
UKRI
Science and Technology Facilities Council
Welcome
Image © Haroon Rafique

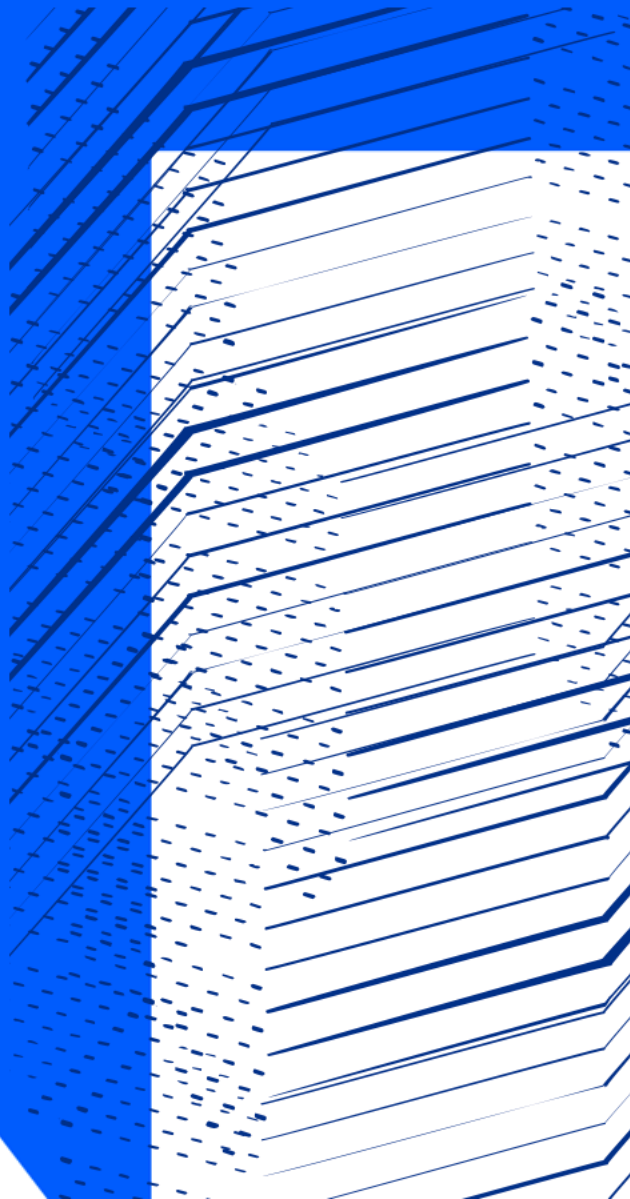
Science and Technology Facilities Council

Accelerator Simulation Framework

Towards end-to-end beam dynamics simulations of the ISIS accelerators

Haroon Rafique

haroon.rafique@stfc.ac.uk



Today’s Journey

0 Introduction

Who’s the new guy and what can he help me with?

1 Common Issues

Saving time. Using each others work. **Doing physics rather than programming.**

2 Approaches

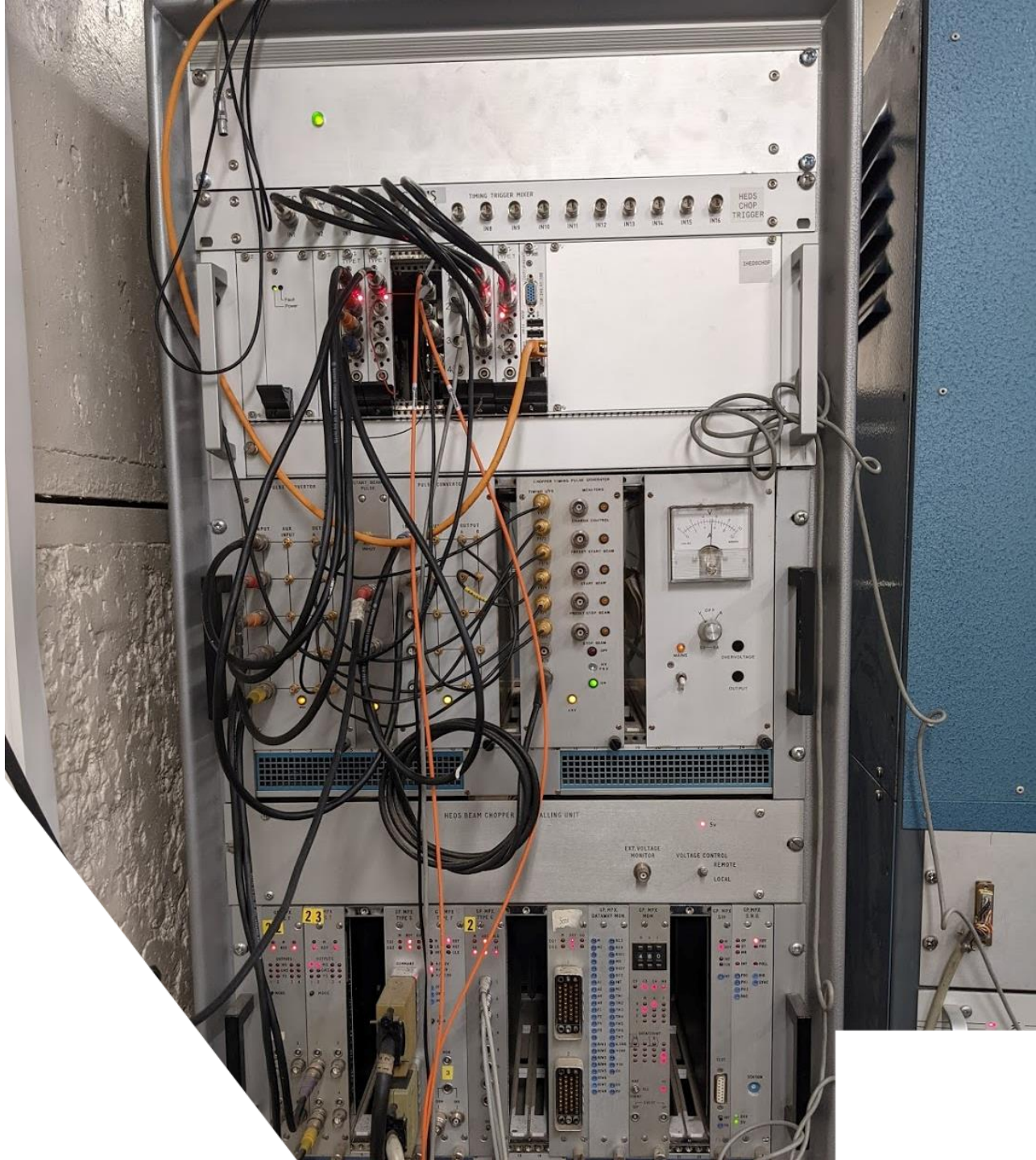
Lessons learnt from personal experience.

3 Opportunities

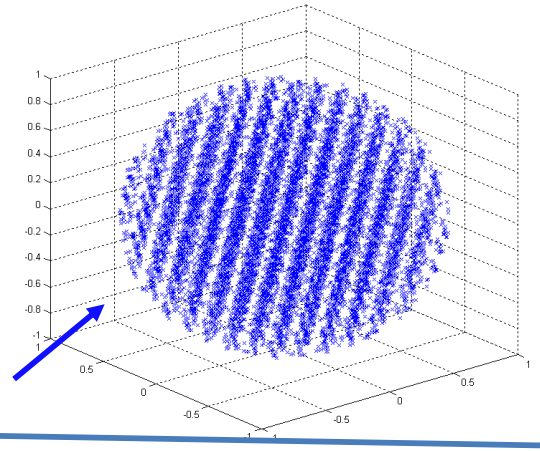
A vision of the future based on the efforts of the past.

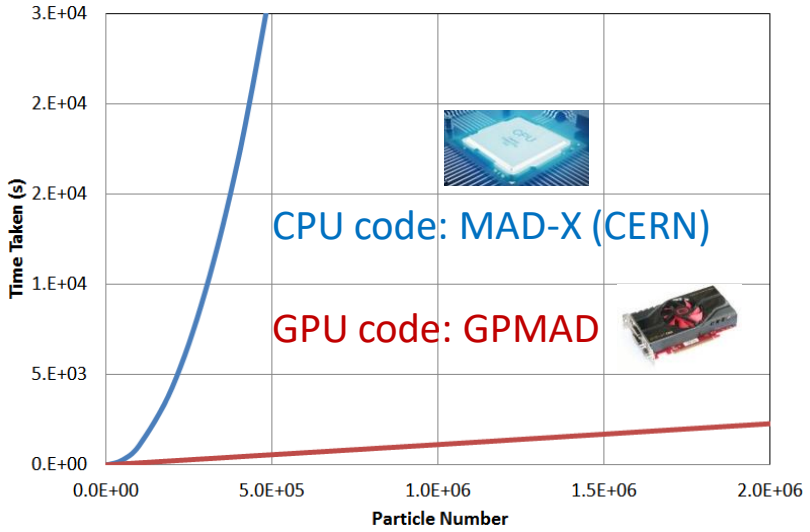
4 Working Together

If we work from the same foundation, we can build a future-proofed, dynamic, and powerful model. Time for debate.



Haroon Rafique





2007-11: MPhys @ Manchester

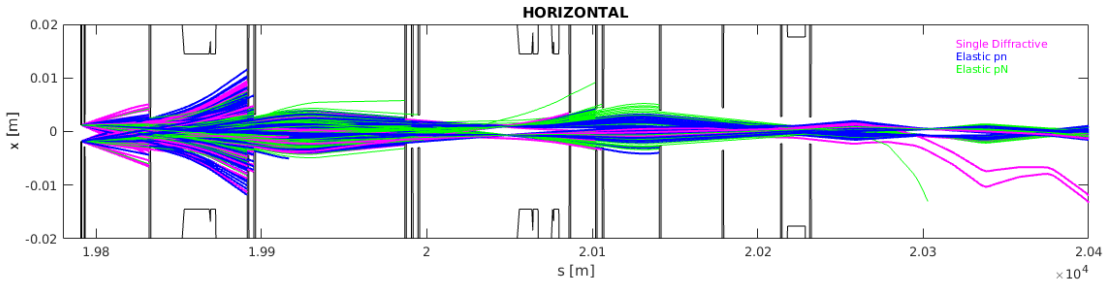
Prof. Rob Appleby, Dr. Hywel Owen

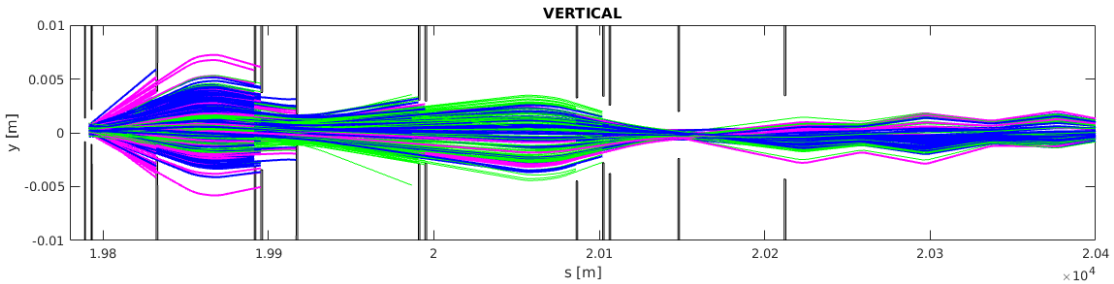
2nd Year theory project: random numbers aren't always random

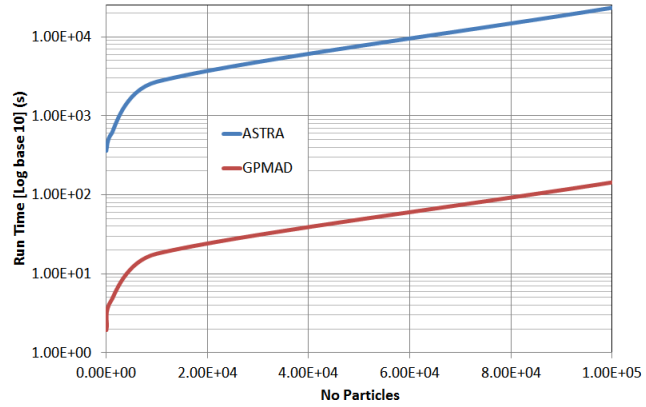
MPhys project: GPUs make particle tracking very fast

Even with space charge (frozen simple)

One of the first scientific uses of CUDA!







