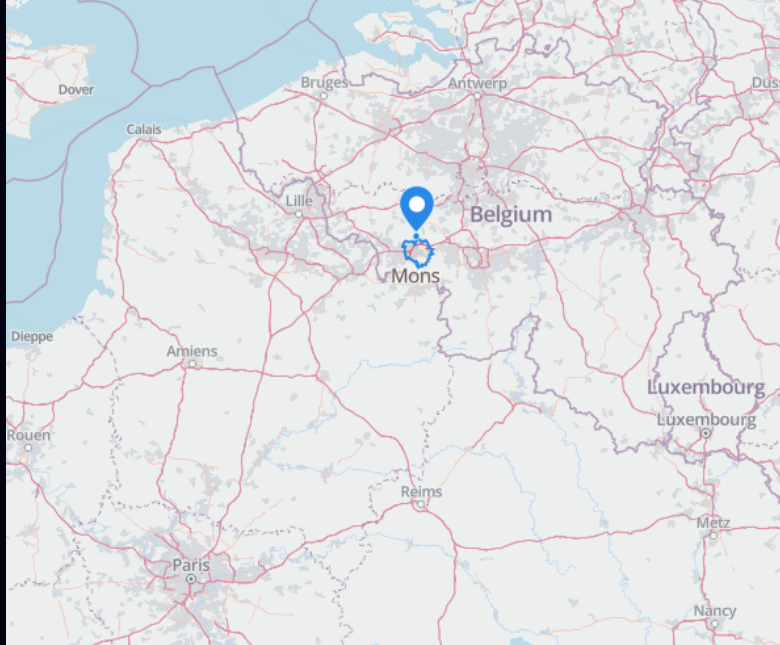


Reproductibilité et environnements de développement

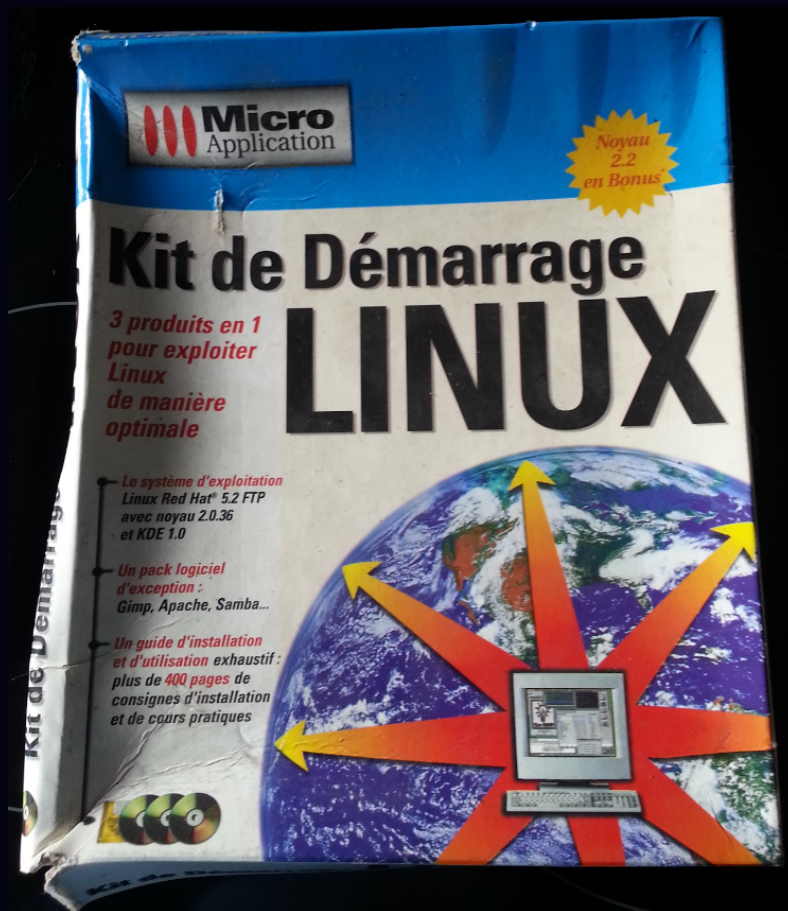
Wébinaire recherche reproductible
23 Mai 2025

