# PTC Documentation

*Release 0.0*

**C.Youngman**

**Feb 13, 2021**

# Contents

Contents:

# CHAPTER 1

# PTC - Introduction

## 1.1 How to read this document

**It's best to treat chapters as buckets whose content was mostly filled when the chapter was first written, but may have received extra content later. Chapters towards the end have usually been written later and may be more up-to-date with the entire PTC concept whose details are changing as more is understood.**

## 1.2 The Project Technical Coordinator aim

The Project Technical Coordinator (PTC) role aims at providing a responsible for coordinating DATA Dept resource usage requests resulting from project implementation procedures.

## 1.3 DATA resources

DATA resource are:

- DATA FTEs (CTRL, DA, DETOP includes CAL, EEE and ITDM) required to perform work, and
- DATA Finances required to pay for DATA items used.

## 1.4 Procedures handled

Procedures are workflows which aim at accompanying an idea through design, implementation and commissioning stages until its final handover as a working object. Procedures should expose an efficient way for idea owners to work out which procedure is best suited for their idea and provide an efficient workflow concept of how to get the work organized. The word idea is used above to underline ownership, it's somebodies idea, and that responsibility remains throughout even though a procedure gets involved with organizing and DATA with implementing. It should also not be surprising that many ideas are running through many procedures concurrently and like all competitions, some will be rejected, others will be cancelled or delayed, and many will succeed. In the remainder of this document *project* will be used rather than *idea*, it's less prosaic.

Procedures which require DATA resources are:

- XO-procedure (launched 2020): improving the operation and preparation for operation of experiments at the facility.
- PMO-procedure (launched Autumn 2019) : focuses on beamline and instrument development projects which exceed a (large) resource usage threshold.
- ERD-procedure (launched Spring 2020): specifies how a single piece of equipment is integrated into DATA services.
- CRD-procedure (relaunched mid-Nov 2020): specifies how a component of many equipment pieces is integrated into DATA services.

The R&D procedure is not included in the list because R&D projects provide their own resources and do not require DATA resources other than daily support. This sounds good, but a definition of daily support is needed and methods for monitoring its usage are required.

The facility's move from contruction and commisioning to operation left a number of instrument completion projects either not planned or not implemented. PMO intends to close and relaunch the latter if still required and allow those nearing completion to proceed with daily support.

Additional DATA resource using procedures should not exist, if they did they would be a resource leak and make the PTC mandate unworkable. This also applies to developments started by DATA groups themselves.

## 1.5 The mandate

Most DATA members will have suggestions for improving coordination of DATA resources. Here are the project procedure points highlighted by the Dept Head.

- Focus on efficient DATA handling of a procedure project.
- Define methods to be used by DATA to handle procedure projects.
- Promote DATA methods into procedures.
- Add or coalesce procedures if needed.
- Support improvements to existing procedures.
- Prevent work side-stepping procedures.
- Act as the gatekeeper for procedure project requests and their handling.
- Monitor DATA resource usage by procedures.
- Define roles, delegation paths, and (re-)establish procedure project communication.
- Report to the Dept. Head.

## 1.6 PTC resources

PTC has no resources, its success depends on supporting contibutions from the Dept as a whole, which means

- Dept members need a clear understanding of how PTC works, i.e. suitable documentation is needed and a reason for reading this document!
- The status of procedure projects needs to be visible to all.

- In addition to procedure project meetings regular review meetings are needed to maintain an even knowledge level.
- Lessons need to be learned from mistakes, i.e. asking what has been missed should be second nature.
- Meeting and management tools, similar to sharepoint, are needed to imrove (PTC) meeting efficiency.

### 1.6.1 Next steps

Let's assume the PTC solution is obvious and, rather than analyse existing inefficiencies first, go straight for the solution in the next 'Big Picture' chapter. Later chapters describe how PTC will work with long and short procedure types defined in the big picture. Long procedures place similar requirements on DATA and as more experience has been gained with the PMO procedure this will be described first with XO inheriting substantial parts. The short procedures ERD followed by CRD are then described. Thereafter the PTC interface is specified followed by a chapter describing PTC and procedure management support software tooling. Issues relating to project visibility and scheduling followed by a glossary are then described.

The glossary is importand as the work set used in project planning is small, e.g. project is (re-)used excessively. To understand the difference between procedure workflow and implementation workflow in this document read the glossary before moving on!

CHAPTER 2

# PTC - Big Picture

An overview of PTC requirements for communication paths and roles is show below.

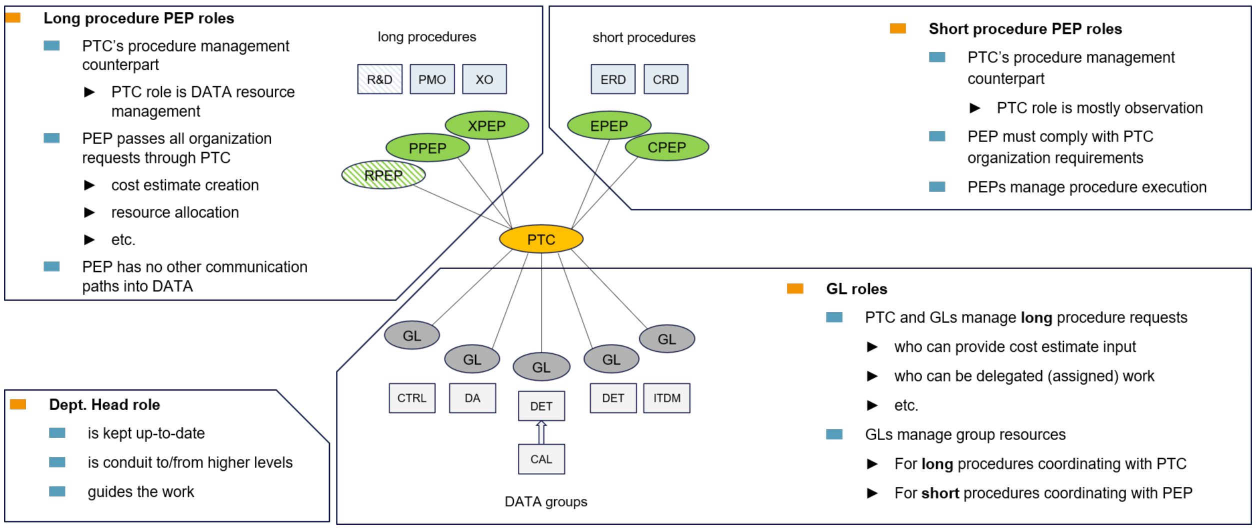

Long procedure PEP roles
PTC's procedure management counterpart
PTC role is DATA resource management
PEP passes all organization requests through PTC
cost estimate creation
resource allocation
etc.
PEP has no other communication paths into DATA
long procedures
R&D
PMO
XO
XPEP
PPEP
RPEP
short procedures
ERD
CRD
EPEP
CPEP
Short procedure PEP roles
PTC's procedure management counterpart
PTC role is mostly observation
PEP must comply with PTC organization requirements
PEPs manage procedure execution
PTC
GL
GL
GL
GL
GL
CTRL
DA
DET
CAL
DET
ITDM
DATA groups
GL roles
PTC and GLs manage long procedure requests
who can provide cost estimate input
who can be delegated (assigned) work
etc.
GLs manage group resources
For long procedures coordinating with PTC
For short procedures coordinating with PEP
Dept. Head role
is kept up-to-date
is conduit to/from higher levels
guides the work

Fig. 1: PTC communication paths and roles

The figure is self-explanatory, but a few strengthening remarks should be added.

Explanation of procedure types

- *Long* procedure projects typically have long implementation completion times, make large resource requests on many DATA groups, and may have additional requirements such as safety.
- *Short* procedure projects typically have short implementation completion times due to making fewer resource requests on fewer DATA groups.
- The long and short definitions additionally avoid using external and internal to classify procedures, which inherently imply a bias that is not intended.

Additional statements regarding roles

- Procedure Entry Points (PEPs) are the working partner of PTC. Each procedure has at least one PEP.
- Long PEPs pass all organization and work requests through PTC. This avoids *punch through* where non-DATA group personnel approach DATA personnel w/o the knowledge of the DATA GLs, TLs, or PTC and, consequently, receive information which is meaningless.
- Long procedure managers, probably the PEPs, manage their procedure.
- Long PEPs see the PTC as the DATA resource manager who maintains DATA's view of the procedure concerned.
- Short PEPs see the PTC mostly as an observer, i.e. they are expected to run their procedure efficiently. The procedure view exposed to PTC must comply with PTC requirements.

## 2.1 Fine tune rather than change

Products that make things easier are an enormous benefit and designing a product for identical robots is simply a question of logic. Fortunately people are not robots and people must see the benefits for a product to be a success. The view of project management at EuXFEL reveals no clear path taken and no significant drive to address the issue on a large scale. Perhaps it is fairer to say that tools are available, but their usage is scattered and not coordinated beyond individual or group requirements. Mandating PTC represents a move towards cost aware resource planning by DATA. However, its startup is difficult because no consistent approach exists between different procedures and the tools used.

The approach followed is fine tune rather than change:

- discuss with procedure workflow owners how their systems work, identify requirements and identify useful PTC interface components.
- with the procedure managers produce dry-run environments of the derived PTC interface and test these with owners and DATA group resource holders.
- align test feedback and implement the PTC interface followed by periodic reviews and modifications.

Completiuon of these items is necessary to allow an understanding of DATA resource capacity and usage which should enable resource scheduling.

## 2.2 The aimed for solution

PTC creates and manages for each long procedure a Redmine project container where project implementation workflows (of FTE tasks) specified in DATA cost-estimates are placed. As the Redmine container belongs to PTC it automatically satisfies the PTC interface requirement.

PTC uses the ERD and CRD short procedure Redmine project containers to perform work with the implementation projects. PTC does not own the container, but the owners must comply with the PTC interface requirements.

PEPs work with PTC to ensure knowledge equality and efficient project scheduling.

PTC tooling aims at satisfying PTC work actions and reducing PEP management load by automating repetitive and error prone operations.

PTC project scheduling should be visible, which may require additional Web tooling if it cannot be provided by Redmine directly.

CHAPTER 3

# PTC - Choosing a procedure

As mentioned above, procedure managers should expose an efficient way of allowing the idea holder to identify which procedure is appropriate. It would be great to have a browser decision form whose questions direct the project requester to the correct procedure, but we do not have this yet. Therefore base the decision on the chart below and ask at the email addresses given.

Below is a flat way of looking at all work coming in, DATA needs to decide is this is the correct view of the world, or propose anothre view.

An immediate issue is how do long procedures apply project acceptance thresholds which require technical input to decide.

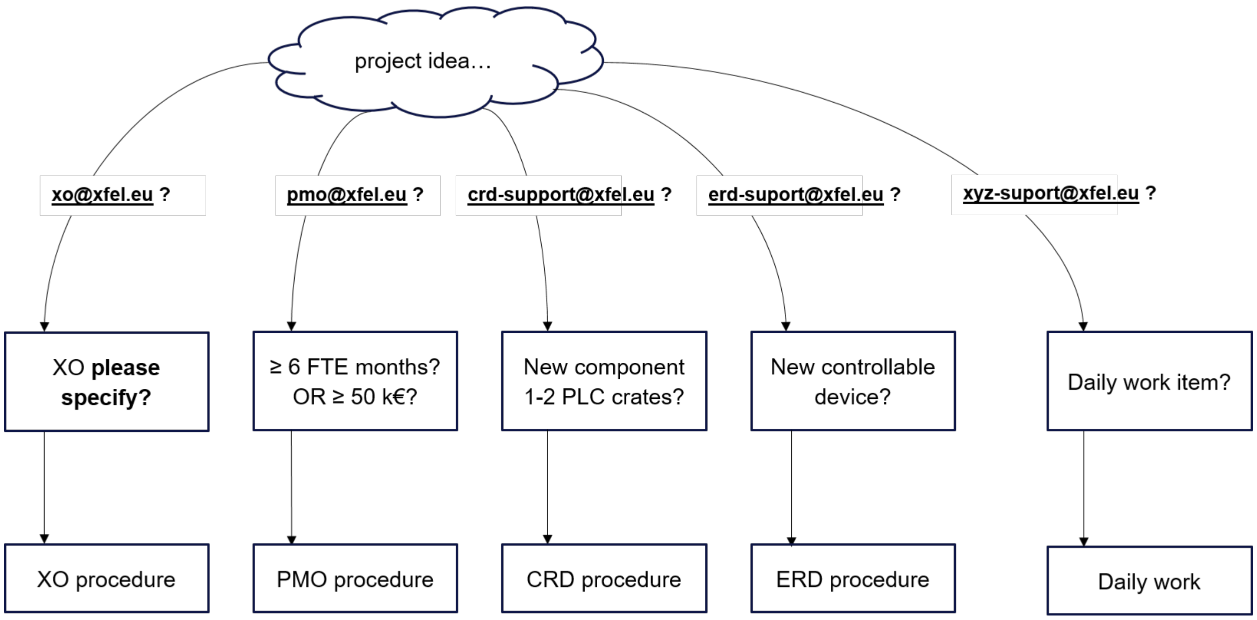


Fig. 1: PTC Choosing a procedure chart, where

## 3.1 DATA Daily work definition - questionnaire

Daily work looks like a procedure, work items can be specified by non-DATA (e.g. by the CTRL-ICI instrument contact or controls-integration@xfel.eu which inserts a ticket into Controls' Redmine) or DATA (e.g. EEE-PLC request to update loop f/w) personnel and the ability of DATA to perform daily work depends on the availability of resources.

The table below are DATA group and team answers to: 'What work would be done w/o the need for it to be in a procedure project?'. The CTRL-ICI example answer given avoids having an empty definition column.

