

Tools for Collaborative Work

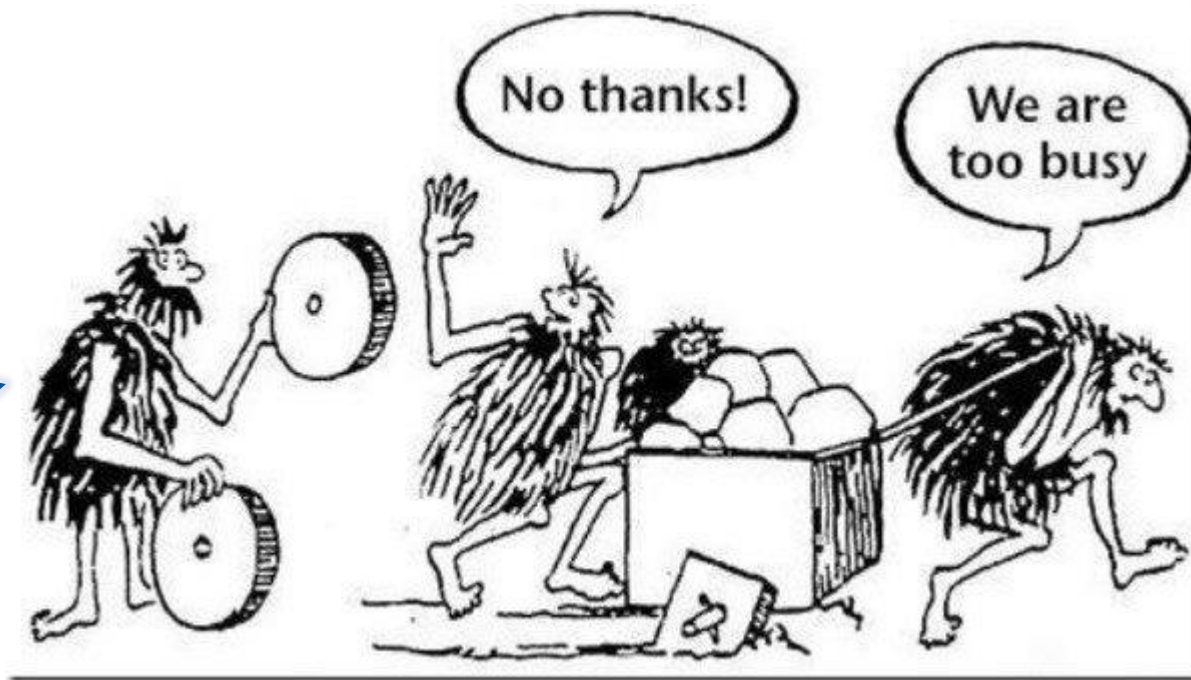
Overview and discussion

Michał Maciejewski on behalf of the STEAM
TE-MPE-PE

**THE
STEAM**

Motivation

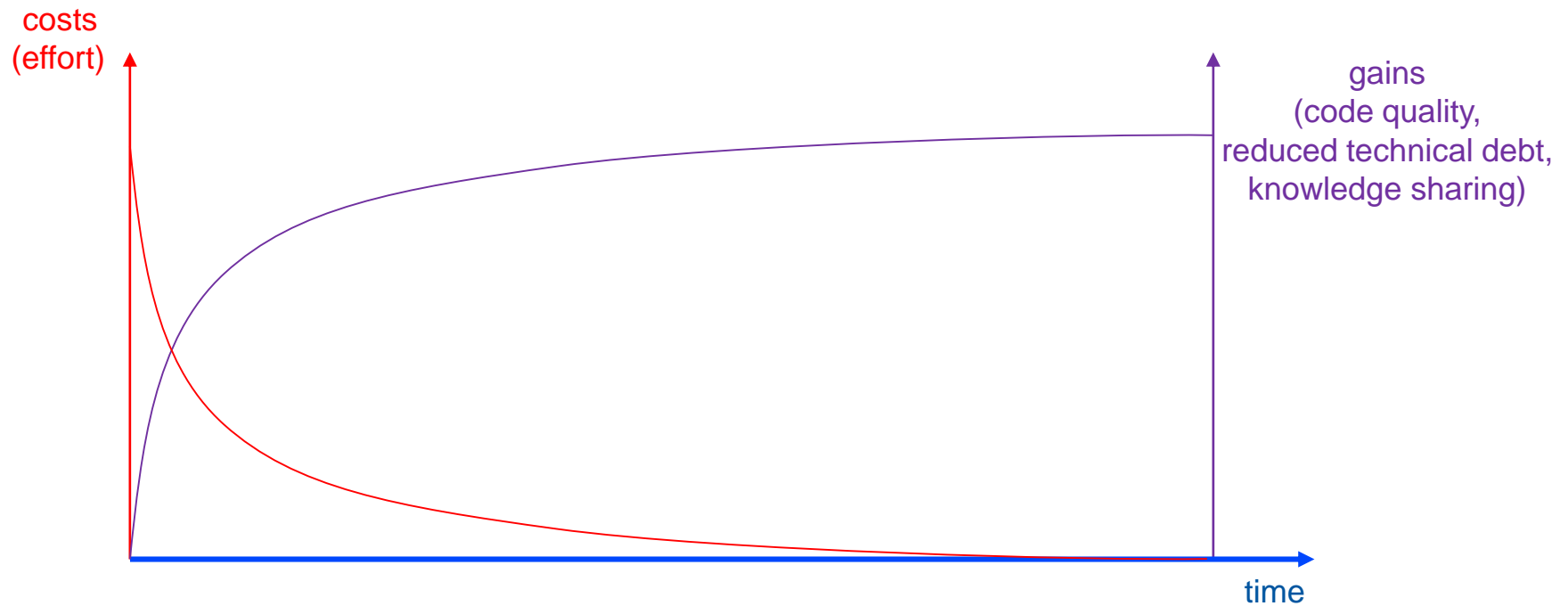
- We have new colleagues in our section
- We have experience using some of the tools
- We (people in general) can be divided into two groups:
*Those who **do** backups and those who **will do** backups*



Source: <https://www.quora.com/I-work-really-very-hard-for-CSIR-life-science-but-still-my-paper-was-not-so-good-What-should-I-have-to-do-so-I-can-crack-it>

Gains vs. costs

- The concepts to be presented are based on the common sense and we've been doing some of them.
- The goal of this presentation is to structure these concepts and discuss a strategy for collaborative work.



Outline

1. Twiki&Indico
2. Sharepoint
3. CernBox
4. Overleaf
5. Coding Conventions
6. GitLab
7. LHC Signal Access
8. SWAN (Service for Web based ANalysis)

Section website (cernbox.cern.ch)

- Our core products are analyses covering the accelerator needs (papers, posters, presentations, internal notes, design reports, ...)
- Great platform to share our activities internally and externally!

The screenshot shows a web browser displaying the Twiki page for the TEMPEPE section. The page title is "Welcome to the 'TE-MPE-PE' Twiki Webpage". The content includes a list of section members, a mandate, and a diagram illustrating the relationship between different technical areas.

Section members:

- STAFF:** Arjan Verweij (S.L.), Bernard Auchmann, Daniel Wolmann, Michael Jonker, Ruediger Schmidt, Zinur Charifoutine (p.1)