■ 🖐 Introduction

- Bonjour, merci !
- Curieux, passionné par l'informatique et la technologie
- Originaire de la région de Mons, en Belgique



- Consultant à la Commission Européenne à Bruxelles
- Collaborateur volontaire à l'Université de Mons
- À la limite du burn-out durant le COVID, j'ai entamé un master en informatique à l'université de Mons afin de m'aider à surmonter cette période difficile et occuper mon esprit différemment.



Date de sortie: Nov 1998

SPIENNES

"Linux days" al'veye commune

Lorsqu'on vous propose des logiciels gratuits, libres, fiables, performants et stables, on se dit généralement qu'il y a là-dessous une arnaque monumentale! Et c'est faux! En effet, depuis déjà de nombreuses années, Linux propose ce genre de produits au grand public car ce système ou plutôt cette philosophie permet de réelles économies en temps et en argent. Pour tenter d'enrayer le phénomène, on a tenté de dénigrer la fiabilité ou encore la convivialité des programmes. Durant deux journées entières, les passionnés de Linux - 95% d'amateurs et 5% de professionnels - sont venus partager leur expérience, leurs connaissances et leur savoir-faire avec les trois cents visiteurs du week end. ■■



J.C. Linux commence à faire des émules

Utilisateur de Linux au quotidien.

Pas très très longtemps sous RedHat, quelques années sous Slackware, Gentoo, FreeBSD, et finalement NixOS depuis 2021.

■ 💰 Carrière professionnelle

- Consultant externe à la Commission Européenne depuis 2012
 - Agence EISMEA (2012 → 2015)
 - Développement de projets PHP
 - DG Digit (2015 → 2025)
 - Développement de projets PHP open-source
 - Développement du PHP Competency Center
 - Recherche et développement de solutions pour des environnements de développement éphémères et reproductibles
 - Développement en Python et I.A.
 - DG EAC (2025 → ...)
 - Architecte
 - DevOps

■ 🙌 Open Source

✍️ Contributeur Open Source depuis environ 2005.

🤖 J'ai toujours aimé l'idée de collaborer de manière transparente et, d'une certaine manière, contribuer à un monde meilleur grâce à l'informatique ouverte et au partage de connaissances. L'open-source occupe donc une place importante dans ma vie. C'est aussi important pour moi de trouver une occupation que j'aime et pour laquelle je me sens utile.

🔪 En 2019-2020, je découvre la programmation fonctionnelle et je commence à programmer différemment grâce aux nouveaux concepts appris dans ce paradigme.

❄️ C'est aussi à ce moment-là que je découvre Nix et NixOS et le monde fascinant de la reproductibilité.

💡 Bien sûr, il m'a fallu du temps pour apprendre... et j'en apprend encore beaucoup tous les jours... et j'adore ça en fait !

🎓 En 2021, je retourne à l'école pour obtenir un master en informatique à l'université de Mons. Ma thèse de master, sera sur la reproductibilité en ingénierie logicielle. Depuis juin 2024, le document est bien évidemment open-source et en accès libre. J'y ai apporté quelques modifications depuis.

■ 🤔 Pourquoi la reproductibilité ?

■ Envie de découvrir de nouvelles choses

- On en parlait de plus en plus autour de moi et je devais remplacer mon ordinateur portable...

■ Envie de lier l'utile à l'agréable

- Pas d'environnements de travail adéquats, pas d'outils adéquats.
- Solutions proposées totalement inadaptées
- L'utilisation de son propre ordinateur était souvent plus efficace
- Projets avec beaucoup trop de prérequis et d'assomptions implicites
- Pas de vision à long terme.
- *etc etc...*

■ Et donc?