The table prompts an additional question: 'Are Group or Team abbreviations correct and are any missing?'. Whilst inserting definitions please update the column. The abbreviations need to be clarified as they are frequently used by the PTC interface, e.g. in the PMO cost estimate.

| Group or Team | Daily work definition |
|---|---|
| CTRL | |
| CTRL-ICI | An item requiring no more than 1/2 a day which can be scheduled during the next 2 weeks. |
| CTRL-DEV | |
| DA | |
| DETOP | |
| DETOP-CAL | |
| EEE | |
| EEE-EDS | |
| EEE-FET | |
| EEE-PLC | |
| ITDM | |
| ITDM-DAQ | |
| ITDM-INFRA | |

CHAPTER 4

# PTC - PMO projects

## 4.1 PMO workflow as seen by DATA

The table below defines the PTC's work interface to PMO. It contains a row ordered list of activities and descriptions performed by PTC during a PMO project lifecycle. Open points are possible suggestions for future improvements.

| activity | description |
|---|---|
| PMO sends email requesting cost-estimate from PTC with project number (e.g. DP021) in the subject and project owner (PO) in cc. | This email starts PTC as DATA's gatekeeper for the project. All subsequent emails and Redmine tickets must contain the project number in the subject, which enables PTC email filters. Some projects may require no DATA resources, but a cost-estimate must be produced to document this. Open: inject a Redmine ticket with email. with the first email. |
| PTC schedules cost-estimate meeting | Participants: PTC, PO and PO's technical support, and representatives of all DATA groups (and SRP, TS… if required). PO working with PTC distribute heads-up relevant info before the meeting. |
| PTC chairs cost-estimate meeting. | The meeting aim is to identify DATA resource requirements and itemize these in the cost-estimate FTE and Financial sheets and to list prerequisites (on PO) and project implementation risks in additional sheets Any outstanding issues, either PO or DATA side, are handled offline and managed by PTC to reduce the need for a second all present meeting. |
| PTC finalizes the cost-estimate and sends it to PMO with the PO in cc. | This requires PTC, PO and DATA (SRP,TS…) groups to agree on spreadsheet content which is noted in the banner sheet. All DATA (SRP,TS…) groups must signoff the cost-estimate. |
| PTC inserts the project into Redmine. | This is the first scheduling for the project. Open: decide on who can move the schedule, currently this is PTC, i.e. DATA only. Opem: inter-procedure dependencies must be handled |
| PTC handles change requests from PMO. | Projects which are cancelled are moved to the cancelled folder w/o changing task statuses, i.e. they disappear from the the in-progress folder. Similarly projects placed on-hold are moved to the on-hold folder w/o change. In-progress projects are scheduled by DATA as the resource owner, priority change request from PMO or higher need to be processed by PTC (with DATA DH,GLs,TLs) Cost-estimate changes require agreement of DATA (SRP,TS…) groups and the modified document must be signed off and distributed to all. |
| PTC handles project closeout requests from PMO. | The cost-estimate is updated for actual resouce cost and a comments sheet filled by DATA groups and distributed to all DATA groups, PO and PMO. The project remains in Redmine in-progress until moving to the completed folder at the end of the year. |

## 4.2 Role definitions

The following role definitions are apparent from the table.

PTC:

- is the gatekeeper and no request from PMO, PM, PO or their delegates should *punch through* to DATA personnel without it being previously delegated to the personnel by PTC and the respective GL or their TL delegate.
- organizes cost-estimates with DATA GLs, their delegates, etc.
- once finalized with DATA and the PO, the cost-estimate is sent to PMO (cc PO) along with a confirmation or not of DATA personnel names should they have appeared as WP leaders in the PMO project form.
- ensures project scheduling is organized according to the cost-estimate workflow definition and DATA resource availability, and input from the Dept head, PMO, etc.
- manages DATA's Redmine implementation workflow representation of projects whereby task developers are required to keep task milestones, see PTC inteface chapter, up-to-date.
- tracks with PMO project status

PMO:

- is responsible for handling project passage through their procedure workflow
- ensures that Project Responsibles (aka Owners) understand the requirement of working with DATA, i.e. the PTC role
- ensures that the PMO project form is completed using DATA's cost-estimate and personnel name (e.g. WP leader delegations) information.
- tracks with PTC in-progress project status
- informs PTC of project reorganization: rejection, cancellation, delays, or holdups.

DH:

- is aware of and participates in scheduling activities.

GL (or their delegates, e.g. TLs):

- participate in cost-estimate gathering and later modifications.
- is aware of and participates in scheduling activities.

SRP,TS… if required:

- participate in cost-estimate gathering and later modifications.
- are aware of and participates in scheduling activities.

## 4.3 DATA cost-estimate production

Cost-estimates are Excel files containing 5 sheets:

- *Banner*, the signoff sheet.
- *Prerequisites*, a list of DATA requirements on the PO.
- *FTEs*, the manpower tasks required from DATA to complete the project.
- *Financial*, the fanancial costs identified by DATA to complete the project.
- *Risks*, DATA statements relating to potential in-progress issues or show stoppers of the project.

The cost-estimate is updated during project rollout using PTC tooling by:

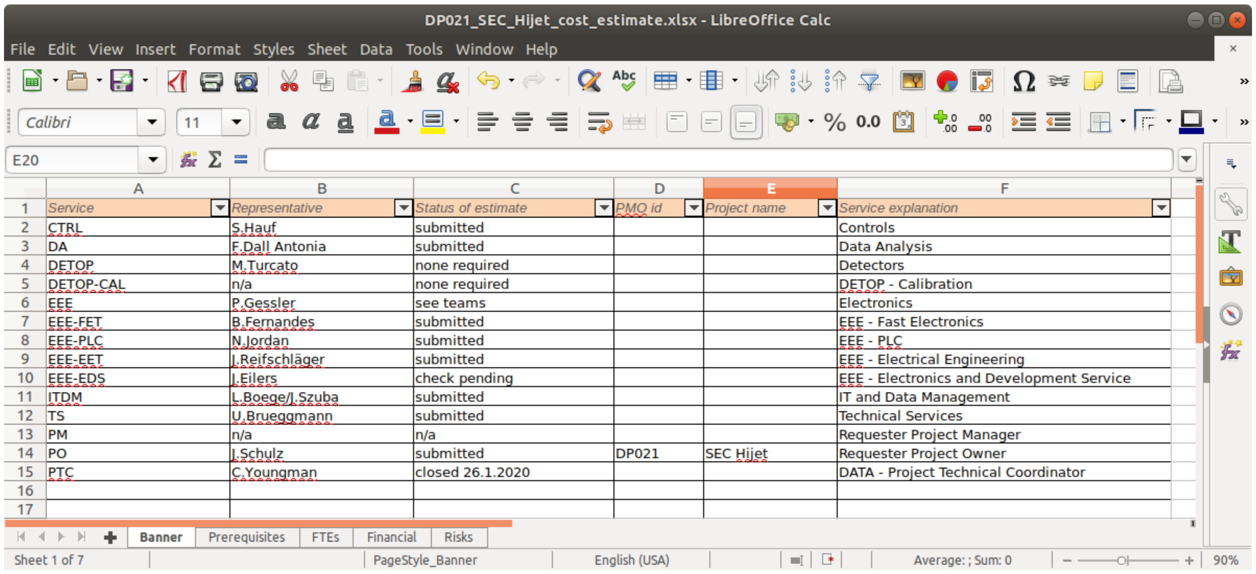
- adding Redmine ticket numbers of tasks
- updating task attribute: status, start_date, due_date, etc.
- tbd: add project milestone and time taken

The cost-estimate is finalized on closeout by:

- adding a *Comments* sheet describing issues, pro and con, encountered by DATA during project rollout.
- adding to *FTEs*, the actual FTE task time spent.

## 4.4 Example cost-estimate sheets

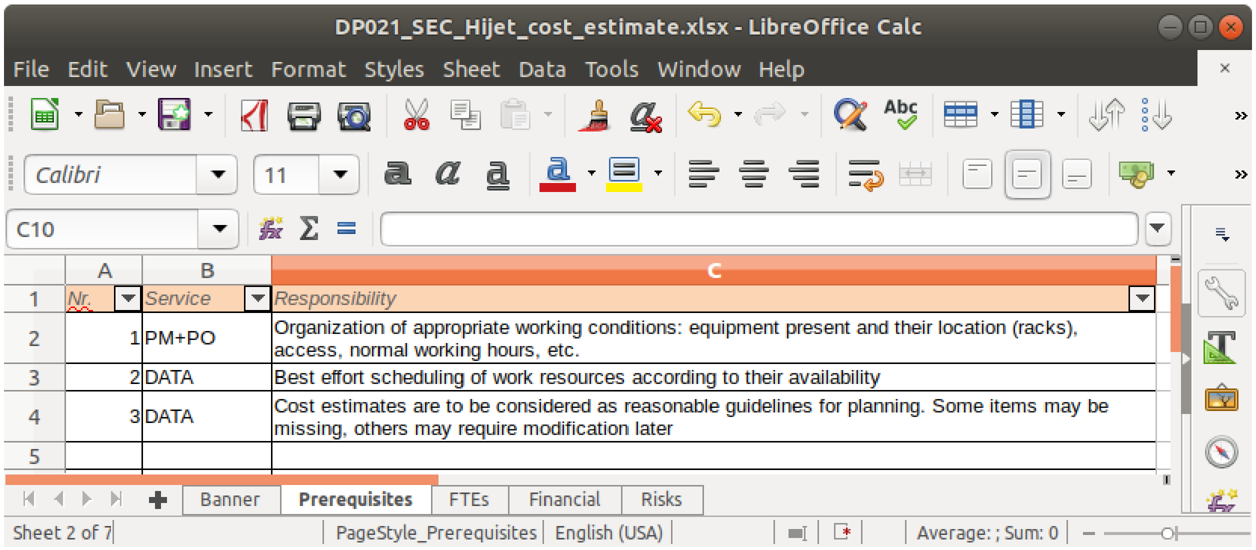
The Banner table above lists PO, PM (the PO's Project Manager) and DATA representatives participating in the cost-estimate. For DATA groups the status of estimate can be none-required, TBD or submitted. Where GL and team leaders of the same DATA group are present the GL status tracks the team leader status and finally reaches submitted or none-required. The GL can determine the status as seen in the example given where DETOP-CAL was not present



| Service | Representative | Status of estimate | PMO id | Project name | Service explanation |
|---|---|---|---|---|---|
| CTRL | S.Hauf | submitted | | | Controls |
| DA | F.Dall Antonia | submitted | | | Data Analysis |
| DETOP | M.Turcato | none required | | | Detectors |
| DETOP-CAL | n/a | none required | | | DETOP - Calibration |
| EEE | P.Gessler | see teams | | | Electronics |
| EEE-FET | B.Fernandes | submitted | | | EEE - Fast Electronics |
| EEE-PLC | N.Jordan | submitted | | | EEE - PLC |
| EEE-EET | J.Reifschläger | submitted | | | EEE - Electrical Engineering |
| EEE-EDS | J.Eilers | check pending | | | EEE - Electronics and Development Service |
| ITDM | L.Boege/J.Szuba | submitted | | | IT and Data Management |
| TS | U.Brueggmann | submitted | | | Technical Services |
| PM | n/a | n/a | | | Requester Project Manager |
| PO | J.Schulz | submitted | DP021 | SEC Hijet | Requester Project Owner |
| PTC | C.Youngman | closed 26.1.2020 | | | DATA - Project Technical Coordinator |

Fig. 1: Cost-estimate Banner sheet

at the meeting as not required. PO, PM and PTC reach completed when no outsatnding issues are present, i.e. the cost estimate is finished.



| Nr. | Service | Responsibility |
|---|---|---|
| 1 | PM+PO | Organization of appropriate working conditions: equipment present and their location (racks), access, normal working hours, etc. |
| 2 | DATA | Best effort scheduling of work resources according to their availability |
| 3 | DATA | Cost estimates are to be considered as reasonable guidelines for planning. Some items may be missing, others may require modification later |

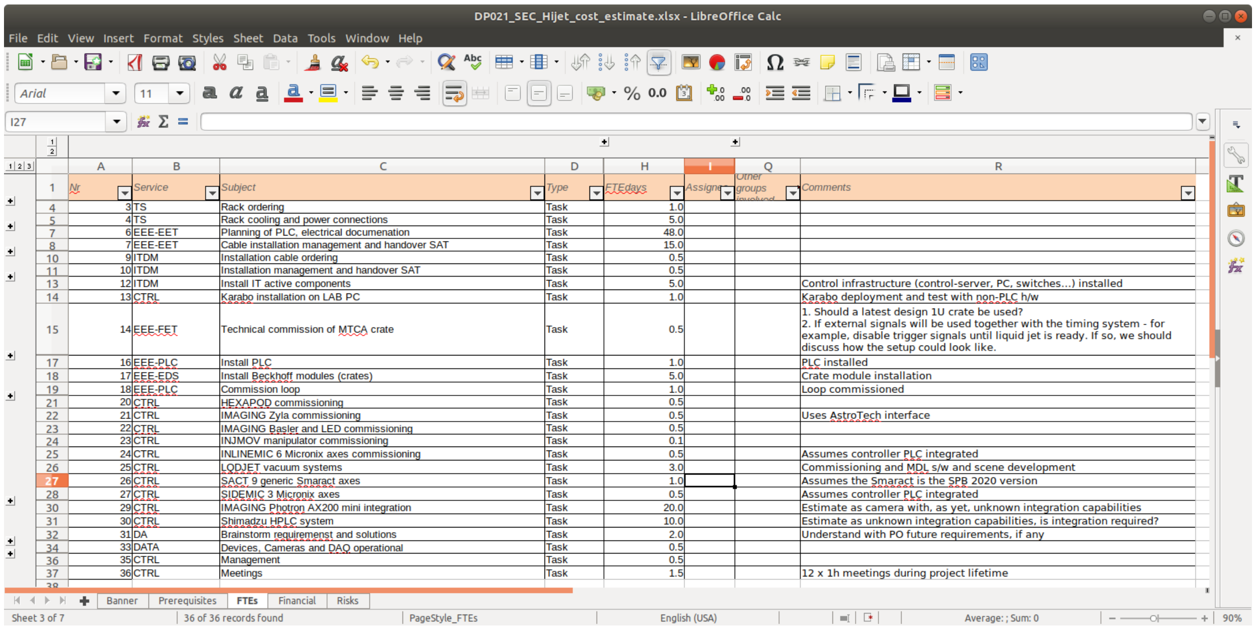
Fig. 2: Cost-estimate Prerequisite sheet

The Prerequisite table above lists requirements from DATA that are to be satisfied by the PO before work can start. Prerequisites are typically: avialability of s/w, f/w and hardware required, installation in a location suitable for working, possibly remotely, etc. Milestones defining when different stages of project work can be started are also prerequisites.