2011-16: PhD @ Huddersfield / NGACDT

Prof. Roger Barlow [in collaboration with CERN]

MERLIN for HL-LHC Collimation

Loss maps, novel composite material scattering, hollow electron lens collimation

Thesis published in 2017 because it was too long – keep it short and sweet to save time

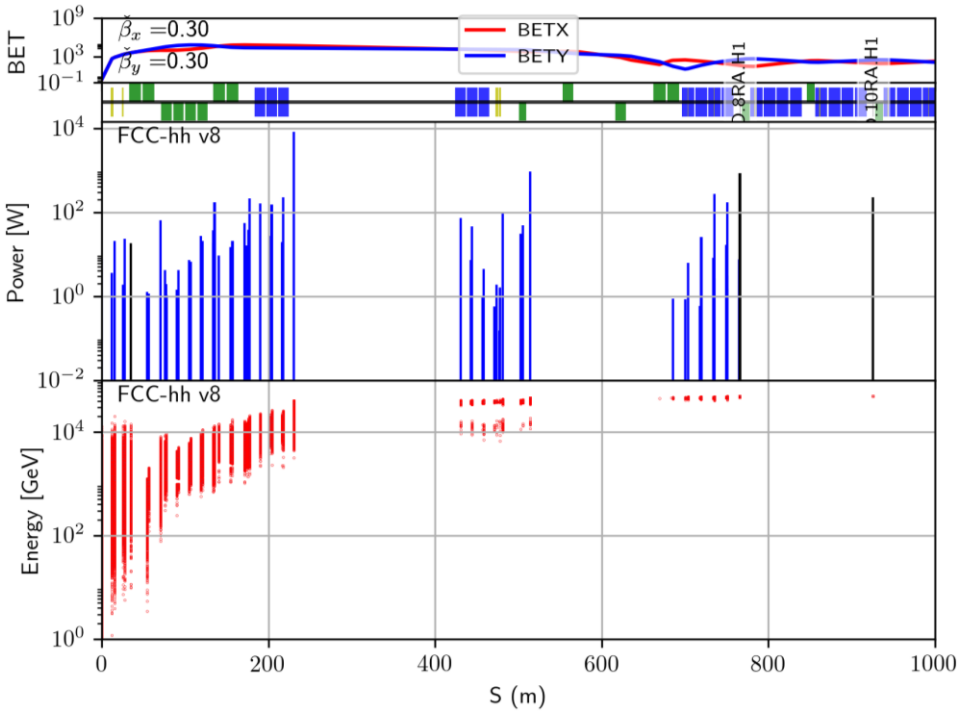
Science and Technology Facilities Council

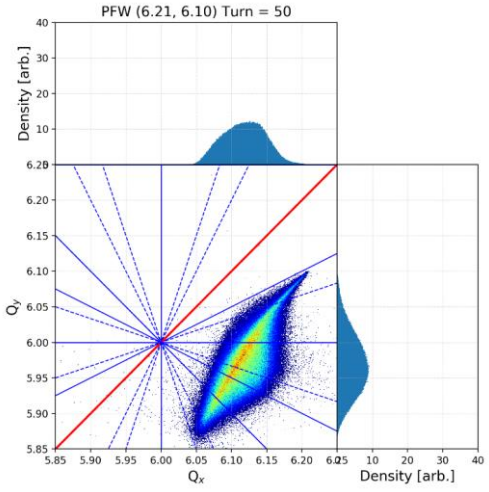
Haroon Rafique

2016-18 PDRA @ Manchester

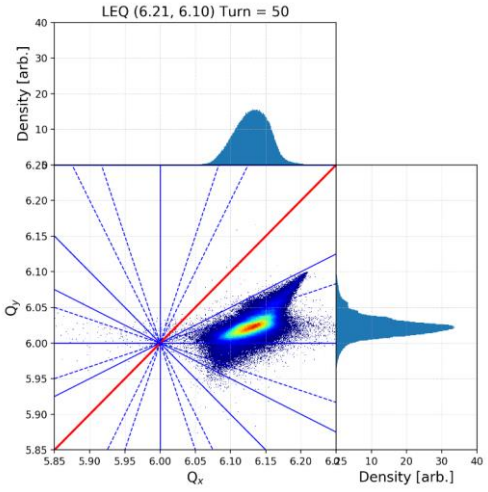
Prof. Rob Appleby [in collaboration with CERN / JAI / EuroCirCol]

FCC-hh 50 TeV collision debris: reduced damage to SC magnets by adding collimators to design (specifications checked in FLUKA)
Muon tracking (analytic & FLUKA) shows no 20 TeV muons reach next detector through standard rock or accelerator tunnel





Control working point with main magnet auxiliary windings



Control working point with low energy quads

2018-20 Senior Fellow @ CERN

Dr. Hannes Bartosik, Dr. Yannis Papaphilipou, Dr. Alex Huschauer

Space Charge @ Proton Synchrotron Injection:
Large horizontal emittance growth between Booster & PS (30-40%)
Implemented infrastructure to perform PTC PyORBIT PIC simulations including:

- Space charge benchmarking campaign with profile measurements
- Measured injection bump tune swing caused by eddy currents
- Measured dispersion mismatch in transfer line
- Initial distribution generated using measurements
- Installation and support on HPC infrastructure (2nd largest HPC user at CERN at >1180 CPU years)

Used to show that **space charge is not expected to be a major contributor to the 30-40% emittance growth**, and possible benefits of alternative tune control in operation. Used to predict performance for upgrades (now in commissioning).

UKRI Science and Technology Facilities Council

Science and Technology Facilities Council

Common Issues

Common to me at least

What do we need when starting a new accelerator project?

Beam

- **Parameters**: Energy, energy spread, charge, intensity etc
- **What does it look like:** 6D Profiles/distributions, measurements etc…

Accelerator Lattice (for operational machine)

- **Sequence of lattice elements:** what is in the accelerator and where
- **Magnet strengths:** constants, ramp functions – what changes and when

Specifics

- **For example:** apertures, operational settings, inputs/outputs for other codes…

Where do we get this information?

People

- **Point me:** provide references
- **Explain to me:** based on experience
  - **Requires: People and their time**

References

- **Documentation:** what year is it from? Is it still relevant? Costs time to read and search for information
- **Examples:** code snippets, simulations etc. Usually require explanation from a person or some time to understand (familiar language etc?)

Blind Search

- **Have a rummage:** look through the shared network drive
  - **Requires: time and access**

Modelling Complex Facilities

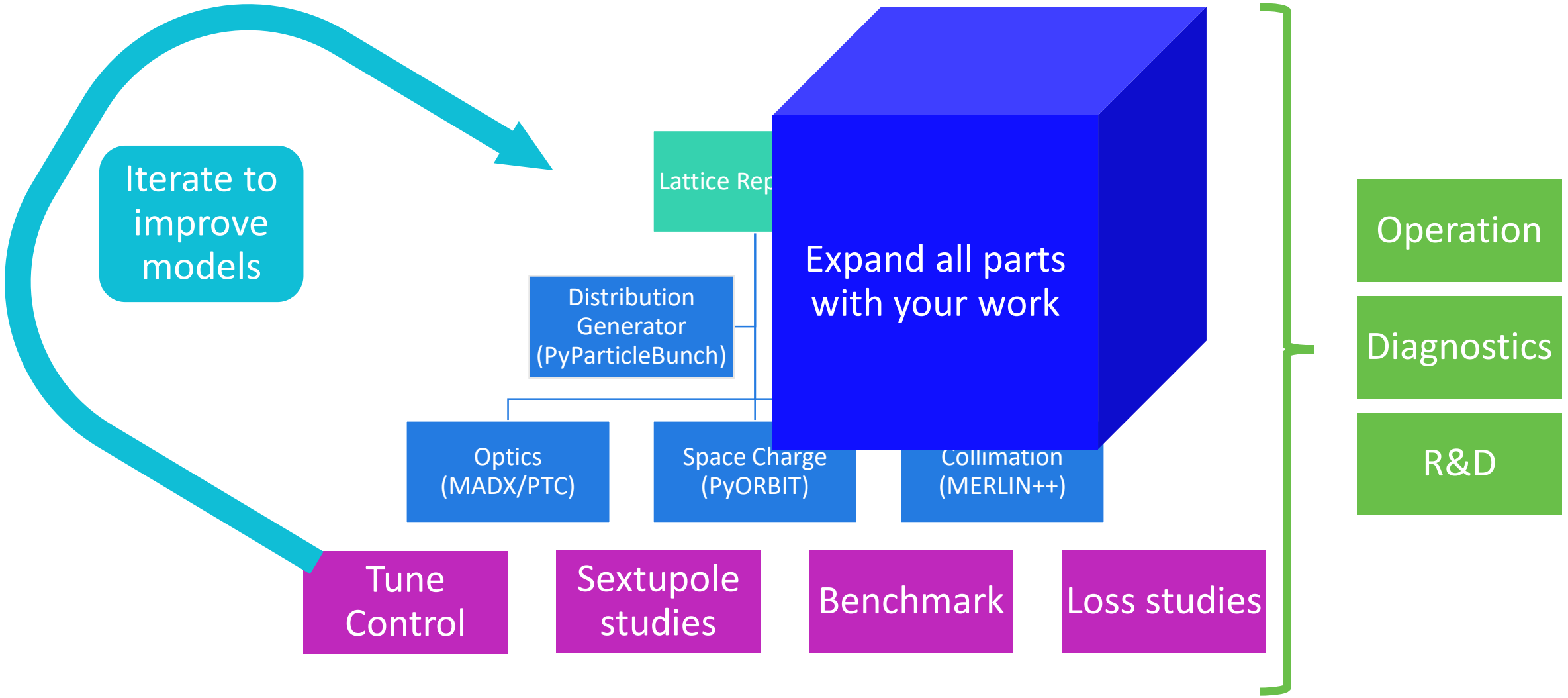
It makes sense to divide modelling into expert areas:

- Ion Source
- RFQ
- **Linac**
- **Transfer lines**
- **Synchrotron**
- Vacuum systems
- Power & **RF systems**
- **Specific systems** (e.g. charge exchange injection)

How do we bring the relevant parts together easily?

Why? We're not that far from an end-to-end ISIS accelerator simulation…

Proposal: Use a shared simulation framework



How do we work together?

Individual models/methods/codes/scripts

- How and where should I **store** important data/information/code/scripts?
- How can I develop my own scripts/code in a way that can be **accessible** to others, avoids **problems with unfinished work, bugs** etc, and **isn't difficult**?

Group framework

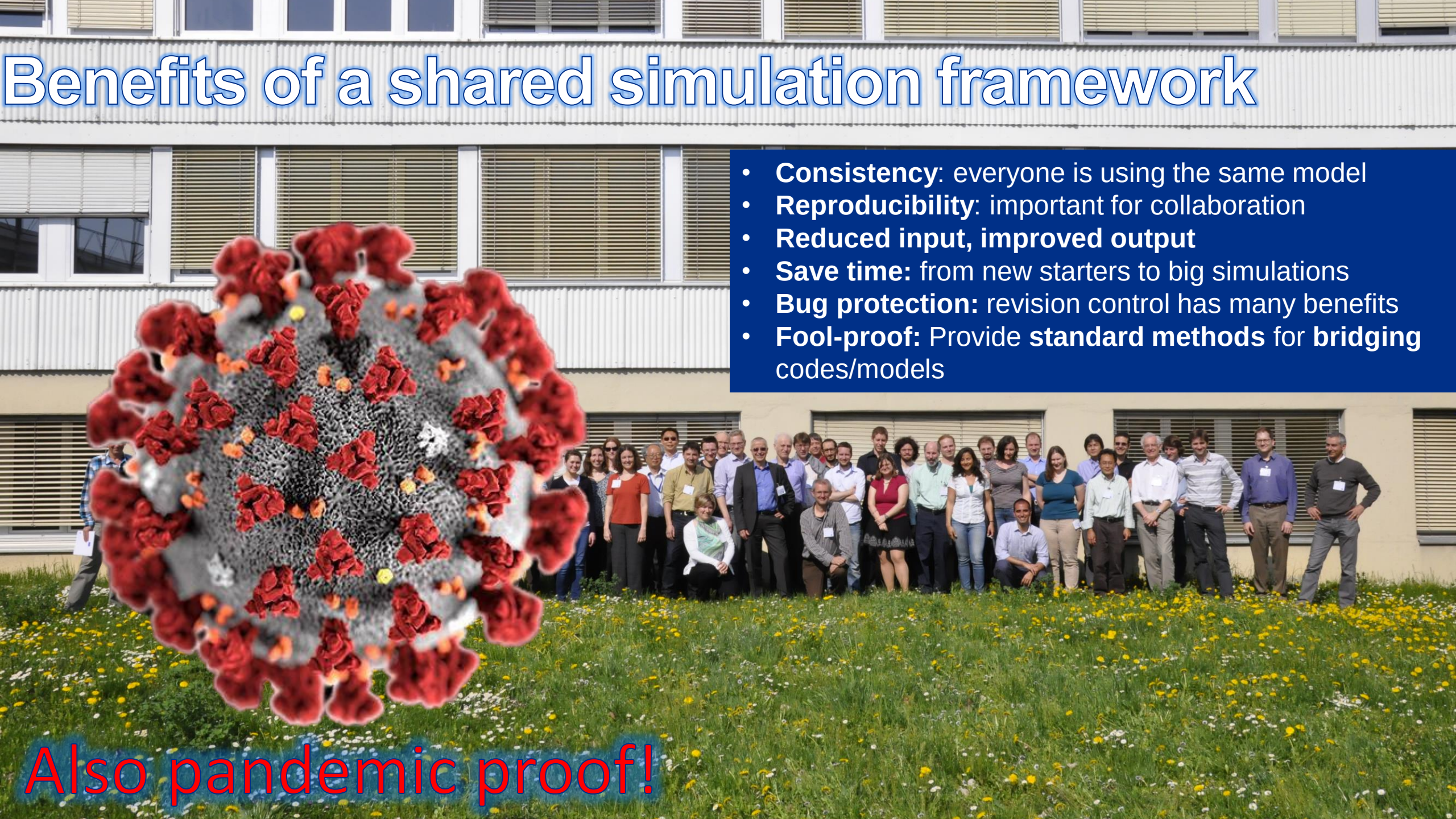
- How should we **structure** everything? Should it be different for **design/operation**?
- What **codes** should we use and **why**? Do we have **alternative codes**?
- How do we all use the same thing but modify our own versions **without headache**?

