# What?

## What is it all about?

1. The artifact shall have a name for reference
2. The artifact shall indicate its context of development (e.g., domain, problem, project)
3. The artifact shall indicate the main functionalities supported (e.g., language support, code generation, model analysis)
4. The artifact shall indicate the relation with its respective paper

## What does it have?

1. The artifact shall provide an overall description of its directory structure and content
2. The artifact shall include everything required for replications (i.e., complete)
3. The artifact shall include no more assets than necessary for replications (i.e., concise)
4. The artifact shall include a preprint of its associated article

## What concepts and technologies underpin the artifact?

1. The artifact shall indicate the theories that underpin its creation (e.g., formalisms, semantics)
2. The artifact shall indicate the modeling languages used to develop it (e.g., UML, xtUML, SysML, BPMN)
3. The artifact shall indicate the meta-modeling languages used to develop it (e.g., EMOF, CMOF, ECore/EMF, KM3)
4. The artifact shall indicate the standards and/or specifications used to develop it (e.g., ISO, CMI, XMI, CWM, HUTN, JMI, DD, OCL)
5. The artifact shall indicate the programming/markup language used to develop it
6. The artifact shall indicate the libraries, dependencies, and frameworks used to develop it and their respective versions (e.g., Eclipse release)
7. The artifact shall rely on open, well-maintained, and documented libraries or dependencies

# Why?

## Why it was created?

1. The artifact shall report the motivation for its development
2. The artifact shall report its objective/goal (e.g., verify claims, replicability, reusability, a whole new library/framework)
3. The artifact shall indicate its advantages and/or novelty (e.g., algorithm, language, method)

# Where?

## Where is it hosted?

1. The artifact shall be hosted in an open, public repository (e.g., GitHub, BitBucket, Zenodo, Figshare)
2. The artifact shall be hosted in a repository indexed and findable by search engines (e.g., GitHub, BitBucket, Zenodo, Figshare)
3. The artifact shall be archived for long-term, permanent access (e.g., Zenodo, Figshare)

## Where shall I cite?

1. The artifact shall provide an explicit format for citation (e.g., in a CITATION or README file)
2. The artifact shall provide citation information in a BibTeX format
3. The artifact shall provide an URL for citation
4. The artifact shall provide a DOI for citation

## Where to find related work?

1. The artifact shall give credit to data obtained from other sources (e.g., authors, paper, repository, benchmark)
2. The artifact shall provide references about key concepts (e.g., papers, surveys, wiki, reports)
3. The artifact shall provide references to studies using it (e.g., known uses, integrated with)
4. The artifact shall provide references to related projects
5. The artifact shall provide references using in-code citation (e.g., code header)

# Who?

## Who could use it?

1. The artifact shall be deposited under an explicit open license (e.g., reported in a LICENSE file)
2. Files shall be made available using open/non-proprietary formats (e.g., JSON, open XML schema)
3. The artifact shall indicate a list of potential users (e.g., professionals, researchers, industry sectors)

## Who are the authors?

1. The artifact shall provide a communication channel for interacting with the community (e.g., forum, mailing list, issue tracker, IRC, Slack, Discord)
2. The artifact shall indicate the names of its authors
3. The artifact shall indicate the institution of its authors
4. The artifact shall indicate the contact details of its authors/collaborators (e.g., email address, ResearchGate, Linkedin, website)
5. The artifact shall indicate the ORCID of its authors/collaborators
6. The artifact shall indicate the level of experience of its authors/collaborators (e.g., bio, degree, position)

## Who funded this project?

1. The artifact shall indicate the funding agencies that supported the project (e.g., ERC, NWO, CNPq, DFG, EPSRC, NSF)

# When?

## When did changes happen?

1. Changes to the artifact shall be tracked using version control (e.g., GitHub, GitLab, BitBucket)
2. Changes to the artifact shall be small (e.g., conciseness, cohesion, clear edit)
3. Changes to the artifact shall be explained (e.g., CHANGELOG, commit messages)
4. The artifact shall allow referencing or retrieving specific versions using tags and/or release identifiers