The FTE table above lists the manpower tasks required to complete the project. The *Subject* column text proceeded by the PMO-id (here DP021) is used as the subject of the corresponding Redmine task.

The Financial table above list the cost of items needed to complete the project. Particular attention should be given to the *Paid by* column.

The Risks table lists items that could prove to be show stoppers or at least performance reducers. The table's impor-

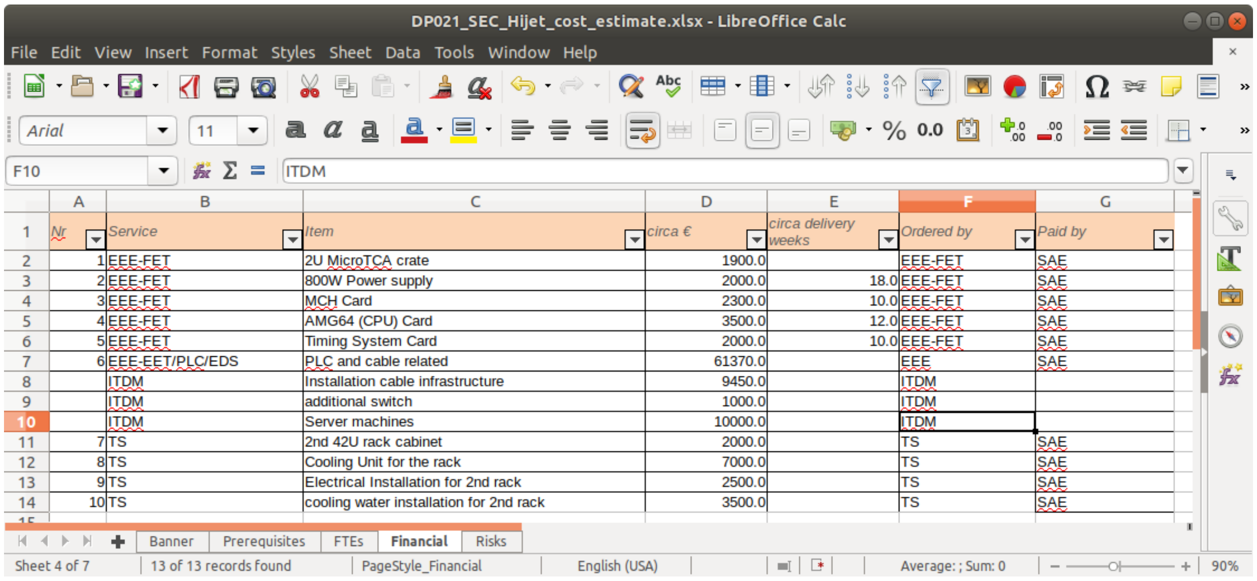


DP021_SEC_Hijet_cost_estimate.xlsx - LibreOffice Calc

| | Nr | Service | Subject | Type | FTEdays | Assigne | Other groups | Comments |
|---|---|---|---|---|---|---|---|---|
| 4 | 3 | TS | Rack ordering | Task | 1.0 | | | |
| 5 | 4 | TS | Rack cooling and power connections | Task | 5.0 | | | |
| 7 | 6 | EEE-EET | Planning of PLC, electrical documenation | Task | 48.0 | | | |
| 8 | 7 | EEE-EET | Cable installation management and handover SAT | Task | 15.0 | | | |
| 10 | 9 | ITDM | Installation cable ordering | Task | 0.5 | | | |
| 11 | 10 | ITDM | Installation management and handover SAT | Task | 0.5 | | | |
| 13 | 12 | ITDM | Install IT active components | Task | 5.0 | | | Control infrastructure (control-server, PC, switches...) installed |
| 14 | 13 | CTRL | Karabo installation on LAB PC | Task | 1.0 | | | Karabo deployment and test with non-PLC h/w |
| 15 | 14 | EEE-FET | Technical commission of MTCA crate | Task | 0.5 | | | 1. Should a latest design 1U crate be used?<br>2. If external signals will be used together with the timing system - for example, disable trigger signals until liquid jet is ready. If so, we should discuss how the setup could look like. |
| 17 | 16 | EEE-PLC | Install PLC | Task | 1.0 | | | PLC installed |
| 18 | 17 | EEE-EDS | Install Beckhoff modules (crates) | Task | 5.0 | | | Crate module installation |
| 19 | 18 | EEE-PLC | Commission loop | Task | 1.0 | | | Loop commissioned |
| 21 | 20 | CTRL | HEXAPOD commissioning | Task | 0.5 | | | |
| 22 | 21 | CTRL | IMAGING Zyla commissioning | Task | 0.5 | | | Uses AstroTech interface |
| 23 | 22 | CTRL | IMAGING Basler and LED commissioning | Task | 0.5 | | | |
| 24 | 23 | CTRL | INJMOV manipulator commissioning | Task | 0.1 | | | |
| 25 | 24 | CTRL | INLINEMIC 6 Micronix axes commissioning | Task | 0.5 | | | Assumes controller PLC integrated |
| 26 | 25 | CTRL | LQDJET vacuum systems | Task | 3.0 | | | Commissioning and MDL s/w and scene development |
| 27 | 26 | CTRL | SACT 9 generic Smaract axes | Task | 1.0 | | | Assumes the Smaract is the SPB 2020 version |
| 28 | 27 | CTRL | SIDEMIC 3 Micronix axes | Task | 0.5 | | | Assumes controller PLC integrated |
| 30 | 29 | CTRL | IMAGING Photron AX200 mini integration | Task | 20.0 | | | Estimate as camera with, as yet, unknown integration capabilities |
| 31 | 30 | CTRL | Shimadzu HPLC system | Task | 10.0 | | | Estimate as unknown integration capabilities, is integration required? |
| 32 | 31 | DA | Brainstorm requiremenst and solutions | Task | 2.0 | | | Understand with PO future requirements, if any |
| 34 | 33 | DATA | Devices, Cameras and DAQ operational | Task | 0.5 | | | |
| 36 | 35 | CTRL | Management | Task | 0.5 | | | |
| 37 | 36 | CTRL | Meetings | Task | 1.5 | | | 12 x 1h meetings during project lifetime |

Sheet 3 of 7 | 36 of 36 records found | PageStyle_FTEs | English (USA) | Average: ; Sum: 0 | 90%

Fig. 3: Cost-estimate FTEs task sheet

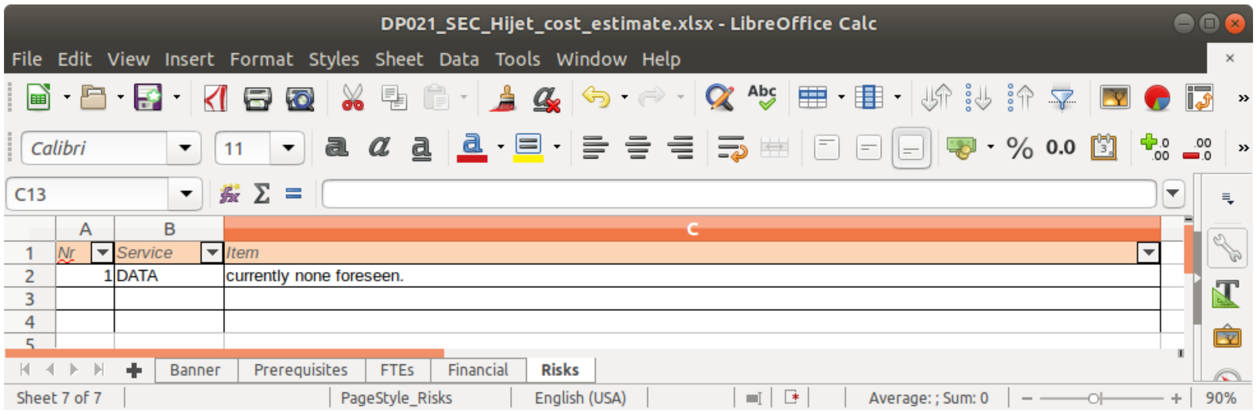


DP021_SEC_Hijet_cost_estimate.xlsx - LibreOffice Calc

| | Nr | Service | Item | circa € | circa delivery weeks | Ordered by | Paid by |
|---|---|---|---|---|---|---|---|
| 2 | 1 | EEE-FET | 2U MicroTCA crate | 1900.0 | | EEE-FET | SAE |
| 3 | 2 | EEE-FET | 800W Power supply | 2000.0 | 18.0 | EEE-FET | SAE |
| 4 | 3 | EEE-FET | MCH Card | 2300.0 | 10.0 | EEE-FET | SAE |
| 5 | 4 | EEE-FET | AMG64 (CPU) Card | 3500.0 | 12.0 | EEE-FET | SAE |
| 6 | 5 | EEE-FET | Timing System Card | 2000.0 | 10.0 | EEE-FET | SAE |
| 7 | 6 | EEE-EET/PLC/EDS | PLC and cable related | 61370.0 | | EEE | SAE |
| 8 | | ITDM | Installation cable infrastructure | 9450.0 | | ITDM | |
| 9 | | ITDM | additional switch | 1000.0 | | ITDM | |
| 10 | | ITDM | Server machines | 10000.0 | | ITDM | |
| 11 | 7 | TS | 2nd 42U rack cabinet | 2000.0 | | TS | SAE |
| 12 | 8 | TS | Cooling Unit for the rack | 7000.0 | | TS | SAE |
| 13 | 9 | TS | Electrical Installation for 2nd rack | 2500.0 | | TS | SAE |
| 14 | 10 | TS | cooling water installation for 2nd rack | 3500.0 | | TS | SAE |

Sheet 4 of 7 | 13 of 13 records found | PageStyle_Financial | English (USA) | Average: ; Sum: 0 | 90%

Fig. 4: Cost-estimate Finances sheet


DP021_SEC_Hijet_cost_estimate.xlsx - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Calibri
11
C13
Banner
Prerequisites
FTEs
Financial
Risks
Sheet 7 of 7
PageStyle_Risks
English (USA)
Average: ; Sum: 0
90%

| Nr | Service | Item |
|---|---|---|
| 1 | DATA | currently none foreseen. |

Fig. 5: Cost-estimate Risk sheet

tance for DATA is as a sheet waver, it list technical shortfalls that the PO may not be aware of initially.

## 4.5 PTC tooling assisted management actions

PTC tooling assists the following management actions:

- pre-injection validation of cost-estimate sheet properties, their attributes and workflow organization
- injection of cost-estimate sheet FTE tasks into Redmine
- updating Redmine tasks attributes into the cost-estimate sheet

## 4.6 Project scheduling

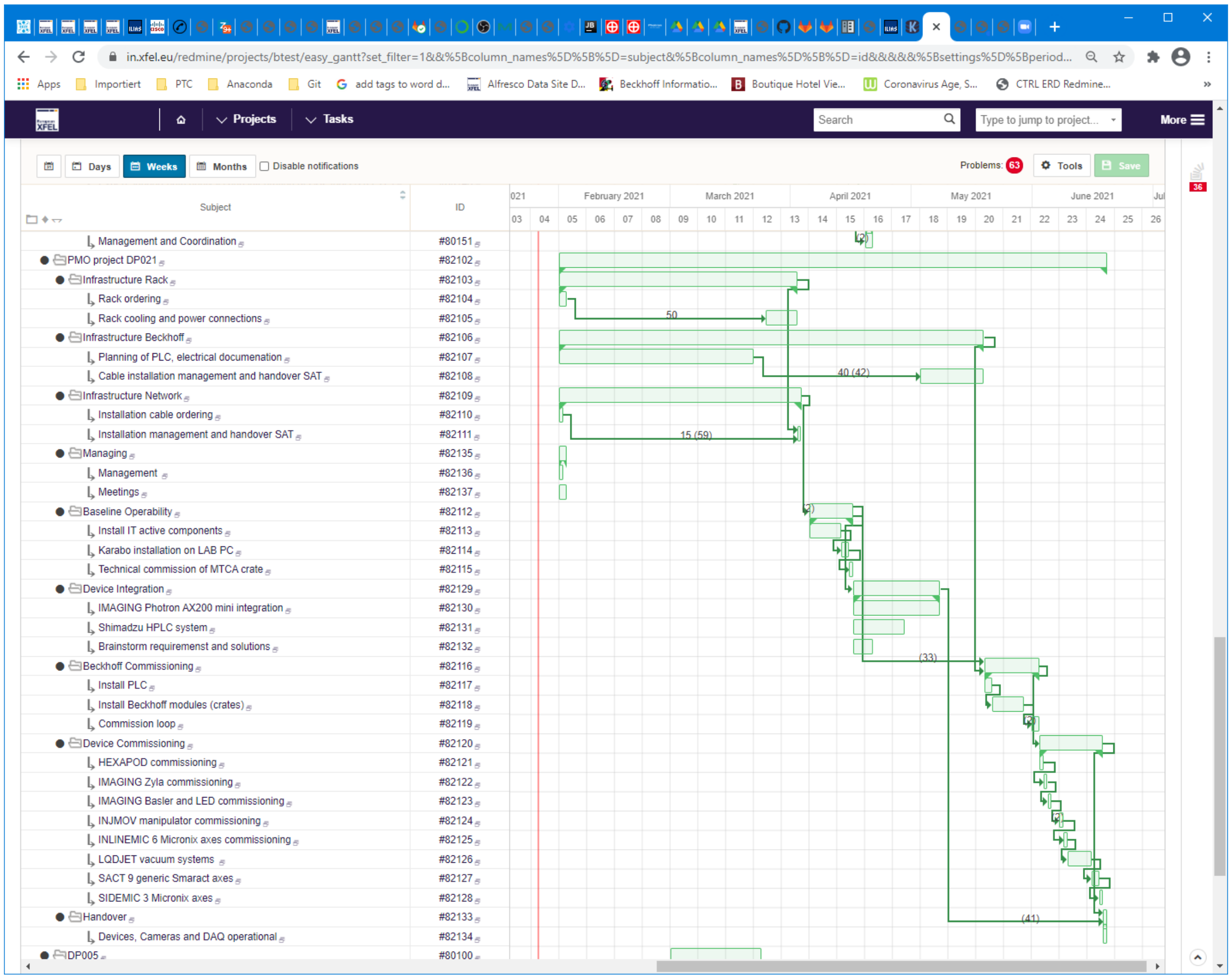
As stated above project scheduling is a DATA activity as it requires allocation of resources. This is discussed in a later chapter.

