



Open-Source Software Development with Industry

Alexander Krimm, SIS100/SIS18

1st Workshop on Open Science @ GSI/FAIR 20.10.2023



Finland



France



Germany



India



Poland



Romania



Russia



Slovenia



Sweden



UK

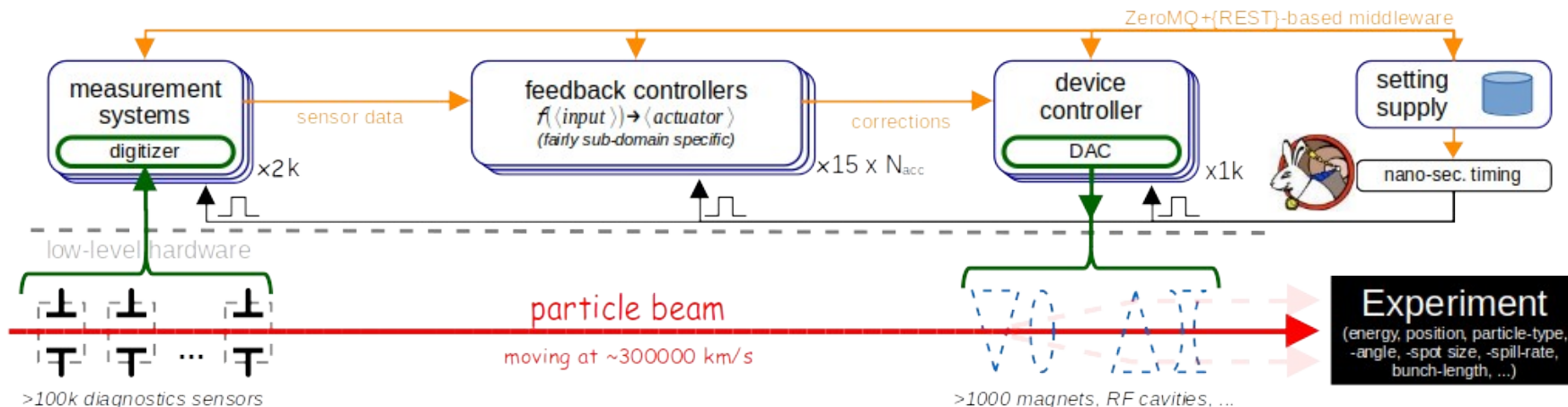
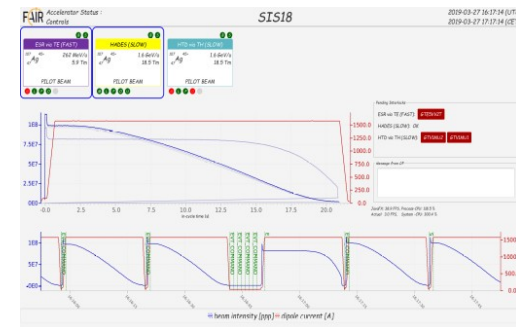




small team in SIS100/SIS18 + help from CIT and ACO working on system integration, beam-based diagnostic and feedback systems for FAIR

- 100k signals from various data-acquisition sources & 100s of systems
- 20+ core services per machine for commissioning, first-line diagnostics and operation + co-use by experiments
- **60+ person-years of scheduled work** → **only possible with external partners**

⇒ huge demand for **sustainable** software development



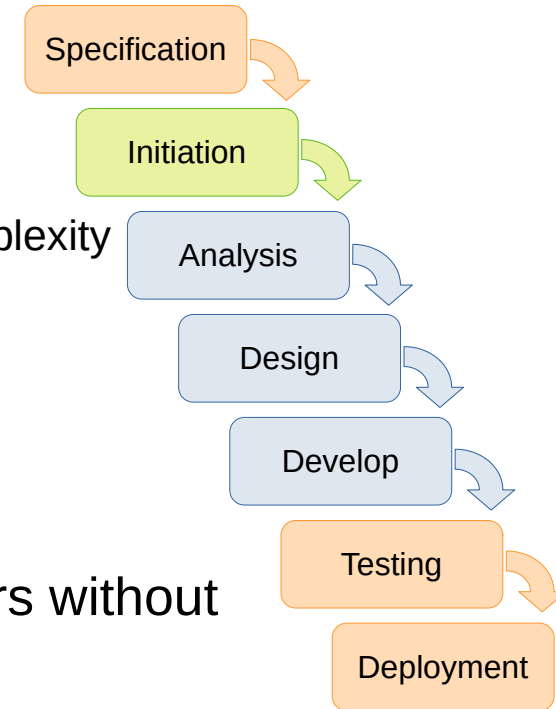
Why have external Partners at all?