- Je me suis rendu compte que la reproductibilité est un sujet essentiel dans le développement de logiciels
- Beaucoup de problèmes rencontrés dans le cadre de mon travail pourraient être résolus en implémentant des processus reproductibles.
- L'informatique moderne devrait être pensée de manière à ce que les choses soient reproductibles **par défaut**.

If everyone on a research team knows that everything they do is going to someday be published for reproducibility , they'll behave differently from day one.

2009, Donoho et al., <https://doi.org/10.1109/MCSE.2009.15>

■ L'utile vers l'agréable

- De frustrations... vers une solution
- Adoption de Nix, une solution initialement européenne !
- Écriture de ma thèse de master sur ce sujet.

Reproducibility In Software Engineering

zenodo

Search records...

Q

CommunitiesMy dashboard

+

▼

pol.dell...

▼

Software Engineering Research

Published May 1, 2025 | Version v5

ThesisOpen

Edit

New version

Share

18K VIEWS

2K DOWNLOADS

Show more details

Versions

Version v5	May 1, 2025
10.5281/zenodo.15315531	
Version v4	Oct 5, 2024
10.5281/zenodo.13894231	
Version v3	Sep 5, 2024
10.5281/zenodo.13693009	
Version v2	Aug 5, 2024
10.5281/zenodo.13208605	
Version v1	Jul 5, 2024
10.5281/zenodo.12666899	

Reproducibility in Software Engineering

Dellaiera, Pol (Researcher)

Contributors

Supervisor: Mens, Tom

The concept of reproducibility has long been a cornerstone in scientific research, ensuring that results are robust, repeatable, and can be independently verified. This concept has been extended to computer science, focusing on the ability to recreate identical software artefacts. However, the importance of reproducibility in software engineering is often overlooked, leading to challenges in the validation, security, and reliability of software products.

This master's thesis aims to investigate the current state of reproducibility in software engineering, exploring both the barriers and potential solutions to making software more reproducible and raising awareness. It identifies key factors that impede reproducibility such as inconsistent environments, lack of standardisation, and incomplete documentation. To tackle these issues, I propose an empirical comparison of tools facilitating software reproducibility.

To provide a comprehensive assessment of reproducibility in software engineering, this study adopts a methodology that involves a hands-on evaluation of four different methods and tools. Through a systematic evaluation of these tools, this research seeks to determine their effectiveness in establishing and maintaining identical software environments and builds.

This study contributes to academic knowledge and offers practical insights that could influence future software development protocols and standards.

<https://doi.org/10.5281/zenodo.12666898>

Issues Pull requests Milestones Explore Donate

p1ld7a / master-thesis

<> Code Issues Pull requests Activity + Enable more Settings

Pol Dellaiera's Master Thesis - Reproducibility In Software Engineering

Manage topics

131 commits 3 branches 0 tags 1.6 MiB

main Find a file Add file HTTPS SSH https://codeberg.org/p1ld7a/master-thesis.gtt

Search code... Exact

Pol Dellaiera	152f6609d0	fix: add missing references to chapter 2	3 weeks ago
.github		chore(deps): bump cachix/install-nix-action from 30 to 31	last month
lib		chapter 3: fix version of build-base in Dockerfile	8 months ago
nix		flake: refactor and simplify	4 months ago
resources		chore: minor cosmetic change in a table	3 weeks ago
src/thesis		fix: add missing references to chapter 2	3 weeks ago
.devcontainer.json		Initial commit	11 months ago
.editorconfig		Initial commit	11 months ago
.envrc		Initial commit	11 months ago
.gitignore		Initial commit	11 months ago

<https://codeberg.org/p1ld7a/master-thesis>

drupol / master-thesis

Q Type 17 to search

 |  |  |  |  | 

 Code  Issues  Pull requests  Discussions  Actions  Security  Insights  Settings

 master-thesis Public

 Unpin  Unwatch 1  Fork 1  Star 27

 main 4 Branches 60 Tags  Add file  Code

 drupol fix: add missing references to chapter 2 152f660 · 3 weeks ago 131 Commits