- FELL:** Andrea Apollonio, Lorenzo Bortot, Marco Priol, Matthieu Valette
- DOCT:** Jonas Ghini, Laura Grob, Michal Maciejewski, Odel Rey Orozco, Oliver Stein, Vivien Raginel
- FTEC:** Alejandro Fernandez Navarro
- TECH:** David Kleiven, Tobias Griesemer
- COAS:** Arto Niemi, Naeem Tahir

Mandate:

- Studies of protection issues for LHC superconducting magnet circuits.
- Monitoring of the evolution of the magnet circuit performance.
- Development of tools to understand circuit electrical and protection behaviour.
- Machine protection studies. Analyse coherence of MP systems across systems. Establish quench and damage levels due to beam loss. Study the reliability of the machine protection systems.
- Assist commissioning of the powering system, machine protection systems and provide support to operation with and without beam.
- Coordination of the studies for CLIC machine protection systems and other future accelerators.

Diagram: A diagram with three main colored boxes: "Circuit Modeling" (yellow), "Beam Impact & Machine Protection" (green), and "Fast beam" (red). The "Circuit Modeling" box lists: Quench studies (LHC, HiLumi, FCC), LHC circuit issues (shorts, voltage transients, etc), CLIQ analysis, and STEAM. The "Beam Impact & Machine Protection" box lists: Damage limits on superconductors, Machine protection (LHC, HiLumi, FCC, CLIC), and Hydrodynamic tunnelling. The "Fast beam" box lists: UFO studies, Beam induced quenches, BLM thresholds, and Fast beam. Arrows indicate interactions between these areas.