## 4.7 Project closout on completion

The cost-estimate can be filled with Redmine information using PTC tooling to evaluate the actual FTE cost of a task. Then, the *Comments* sheet should be filled out by DATA groups after which the updated cost-estimate is signed off by all DATA groups and sent to PMO and PO.

## 4.8 PTC Project time budgets

Initial experience with projects in 2020 (Schimadzu, AGIPD mini-half, HED pulsar in A.11, PES at PETRA, etc.) which are considered to be a representative sample w.r.t. DATA resource usage lead to the time budgets tabulated below. The table does not include time spent on project meetings, these are tasks in the cost-estimate.



| Subject | ID |
|---|---|
| Management and Coordination | #80151 |
| PMO project DP021 | #82102 |
| Infrastructure Rack | #82103 |
| Rack ordering | #82104 |
| Rack cooling and power connections | #82105 |
| Infrastructure Beckhoff | #82106 |
| Planning of PLC, electrical documenation | #82107 |
| Cable installation management and handover SAT | #82108 |
| Infrastructure Network | #82109 |
| Installation cable ordering | #82110 |
| Installation management and handover SAT | #82111 |
| Managing | #82135 |
| Management | #82136 |
| Meetings | #82137 |
| Baseline Operability | #82112 |
| Install IT active components | #82113 |
| Karabo installation on LAB PC | #82114 |
| Technical commission of MTCA crate | #82115 |
| Device Integration | #82129 |
| IMAGING Photron AX200 mini integration | #82130 |
| Shimadzu HPLC system | #82131 |
| Brainstorm requiremenst and solutions | #82132 |
| Beckhoff Commissioning | #82116 |
| Install PLC | #82117 |
| Install Beckhoff modules (crates) | #82118 |
| Commission loop | #82119 |
| Device Commissioning | #82120 |
| HEXAPOD commissioning | #82121 |
| IMAGING Zyla commissioning | #82122 |
| IMAGING Basler and LED commissioning | #82123 |
| INJMOV manipulator commissioning | #82124 |
| INLINEMIC 6 Micronix axes commissioning | #82125 |
| LQDJET vacuum systems | #82126 |
| SACT 9 generic Smaract axes | #82127 |
| SIDEMIC 3 Micronix axes | #82128 |
| Handover | #82133 |
| Devices, Cameras and DAQ operational | #82134 |
| DP005 | #80100 |

Fig. 6: Gantt view of the implementation workflow injected into Redmine.

| action | time budget [parties] |
|---|---|
| PTC sends organize DATA cost-estimate meeting including discision with PO on heads-up material required | 2 hr [PTC & PO] |
| DATA group representatives for cost-estimate read heads-up information. | 1 hr each [PTC and DATA groups] |
| PTC chaired cost-estimate meeting attendance. | 1.5 hr each [PTC, PO, & DATA groups, those w/o requirements can exit after confirming no resources are required at meeting] |
| Offline cost-estimte finalization | 2 hr [PTC, PO and DATA groups] |
| PTC sends final cost estimate to PMO | 1 hr [PTC] |
| PTC closeout cost-estimate updated | 1 hr [PTC & DATA groups] |
| PTC sends closeout to PMO | 2 hr [PTC] |

## 4.9 PMO meetings

To maintain communication a monthly meeting is timed to proceed the PMO - MB meeting by one week. The PTC aim for the meeting is to align in-progress project statuses and scheduling, receive notification of new projects requiring cost-estimates, and participate in project closeout summaries. PMO should have similar requirements for the meeting.

## 4.10 PTC email lists

The SYMPA mailing list service is used to assist communication by defining

- ptc-DATA, the list of DATA group contacts, e.g. for cost-estimate gathering
- ptc-DPnnn-po, the PO contact for a project
- ptc-DPnnn, the PO side list of contacts for a project
- ptc-pmo, the PMO contacts

Mail filters applied to the email subject are used to shape PTC activity.

- DPnnn, for project emails and Redmine tickets - do not include multiple poject IDs in a mail subject!
- PTC-PMO, for PTC communication

CHAPTER 5

# PTC - XO projects

The workflow provided by XO is shown below.

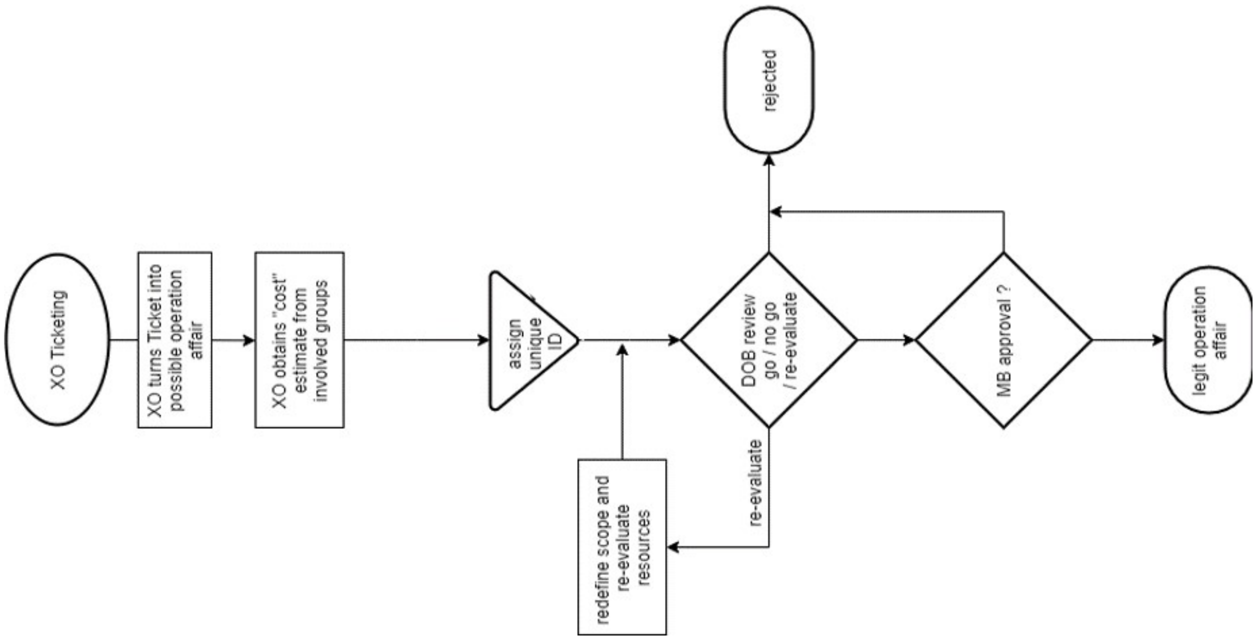

XO Ticketing
XO turns Ticket into possible operation affair
XO obtains "cost" estimate from involved groups
assign unique ID
redefine scope and re-evaluate resources
re-evaluate
DOB review go / no go / re-evaluate
rejected
MB approval ?
legit operation affair

Fig. 1: XO procedure workflow

## 5.1 XO workflow as seen by DATA

Long procedures have the same requirements on DATA, therefore the text describing DATA's handling of the XO procedure is largely the same as the PMO procedure text and only differences are noted below. If entire sub-sections are missing they are identical to the PMO versions.

Differences regarding the workflow are:

- the unique project identifier for XO projects is OAnnn (Operations Affair)
- the project identifier **must be assigned before cost-estimates gathering.** and not as shown after it.
- Ligit means to be implemented.

## 5.2 XO meetings

In addition to the XO-Facility weekly operations meetings it makes sense to have a regular monthly meeting with the aim of maintaining information exchange on project status, resource availability, future work, etc.

## 5.3 PTC email lists

The SYMPA mailing list service is used to assist communication by defining

- ptc-OAnnn-po, the PO contact for a project
- ptc-OAnnn, the PO side list of contacts for a project
- ptc-xo, the XO contacts

Mail filters applied to the email subject are used to shape PTC activity.

- OAnnn, for project emails and Redmine tickets
- PTC-XO, for PTC communication

Examples of PTC procedure email folders and filters are shown below.

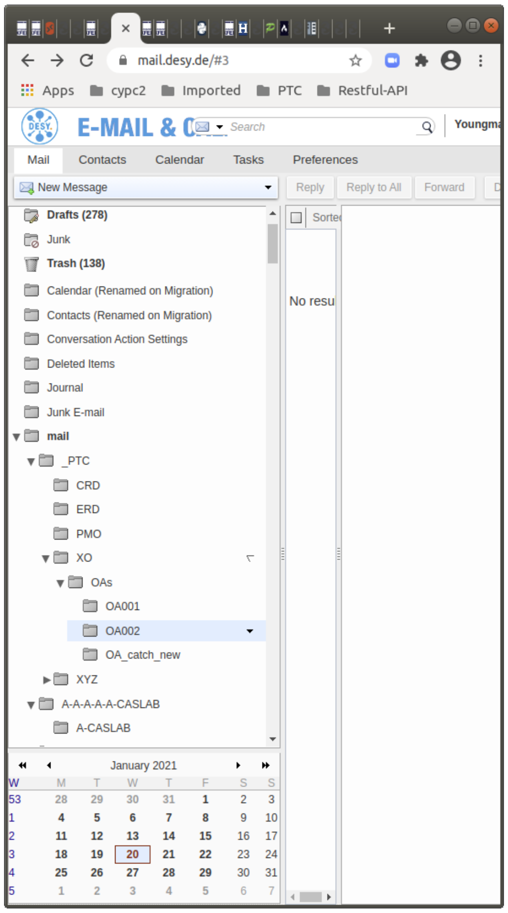
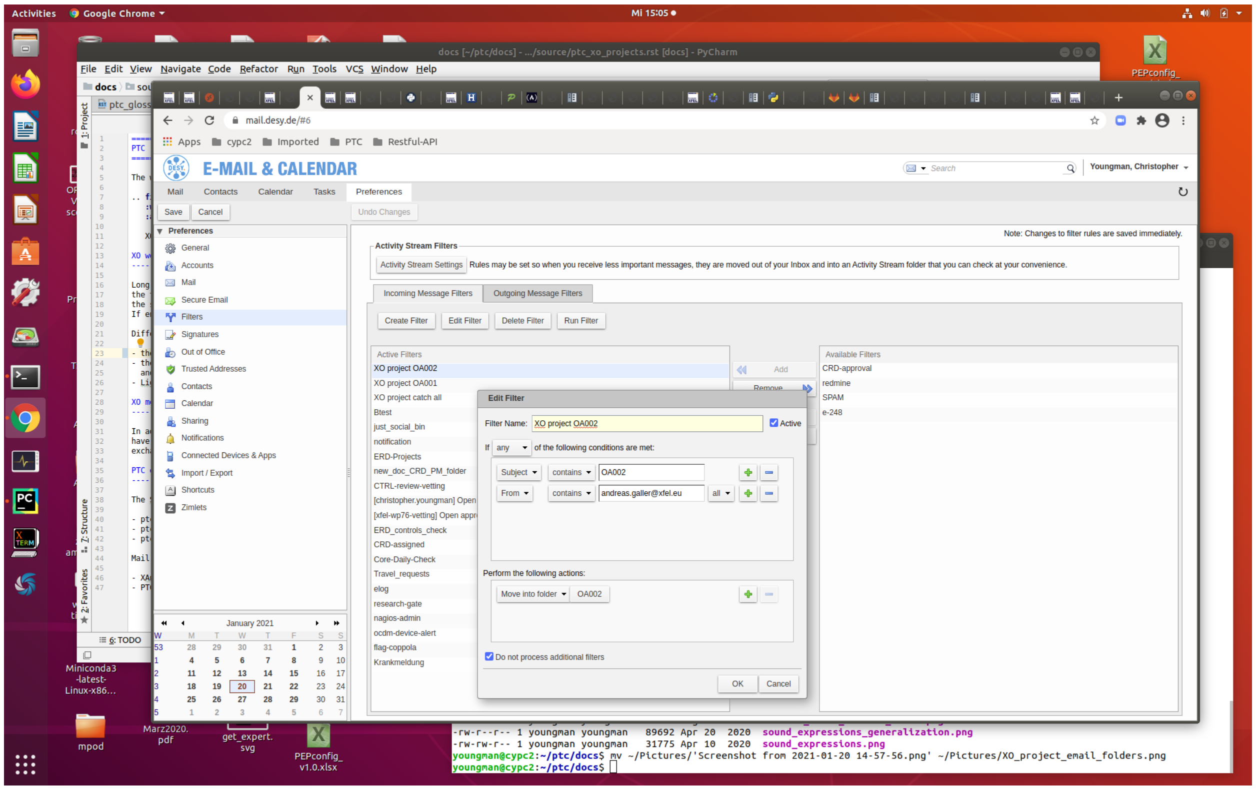


Fig. 2: Example PTC-XO email folders



Note: Changes to filter rules are saved immediately.

Activity Stream Filters

Activity Stream Settings: Rules may be set so when you receive less important messages, they are moved out of your Inbox and into an Activity Stream folder that you can check at your convenience.

Active Filters: XO project OA002, XO project OA001, XO project catch all, Btest, just_social_bin, notification, ERD-Projects, new_doc_CRD_PM_folder, CTRL-review-vetting, [christopher.youngman] Open, [xfel-wp76-vetting] Open appr, ERD_controls_check, CRD-assigned, Core-Daily-Check, Travel_requests, elog, research-gate, nagios-admin, ocdm-device-alert, flag-coppola, Krankmeldung

Available Filters: CRD-approval, redmine, SPAM, e-248

Edit Filter

Filter Name: XO project OA002 [✓ Active]

If any of the following conditions are met:

- Subject contains OA002
- From contains andreas.galler@xfel.eu all

Perform the following actions:

- Move into folder OA002

[✓] Do not process additional filters

```
-rw-r--r-- 1 youngman youngman   89692 Apr 20  2020  sound_expressions_generalization.png
-rw-rw-r-- 1 youngman youngman   31775 Apr 10  2020  sound_expressions.png
youngman@cypc2:~/ptc/docs$ mv ~/Pictures/'Screenshot from 2021-01-20 14-57-56.png' ~/Pictures/XO_project_email_folders.png
youngman@cypc2:~/ptc/docs$ 
```

Fig. 3: Example PTC-XO email filter

CHAPTER 6

# PTC - ERD projects

The ERD procedur workflow is shown below and a simplified PTC version follows.