	.github	chore(deps): bump cachix/install-nix-action from 3...	last month
	lib	chapter 3: fix version of build-base in Dockerfile	8 months ago
	nix	flake: refactor and simplify	4 months ago
	resources	chore: minor cosmetic change in a table	3 weeks ago
	src/thesis	fix: add missing references to chapter 2	3 weeks ago
	.devcontainer.json	Initial commit	11 months ago
	.editorconfig	Initial commit	11 months ago
	.envrc	Initial commit	11 months ago
	.gitignore	Initial commit	11 months ago
	.prettierignore	Initial commit	11 months ago
	.prettierrc	Initial commit	11 months ago
	CITATION.cff	chore: update DOI, always resolve to the latest do...	10 months ago
	LICENSE	Initial commit	11 months ago
	README.md	chore: update DOI, always resolve to the latest do...	10 months ago

About

Pol Dellaiera's Master Thesis - Reproducibility In Software Engineering

codeberg.org/p1ld7a/master-thesis

master-thesis unons

 Readme  View license  Cite this repository  Activity

 27 stars  1 watching  1 fork

Releases 60

 Release 87 (ec1f625bf3aeb0... Latest on Apr 10

+ 59 releases

Packages

No packages published

[Publish your first package](#)

<https://github.com/drupol/master-thesis>

■ Reproducibility In Software Engineering

- Initialement un repository privé, partagé avec mon superviseur Tom Mens
- Reproductible par défaut, avec environnement de développement
- Écrite avec Typst, une alternative moderne à LaTeX
- Composition
 - Introduction
 - Reproductibilité
 - Partie théorique
 - Historique
 - Définitions
 - Importance
 - Évaluation logicielle
 - Partie pratique
 - Pas d'outils
 - Docker
 - Guix
 - Nix
 - Conclusion

■ 🛠 Environnements Reproductibles

Tout comme on dort toujours mieux dans son propre lit, ...

On est toujours plus efficace dans un environnement de travail familier et avec ses propres outils.

■ Le passé

- Approche convergente qui rétablit au mieux une situation donnée en cas de divergence
- Systèmes muables

- Exemples de solutions
 - *"dotfiles"*
 - Puppet
 - Chef
 - Ansible

■ Le futur

- Approche congruente où on ne laisse pas de place à l'improvisation
- Systèmes immuables

- Exemples de solutions
 - Nix
 - Guix

■ Et Docker/Podman?

- Ces outils permettent de construire et d'interpréter des images de conteneurs
- Ces images sont immuables certes, mais pas forcément reproductibles
- L'utilisation de ces outils ne garantit pas la reproductibilité
 - Pas de garantie que l'environnement de développement soit le même sur toutes les machines
 - Pas de garantie que l'environnement de production soit le même sur toutes les machines

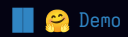
■ Pourquoi?

```
1 FROM buildpack-deps:bookworm
2 # ...
3 RUN set -eux; \
4     apt-get update; \
5     apt-get install -y --no-install-recommends \
6         libbluetooth-dev \
7         tk-dev \
8         uuid-dev \
9     ; \
10    rm -rf /var/lib/apt/lists/*
11 # ...
```

Un extrait du Dockerfile utilisé pour construire l'image officielle de Python.

■ Environnements de Développement Reproductibles

- Partons de l'hypothèse qu'un environnement de développement est le résultat d'un build
- Nix est un builder performant qui me permet de partager des "*builds*"
- Si ça marche sur ma machine, il y a de fortes chances que ça marche ailleurs
- Grâce à une syntaxe principalement déclarative, pas de dépendances implicites ou cachées
- Force les développeurs à déclarer les dépendances de leurs projets, à mieux connaître et comprendre leurs outils