Group consensus

- Everyone has their **own valid methods**
- The idea is to **share and standardise to make our lives easier** not re-do everything

Benefits of a shared simulation framework

- **Consistency**: everyone is using the same model
- **Reproducibility**: important for collaboration
- **Reduced input, improved output**
- **Save time:** from new starters to big simulations
- **Bug protection:** revision control has many benefits
- **Fool-proof:** Provide **standard methods** for **bridging** codes/models



Also pandemic proof!

Science and Technology Facilities Council

Approaches

How do we resolve common issues?

Start with a strong foundation

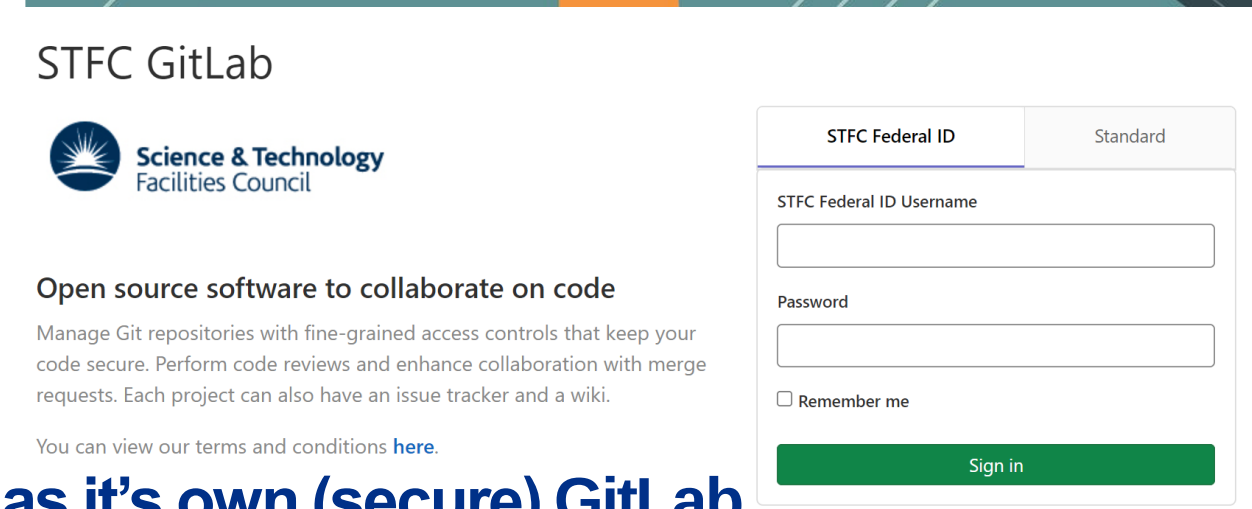
Git is software for **tracking changes in any set of files**, usually used for **coordinating work** among programmers **collaboratively developing** source code during software development.
Its goals include **speed, data integrity, and support for distributed, non-linear workflows**.

GitHub: Microsoft owned public service
GitLab: Open core company offered service (has more advanced features)

HR's GitHub: https://github.com/HaroonRafique

HR's STFC GitLab: https://gitlab.stfc.ac.uk/mmf67433

See the following talk for more details on using Git

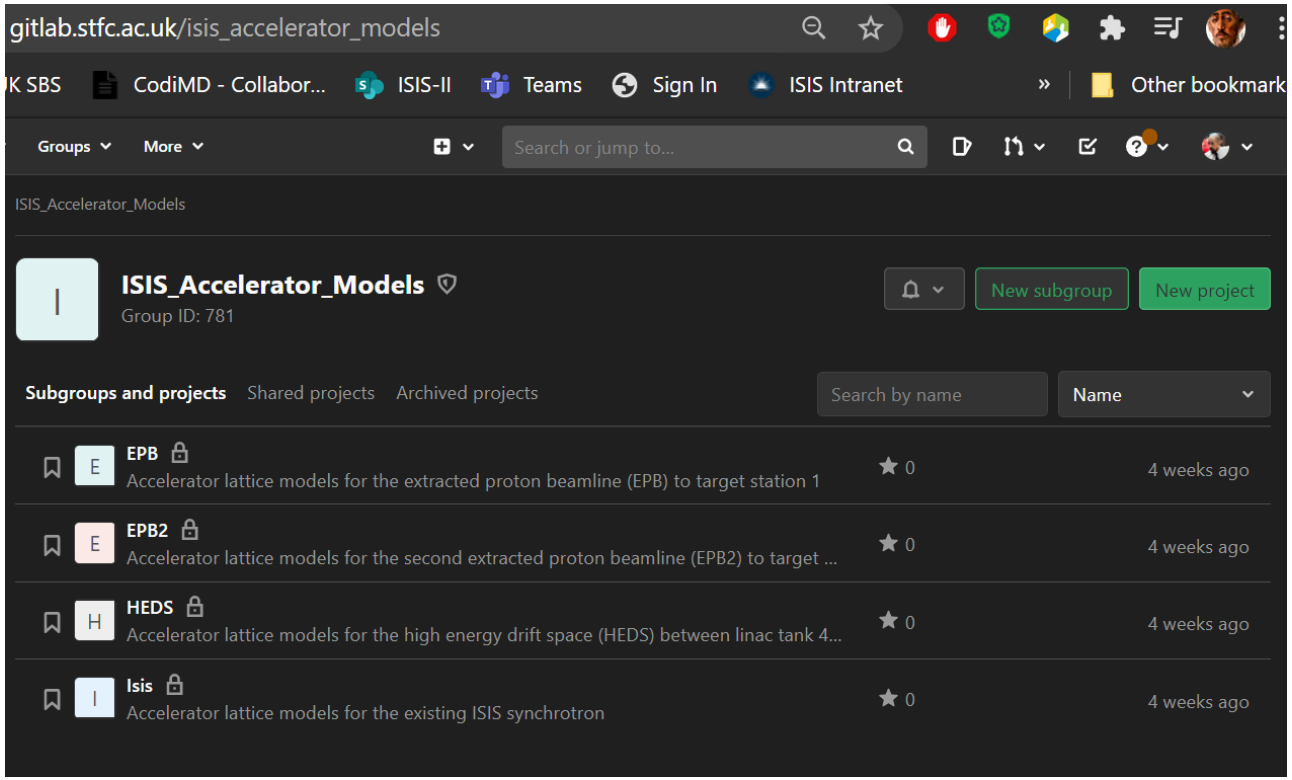


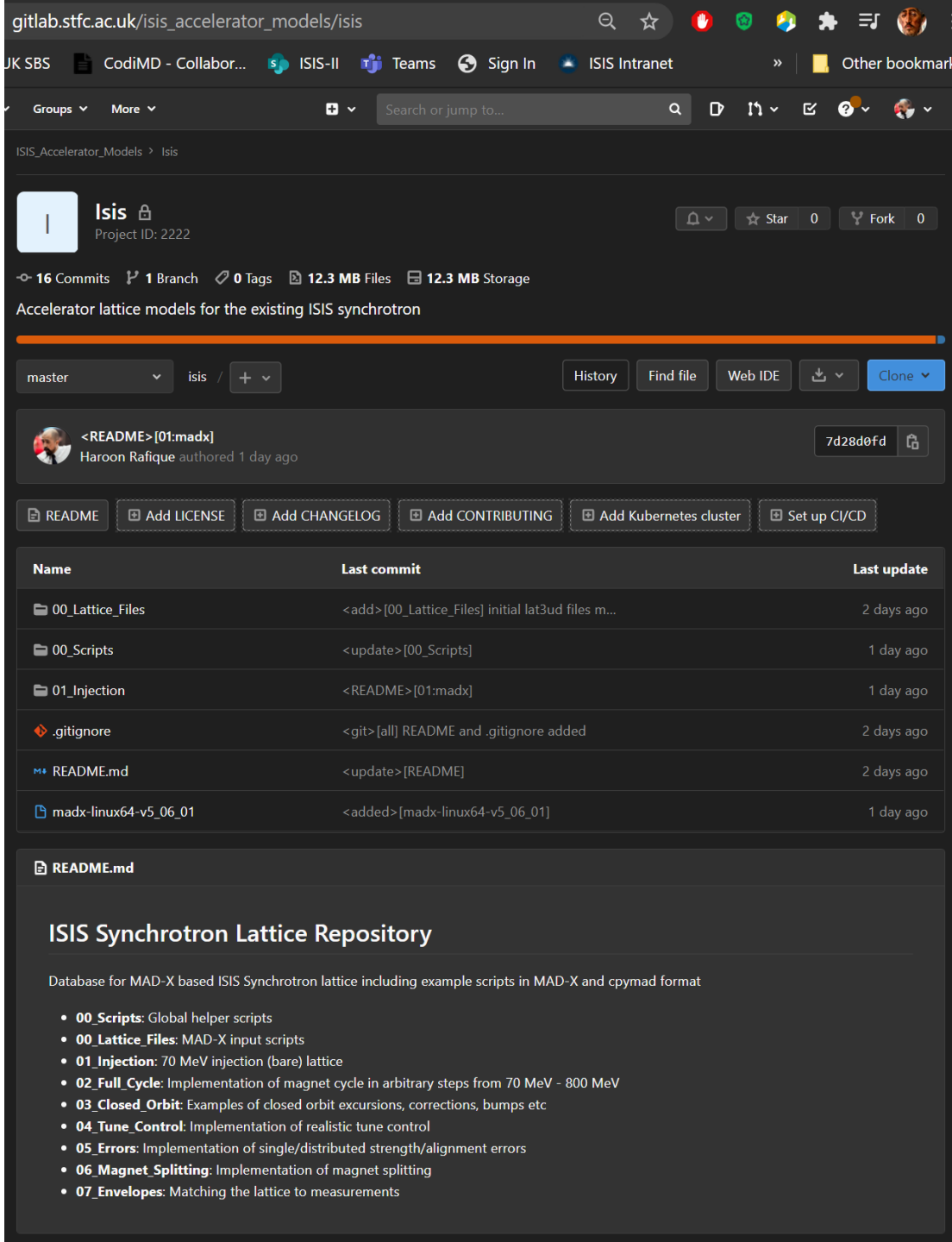
STFC has it's own (secure) GitLab

Science and Technology Facilities Council

ISIS Accelerator Lattice Repository

https://gitlab.stfc.ac.uk/isis_accelerator_models (full tree)
https://gitlab.stfc.ac.uk/isis_accelerator_models/isis





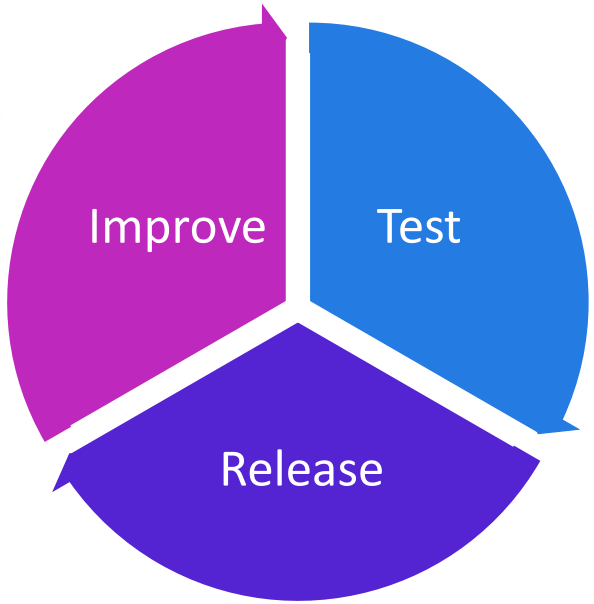
Science and Technology Facilities Council

Note use of markdown for simple documentation:
https://www.markdownguide.org/basic-syntax/

ISIS Accelerator Lattice Repository: How to use

0 Request access on the STFC GitLab

Use secure internal GitLab for work, in future we can copy things to Github for ease of access/teaching etc



1 Clone the repository onto your machine

Create a local copy for you to use

2 Run the examples to check you have the requirements

Python, ipython notebook, cpymad, tfs, madx, etc

3a Basic uses (if you add/upgrade features consider sharing):

- Generate MAD-X/PTC format inputs for another code (e.g. PyORBIT)
- Look at optics changes; closed orbit errors, gradient errors etc…

3b Advanced uses (fork/branch the code to create your own copy first):

- Implement your fancy scripts, where possible breaking them down into universal scripts we can all use – ONLY work that is relevant to the lattice repository (MAD – related lattice modifications).
- Example: Pete's tune control mapping, Hayley's optics measurements, … (I'll help of course)

4 Create a pull request (pull your work into the main branch for everyone)

Short and sweet: Your modifications, why they are needed, your proof that they don't break everything and work as expected

