised applications (e.g., Docker).

Category: Deployment practices

40

What are the (estimated) system requirements for running your model? *

e.g., operating system requirements (Windows, MacOS, Linux), computational power, memory, disk space, etc.

41

How relevant are practices for system requirements management to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Continuous deployment

Continuous deployment typically involves a series of automation processes that aim to reduce the time between development and release of a new software version. In the case of IAMs, the production system can take the form of a Docker image that can be downloaded and run by any user (https://www.docker.com).

Category: Deployment practices

42

Have you ever used Docker in the context of DIAMOND or in general? *

43

How relevant are continuous deployment practices to the development and uptake of your model? *

If you are unsure, please provide a best guess.

- ◯ Very relevant
- ◯ Relevant
- ◯ Moderately relevant
- ◯ Slightly relevant
- ◯ Not relevant

Feedback and next steps

44

What other practices are we missing from this framework? *

Model development	Documentation	Evaluation	Versioning and collaboration	Deployment
Requirements Engineering	Conceptual documentation	Code testing	Version control	Dependency management
Practices for data processing	Code documentation	Model validation	Issue tracking	System requirements
Programming paradigms, patterns, and standards	Data documentation	Continuous integration	Open-source development	Continuous deployment
Development lifecycle models	Tutorials and case studies			
Practices for refactoring and adding new features				

45

Is there any other issue that you would need help with in terms of preparing and releasing the first model version in July 2025 and, afterward, the final model version? *