Research and Software Engineering at the Netherlands eScience Center

Carlos Martinez-Ortiz
24th February 2021
What is the eScience center

NWO

Ministry of Education, Culture and Science

SURF

Project funding

Research software

Office, coordination & management
eScience Center main activities

**Collaborative projects**
with selected academic researchers

**Sharing expertise**
with research communities
Multiscale simulations of excitation dynamics in molecular materials for sustainable energy applications

- Understand how molecular systems harvest energy from the sun and how energy propagates in these materials
- Quantum Chemistry for energy applications
Quantum Chemistry for Photovoltaics

Focus on the excited state properties of small molecular systems using GW-BSE

- Reduce memory requirements
- Accelerate diagonalization scheme
- Off-load heavy calculation to GPU

Harvesting solar energy in organic materials
Quantum Chemistry for Photovoltaics

Diagonalization of large matrices
Nicolas Renaud

Development of matrix-free eigensolver (Jacobi-Davidson) and integration in the open source library Eigen

GPU programming
Felipe Zapata

Development of CUDA kernels for the calculations of tensor-matrix product
Involved in 3-center integral calculations

https://orcid.org/0000-0001-5565-7577
Real Time National Policy Adjustment and Evaluation on the basis of a computational model for COVID19

- Economic & social costs of social distancing measures
Real Time National Policy Adjustment and Evaluation on the basis of a computational model for COVID19
EnKF: Python implementation

- Python based implementation developed for OMUSE/HyMUSE
- based on driver class and a number user implementable classes

→the implementation and use of EnKF is easy, but the real devil is in the details of the choice of the state variables, observation function and the model for observation and state errors.
Special Interest Groups

• Stay up to date with the state of the art
• Coordinate use of technology in projects
• Recently: more external engagement
• Knowledge dissemination
  • workshops, hackathons, tutorials, etc.
Research Software Best Good Practices, building expertise and knowledge sharing

Mateusz Kuzak
24th February 2021
The Netherlands eScience Center is the National Expertise Center for Research Software Engineering.

That does not automagically make software developed at the Center better than that developed at other research institutions.

Q1. How do we know we know the software is of good quality?
Q2. What does it mean good quality software?

Answer: Software Quality Checklist
Collaborations Workshop 2016 (CW16), Royal College of Surgeons of Edinburgh
Lightning talk CW16

Mateusz Kuzak
eScience research engineer

- nlesc
  /estep-checklist

- SoftDev4LS
  /open-source-software
  /good-enough-practices
  /software-development-metrics
The checklist grew into a guide

https://guide.esciencecenter.nl/
The **bare minimum** that every software project should do, from the start, is:

- Pick & include an [open source license](#)
- Use [version control](#)
- Use a [publicly accessible](#) version controlled repository
- Add a [readme describing the project](#)

We recommend that you also do the following (from the start of the project):

- Use [code quality tools](#)
- [Testing](#)
- Use [standards](#)

Additional steps depend on the goal of the software (zero or more can apply):

- [I'm publishing a paper](#)
- [I'm expecting users](#)
- [I'm expecting contributors](#)
Guide in 2019

- Version Control
- Code Quality
- Code Review
- Licensing
- Communication
- Testing
- Releases

- Documentation
- Standards
- UX
- Specific language guides (Java, Python, C and C++, Fortran)
- Intellectual property
- ... and more
Growing our own content

Vs

Referencing external content
The Turing Way

- largely similar content
- goes beyond research software
- large diverse international community of contributors
- opportunities for contributions by the Center, based on our expertise

What's in the Guide

How we do it at the Center
Internal Guide

Universal information
for all RSEs and researchers writing code
In 2019 we decided to the work

Guide
• How we do it at the Center
• Topics not covered by The Turing Way

The Turing Way Contribution
• Universal content
How is the work organized?

- Written by engineers, based on their expertise
- One Guide maintainer
- Coordinated by Software Sustainability SIG (Special Interest Group)
- Progress via sprints and async contributions
Software Sustainability SIG

- Knowledge sharing on
  - Good practices
  - Sustainability
  - Reusability
  - Reproducibility
- Internal training (version control, CI/CD etc.)
- Developing guidelines
- Contribution to the Guide and the Turing Way

Work in small teams, project based in sprints.