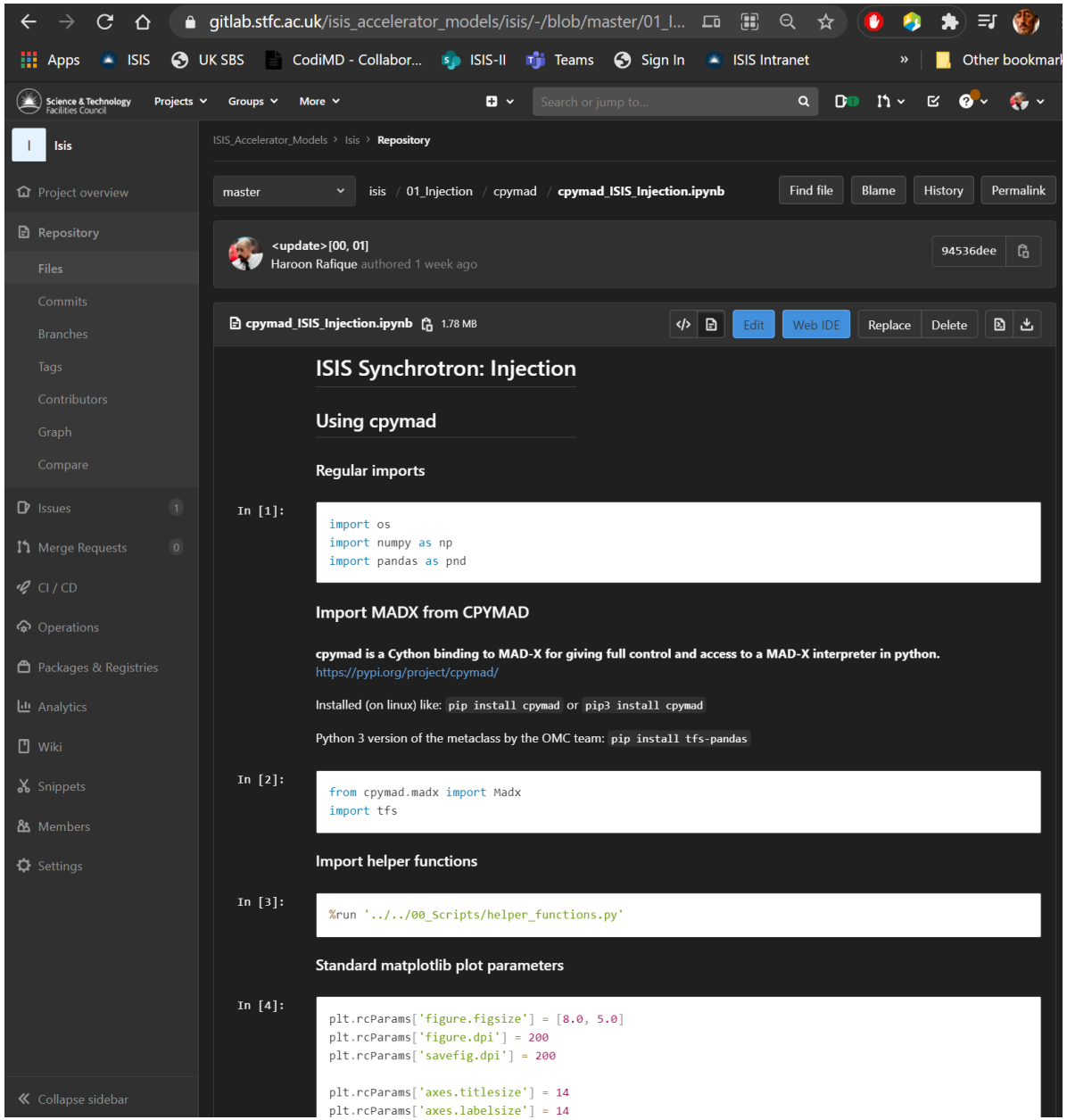
ISIS Accelerator Lattice Repository: cpymad example

Folder:
https://gitlab.stfc.ac.uk/isis_accelerator_models/isis/-/blob/master/01_Injection/cpymad/

iPython Notebook:
https://gitlab.stfc.ac.uk/isis_accelerator_models/isis/-/blob/master/01_Injection/cpymad/cpymad_ISIS_Injection.ipynb

Useful link: Jupyter Online Notebook Viewer (note this won't work with STFC GitLab notebooks as they aren't public):
https://nbviewer.jupyter.org/

Time for an interactive example…



UKRI Science and Technology Facilities Council

ISIS Accelerator Lattice Repository: Ideas

Optics:

Static:

- Generate matched lattice for given parameters/constraints
- Investigate closed orbit errors/correction
- Investigate gradient errors/correction
- Benchmark lattice models using measurements

Dynamic:

- Generate magnet ramping tables
- Generate matched lattice for given parameters/constraints over time

Operation:

- Improved and up to date model for apps such as LOCO
- Starting point for implementing corrections such as Pete's advanced tune controls

Simulation:

- Standard methods to produce MADX/PTC etc outputs required as input to other codes
- Use together with 'accelerator beam distribution' repository to perform bunch tracking in MAD-X and PTC

Lattice Repository: How to start developing

1. Sign in to create an STFC GitLab account: https://gitlab.stfc.ac.uk/

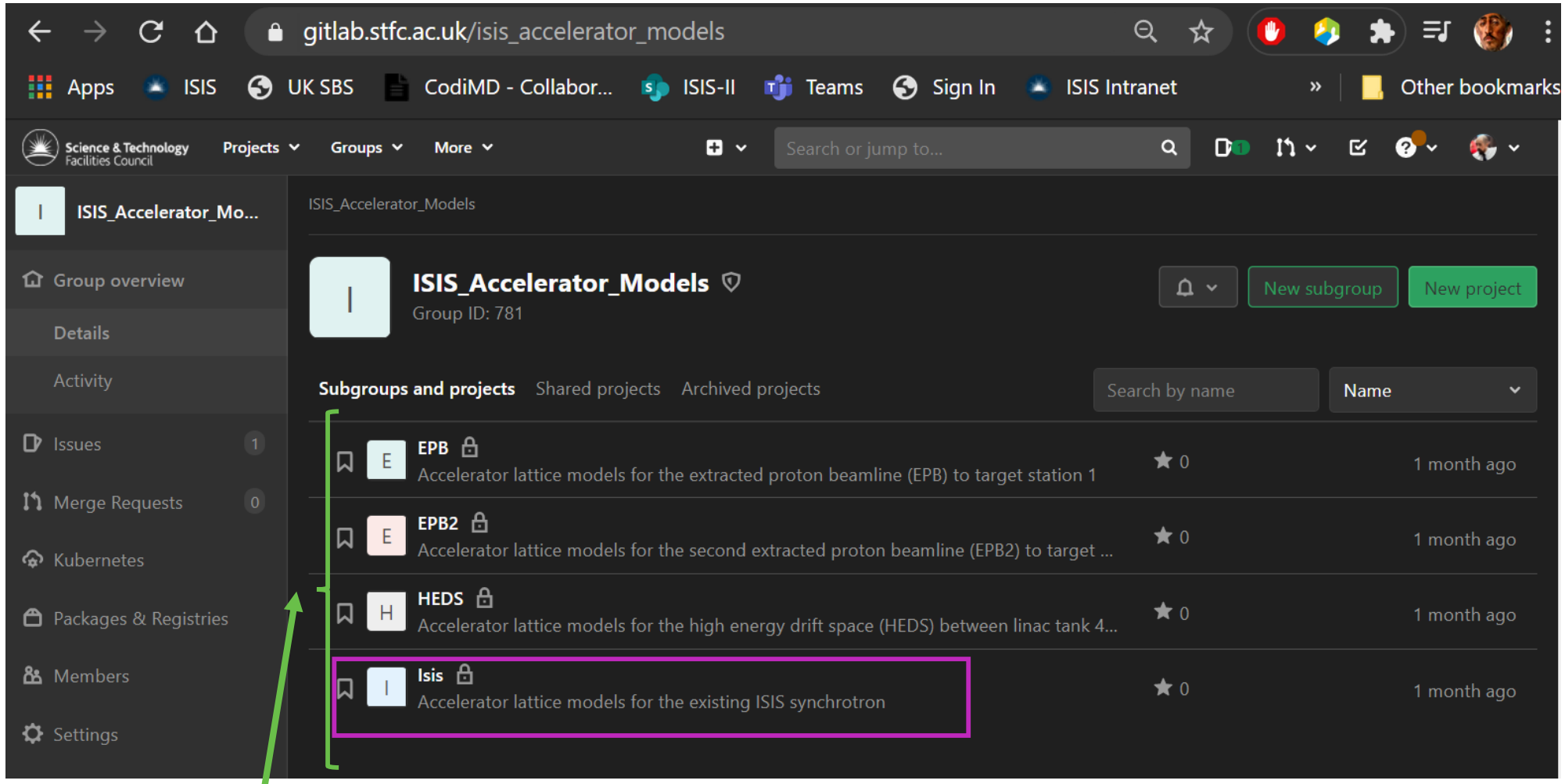
2. Request access to the Project: https://gitlab.stfc.ac.uk/isis_accelerator_models

3. Get stuck in using the issues boards:
   - General lattice repo stuff: https://gitlab.stfc.ac.uk/groups/isis_accelerator_models/-/boards
   - Specific ISIS synchrotron repo stuff: https://gitlab.stfc.ac.uk/isis_accelerator_models/isis/-/boards/80

4. Assign tasks to yourself and start forking/branching/pushing/pulling etc!

Lattice Repository: Group Page

We start at Details



Let's take a look at the ISIS synchrotron lattice repository

Each lattice has it's own repository

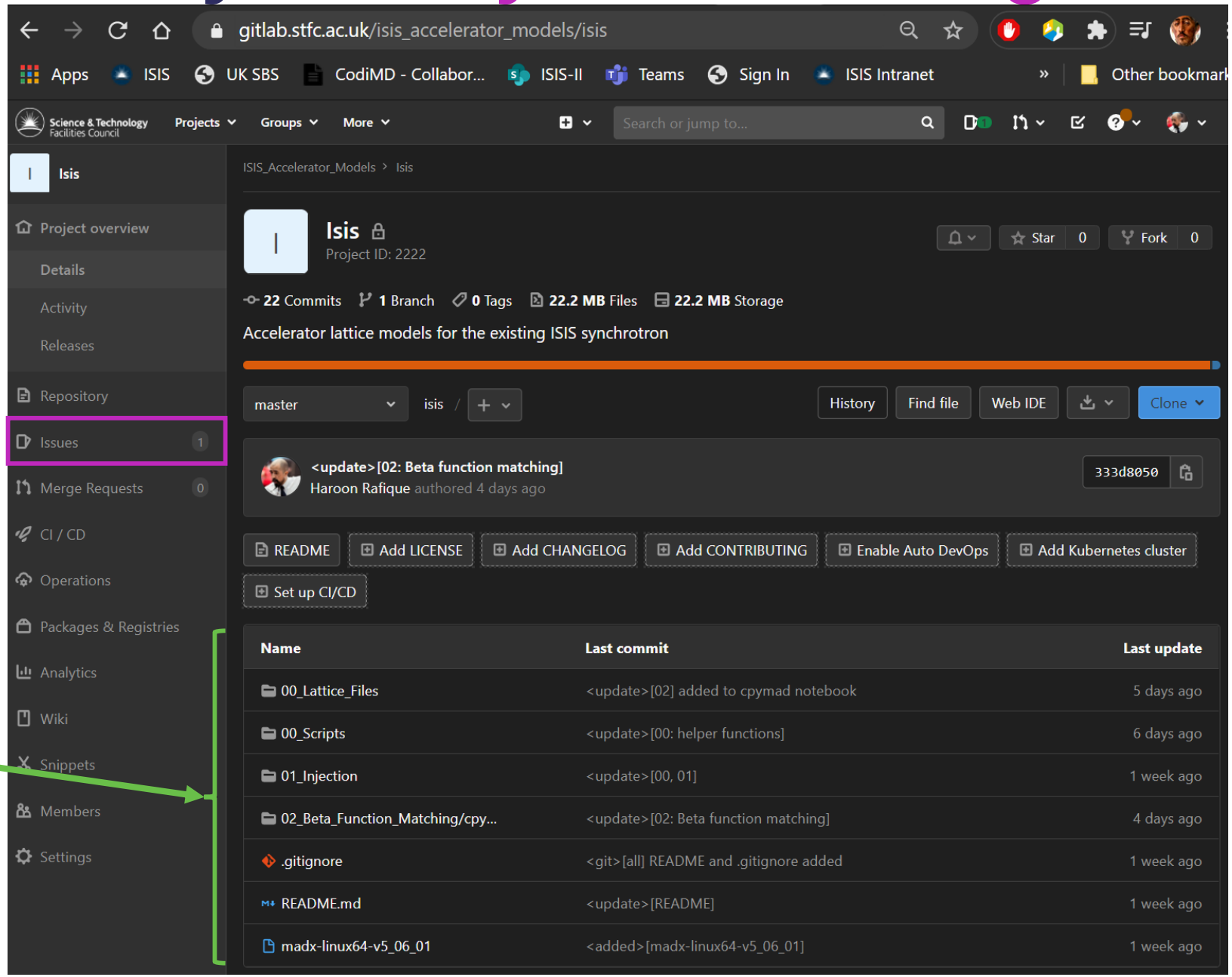
Science and Technology Facilities Council