- **lack of in-house resources:** freeing in-house team to focus on FAIR-accelerator-specific tasks.
- 1) Mitigating Risks → ‘agile’ and ‘lean’ Framework Contract
 - general FAIR contract designed for established technologies but unsuitable for SW development
 - 2) Software Quality and Efficiency mirrors Communication Structure (→ M. Conway’s Law)
 - **collaborations must be based on trust + open and transparent communication**
 - 3) Public Documentation of Technologies & Outreach
 - fostering seamless collaboration & attract a diverse talent pool
 - break free from vendor lock-in, circumvent single service-provider problem

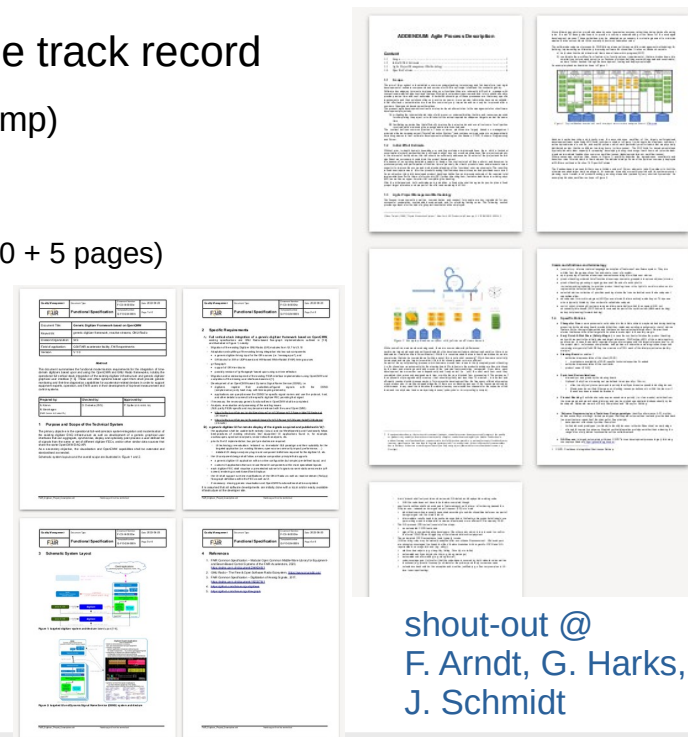
Creating Shared Value through creating public ‘clean’ and ‘lean’ Code-Base

- efficient response to 24h/7 operational needs and sustainable long-term maintenance
- FAIR as technology forge: develop technologies critical to FAIR & enable industry and general public in new emerging key technologies.

- “old contract”: ‘waterfall’ design process with few pros and mostly cons
 - strong tendency to over- or underspecification for software
 - overspecification time consuming for us and industry partners
 - design effort nearly identical to implementation effort
 - asymmetry between in-house and external understanding of project complexity
 - initial over- or underestimation of the project effort
 - slow feedback loops and expensive ECRs
 - “code-dump”: quality control and future maintainability
- closed in-house solutions: complicate onboarding
- hard to assess qualities and prior work of new industry partners without reliable public track record

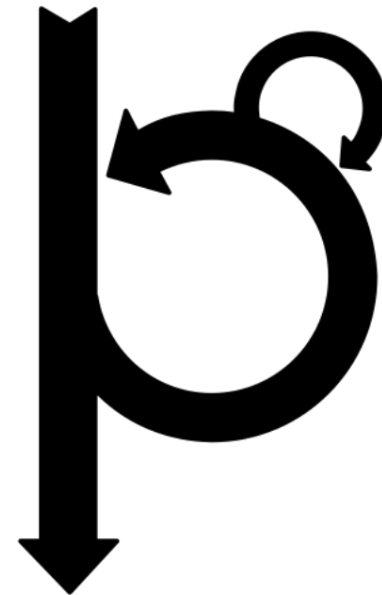


- enables fast and lightweight dispatch of projects to a pre-qualified pool of industry partners.
- pool selection process:
 - demonstrated skills and expertise proven by public open-source track record
 - quality of work & sustainability (actively maintained projects vs code-dump)
 - style of community engagement
- short-form legal contract + agile process description (10 + 5 pages)
 - small, re-usable, and describes the collaborative process
 - live iterations: 'change for free' policy
 - incentive for efficiency: 'early finish' & 'risk share'
- project description (2-10 pages, by FAIR)
 - outlines scope of work that should be performed
 - details provided in referenced public resources