► Attachments

[Edit](#) | [Attach](#) | [Watch](#) | [Print version](#) | [History: r17 < r16 < r15 < r14 < r13](#) | [Backlinks](#) | [Raw View](#) | [Raw edit](#) | [More topic actions](#)

Sharepoint* (cern.ch/steam)

Simulation of Transient Effects in Accelerator Magnets

Welcome to the website of the STEAM project. The project aims to re-shape, for the life-time of the LHC and beyond, the foundations of simulations of transient effects in superconducting magnets and circuits.

Circuits of superconducting accelerator magnets are complex systems with components combining technologies from several engineering fields, show strong mutual interactions and can contain up to thousands of components. The transient phenomena occurring in the circuit can be consistently captured only if the simulation includes the components' mutual influence. It is desirable to solve this type of multi-physics, multi-scale, and multi-rate problem using an intrinsically consistent monolithic approach. However, this can lead to non-acceptable computational times. At the same time, it is worth noting that no currently available multi-physics simulation tool covers the full range of phenomena: high-performance tools tend to be specialized on only a subset of physical domains.

Instead, these complex phenomena are decomposed into smaller subproblems. These subproblems are solved with dedicated, validated tools. In particular, netlist-based circuit solvers are used to model electrical networks, finite element method is employed to represent magneto-thermal and mechanical phenomena, and fixed-time stepping solvers are used to study controller behavior.

The STEAM project has been developed in order to tackle these challenges and is based on two pillars (see figure below):

1. Hierarchical co-simulation framework providing a common interface to run cooperative simulations of the validated models.
2. Software packages for automated generation of both electrical circuit and finite element models.

Diagram: Hierarchical Co-simulation Architecture

The diagram illustrates the architecture of the hierarchical co-simulation framework. At the top, a **User** interacts with a **Co-simulation Scenario**. This scenario feeds into a **Hierarchical Co-simulation** layer, which uses **Waveform Relaxation / Weak Coupling**. This layer is connected to **Tool Adapters**, which interface with three specialized simulation tools: **Field** (using COMSOL), **Circuit**, and **Controller**. These tools are supported by an **Automated Model** layer, which is also accessible to the **User**. The Java logo is present, indicating the platform used for the tool adapters.

*NB: CERN is in the process of Mexit



CERNBox (cernbox.cern.ch)

- Use of google drive, onedrive, dropbox, etc. is discouraged
- 1 TB per user (also personal files) / project
- Multiplatform (Android, iOS, Windows, OS, linux)
- Synchronises across multiple locations
- Stores 10 latest versions of a file



Multiple users in our section: Arjan, Bernhard, Matthieu, Michał, Christoph, Per, Zinur, Akrivi, Lorenzo, Marco, ...

Overleaf (<http://overleaf.com>)

- A platform for cooperative writing of papers + paper repository!
- No more hand written corrections, multiple versions of files:
Nature_paper_MM.pdf, Nature_paper_MM_v1.pdf,...

The screenshot shows the Overleaf dashboard in a web browser. The browser's address bar displays "https://www.overleaf.com/dash". The page header includes the Overleaf logo, navigation links for "FEATURES & BENEFITS", "TEMPLATES", "PRICING", "COMPANY", and "HELP", and a user profile for "Michal".

The main content area is divided into several sections:

- NEW PROJECT**: A green button with an upload icon.
- Search**: A search bar with a "SEARCH" button.
- Filters**: A sidebar on the left with filters for "Active" (29), "Starred" (0), "Archived" (15), and "Trash" (21). Below these are project tags: "thesis" (7), "playground" (5), "paper" (3), "abstract" (2), and "mpcci" (2).
- Project List**: A list of projects with details such as title, edit time, and user. Each project has icons for trash, download, star, and share. The projects listed are:
 - Co-Simulation of Transient Effects in Superconducting Accelerat... (16378480...jcb)
 - Bond Graph model of an SC magnet (12619887...zfq)
 - Modified RB 11T Cryo-assembly (17602726...jbd)
 - Corrected schematics of electrical circuits (15358290...ysh)
 - CERN blue text (16919469...rsr)
- Advertisement**: A banner for the European Organization for Nuclear Research (CERN) with the CERN logo and text: "You are a member of European Organization for Nuclear Research (CERN) on Overleaf, which grants you extra features. LEARN MORE".
- Pro+ Account Promotion**: A central box titled "Find out more about your Pro+ account..." with three options: "Protected Projects", "Full Project History", and "Higher Limits and More".