Lattice Repository: ISIS Synchrotron Page

We start at Details

Let's take a look at the issue board

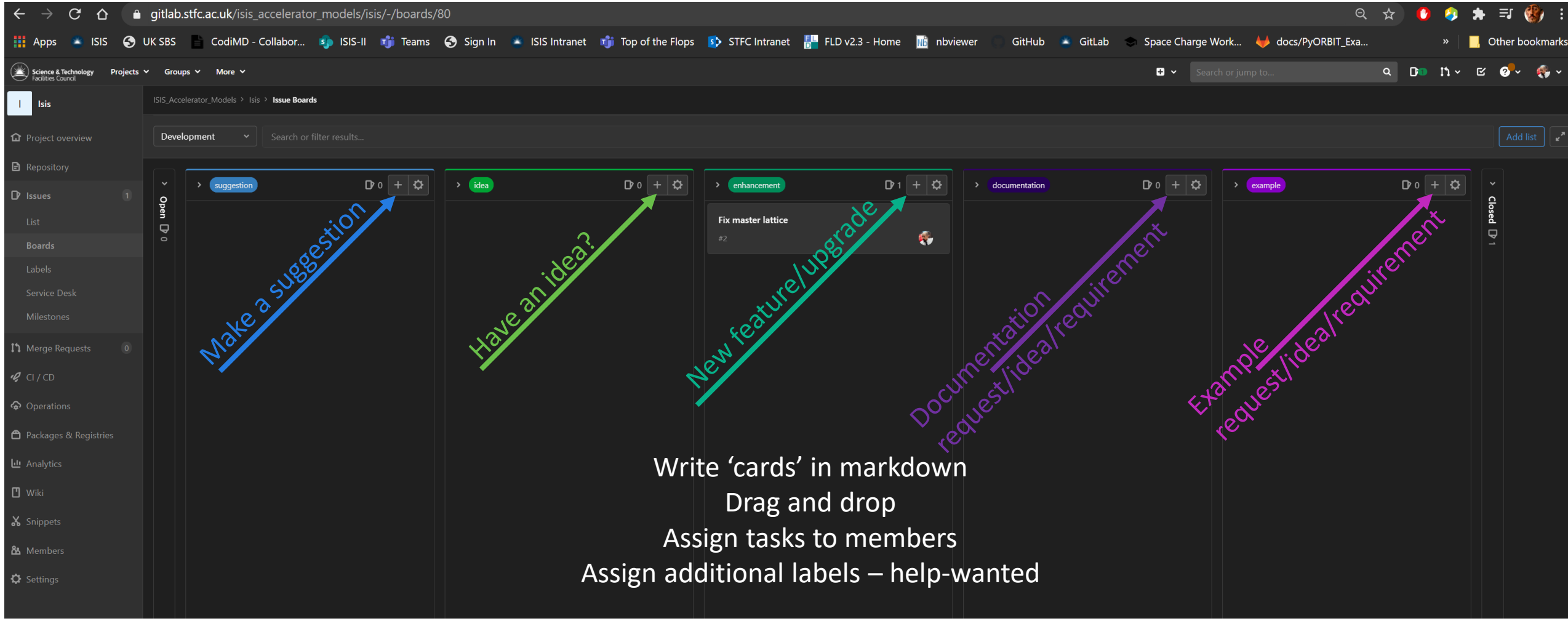
File contents of the repository



Markdown README under this

Science and Technology Facilities Council

Lattice Repository: ISIS Synchrotron Issue Board



Write ‘cards’ in markdown
Drag and drop
Assign tasks to members
Assign additional labels – help-wanted

Science and Technology Facilities Council

Lattice Repository: Labels

Labels can be applied to issues and merge requests.
Star a label to make it a priority label. Order the prioritized labels to change their relative priority, by dragging.

Prioritized Labels

idea	A thought or suggestion as to a possible course of action ISIS_Accelerator_Models	Issues · Merge requests · Prioritized label	Unsubscribe
enhancement	A new feature or improvement to an existing feature ISIS_Accelerator_Models	Issues · Merge requests · Prioritized label	Unsubscribe
bug	Something is not behaving as expected ISIS_Accelerator_Models	Issues · Merge requests · Prioritized label	Unsubscribe
example	Please provide an example of how to use this ISIS_Accelerator_Models	Issues · Merge requests · Prioritized label	Unsubscribe
documentation	Please can you explain how this works ISIS_Accelerator_Models	Issues · Merge requests · Prioritized label	Unsubscribe

Lattice Repository: Labels

Other Labels

discussion	This is up for debate ISIS_Accelerator_Models	Issues · Merge requests	Unsubscribe
help-wanted	I need help with something ISIS_Accelerator_Models	Issues · Merge requests	Unsubscribe
Level: critical	Needs immediate attention: main features are affected ISIS_Accelerator_Models	Issues · Merge requests	Unsubscribe
Level: low	Problem does not affect code features: e.g. incomplete documentation ISIS_Accelerator_Models	Issues · Merge requests	Unsubscribe
Level: moderate	Problem only affects non-critical features or extended functions: e.g. change in external code requires update for extra features of external code ISIS_Accelerator_Models	Issues · Merge requests	Unsubscribe
suggestion	Maybe this would work better a different way. Maybe not ISIS_Accelerator_Models	Issues · Merge requests	Unsubscribe

Accelerator Simulation Framework Project

github.com/users/HaroonRafique/projects/1

Apps | ISIS | UK SBS | CodiMD - Collabor... | ISIS-II | Teams | Sign In | ISIS Intranet | Top of the Flops | STFC Intranet | FLD v2.3 - Home | nbviewer | GitHub | GitLab | Space Charge Work... | docs/PyORBIT_Exa... | Other bookmarks

Search or jump to... / Pull requests Issues Marketplace Explore

HaroonRafique

Repositories 27 | Projects 1

Accelerator_Simulation_Framework

Updated 5 minutes ago

Filter cards

8 Idea

- PyParticleBunch::Formats::PTC — Added by HaroonRafique
- PyParticleBunch::Formats::MADX — Added by HaroonRafique
- PyParticleBunch::Distributions::Poincare — Added by HaroonRafique
- PyParticleBunch::Distributions::Uniform — Added by HaroonRafique
- PyParticleBunch::Distributions::Joho — Added by HaroonRafique
- PyParticleBunch::Distributions::Waterbag — Added by HaroonRafique
- PyParticleBunch::Distributions::KV — Added by HaroonRafique
- PyParticleBunch::Distributions::Gaussian — Added by HaroonRafique

2 To do

- Examples: — 0 of 4 — ParticleBunch#3 opened by HaroonRafique — bug, enhancement, help wanted
- Generate a particle distribution — 0 of 9 — ParticleBunch#1 opened by HaroonRafique — bug, enhancement, help wanted

Automated as To do — Manage

3 In progress

- Document physics processes behind particle generation and normalisation used in the code: — 0 of 3 — ParticleBunch#2 opened by HaroonRafique — bug, enhancement, help wanted
- Create Git usage instructions — Accelerator_Simulation_Framework#2 opened by HaroonRafique — documentation, good first issue, help wanted
- Create a Git style guide — 0 of 1 — Accelerator_Simulation_Framework#1 opened by HaroonRafique — documentation, good first issue, help wanted

Automated as In progress — Manage

0 Done

Automated as Done — Manage

+ Add column

Markdown let's us make checklists

Github (public) because:

- PyORBIT/MERLIN etc on GitHub
- Doesn't contain sensitive code
- Easier to access

Let's take a look at the full issue page

Create Git usage instructions #2

Opened in HaroonRafique/Accelerator_Simulation_Framework

HaroonRafique commented 2 minutes ago

Browser

- [ ] Edit markdown files

Linux

- [ ] Basic Git commands
- [ ] Advanced IGit commands

Windows

- [ ] Git GUI instructions ?