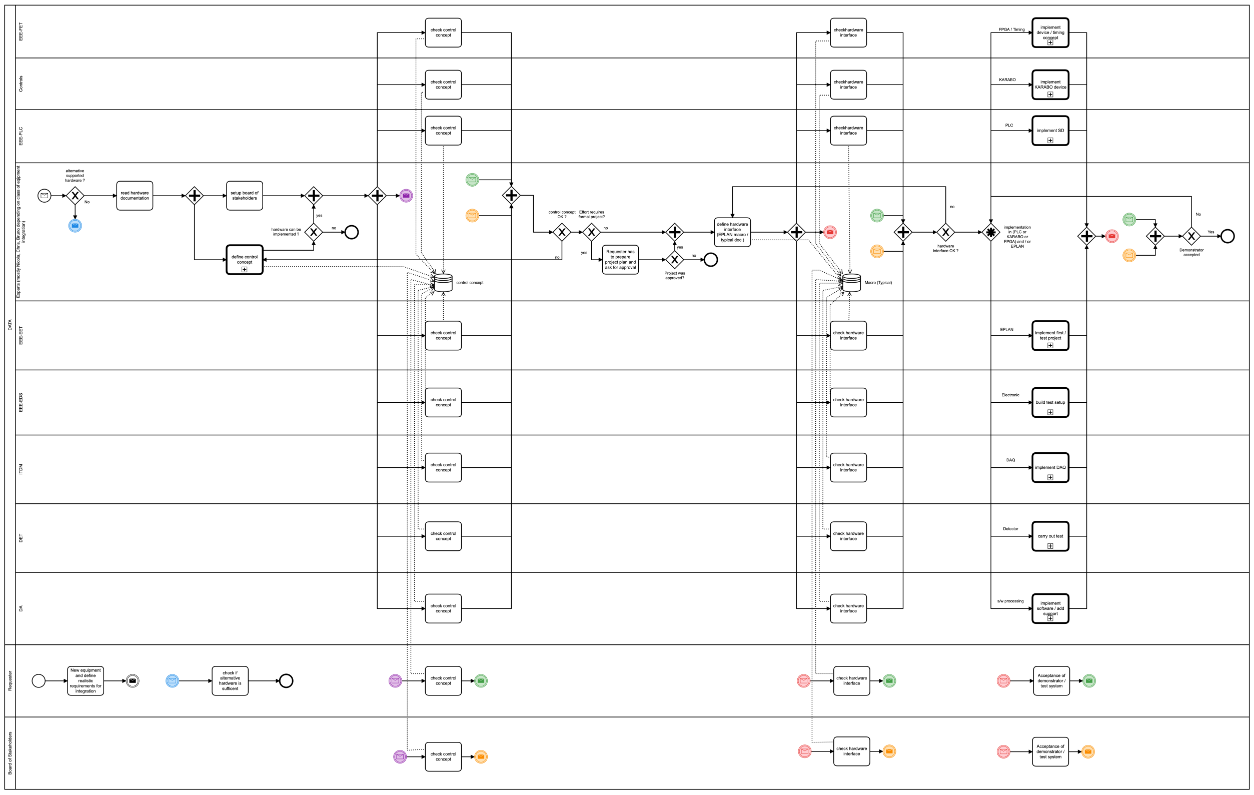

DATA
EEE-FET
Controls
EEE-PLC
Experts (mostly Nicola, Chris, Bruno depending on class of eqipment integration)
EEE-EET
EEE-EDS
ITDM
DET
DA
Requester
Board of Stakeholders
alternative supported hardware ?
No
read hardware documentation
setup board of stakeholders
define control concept
hardware can be implemented ?
yes
no
check control concept
control concept
control concept OK ?
Effort requires formal project?
no
yes
Requester has to prepare project plan and ask for approval
Project was approved?
define hardware interface (EPLAN macro / typical doc.)
checkhardware interface
check hardware interface
Macro (Typical)
hardware interface OK ?
implementation in (PLC or KARABO or FPGA) and / or EPLAN
FPGA / Timing
implement device / timing concept
KARABO
implement KARABO device
PLC
implement SD
EPLAN
implement first / test project
Electronic
build test setup
DAQ
implement DAQ
Detector
carry out test
s/w processing
implement software / add support
Demonstrator accepted
Yes
New equipment and define realistic requirements for integration
check if alternative hardware is sufficent
Acceptance of demonstrator / test system

Fig. 1: EEE's ERD procedure workflow

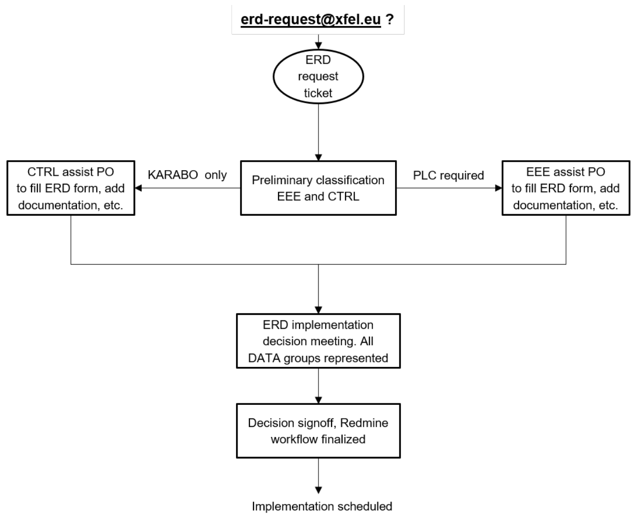

erd-request@xfel.eu ?
ERD request ticket
Preliminary classification EEE and CTRL
KARABO only
PLC required
CTRL assist PO to fill ERD form, add documentation, etc.
EEE assist PO to fill ERD form, add documentation, etc.
ERD implementation decision meeting. All DATA groups represented
Decision signoff, Redmine workflow finalized
Implementation scheduled

Fig. 2: PTC's simplified ERD procedure workflow

## 6.1 ERD workflow as seen by DATA

As stated in the 'big picture' chapter PTC acts as an observer to the ERD and CRD short procedures. The single proviso is that the short procedures must comply with PTC's inteface requirements. For ERD this should be quickly achievable as PTC tooling was initially developed by dry-running a ERD like workflow definition with the CRD workflow joining later. The PTC interface requirements on ERD are related to the following points:

- the Redmine representation of a request task is defined to be a single task injected into Redmine via an email (address?) with template content specifying key elements of the requested project.
- the Redmine task and sub-task structure is defined for the different equipment type implementations (Karabo s/w and no PLC, PLC with existing Karabo s/w interface, and PLC with new Karabo s/w interface).
- the 'Elvis' Redmine *tracker* type is used for workflow tasks with its statuses and developer and manager allowed transition definitions.

## 6.2 Role definitions

PTC:

- is an observer viewing the ERD procedure through the PTC interface
- performs DATA resource scheduling with ERD PEPs and GL using ERD and other procedure views

PEPs:

- ensure that new equipment is fed through the ERD procedure
- comply with the PTC interface requirements
- ensure that the ERD procedure (tasks execution, meetings, etc.) is efficiently and managed
- assign developers to tasks based on DATA resource scheduling decisions

## 6.3 PTC tooling assisted management actions

PTC tooling should be used to assist management actions.

The Redmine task count in 50 ERD implementation workflows analysed currently lies in the range 10 - 30. Manually creating and managing these tasks is a significant workload which can be reduced with:

- workflow_injection to inject workflow tasks (subject, assignee, etc.) defined in a spread sheet into Redmine.
- workflow_update to update the spreadsheet from Redmine during and on closeout of the project.

Other assist options are:

- migrating Cancelled and Rejected projects to their final Redmine locations
- tune the PTC interface configuration options to match the needs of the procedure to produce project status summaries, meeting agendas, etc.

## 6.4 Project scheduling

This activity is a responsibility of the PEPs concerned albeit with the requirement that a consensus of scheduling is reached with DATA's PTC.

## 6.5 Project closeout

ERD projects should be relatively short, this should be confirned by ensuring that time spent information is aggregated during project closeout.

## 6.6 ERD meetings

ERD decision meetings are currently held twice per month and PTC will attend these meetings.

## 6.7 PTC email lists

The SYMPA mailing list service is used to assist communication by defining

- ptc-erd, the ERD-PEP contacts

Mail filters applied to the email subject are used to shape PTC activity.

- PTC-ERD, for PTC communication

CHAPTER 7

# PTC - CRD projects

## 7.1 CRD workflow as seen by DATA

As mentioned in the ERD chapter PTC's view of the CRD procedure followed on from ERD, therefore the text describing PTC's handling of the CRD procedure is largely the same and only differences are noted below.

- completion of 'Elvis' *tracker* tests for CRD are needed.
- the CRD workflow shown is new and replaces the one used until mid-Nov 2020
- the new procedure calls the ERD procedure when a device in the CRD has no existing ERD.
- additionally the new procedure flattens the decision structure by ensuring each DATA groups has to signoff the request, and removes the requirement on CTRL to vet the CRD before submission which is now the respoonsibility of project owner.

## 7.2 Role definitions

PEPs:

- establish a CRD Redmine representation compliant with the PTC interface requirements for the procedure workflow.

## 7.3 Project Scehduling

This activity is a responsibility of the PEPs concerned albeit with the requirement that a consensus of scheduling is reached with DATA's PTC.

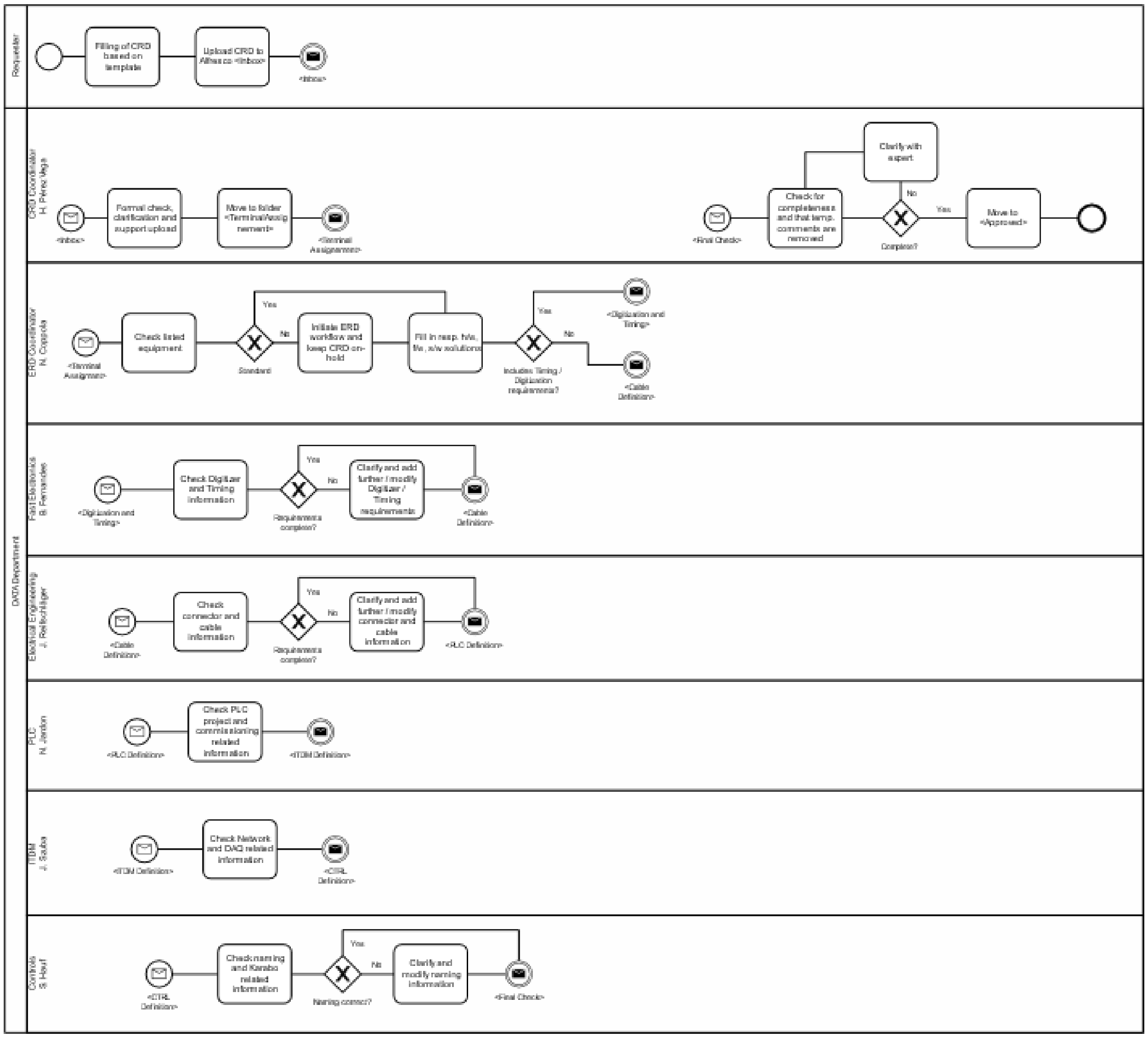

Requester
Filling of CRD based on template
Upload CRD to Alfresco <Inbox>
<Inbox>
DATA Department
CRD Coordinator H. Pérez Vega
<Inbox>
Formal check, clarification and support upload
Move to folder <TerminalAssignment>
<Terminal Assignment>
<Final Check>
Check for completeness and that temp. comments are removed
Clarify with expert
No
Yes
Complete?
Move to <Approved>
ERD Coordinator N. Coppola
<Terminal Assignment>
Check listed equipment
Yes
No
Standard
Initiate ERD workflow and keep CRD on-hold
Fill in resp. h/w, f/w, s/w solutions
Yes
No
Includes Timing / Digitization requirements?
<Digitization and Timing>
<Cable Definition>
Fast Electronics B. Fernandes
<Digitization and Timing>
Check Digitizer and Timing information
Yes
No
Requirements complete?
Clarify and add further / modify Digitizer / Timing requirements
<Cable Definition>
Electrical Engineering J. Reifschläger
<Cable Definition>
Check connector and cable information
Yes
No
Requirements complete?
Clarify and add further / modify connector and cable information
<PLC Definition>
PLC N. Jardon
<PLC Definition>
Check PLC project and commissioning related information
<ITDM Definition>
ITDM J. Szuba
<ITDM Definition>
Check Network and DAQ related information
<CTRL Definition>
Controls S. Hauf
<CTRL Definition>
Check naming and Karabo related information
Yes
No
Naming correct?
Clarify and modify naming information
<Final Check>

Fig. 1: EEE's CRD procedure workflow

## 7.4 Project closeout

CRD project time spends should be longer than ERD and shorter than PMO projects. Project closeouts should be monitored to confirm this.