shout-out @
F. Arndt, G. Harks,
J. Schmidt

- **function over form**
 - focus on design, function and intended use
 - ... rather than overspecified implementation details that are impossible to know at project start
- **prefer building upon or extending existing projects** rather than creating new ones
 - sharing of expertise and ideas
 - spread maintenance on more shoulders
- **co-development**
 - lightweight specification + iterative process
 - allow re-priorisation and agree on a fair sharing of risks
- **shared code ownership**
 - early integration of individual functionalities
 - review process of small functional units
 - code and development process in the open as much as possible
 - for direct communication → document results publicly
- **collaboration must be based on trust and open communication**
 - communicate early on wrong estimates



FAIR/GSI

Industry Partners

CALL: agile contract
& project description

OFFER(s): coarse backlog
with storypoint estimate

select best offer
(multiple companies)

public story-point
estimation & prioritisation
with all partners

20%

80%

review work and
track storypoints

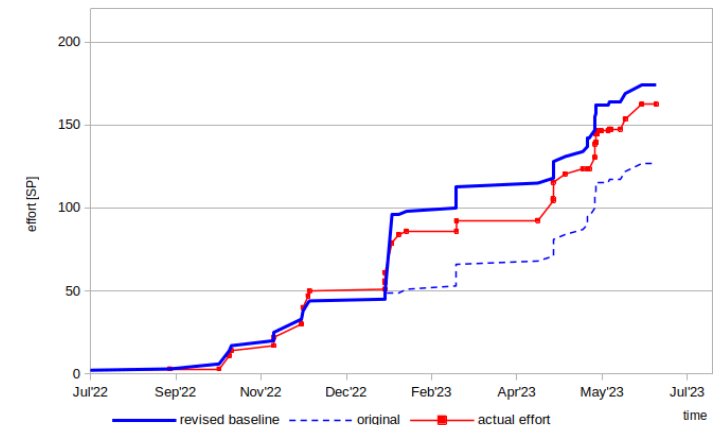
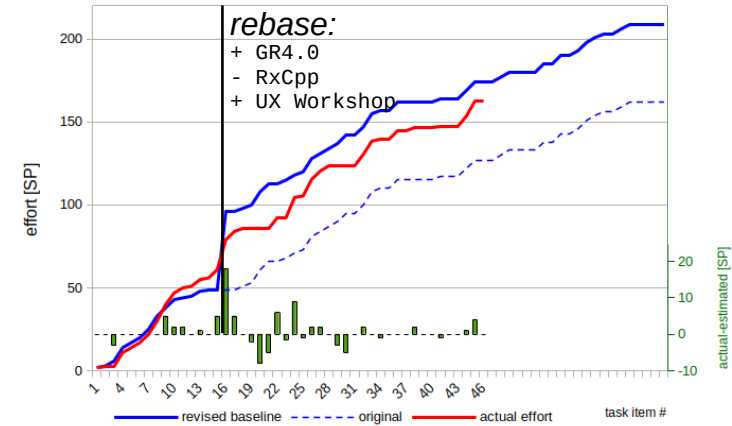
Kanban Project Board

implement selected issues

re-prioritisation

Pull-Request

- story-point: one unit of work, approximately one workday
- [estimated, actual] tracking on kanban board issues
 - functional and financial records
 - but more important: continuous feedback and improvement on shared understanding of task and project complexity



Digitizer Reimplementation

Backlog | Call#4 | Details | Call#1 | OperationalDeploy | + New View

Filter by keyword or by field

Columns: Ideas (w), Backlog, Selected (3), In progress, Finished Implementation (2), QA-Accepted/Merged (w)

Items in 'Selected (3)':

- graph-prototype #162: Implement up-/downconversion filter (i.e. 'FrequencyXLating' filter)
- graph-prototype #161: EPIC: List of GR 3.10 blocks & Others to be ported to GR 4.0
- graph-prototype #125: [2pt] graph-prototype: Fix issues with uT and dynamic libraries, plugins