Show less — 0

Assignees: HaroonRafique

Labels: documentation, good first issue, help wanted

Projects: Accelerator_Simulation_Framework — In progress

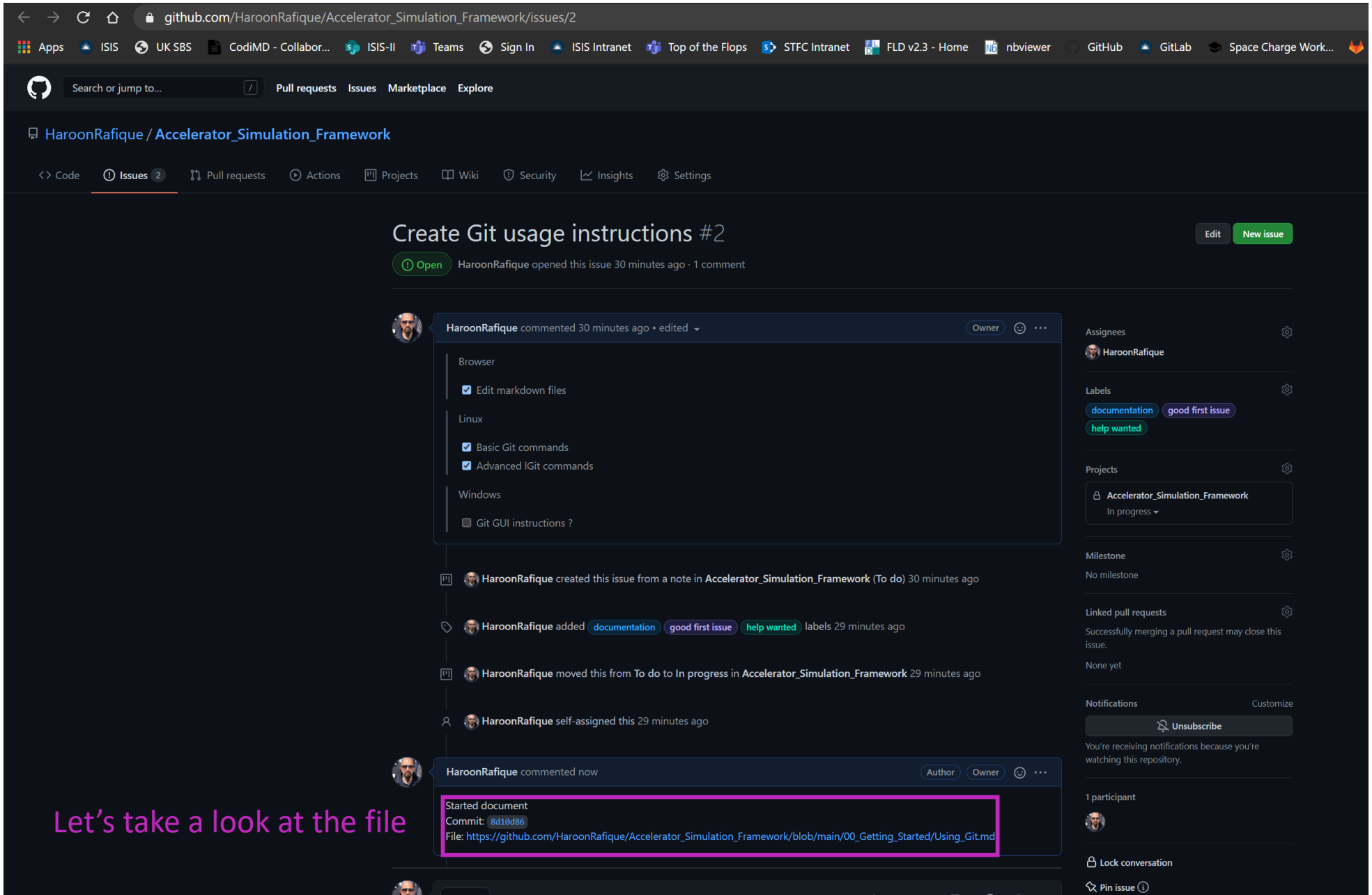
Milestone: No milestone

Go to issue for full details

Close issue

Facilities Council

Accelerator Simulation Framework: Git usage instructions issue

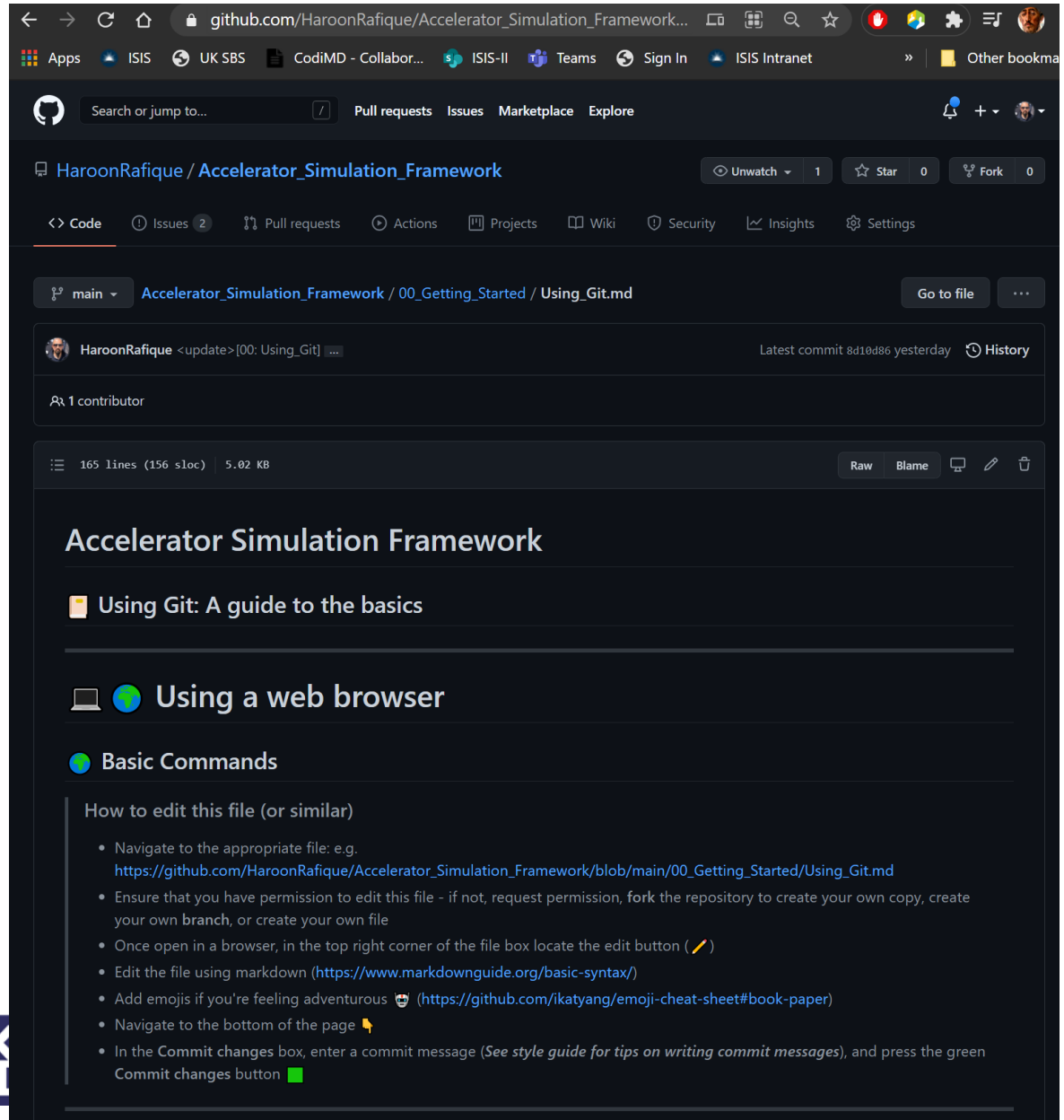


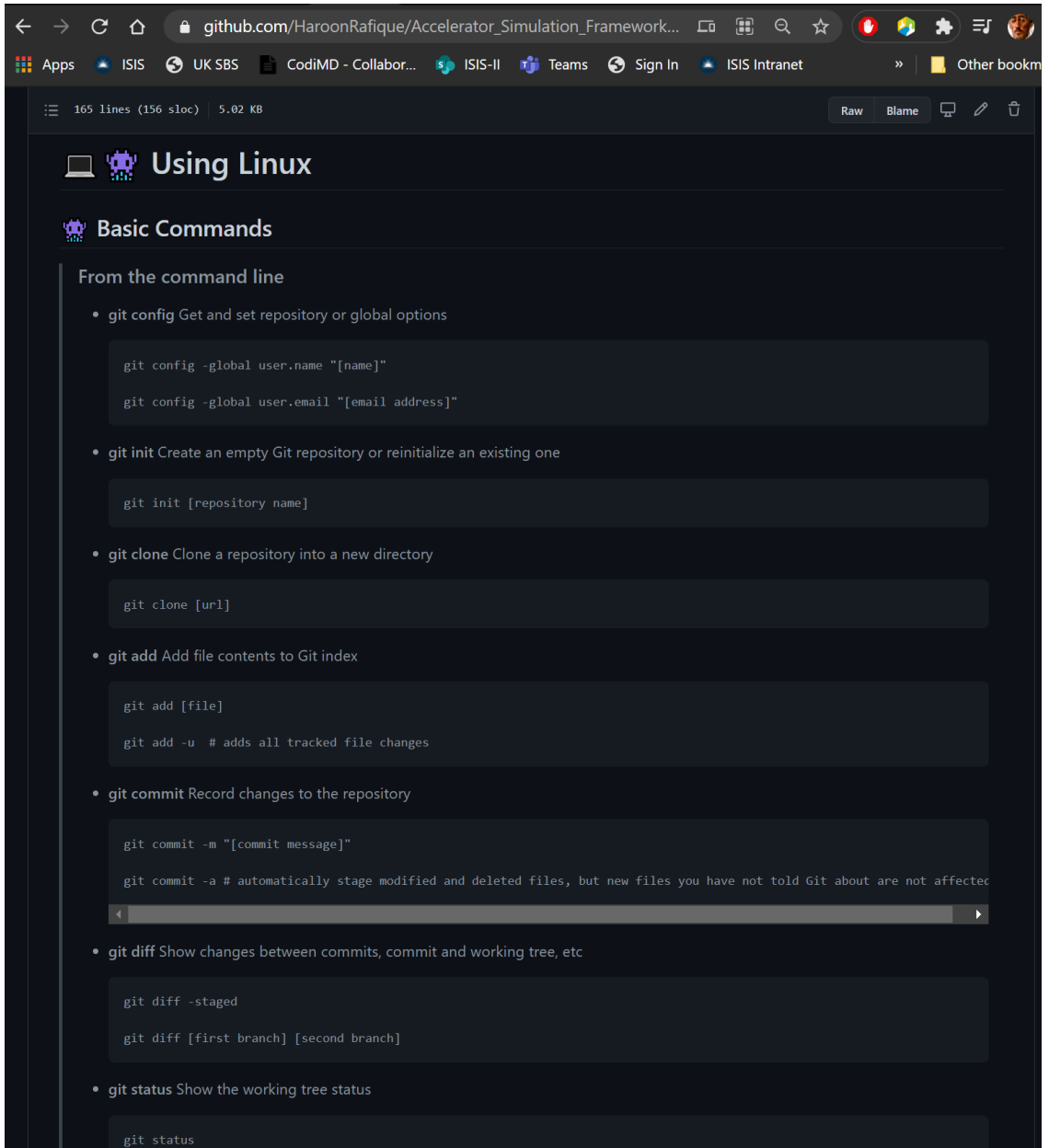
Let's take a look at the file

Science and Technology Facilities Council

Accelerator Simulation Framework: Git usage instructions

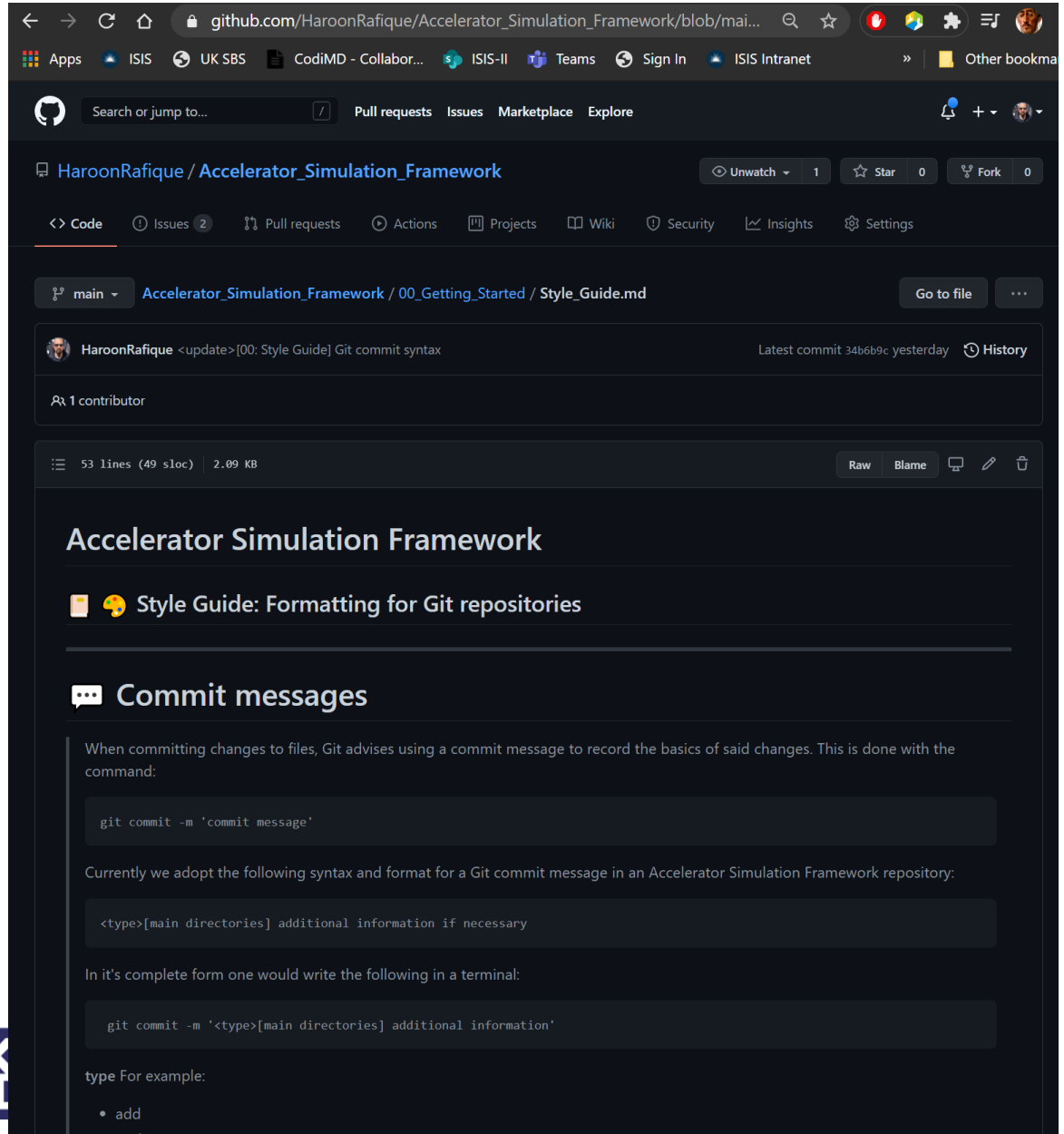
https://github.com/HaroonRafique/Accelerator_Simulation_Framework/blob/main/00_Getting_Started/Using_Git.md





Accelerator Simulation Framework: Git style guide

https://github.com/HaroonRafique/Accelerator_Simulation_Framework/blob/main/00_Getting_Started/Style_Guide.md



github.com/HaroonRafique/Accelerator_Simulation_Framework/blob/mai...

Apps ISIS UK SBS CodiMD - Collabor... ISIS-II Teams Sign In ISIS Intranet Other bookma

Search or jump to... Pull requests Issues Marketplace Explore

HaroonRafique / Accelerator_Simulation_Framework

Unwatch 1 Star 0 Fork 0

Code Issues 2 Pull requests Actions Projects Wiki Security Insights Settings

main Accelerator_Simulation_Framework / 00_Getting_Started / Style_Guide.md Go to file

HaroonRafique <update>[00: Style Guide] Git commit syntax Latest commit 34b6b9c yesterday History

1 contributor

53 lines (49 sloc) 2.09 KB Raw Blame

Accelerator Simulation Framework

Style Guide: Formatting for Git repositories

Commit messages

When committing changes to files, Git advises using a commit message to record the basics of said changes. This is done with the command:

```
git commit -m 'commit message'
```

Currently we adopt the following syntax and format for a Git commit message in an Accelerator Simulation Framework repository:

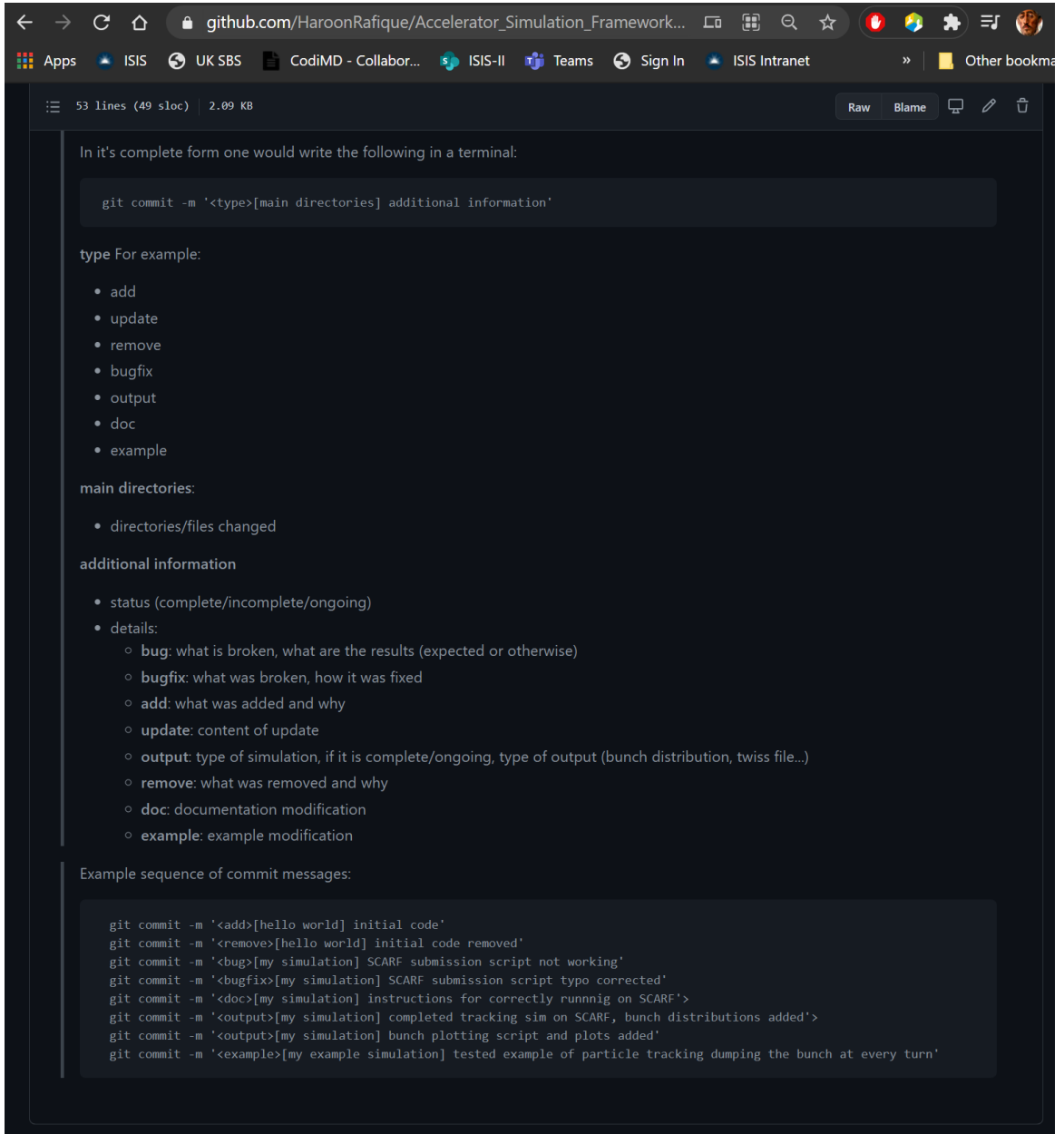
```
<type>[main directories] additional information if necessary
```

In it's complete form one would write the following in a terminal:

```
git commit -m '<type>[main directories] additional information'
```

type For example:

- add



github.com/HaroonRafique/Accelerator_Simulation_Framework...