## 7.5 CRD meetings

CRD decision meetings aiming at R3dmine implementation and PTC interface compliance are currently held 'as required' with the PEP and other CRD involved collegues.

## 7.6 PTC email lists

The SYMPA mailing list service is used to assist communication by defining

- ptc-crd, the ERD-PEP contacts

Mail filters applied to the email subject are used to shape PTC activity.

- PTC-CRD, for PTC communication

# CHAPTER 8

# PTC interface

This section develops the PTC interface exposed to project procedures. Discussion with procedure owners showed that they are either already using Redmine (ERD), intend to (CRD), or are willing (PMO and XO) to use it for the project procedure workflow management interface to PTC.

## 8.1 Redmine software at EuXFEL

Redmine version 3.3.3 project management software is available and is used for single tasks, e.g. email driven ticket creation addressing daily operating activities, through to multi-task workflow management.

## 8.2 Procedure-Redmine name clashes

Unfortunately Redmine and EuXFEL names for things often clash. The following table defines the name meanings used throughout this document.

| name | meaning in this document |
|---|---|
| *project* | Project owners objective, e.g. Integrate a commercial camera into EuXFEL control, timing synchronization, and DAQ system. It is **not** the Redmine definition. |
| *procedure* | The workflow triggered by the project owner to implement the *project*, e.g ERD, PMO… |
| *workflow* | Each *project* in a procedure runs through a set of tasks which result in its completion. The set is the *workflow* and it can be the same for different projects, but does not have to be. |
| *task* | A single work deliverable in a project's *workflow*, e.g. implement external trigger. |
| *issue* | A Redmine *issue* is similar to a *task*, but issues are implemented by Redmine as different *tracker* types. |
| *tracker* | Different Redmine *tracker* types express different *issue* properties, e.g. *status* and *transition*, needed to allow *issue* to work in a *workflow* |
| *status* | The Redmine state of an *issue*, e.g. Open , In progress, Closed… |
| *transition* | The Redmine transition to another *issue* state *status* which are allowed for the *tracker* implemented. |
| *task-group* | A set of tasks which are grouped together for implementation handling. Their Redmine equivalent is an *issue* with sub-issues (aka children). |
| *request-task* | A *task* w/o parent or children tasks is a request for a project. |
| *project-anchor* | A *task-group* w/o parent is the top-level *task* of a *project*. Remember that *task-groups* have children! |
| *developer* | A worker who implements a *task*. |
| *manager* | A manager of *project* progress. |

## 8.3 Fine tuning requirements

A number of fine tuning changes w.r.t. to Redmine, both usage and internal definitions, are needed to allow PTC to expose a single interface to all procedures.

## 8.4 PTC interface workflow requirements

These are few and largely associated with formalizing the initial request and review process. For CRD and ERD procedures this requires:

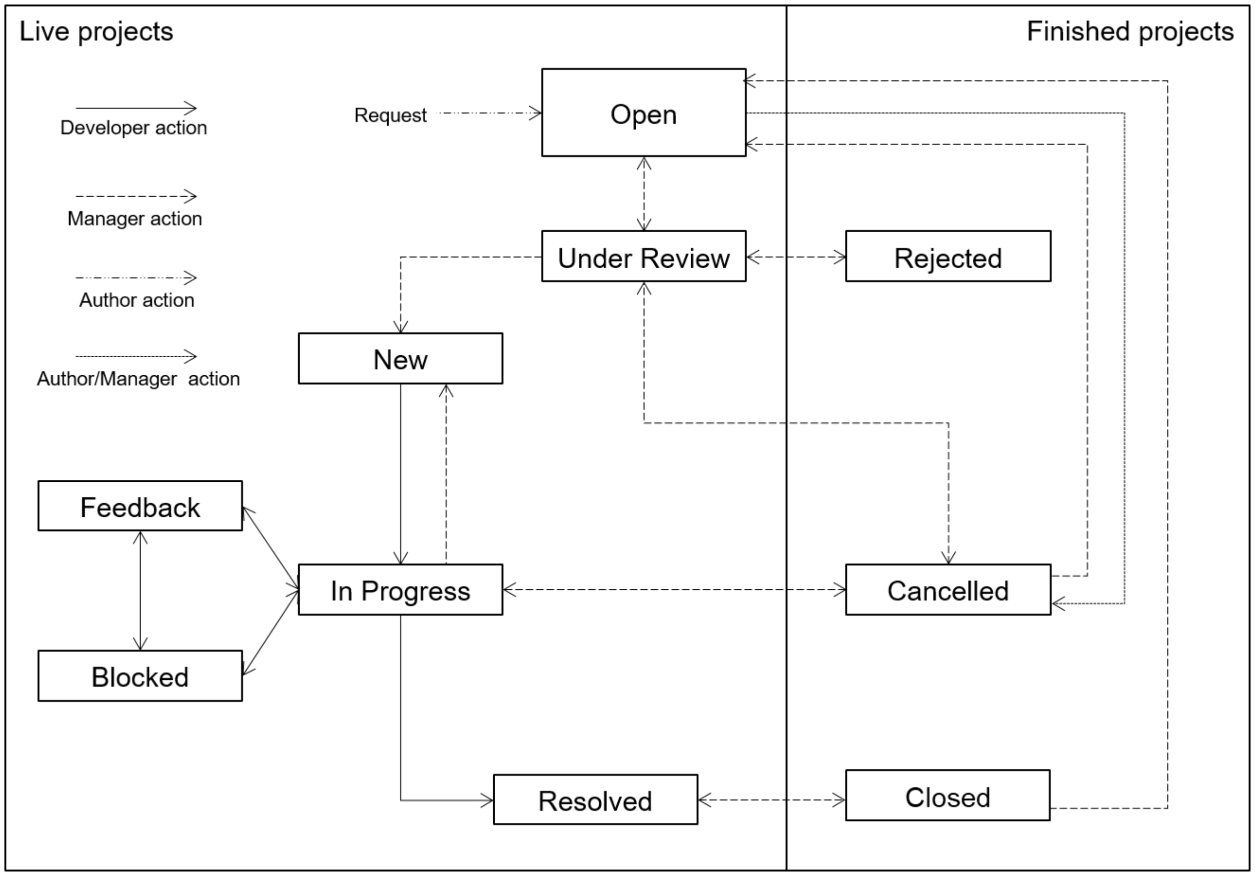
- *request-task* creation should be created via a browser form and failing that via email.
- the *request-task* should remain a parentless and childless *task* whilst *status* is Open. In the ERD procedure Open is used by the PEPs to idenitify who assists the project owner filling out the ERD document. For the CRD procedure **text needed**
- the *request-task* migrates to a *project-anchor* when the *status* is changed to Under Review and sufficient sub-task structure is added for the review process. For the ERD procedure the review meeting requires all DATA groups to participate. For the CRD proceedure **text needed**
- the *project-anchor* status is changed to New when the *project* is accepted, otherwise it is Rejected.
- the sub-task structure of the *project-anchor* is expanded to satisfy the requirements of the project by the procedure manager, the PTC interface requires only that the *project-anchor status* and *milestone* align with the *project* state. This can be update is rule driven and can be automated.

- Rejected, Cancelled, and Closed *projects* are moved out of the Redmine *project-id* to the Finished *project-id* with the *project-anchor status* and *milestone* properties updated leaving lower level properties unchanged.

For PMO and probably any other long procedures there is no PTC participation in the review process and the PTC creates a *project-anchor* and associated sub-task structure that represent the *project* w.r.t. DATA resource tasks needed. Otherwise the requirements for long procedures are identical (is PMO's on-hold needed) to those of CRD and ERD processes.

## 8.5 Workflow tracker selection

Following discussion with CRD and ERD procedure owners the Redmine 'Task' tracker previously used is replaced by the 'Elvis' tracker which implements the workflow shown below.



Additionally Elvis requires the use of:

- the (Redmine) *Category* which is a string associated with a workflow task with content selected from a pull-down menu of predefined sentences. The aim is to allow the developer to expose the current state of the work on the task to workflow viewers. This mechanism is better suited to this purpose than other possibilities like %_done which is too subjective when details of the task are unknown to the non-developer. The concept follows a requirement specified by PMO.
- the (Redmine) integer custom-field *Duration (hrs)* in which the developer inserts the integrated hours spent on the task. This field allows the actual time spent on the task between the *start_date* and the *due_date*. Duration allows the comparison of expected task time and actual time to be automatically calculated at project closeout. It does not allow the efficiency of the deveolper to be calculated. The Redmine time spent plugin is not used.

## 8.6 How different roles work with Elvis

Developer:

- When the task is assigned to a developer the status is *New*. Note that if a change of developer is required the status may not be *New* when they start
- When working on the task the status is *In Progress*
- If work is stopped due to an internal cause like the PM must act (e.g. provide the test hardware, repair broken hardware, report on a developer implementation question, etc.) or a Dept group must act (e.g. fix faulty s/w, f/w or h/w) set the status to *Blocked*
- If work is stopped for an external reason (e.g. a design issue must be resolved by PM or PO, h/w vendor is required to update f/w, etc.) set the status to *Feedback*
- If the task is finished the status should be set to *Resolved*

Redmine procedure managers (PTC for long and PEPs for short procedures):

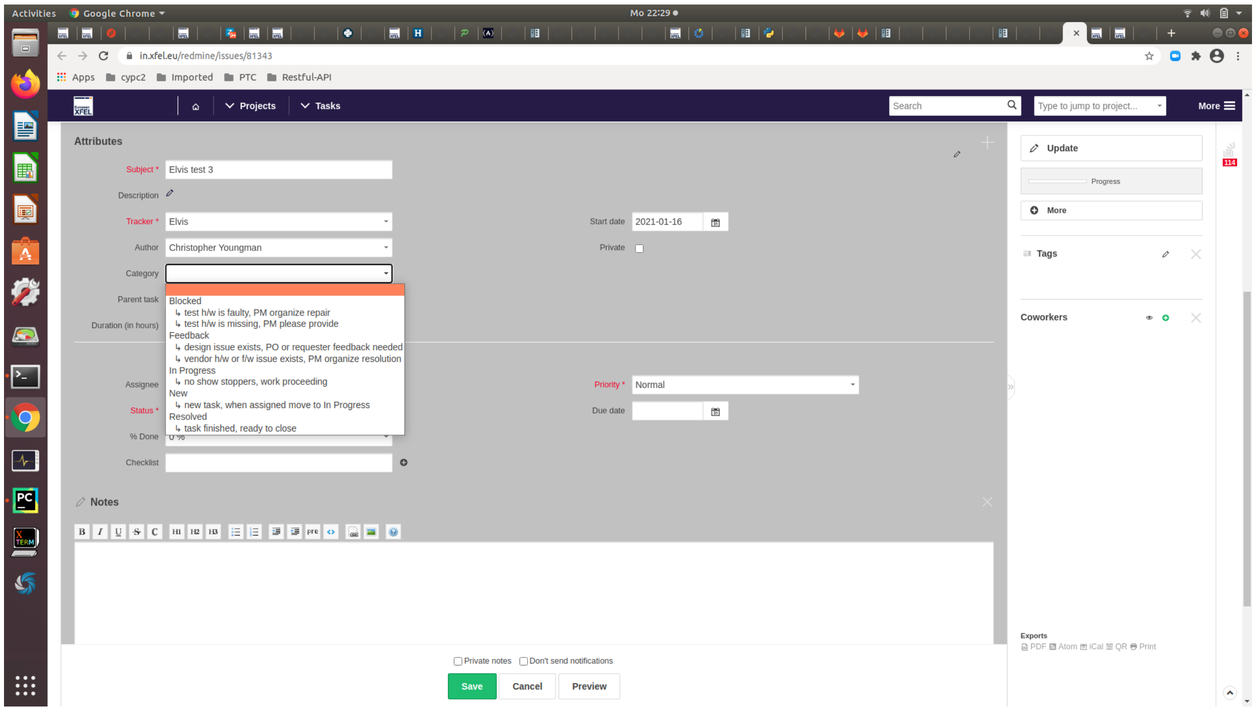
- When the request is injected the status is *Open*
- During Open any tobe dones with the request (e.g. documentation) should be resolved and when ready the task status set *Under Review* with sufficient workflow task extension to allow the review (e.g. the ERD stakeholder board)
- after review the status should be set *New* or *Rejected*.
- For *New* additional workflow tasks should be added (e.g. ERD addition of implementation tasks) w/o Assignee assignment. The workflow is updated in Redmine w/o assignees until it can be scheduled.
- Managers should remember that there may be many implementation tasks (each in *New*, *In Progress*, *Feedback*, *Blocked*, or *Resolved*) and managers must ensure that implementation task status is correctly reflected by the anchor task status by using *RAL* aspects of the tooling.

Both developers and managers must set task (Redmine) categories when tasks statuses are changed by using an appropriate Category sub-class item. There is a sub-class for each Elvis status value and a category corresponding to the status value should be set - unfortunately it is not possible to liit the choice to the actual status value. The following figure illustrates setting a category.