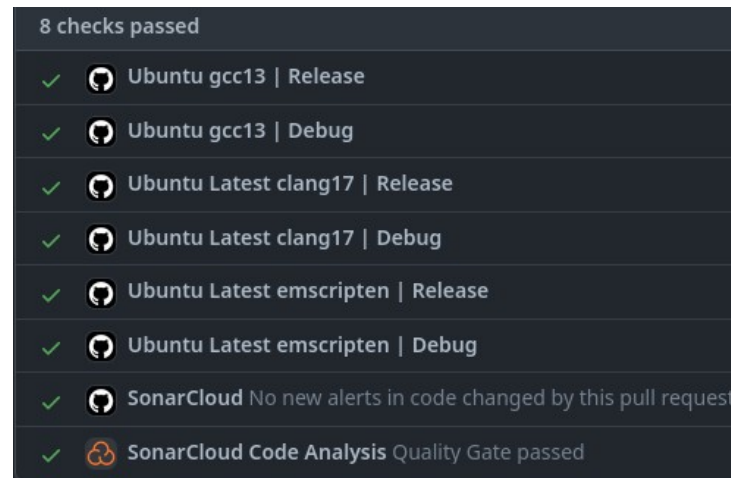
Items in 'In progress':

- opendigitizer #90: [2pt] See what the problem is with memory relocating ports after connection
- opendigitizer #37: [5pt] Non-distributed DNSs with persistence
- opendigitizer #15: [5pt] UI: Storing, clearing and reloading the flow-graph and the UI configuration
- opencmc-cpp #300: 37 dns with persistence
- opendigitizer #119: [5pt] Base UI for signal selection

Items in 'QA-Accepted/Merged (w)':

- graph-prototype #187: [2SP3SP] Add the support for arrays/vectors to the connect syntax
- opendigitizer #91: [0pt] Investigate all the node instances we use in tests and extract into a library
- graph-prototype #69: [0pt, 8pt] implement selector block/node (switch matrix like run-time plumbing between input-output ports)
- graph-prototype #159: Improve performance of non-fftlib implementation of FFT
- opencmc-cpp #191: [2pt, 7pt] Event store: Evaluate the current system settings and extend if needed, transfer

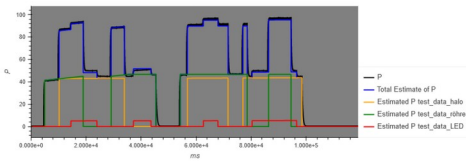
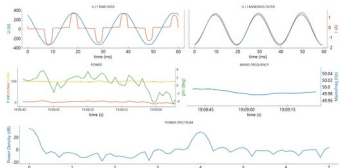
- issues on the Kanban board have a predefined **deliverable** and story-point **estimate**. Once done,
 - reviewed by GSI/FAIR,
 - deliverable merged, and
 - story points get tracked (triggers payment)
- **PR-reviews** are assisted by CI:
 - must build without compiler warnings,
 - must pass static code analysis tools (aka. ‘linters’),
 - must pass unit tests and code coverage criteria,
 - automated code formatting, ...
- for big differences between estimated and used story points, discuss causes and possible ways to get back on track



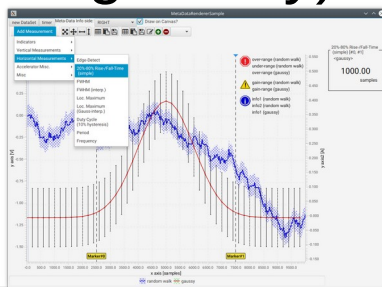
- digitizer framework + GNURadio + KDAB (3 x 180SP, 2019-ongoing)



- AI-based pulsed-power monitoring + Infoteam AG (2 x 90SP)



- chart-fx (Java charting library) + HEBI Robotics (20SP)



Why have external Partners at all?

- **lack of in-house resources:** freeing in-house team to focus on FAIR-accelerator-specific tasks.
- 1) Mitigating Risks → ‘agile’ and ‘lean’ Framework Contract
 - general FAIR contract designed for established technologies but unsuitable for SW development
 - 2) Software Quality and Efficiency mirrors Communication Structure (→ M. Conway’s Law)
 - **collaborations must be based on trust + open and transparent communication**
 - 3) Public Documentation of Technologies & Outreach
 - fostering seamless collaboration & attract a diverse talent pool
 - break free from vendor lock-in, circumvent single service-provider problem

Creating Shared Value through creating public ‘clean’ and ‘lean’ Code-Base

- efficient response to 24h/7 operational needs and sustainable long-term maintenance
- FAIR as technology forge: develop technologies critical to FAIR & enable industry and general public in new emerging key technologies.



Thanks for your Attention!

Do you use/develop/manage open-source projects? What are your experiences and approaches?

Alexander Krimm, SIS100/SIS18

Open-Source Software Development with Industry, 1st Workshop on Open Science @ GSI/FAIR 20.10.2023



Finland



France



Germany



India



Poland



Romania



Russia



Slovenia



Sweden



UK