Apps ISIS UK SBS CodiMD - Collabor... ISIS-II Teams Sign In ISIS Intranet Other bookma

53 lines (49 sloc) 2.09 KB Raw Blame

In it's complete form one would write the following in a terminal:

```
git commit -m '<type>[main directories] additional information'
```

type For example:

- add
- update
- remove
- bugfix
- output
- doc
- example

main directories:

- directories/files changed

additional information

- status (complete/incomplete/ongoing)
- details:
  - **bug**: what is broken, what are the results (expected or otherwise)
  - **bugfix**: what was broken, how it was fixed
  - **add**: what was added and why
  - **update**: content of update
  - **output**: type of simulation, if it is complete/ongoing, type of output (bunch distribution, twiss file...)
  - **remove**: what was removed and why
  - **doc**: documentation modification
  - **example**: example modification

Example sequence of commit messages:

```
git commit -m '<add>[hello world] initial code'
git commit -m '<remove>[hello world] initial code removed'
git commit -m '<bug>[my simulation] SCARF submission script not working'
git commit -m '<bugfix>[my simulation] SCARF submission script typo corrected'
git commit -m '<doc>[my simulation] instructions for correctly runnnig on SCARF'>
git commit -m '<output>[my simulation] completed tracking sim on SCARF, bunch distributions added'>
git commit -m '<output>[my simulation] bunch plotting script and plots added'
git commit -m '<example>[my example simulation] tested example of particle tracking dumping the bunch at every turn'
```

Science and Technology Facilities Council

Opportunities

What could we do with a shared framework?

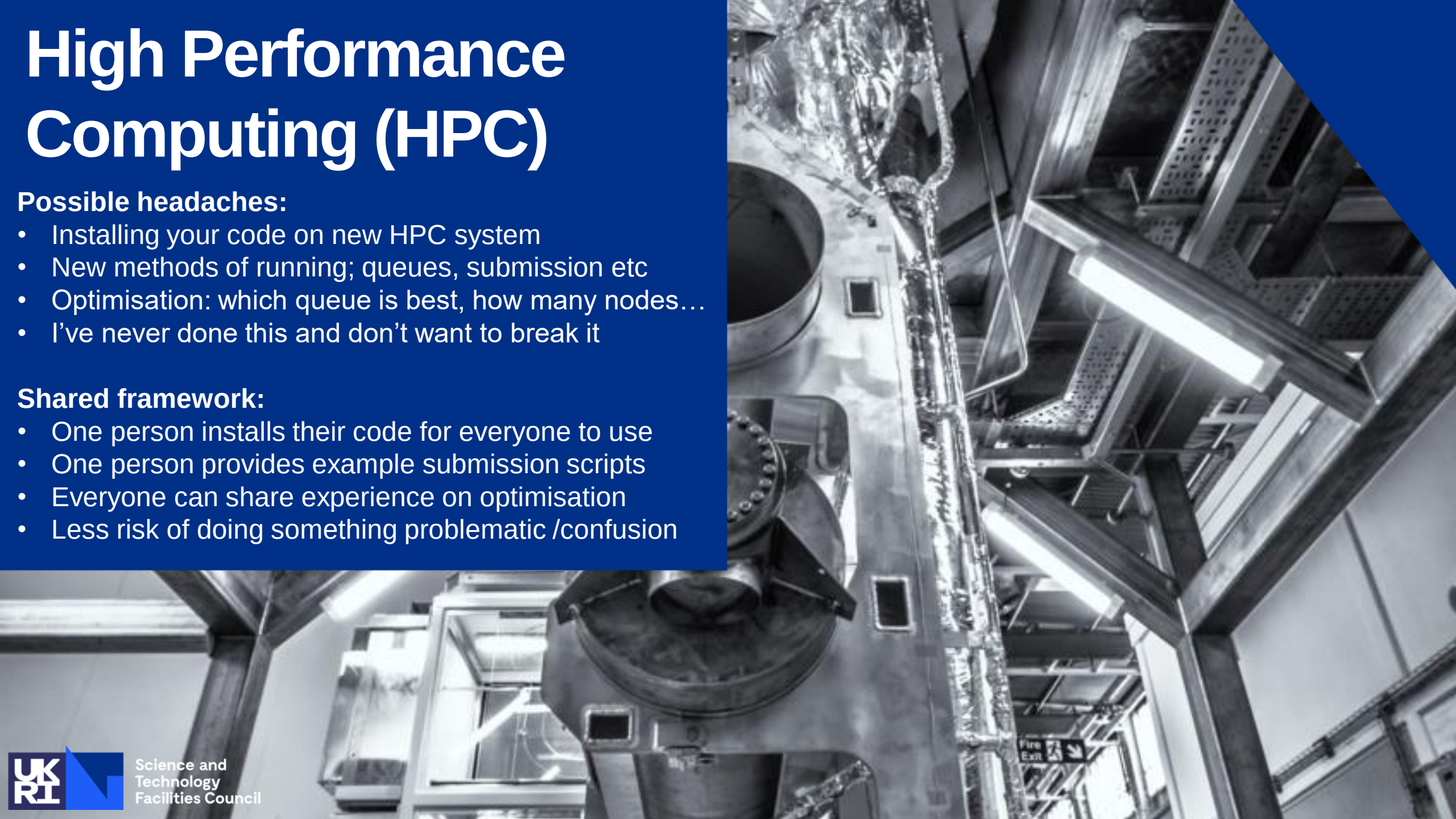
High Performance Computing (HPC)

Possible headaches:

- Installing your code on new HPC system
- New methods of running; queues, submission etc
- Optimisation: which queue is best, how many nodes…
- I've never done this and don't want to break it

Shared framework:

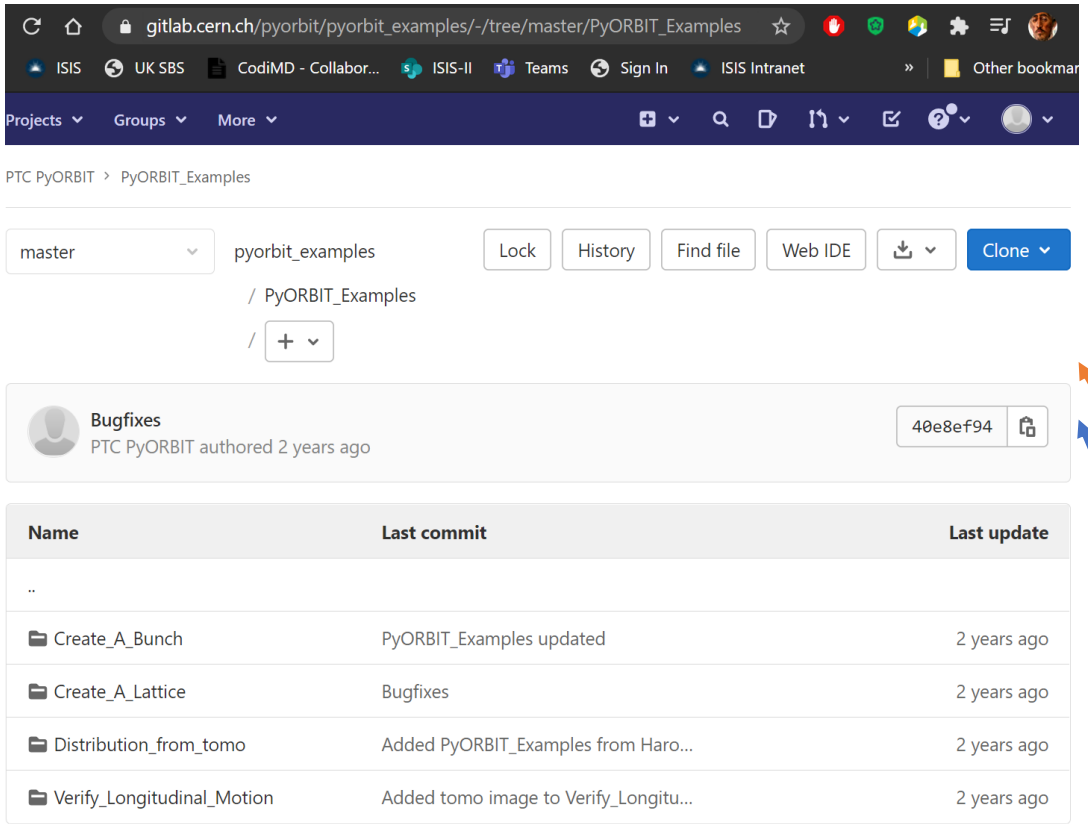
- One person installs their code for everyone to use
- One person provides example submission scripts
- Everyone can share experience on optimisation
- Less risk of doing something problematic /confusion