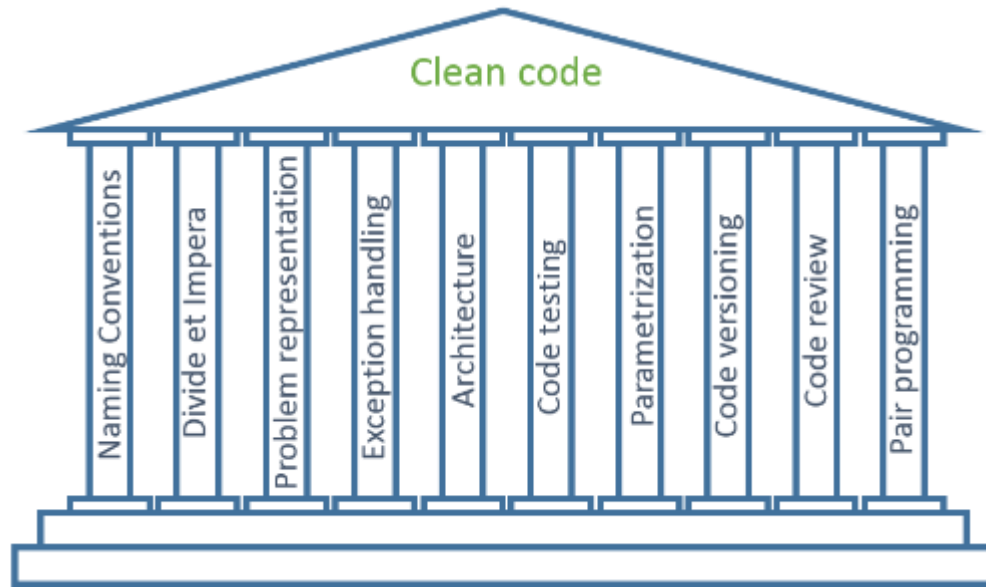
Overleaf (<http://overleaf.com>) + ShareLaTeX

- A platform for cooperative writing of papers + paper repository!
- No more hand written corrections, multiple versions of files:
Nature_paper_MM.pdf, Nature_paper_MM_v1.pdf,...