## When do future changes shall happen?

1. The artifact shall provide a timeline for future goals and planned updates (e.g., frequency, next steps, future work plans)
2. The artifact shall be open for change requests and receiving feedback from users (e.g., bug fixes, pull requests, collaboration)

# How?

## How is it organized?

1. Tabular data files shall follow analysis-friendly formats (e.g., the column is variable, the row is observation, data dictionary, the meaning of column/row headers)
2. Files and folders shall have self-explaining names matching content, meaning, and human abstractions (e.g., doc/, src/, results/, src/, bin/)
3. The artifact shall indicate best practices used (e.g., naming or code conventions, guidelines/checklists)
4. Useful metadata shall be used as part of filenames for pattern matching (e.g., yyyymmdd), where meaningful
5. The artifact shall be compliant with ICT accessibility standards (e.g., Section 508, WAI)
6. The experiment workflow shall be broken-down into small and simple procedures to facilitate reuse (e.g., scripts, functions)
7. The artifact shall include a website or wiki-page

## How to set up a running environment?

1. The artifact shall provide instructions for downloading
2. The artifact shall provide the open-source code
3. The artifact shall provide a binary/compiled version
4. The artifact shall provide a container for freezing dependencies and quickly setting up a running environment (e.g., VM, Docker)
5. The artifact shall provide a step-by-step tutorial of how to build the source code
6. The artifact compilation shall rely on build automation tools (e.g., make, ant)
7. The artifact compilation shall rely on dependency management tools (e.g., maven, pip)
8. The artifact shall provide instructions to install it

## How to get started?

1. The artifact shall include instructions for running it on minimal test data (e.g., quick run, smoke testing)
2. The artifact shall indicate the most relevant and interesting parts of the source code/artifact
3. The artifact shall include step-by-step instructions for running it (e.g., README, PDF)
4. The artifact shall include a video tutorial with the step-by-step for running it (e.g., Youtube, Vimeo)

## How to replicate the experiment?

1. The artifact shall provide instructions for manual/automated pre-processing of raw data for experiments (e.g., bash, python, R script)
2. The artifact shall provide instructions for manual/automated replication of the complete (or at least a subset) experiment as in the paper (e.g., bash, python, R script)
3. The experiment workflow shall be fully automated (including raw data processing, experiment execution, figures plotting)
4. The artifact shall include the experiment results shown in its associated paper in tabular and machine-readable format (e.g., .csv, .tab)
5. The artifact shall include log files produced

## How to run the analysis of results?

1. The artifact shall include scripts for the automated analysis of results as in the paper (e.g., R scripts, python scripts, Jupyter notebooks)
2. The artifact shall include scripts for drawing figures and/or plots as in the paper (e.g., R scripts, python scripts, Jupyter notebooks)
3. The artifact shall provide a clear description of the measurement procedures and metrics used in the paper

## How could it be repurposed?

1. The artifact shall indicate suggestions for contributions (e.g., notes.txt, todo.txt, ways it could be repurposed, wishlist)
2. The artifact documentation shall be designed considering users with minimal expertise
3. The artifact shall provide details about ethical concerns in replications
4. The artifact source code shall be documented (e.g., in-code comments, Javadoc)
5. The artifact shall provide instructions for integration/chaining with commercial tools (e.g., Matlab, DOORS)
6. The artifact shall report known issues, bugs, and limitations (e.g., issue tracker)

# How much/many?

## How many resources does it need?

1. The artifact shall indicate the system and environment where it was successfully evaluated (e.g., OS, CPU, RAM, GPU, Disk)
2. The artifact shall indicate the minimum system and environment requirements for usage (e.g., OS, CPU, RAM, GPU, Disk)
3. The artifact shall indicate the skills and/or settings required for usage (e.g., team configuration, users' skills)
4. The artifact shall indicate the approximate amount of time needed to replicate the experiment