Demo 1 - Les devShells

- Créer un environnement de développement reproductible
- Contenant les logiciels:
 - PHP
 - Typst

```
1 # shell.nix
2 {
3   pkgs ? import (fetchTarball {
4     url = "https://github.com/NixOS/nixpkgs/archive/e314d5c6d3b3a0f40ec5bcbc007b0cbe412f48ae.tar.gz";
5     sha256 = "049zgk55rk6x3m0v7xdcabnayc0l1rbbfvfg9fr9ky92f9g2wl12";
6   }) { }
7 }:
8 pkgs.mkShell {
9   packages = [
10     pkgs.php
11     pkgs.typst
12   ];
13 }
```

```
nix-shell ./demos/demo-1-a/ --run "php --version"
```

```
nix-shell ./demos/demo-1-a/ --run "typst --version"
```

[finished]

```
PHP 8.4.7 (cli) (built: May  6 2025 12:31:58) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.4.7, Copyright (c) Zend Technologies
    with Zend OPcache v8.4.7, Copyright (c), by Zend
Technologies
typst 0.13.1 (unknown hash)
```

```

1 # flake.nix
2 {
3   inputs = {
4     nixpkgs.url = "github:NixOS/nixpkgs/nixpkgs-unstable";
5   };
6
7   outputs = inputs:
8     let
9       system = "x86_64-linux";
10      pkgs = import inputs.nixpkgs { inherit system; };
11    in
12    {
13      devShells.${system}.mon-php-shell = pkgs.mkShell {
14        packages = [
15          pkgs.php
16          pkgs.typst
17        ];
18      };
19    };
20 }

```

```
nix flake show ./demos/demo-1-b
```

[finished]

```

evaluating ''...
git+file:///home/pol/Code/drupal/presentation-2025-05-23?dir=src/2025-05-23-wrr/demos/demo-1-b
evaluating 'devShells'...
└─ devShells
evaluating 'devShells.x86_64-linux'...
└─ x86_64-linux
evaluating 'devShells.x86_64-linux.mon-php-shell'...
└─ mon-php-shell: development environment 'nix-shell'

```

```

1 # flake.nix
2 {
3   inputs = {
4     nixpkgs.url = "github:NixOS/nixpkgs/nixpkgs-unstable";
5     flake-parts.url = "github:hercules-ci/flake-parts";
6   };
7
8   outputs = inputs: inputs.flake-parts.lib.mkFlake { inherit inputs; } {
9     systems = [ "x86_64-linux" "x86_64-darwin" "aarch64-linux" "aarch64-darwin" ];
10
11     perSystem = { pkgs, ... }: {
12       devShells.mon-php-shell = pkgs.mkShell {
13         packages = [
14           pkgs.php
15           pkgs.typhst
16         ];
17       };
18     };
19   };
20 }

```

```
nix flake show ./demos/demo-1-c
```

[finished]

```

evaluating ''...
git+file:///home/pol/Code/drupal/presentation-2025-05-23?dir=src/2025-05-23-wrr/demos/demo-1-c
evaluating 'devShells'...
└─ devShells
  evaluating 'devShells.aarch64-darwin'...
  └─ aarch64-darwin
    evaluating 'devShells.aarch64-darwin.mon-php-shell'...
    └─ mon-php-shell omitted (use '--all-systems' to show)
  evaluating 'devShells.aarch64-linux'...
  └─ aarch64-linux
    evaluating 'devShells.aarch64-linux.mon-php-shell'...
    └─ mon-php-shell omitted (use '--all-systems' to show)
  evaluating 'devShells.x86_64-darwin'...
  └─ x86_64-darwin
    evaluating 'devShells.x86_64-darwin.mon-php-shell'...
    └─ mon-php-shell omitted (use '--all-systems' to show)
  evaluating 'devShells.x86_64-linux'...
  └─ x86_64-linux
    evaluating 'devShells.x86_64-linux.mon-php-shell'...
    └─ mon-php-shell omitted (use '--all-systems' to show)

```

Live demo

Projets montrés:

- <https://code.europa.eu/ecphp/devs-profile/>
- <https://code.europa.eu/pol/ec-lib/>



That's all Folks!