ShareLaTeX

Good practices, continuous integration workflow



GitLab

Code repository
Code review
Continuous integration



Scrum methodology
Mon-Wed-Fri stand-up
meetings @10AM

Best Practices for Scientific Computing

<http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745>

Box 1. Summary of Best Practices

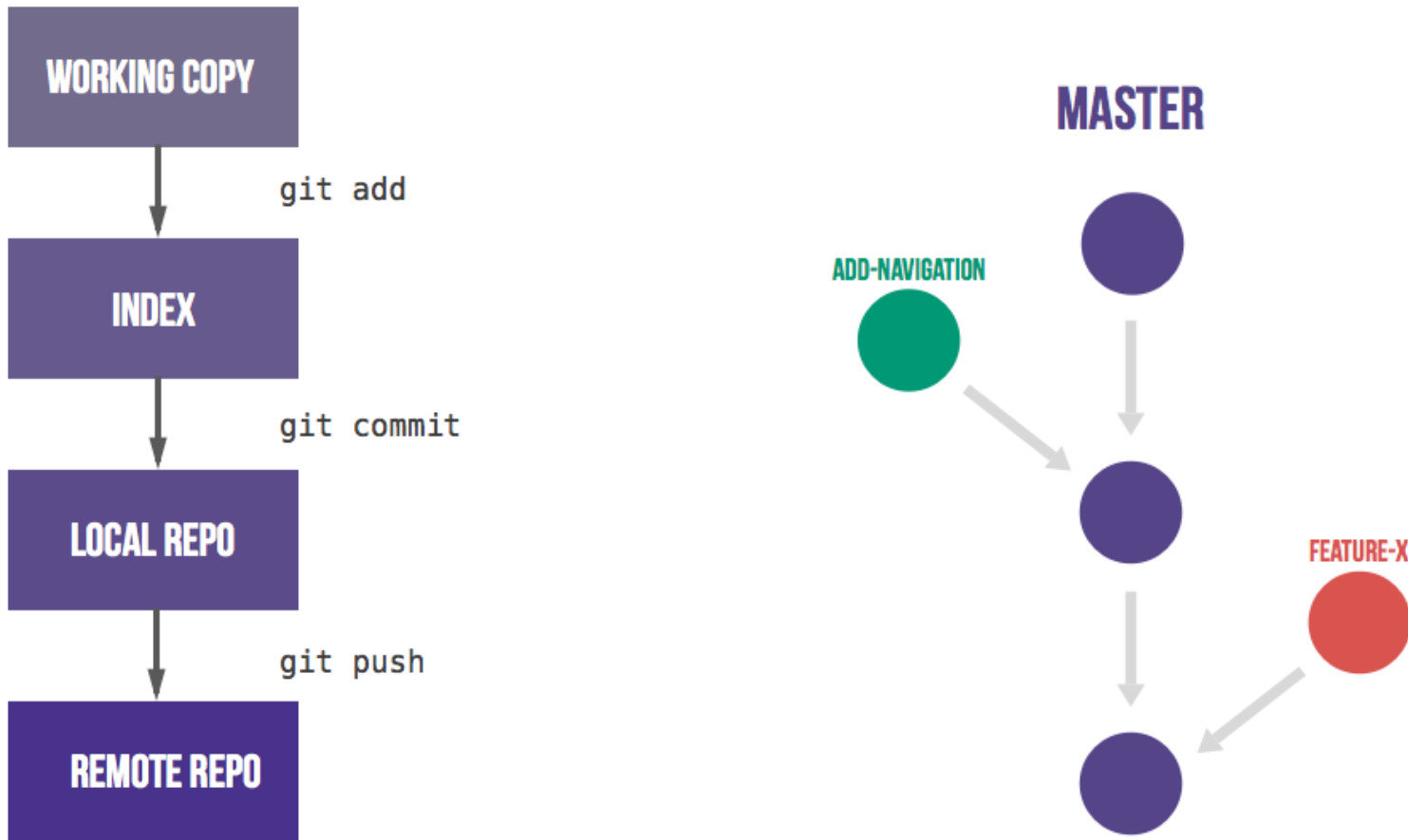
1. Write programs for people, not computers.
 - a. A program should not require its readers to hold more than a handful of facts in memory at once.
 - b. Make names consistent, distinctive, and meaningful.
 - c. Make code style and formatting consistent.
2. Let the computer do the work.
 - a. Make the computer repeat tasks.
 - b. Save recent commands in a file for re-use.
 - c. Use a build tool to automate workflows.
3. Make incremental changes.
 - a. Work in small steps with frequent feedback and course correction.
 - b. Use a version control system.
 - c. Put everything that has been created manually in version control.
4. Don't repeat yourself (or others).
 - a. Every piece of data must have a single authoritative representation in the system.
 - b. Modularize code rather than copying and pasting.
 - c. Re-use code instead of rewriting it.
5. Plan for mistakes.
 - a. Add assertions to programs to check their operation.
 - b. Use an off-the-shelf unit testing library.
 - c. Turn bugs into test cases.
 - d. Use a symbolic debugger.
6. Optimize software only after it works correctly.
 - a. Use a profiler to identify bottlenecks.
 - b. Write code in the highest-level language possible.
7. Document design and purpose, not mechanics.
 - a. Document interfaces and reasons, not implementations.
 - b. Refactor code in preference to explaining how it works.
 - c. Embed the documentation for a piece of software in that software.
8. Collaborate.
 - a. Use pre-merge code reviews.



Simplicity is the art of maximizing the work not done

- KISS – Keep It Simple, Stupid!
- Recognition and use of design patterns
- Code review with experts (TE-MPE-MS, EN-ACE-EDM)
- Search for canonical, math-based problem representation
- Internal code refactoring
- Static code analysis with **sonar qube** (code duplications, code smells, complexity)
- **Humility in programming (complicated solution is not impressive...)**
 - <http://labviewjournal.com/2013/05/humility-1/>
 - <http://labviewjournal.com/codereviews/Code%20Review%20Presentation.pdf>
 - <https://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>

Gitlab (gitlab.cern.ch) – versioning!



Several users in our section: Matthieu, Martyna*, Philippe*, Michał, Christoph, Per, Zinur, Akrivi, Laura, Lorenzo, Marco, Jonas, Alejandro*, ...