## 8.7 Additional PTC interface requirements

These aim at improving PTC work efficiency and use the python tooling described later. The functionality provided is:

- rule based automation of Redmine procedure handling
- executive summaries of all project of a procedure.
- rule based automation of meeting agendas
- injection of project work flow tasks defined by rules or in appropriately structured Excel spreadsheets. The latter are particularly suited for PMO *projects*, where cost-estimate FTE task estimates also specify resource time usage.


in.xfel.eu/redmine/issues/81343

Attributes

Subject * Elvis test 3

Description

Tracker * Elvis

Author Christopher Youngman

Category

Blocked
↳ test h/w is faulty, PM organize repair
↳ test h/w is missing, PM please provide
Feedback
↳ design issue exists, PO or requester feedback needed
↳ vendor h/w or f/w issue exists, PM organize resolution
In Progress
↳ no show stoppers, work proceeding
New
↳ new task, when assigned move to In Progress
Resolved
↳ task finished, ready to close

Start date 2021-01-16

Private

Priority * Normal

Due date

Fig. 1: Redmine task update Category selection (preliminary content)

## 8.8 Further Alfresco usage

Alfresco usage by procedure owners is not affected by the PTC-Redmine interface and the use of tags to allow jquery network identification of the folder of procedure project documents should continue.

CHAPTER 9

# PTC - Redmine tooling

## 9.1 Tooling aims

The PTC role requires a uniform interface to the various project procedures which, as detailed previously, are provided by Redmine. The extra tooling required aims at:

- applying rules to workflow creation and operation, e.g. task names, task and sub-task structures
- automating activity that is repetitive, e.g. creating workflow task structures
- levering the information in Redmine to remove secretary work, e.g. produce status summaries and agendas for meetings.

The **Rules, Automation, and Levering** aim will be a key feature of the tooling provided and the **RAL** concept will accompany much of the work required by PTC. RAL is a good catch-phrase, reality may fall short.

The above are typical *management* requirements on project management software and presumably available in Redmine, but may require additional plugins or the Redmine Pro installation which are not available to us. Listening to comments made by project procedure contacts and ITDM experts, this seems to be the case. Therefore keeping to the *fine tune rather than change* direction the decision was made to investigate providing the tooling using external python tools, which had the advantage of seeing how to work with Redmine from a new viewpoint. The work model which crystallized out being:

- use the Redmine browser interface for single task activities typical of those performed by assigned task workers (developers)
- use the python tooling for assisting manager activities
- PTC uses the tooling to simplify viewing of projects in multiple procedure to enable summary and schedule work

## 9.2 Tooling status

Usable but in development is a fitting description. Most of the features required to develop a script based interface toolset have been worked with. Next steps are focus on code cleaning, fine tuning development of the tooling and interface working with the PEPs, and aligning Redmine access to the PTC-PEP requirements. The final step is to see if the PTC role of *coordinating* project resources, that is summarizing and efficient scheduling, can be achieved.

## 9.3 Python-Redmine library

Redmine.org's Restful-API. lists a number of candidate libraries.

PTC uses Python-Redmine as this is referred to as a library which supports 100% features of Redmine's REST API on the Redmine.org site. It provides a simple but powerful Pythonic API inspired by a well-known Django ORM and is thoroughly tested.

## 9.4 Installing the library and needed extras

To get a working environment:

- python-redmine, openPyxl and numpy libraries must be installed into with Python 3.5+
- access to EuXFEL's Gitlab PTC project must be established and Git-client s/w installed.

## 9.5 Linux

Most Unix users will be familiar with the pip installation of python packages, simply:

- *pip install python-redmine*
- *pip install openpyxl* (for Excel-Python)
- *pip install numpy* (for extra functionality, e.g. workdays per week)

An attractive way of getting 2-for-1 is to install the Karabo GUI, and perform the above pip installs in it.

Git seems to come with Ubuntu, for me this client works:

```
$ git --version
git version 2.17.1
```

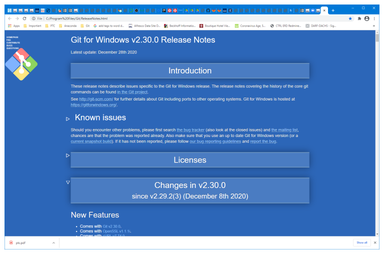
## 9.6 Windows 10

Finally a Windows 10 Python installation from scratch could also be used by perform the following steps.

- To get a running installation of python installed on windows 10 install MiniConda from conda For my test the MiniConda location was C:\Users\account\Miniconda3 and Python version 3.8 was installed. For a user-friendly Python on Windows intro try here
- Now hit the windows button, the one between Control and Alt on English keyboards, under 'A' the entry Anaconda3 (64 bit) use the pulldown menu to select Anaconda Prompt (miniconda3) and start a prompt terminal. The terminal knows where the python is which you've just installed.
- To test but also see which packages are installed with miniconda enter 'pip list'. Pip is there, but not python-redmine.
- Install python-redmine and the extra pips listed above by performing the pip installs.

Note where the Miniconda installation is (e.g. C:UsersusernameMiniConda)

A nicely featured Git-client can be installed from git-client



Git for Windows v2.30.0 Release Notes

Latest update: December 28th 2020

Introduction

These release notes describe issues specific to the Git for Windows release. The release notes covering the history of the core git commands can be found in the Git project.

See http://git-scm.com/ for further details about Git including ports to other operating systems. Git for Windows is hosted at https://gitforwindows.org/.

Known issues

Should you encounter other problems, please first search the bug tracker (also look at the closed issues) and the mailing list, chances are that the problem was reported already. Also make sure that you use an up to date Git for Windows version (or a current snapshot build). If it has not been reported, please follow our bug reporting guidelines and report the bug.

Licenses

Changes in v2.30.0
since v2.29.2(3) (December 8th 2020)

New Features

- Comes with Git v2.30.0.
- Comes with OpenSSL v1.1.1i.

Fig. 1: Git for Windows fly sheet.

To install:

- download the exe and follow the instructions taking “next” recommended answers all the way.
- hit the windows button and start the “Git Bash” shell terminal. This is magnificent - it’s a Windows CLI!
- now download the PTC project’s script and templates projects from Gitlab into a folder of your choice (this assumes you’ve been added as a member to the PTC project in EuXFEL’s Gitlab). In the example below the folder is created with mkdir, but you could have added the folder using the usual windows new folder method.

```
$ pwd
$ mkdir pep_stuff
$ cd pep_stuff
$ git clone xyz
```

Where xyz you need to find by looking in https://git.xfel.eu/gitlab/ptc/scripts for the clone https address of the scripts sub-project, currently (but may change) it’s “https://git.xfel.eu/gitlab/ptc/scripts.git”

To use the Miniconda Python and pip extras simply add the path to it at the front of the PATH and check that pep_tool.py runs with:

```
$ export PATH="/c/Users/username/MiniConda:"$PATH
$ python pep_tool.py your-username your-password -h
```

If successful you should see the pep_tool.py sub-commands listed, see below.

## 9.7 pep_tool.py application

pep_tool aims at providing *RAL* functionality to PTC. As a first interaction with Python-Redmine and its interface to Redmine’s REST API it acts as a trial-and-error testbed, with stable functionality exposed and provisional or dangerous functionality not.

A look at the code will show how unfinished the application is.

- the config(uration) file is foreseen for ease customizing to the different procedures. Additional work is required.
- the API used for Redmine access has no session context and one has to remember this when using thge application.
- the trial-and-error testing of the API have left some half-baked ideas in the code, in particular the slim representation of Redmine issues which is foreseen as a way of handling missing property values and similar.

## 9.8 pep_tool configuration

Difference between procedures, e.g. the Redmine container project-id, are configured in a json formated configuration file.

## 9.9 pep_tool sub-command functionality groups

- ERD and CRD management: workflow_validate, workflow_inject, workflow_update
- PMO and XO cost-estimate management: cost_estimate_validate, cost_estimate_inject, cost_estimate_update.
- task snapshot: diff_snapshot, save_snapshot
- property query related: dump_custom_fields, dump_project, dump_sets, dump_tracker_statuses
- issue related: delete_issue, force_tracker, peek, set_status, traverse_issue
- slim related: slims_all, slims_ctrl, slims_without_parent, traverse_slim
- general workflow handling: add_anchor, add_request, dump_signatures, request_to_anchor, run_test
- summary and meeting related: meeting, summary

## 9.10 pep_tool sub-command options

Currently everything is in one python script - terrible but this will be broken down soon - but has the advantage of having only one place to look at!

pep_tool uses argparse to provide a user-friendly command-line interfaces which validates the arguments used, performs input string to type conversion, and provides help and usage messages and issues errors when give invalid arguments. Additionally its use allows only tested and non-dangerous functionality to be exposed.

```
$ python pep_tool.py youngman password -h
usage: pep_tool.py [-h]
               username password
               {add_anchor,add_request,delete_issue,diff_snapshot,
               dump_custom_fields,dump_project,dump_sets,dump_signatures,
               ...
               summary,traverse_issue,traverse_slim}
               ...

positional arguments:
    username              account with access rights to EuXFEL Redmine
    password              account password
    {add_anchor,add_request,delete_issue,diff_snapshot,dump_custom_fields,
    ...
    summary,traverse_issue,traverse_slim}

    add_anchor          add anchor task to project
    add_request         add request task to project
    delete_issue        delete Redmine task
    diff_snapshot       output differences current Slims to those recorded in
                        last save_snapshot
    dump_custom_fields  output redmine project custom fields
    dump_project        output the Redmine project content
```

(continues on next page)

(continued from previous page)

```
    dump_sets           output task sets: all, top, with parents, anchor and
                        lone
    dump_signatures     output signature of top-level tasks
    dump_tracker_statuses
                        output tracker statuses
    dump_users          output known Redmine user names and ids
    workflow_validate   validate DATA workflow xlsx
    workflow_compare    compare workflows
    workflow_update     update DATA workflow xlsx from Redmine
    workflow_inject     inject DATA workflow xlsx into Redmine
    workflow_leafs      returns leaf (id, start_date, due_date) list of anchor
    workflow_shift      shifts anchor workflow to new date
    workflow_push       push to DATA workflow xlsx
    dump_transitions    test transition finding
    meeting             create minutes of next meeting
    peek                output Redmine issue and Slim content
    request_to_anchor   transforms request to anchor by adding anchor tree
    run_test            run test: delete all tasks in project, create new
    save_snapshot       save Slims to json file
    set_status          set Redmine task status
    set_start_date      set Redmine task start_date
    set_due_date        set Redmine task due_date
    set_estimated_days  set Redmine task estimated_days
    set_relationship    set Redmine task relationship
    delete_relationship
                        delete Redmine task relationship
    set_tracker         set Redmine task to config tracker
    force_tracker       force all Redmine tasks to config tracker
    slims_all           output Slims in project, 1 csv row per slim
    slims_ctrl          output KARABO work slim in ERD project, 1 csv row per
                        slim
    slims_without_parent
                        output Slims w/o parents in project, 1 csv row per
                        slim
    slims_anchor        output project anchors
    slims_request       output project requests
    summary             output summary of project
    traverse_issue      traverse Redmine task tree
    traverse_slim       traverse Slim task tree

optional arguments:
   -h, --help            show this help message and exit
```

## 9.11 pep_tool sub-command option qualifiers

To see if a sub-command requires a qualifier run the sub-command w/o and see if a missing qualifier error is flagged, e.g.

```
python pep_tool.py username password peek
usage: pep_tool.py username password peek [-h] taskID
pep_tool.py username password peek: error: the following arguments are
required: taskID
```

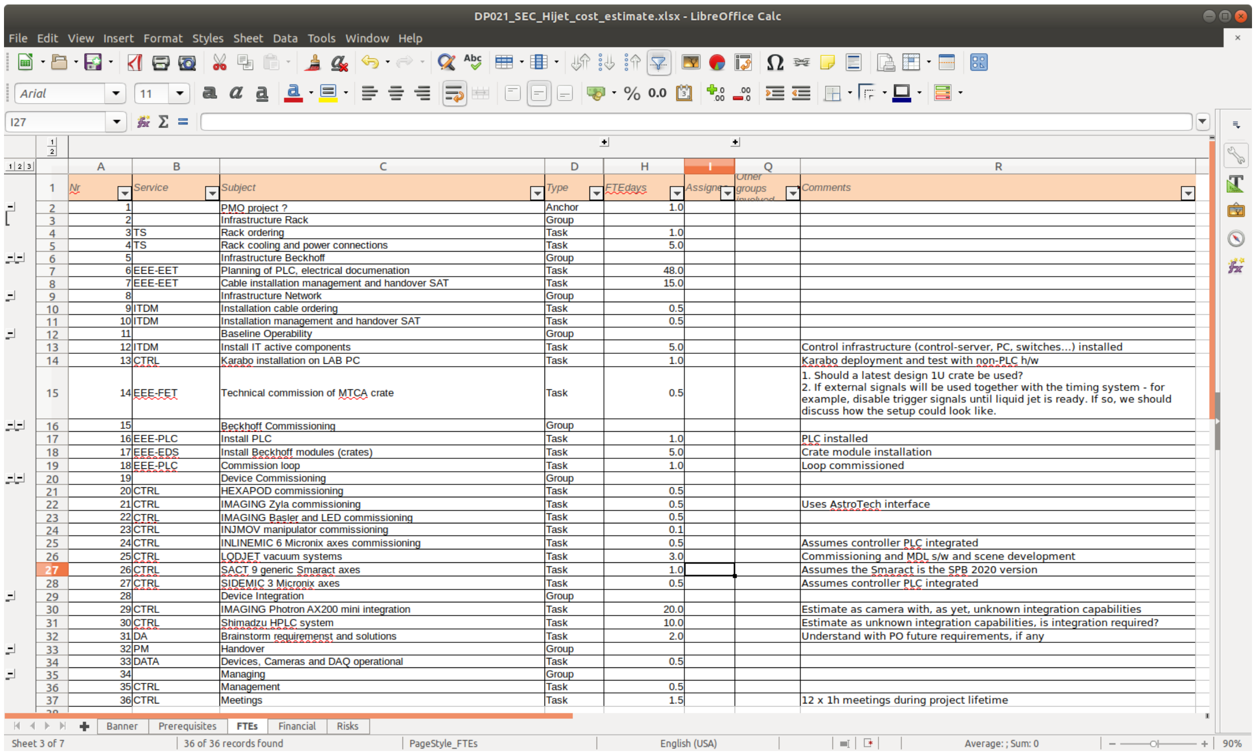
Or add a *-h* behind the sub-command which has the advantage of showing the applications type conversion applied to the qualifier value string provided, e.g.

```
python pep_tool.py username password peek -h
usage: pep_tool.py username password peek [-h] taskID

positional arguments:
  taskID      task (aka ticket) ID, type <class 'int'>

optional arguments:
  -h, --help  show this help message and exit
```

## 9.12 Project implementation workflow programming

tbd, but read the last paragraph in the next section and the cost-estimate section in the PMO procedure chapter.