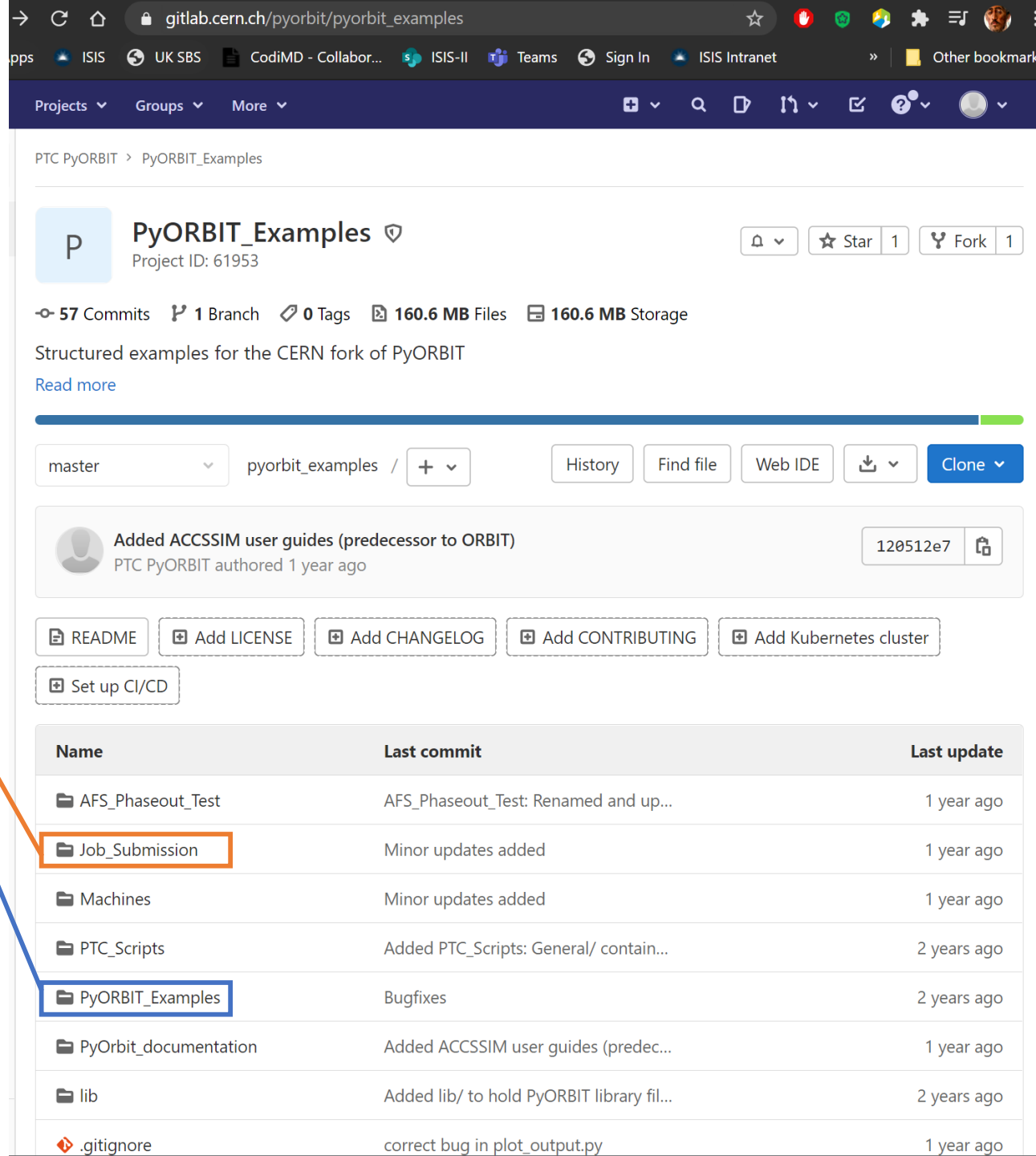


UKRI Science and Technology Facilities Council

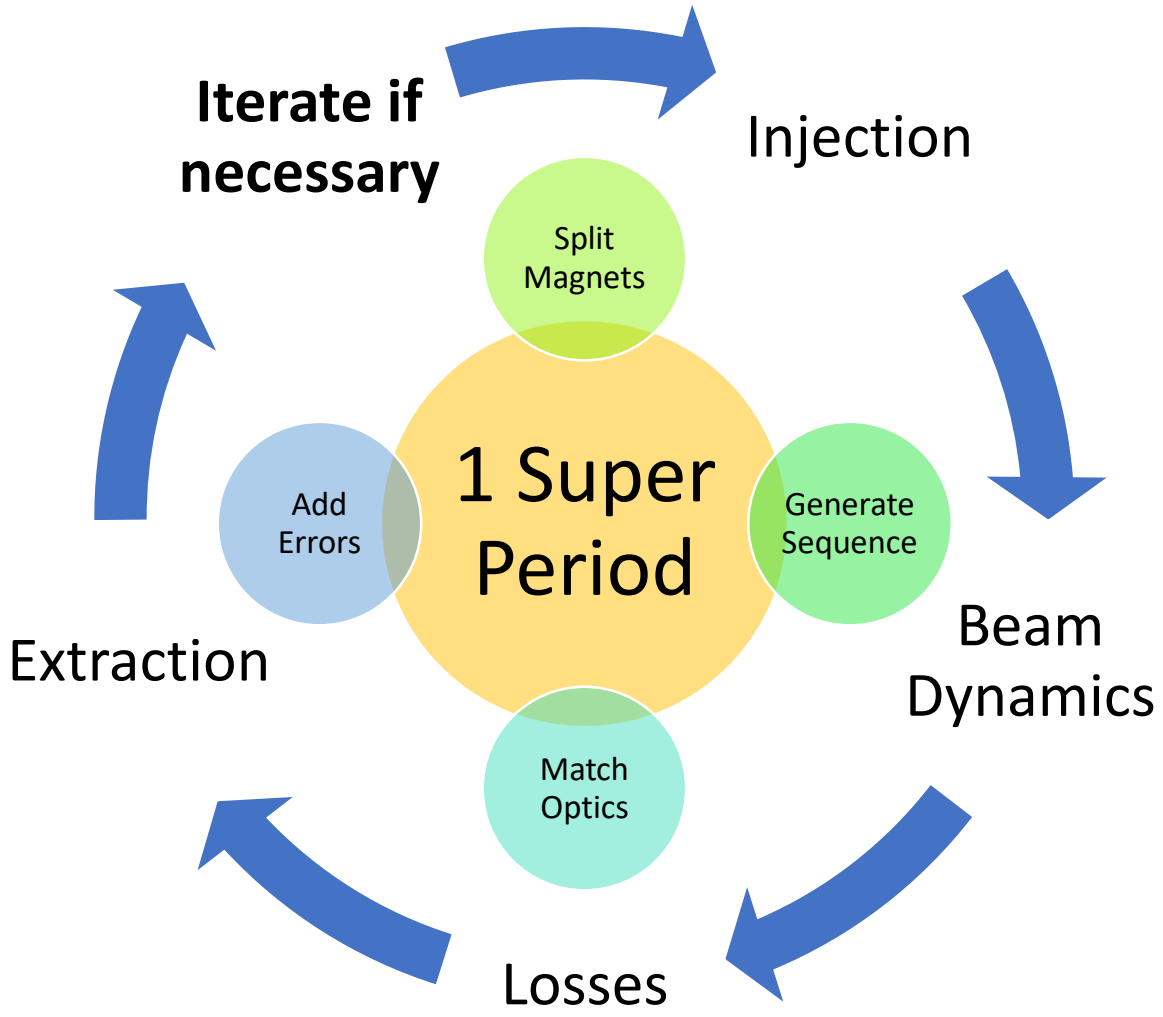
PyORBIT Examples

Created primarily for CERN Proton Synchrotron and Booster space charge simulations on HPC





ISIS-II Accelerator Design Repository: Ideas

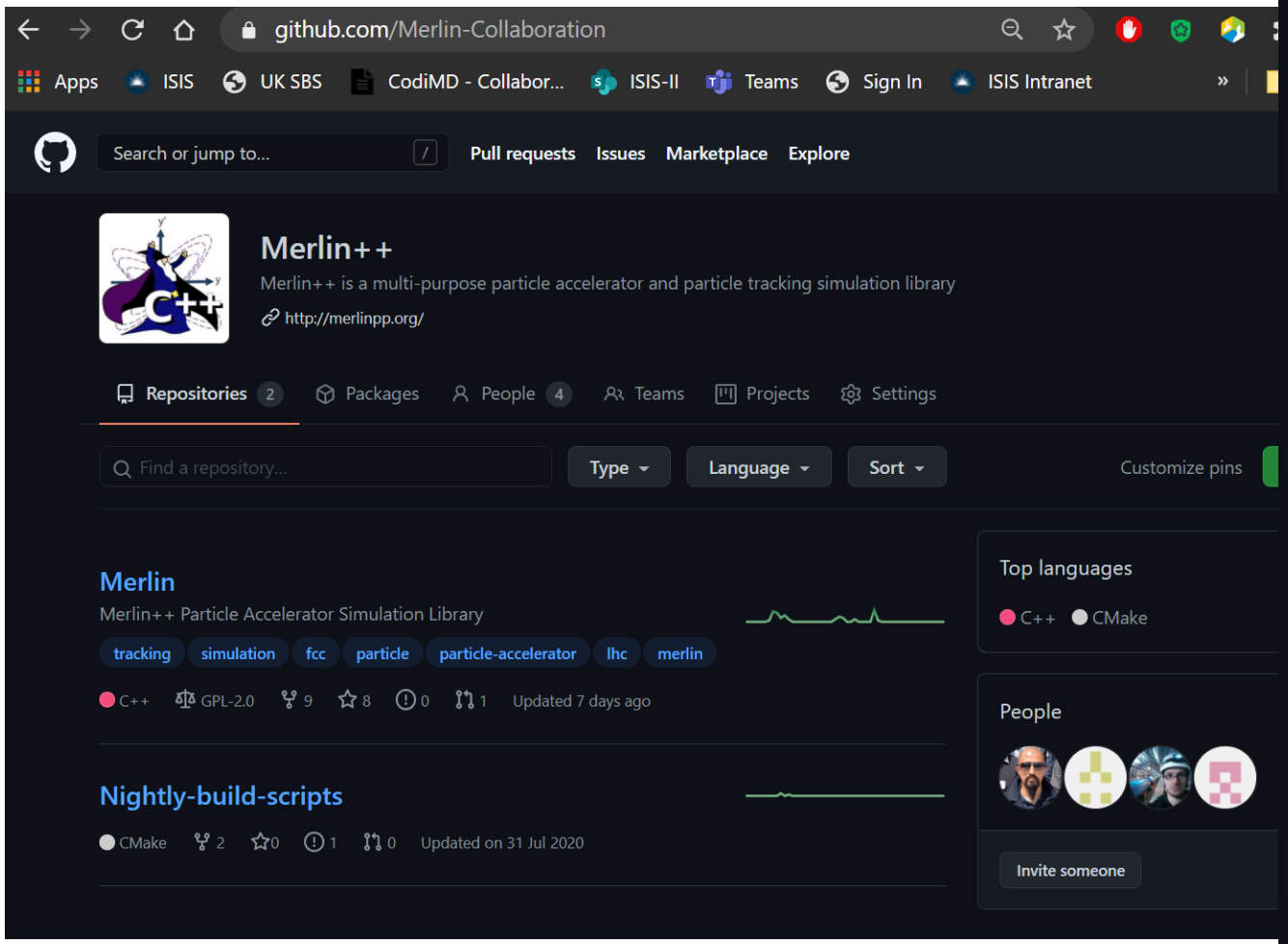


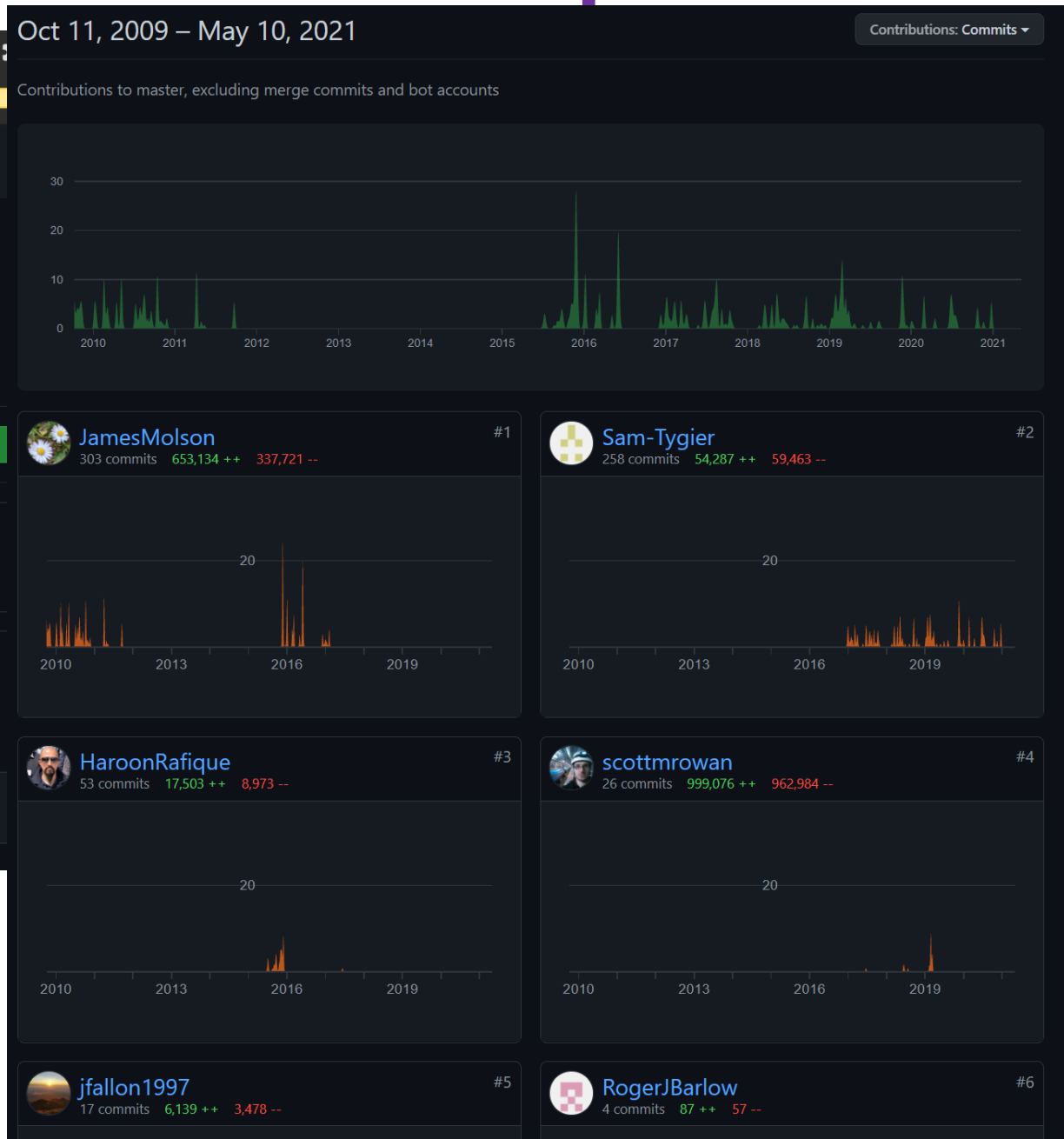
Use a single basis of instructions/examples for multiple design options

Rapid Cycling Synchrotron

Accumulator

Long Term (Code) Development: An example





https://github.com/Merlin-Collaboration/Merlin/network
https://github.com/HaroonRafique/MERLIN/network

Science and Technology Facilities Council

Debate

Working Together:

Problems? Ideas? Questions?

A Shared Framework **Should** Make Working Easier

If your code/simulation is large/complex or otherwise un-git-able

- Store input files etc required to reproduce important simulations
- Save (small) output files and plotting scripts to ease the write-up process (changing plots months after simulation is easy)

If your code/simulation is not useful to others

- Your work may be useful in the future, even in parts, and can save time as a reference at minimum
- Using a repository is a safe way of organising code, and can be useful when revisiting things years later

The time it would take to put it onto GitLab/GitHub is similar to the work itself

- **Git becomes easier with use**, and if we all use it as a default then this **should** save time in the long term
- Spending time to future-proof and make your hard work accessible should benefit the work of the group
- It is up to you to decide how best to spend your time ☺

Most of my work is messy

- Keep your messy work in your own branch/fork or even make your own repository/repositories that are more private and not designed for others to use
- **Only contribute tested, well documented work to shared repositories/projects**

Issues not solved by Git

Large file storage

- Git has a file limit of 50 megabytes
- Can we use **IDAaaS** / neutron user infrastructure
- Should we use **internal file storage**?

Overcoming the fear of change

- **Take a leap of faith?**
- Things are **likely to become more complex** with time (with regards to computing infrastructure, codes, tools, languages)
- Any problem is an opportunity to improve how things are done
- You **can't do everything on windows** (sorry-not-sorry STFC)
- I have **supported user codes / git repositories / HPC installations** for years – **happy to help** with anything
- **Don't know how to do something?** No matter how insignificant/stupid it may seem – **opportunity to contribute**
  - Put it in the README/documentation/ideas/suggestions/issues/bugs
  - Maybe ask Prof. Google

Remember

- **Time saving**, **future proofing**, **collaborating**, **pushing simulation complexity**, **exploiting HPC**, **bug-fixing**, **idiot-proofing**, **pandemic-proofing**, …

Science and Technology Facilities Council

Thank you

haroon.rafique@stfc.ac.uk