<https://twiki.cern.ch/twiki/bin/view/TEMPEPE/CodingConvention>

LHC Signal Access

- In our section we are often times querying signals from the LHC (FPA, QPS, FGC, BLM and many more cryptic abbreviations)
- There is a number of custom tools to access Post Mortem (Java, LabVIEW) or CALS (TIMBER)
- Within our group (MPE-MS) there is a PM REST API
- BE-CO develops NXCALS
- We are creating a light-weight API to homogenize signals

<http://pm-api-pro/v2/>

[http://pm-api-pro/v2/
pmdata/signal?system=FGC&
className=51_self_pmd&
source=RPTE.UA27.RB.A23
×tampInNanos=1529792241640000000
&signal=STATUS.I_MEAS](http://pm-api-pro/v2/pmdata/signal?system=FGC&className=51_self_pmd&source=RPTE.UA27.RB.A23×tampInNanos=1529792241640000000&signal=STATUS.I_MEAS)

Several users in our section: Matthieu, Philippe, Michał, Christoph, Per, Zinur, Akrivi



SWAN (<http://swan.cern.ch>)

- Analyse data without the need to install any software
- Access experiments' and user data in the CERN cloud
- Share with colleagues
- Notebook = code + output (in one file!)

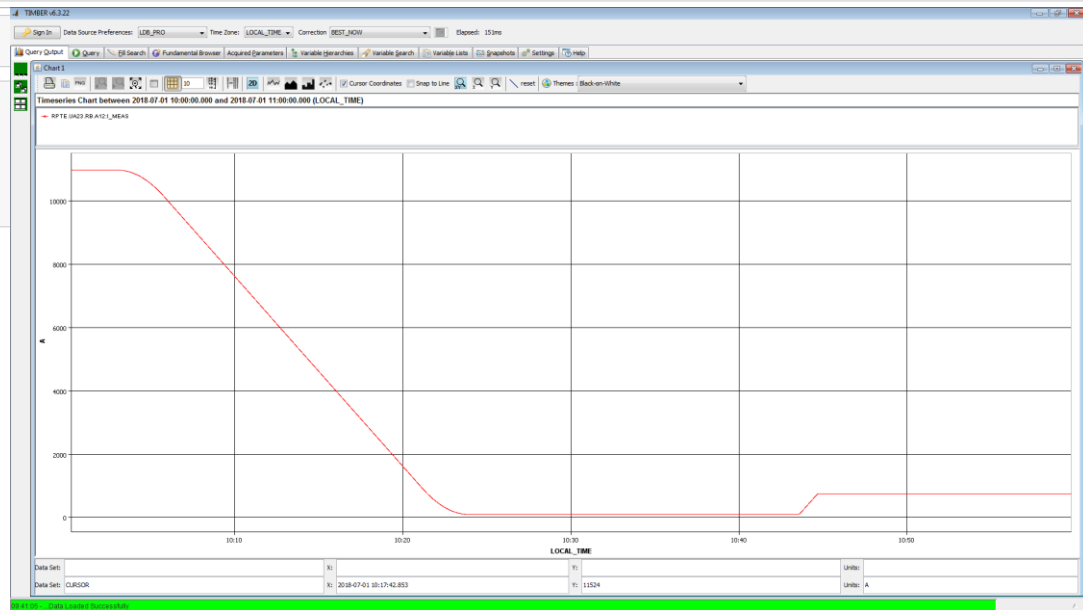
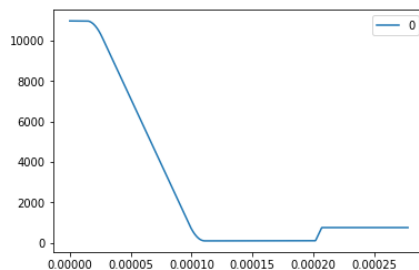
Retrieve and plot DFLAS.7L2.RB.A12.LD1:U_RES in sector 12 at 2018-07-01 10:00 - 11:00

```
In [22]: # RPTE.UA23.RB.A12:I_MEAS
t1 = '2018-07-01 10:00:00.000'
t2 = '2018-07-01 11:00:00.000'
d = ldb.get('RPTE.UA23.RB.A12:I_MEAS', t1, t2)
```

```
In [28]: time = d.get('RPTE.UA23.RB.A12:I_MEAS')[0]
current = d.get('RPTE.UA23.RB.A12:I_MEAS')[1]
```

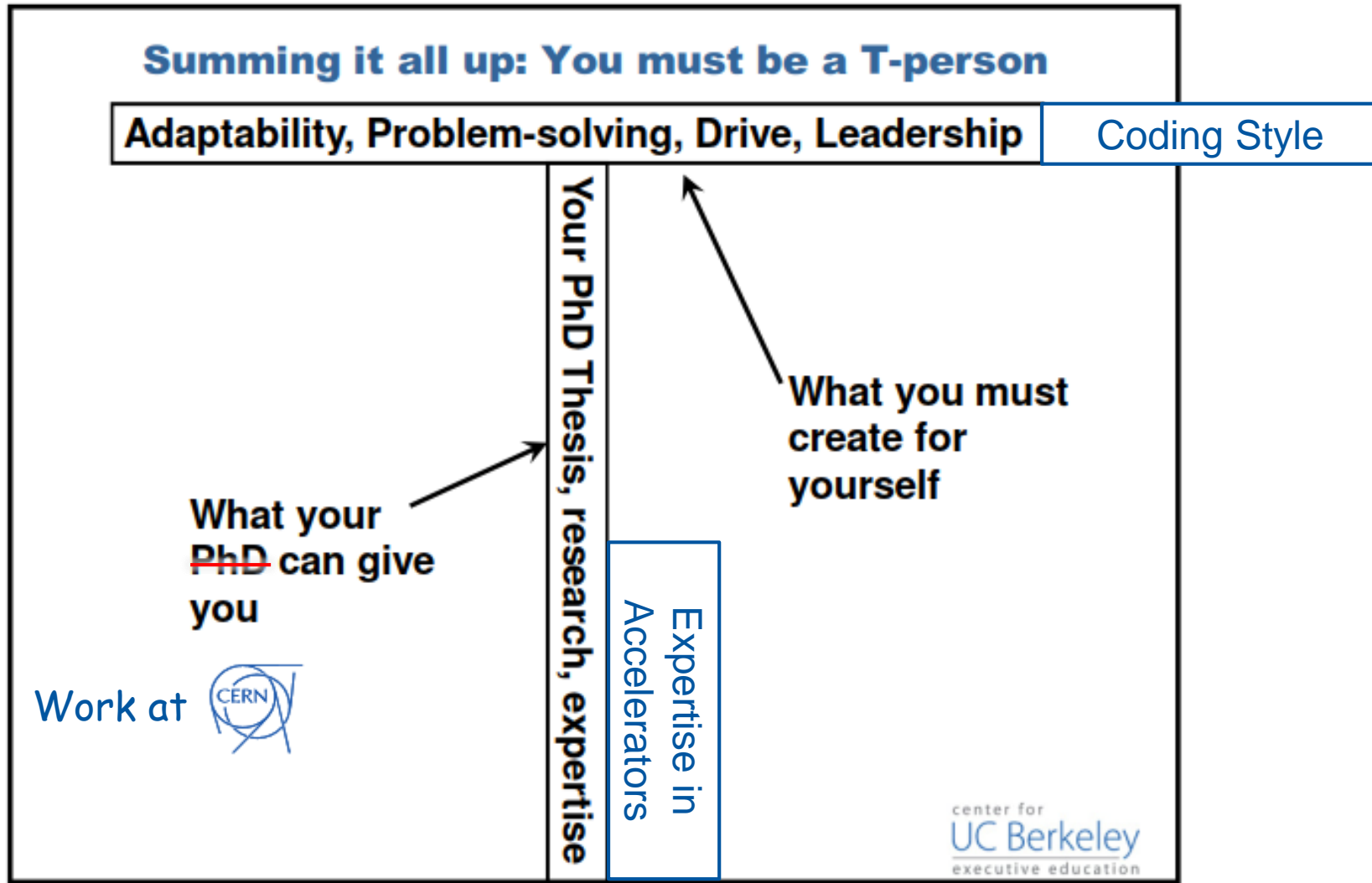
```
In [29]: time[:] = [t - time[0] for t in time]
# Convert from s to h
time[:] = [t/3600 for t in time]
# Create a data frame
df = pd.DataFrame(current, time)
# plot
df.plot()
```

```
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7ffa9d072a58>
```



Several users in our section: Matthieu, Michał, Christoph, Per, Zinur, Akrivi

Be a T-shaped person



Courtesy: Prof. Peter Fiske, UC Berkeley