The *Group* task entry allows grouping tasks with similar requirements. Specifying a *PrevGroup* as a diffrent *Group* allows a staged rollout of task groups - note that only one task in the group need have PrevGroup set and that the PrevGroup entry with the largest *Nr* is used. The *PrevTask* defines the relation of the task to the previous task, if blank the task starts at the same time as the Group task and if set `follows` it follows the previous task.



DP021_SEC_Hijet_cost_estimate.xlsx - LibreOffice Calc

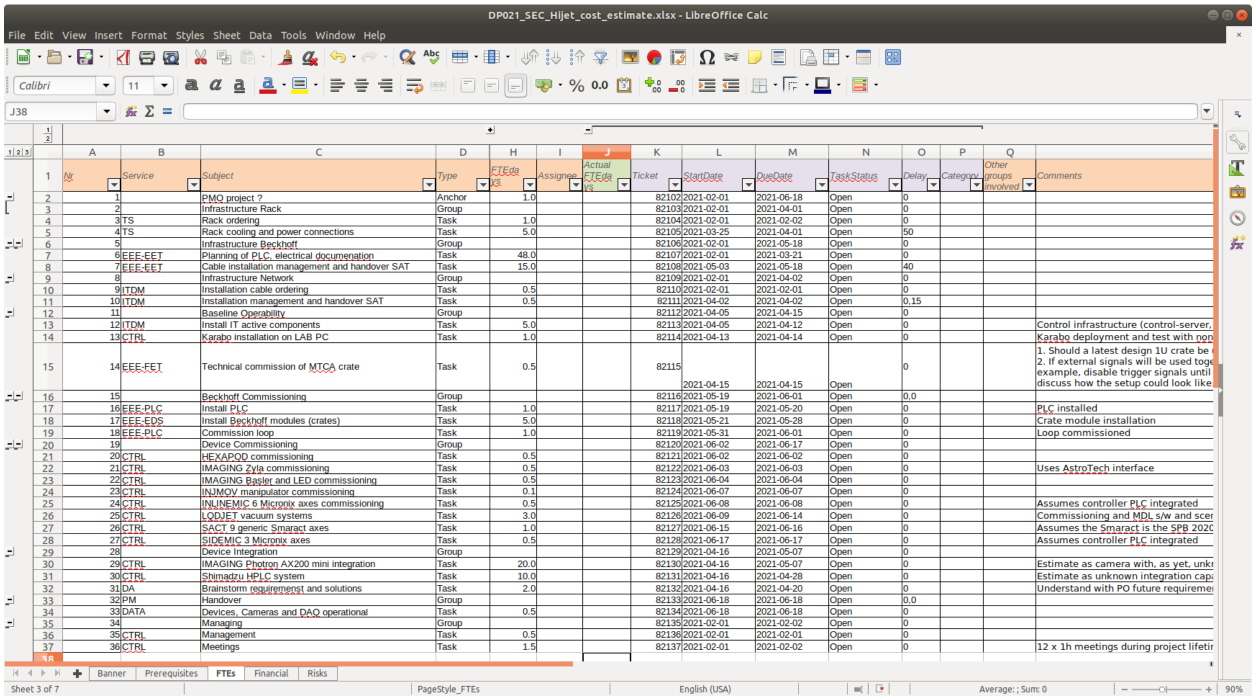
| | A | B | C | D | H | I | Q | R |
|---|---|---|---|---|---|---|---|---|
| 1 | Nr | Service | Subject | Type | FTEdays | Assignee | Other groups involved | Comments |
| 2 | 1 | | PMO project ? | Anchor | 1.0 | | | |
| 3 | 2 | | Infrastructure Rack | Group | | | | |
| 4 | 3 | TS | Rack ordering | Task | 1.0 | | | |
| 5 | 4 | TS | Rack cooling and power connections | Task | 5.0 | | | |
| 6 | 5 | | Infrastructure Beckhoff | Group | | | | |
| 7 | 6 | EEE-EET | Planning of PLC, electrical documenation | Task | 48.0 | | | |
| 8 | 7 | EEE-EET | Cable installation management and handover SAT | Task | 15.0 | | | |
| 9 | 8 | | Infrastructure Network | Group | | | | |
| 10 | 9 | ITDM | Installation cable ordering | Task | 0.5 | | | |
| 11 | 10 | ITDM | Installation management and handover SAT | Task | 0.5 | | | |
| 12 | 11 | | Baseline Operability | Group | | | | |
| 13 | 12 | ITDM | Install IT active components | Task | 5.0 | | | Control infrastructure (control-server, PC, switches...) installed |
| 14 | 13 | CTRL | Karabo installation on LAB PC | Task | 1.0 | | | Karabo deployment and test with non-PLC h/w |
| 15 | 14 | EEE-FET | Technical commission of MTCA crate | Task | 0.5 | | | 1. Should a latest design 1U crate be used?<br>2. If external signals will be used together with the timing system - for example, disable trigger signals until liquid jet is ready. If so, we should discuss how the setup could look like. |
| 16 | 15 | | Beckhoff Commissioning | Group | | | | |
| 17 | 16 | EEE-PLC | Install PLC | Task | 1.0 | | | PLC installed |
| 18 | 17 | EEE-EDS | Install Beckhoff modules (crates) | Task | 5.0 | | | Crate module installation |
| 19 | 18 | EEE-PLC | Commission loop | Task | 1.0 | | | Loop commissioned |
| 20 | 19 | | Device Commissioning | Group | | | | |
| 21 | 20 | CTRL | HEXAPOD commissioning | Task | 0.5 | | | |
| 22 | 21 | CTRL | IMAGING Zyla commissioning | Task | 0.5 | | | Uses AstroTech interface |
| 23 | 22 | CTRL | IMAGING Basler and LED commissioning | Task | 0.5 | | | |
| 24 | 23 | CTRL | INJMOV manipulator commissioning | Task | 0.1 | | | |
| 25 | 24 | CTRL | INLINEMIC 6 Micronix axes commissioning | Task | 0.5 | | | Assumes controller PLC integrated |
| 26 | 25 | CTRL | LQDJET vacuum systems | Task | 3.0 | | | Commissioning and MDL s/w and scene development |
| 27 | 26 | CTRL | SACT 9 generic Smaract axes | Task | 1.0 | | | Assumes the Smaract is the SPB 2020 version |
| 28 | 27 | CTRL | SIDEMIC 3 Micronix axes | Task | 0.5 | | | Assumes controller PLC integrated |
| 29 | 28 | | Device Integration | Group | | | | |
| 30 | 29 | CTRL | IMAGING Photron AX200 mini integration | Task | 20.0 | | | Estimate as camera with, as yet, unknown integration capabilities |
| 31 | 30 | CTRL | Shimadzu HPLC system | Task | 10.0 | | | Estimate as unknown integration capabilities, is integration required? |
| 32 | 31 | DA | Brainstorm requiremenst and solutions | Task | 2.0 | | | Understand with PO future requirements, if any |
| 33 | 32 | PM | Handover | Group | | | | |
| 34 | 33 | DATA | Devices, Cameras and DAQ operational | Task | 0.5 | | | |
| 35 | 34 | | Managing | Group | | | | |
| 36 | 35 | CTRL | Management | Task | 0.5 | | | |
| 37 | 36 | CTRL | Meetings | Task | 1.5 | | | 12 x 1h meetings during project lifetime |

Fig. 2: DP021 cost-estimate's row expanded implementation workflow

## 9.13 Example project workflow handling

Injecting the GENTEC_SLINK1 ERD project into Redmine is described.

The CTRL project workflow spreadsheet is extracted from Gitlab and renamed as required. CTRL because there are presumably only three ERD workflows associated with PLC only work required, PLC and CTRL joint work, and CTRL only.



DP021_SEC_Hijet_cost_estimate.xlsx - LibreOffice Calc

|  | Nr | Service | Subject | Type | FTEdays | Assignee | Actual FTEdays | Ticket | StartDate | DueDate | TaskStatus | Delay | Category | Other groups involved | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 |  | PMO project ? | Anchor | 1.0 |  |  | 82102 | 2021-02-01 | 2021-06-18 | Open | 0 |  |  |  |
| 3 | 2 |  | Infrastructure Rack | Group |  |  |  | 82103 | 2021-02-01 | 2021-04-01 | Open | 0 |  |  |  |
| 4 | 3 | TS | Rack ordering | Task | 1.0 |  |  | 82104 | 2021-02-01 | 2021-02-02 | Open | 0 |  |  |  |
| 5 | 4 | TS | Rack cooling and power connections | Task | 5.0 |  |  | 82105 | 2021-03-25 | 2021-04-01 | Open | 50 |  |  |  |
| 6 | 5 |  | Infrastructure Beckhoff | Group |  |  |  | 82106 | 2021-02-01 | 2021-05-18 | Open | 0 |  |  |  |
| 7 | 6 | EEE-EET | Planning of PLC, electrical documenation | Task | 48.0 |  |  | 82107 | 2021-02-01 | 2021-03-21 | Open | 0 |  |  |  |
| 8 | 7 | EEE-EET | Cable installation management and handover SAT | Task | 15.0 |  |  | 82108 | 2021-05-03 | 2021-05-18 | Open | 40 |  |  |  |
| 9 | 8 |  | Infrastructure Network | Group |  |  |  | 82109 | 2021-02-01 | 2021-04-02 | Open | 0 |  |  |  |
| 10 | 9 | ITDM | Installation cable ordering | Task | 0.5 |  |  | 82110 | 2021-02-01 | 2021-02-01 | Open | 0 |  |  |  |
| 11 | 10 | ITDM | Installation management and handover SAT | Task | 0.5 |  |  | 82111 | 2021-04-02 | 2021-04-02 | Open | 0,15 |  |  |  |
| 12 | 11 |  | Baseline Operability | Group |  |  |  | 82112 | 2021-04-05 | 2021-04-15 | Open | 0 |  |  |  |
| 13 | 12 | ITDM | Install IT active components | Task | 5.0 |  |  | 82113 | 2021-04-05 | 2021-04-12 | Open | 0 |  |  | Control infrastructure (control-server, |
| 14 | 13 | CTRL | Karabo installation on LAB PC | Task | 1.0 |  |  | 82114 | 2021-04-13 | 2021-04-14 | Open | 0 |  |  | Karabo deployment and test with non |
| 15 | 14 | EEE-FET | Technical commission of MTCA crate | Task | 0.5 |  |  | 82115 | 2021-04-15 | 2021-04-15 | Open | 0 |  |  | 1. Should a latest design 1U crate be 2. If external signals will be used toge example, disable trigger signals until discuss how the setup could look like. |
| 16 | 15 |  | Beckhoff Commissioning | Group |  |  |  | 82116 | 2021-05-19 | 2021-06-01 | Open | 0,0 |  |  |  |
| 17 | 16 | EEE-PLC | Install PLC | Task | 1.0 |  |  | 82117 | 2021-05-19 | 2021-05-20 | Open | 0 |  |  | PLC installed |
| 18 | 17 | EEE-EDS | Install Beckhoff modules (crates) | Task | 5.0 |  |  | 82118 | 2021-05-21 | 2021-05-28 | Open | 0 |  |  | Crate module installation |
| 19 | 18 | EEE-PLC | Commission loop | Task | 1.0 |  |  | 82119 | 2021-05-31 | 2021-06-01 | Open | 0 |  |  | Loop commissioned |
| 20 | 19 |  | Device Commissioning | Group |  |  |  | 82120 | 2021-06-02 | 2021-06-17 | Open | 0 |  |  |  |
| 21 | 20 | CTRL | HEXAPOD commissioning | Task | 0.5 |  |  | 82121 | 2021-06-02 | 2021-06-02 | Open | 0 |  |  |  |
| 22 | 21 | CTRL | IMAGING Zyla commissioning | Task | 0.5 |  |  | 82122 | 2021-06-03 | 2021-06-03 | Open | 0 |  |  | Uses AstroTech interface |
| 23 | 22 | CTRL | IMAGING Basler and LED commissioning | Task | 0.5 |  |  | 82123 | 2021-06-04 | 2021-06-04 | Open | 0 |  |  |  |
| 24 | 23 | CTRL | INJMOV manipulator commissioning | Task | 0.1 |  |  | 82124 | 2021-06-07 | 2021-06-07 | Open | 0 |  |  |  |
| 25 | 24 | CTRL | INLINEMIC 6 Micronix axes commissioning | Task | 0.5 |  |  | 82125 | 2021-06-08 | 2021-06-08 | Open | 0 |  |  | Assumes controller PLC integrated |
| 26 | 25 | CTRL | LQDJET vacuum systems | Task | 3.0 |  |  | 82126 | 2021-06-09 | 2021-06-14 | Open | 0 |  |  | Commissioning and MDL s/w and scer |
| 27 | 26 | CTRL | SACT 9 generic Smaract axes | Task | 1.0 |  |  | 82127 | 2021-06-15 | 2021-06-16 | Open | 0 |  |  | Assumes the Smaract is the SPB 202( |
| 28 | 27 | CTRL | SIDEMIC 3 Micronix axes | Task | 0.5 |  |  | 82128 | 2021-06-17 | 2021-06-17 | Open | 0 |  |  | Assumes controller PLC integrated |
| 29 | 28 |  | Device Integration | Group |  |  |  | 82129 | 2021-04-16 | 2021-05-07 | Open | 0 |  |  |  |
| 30 | 29 | CTRL | IMAGING Photron AX200 mini integration | Task | 20.0 |  |  | 82130 | 2021-04-16 | 2021-05-07 | Open | 0 |  |  | Estimate as camera with, as yet, unkr |
| 31 | 30 | CTRL | Shimadzu HPLC system | Task | 10.0 |  |  | 82131 | 2021-04-16 | 2021-04-28 | Open | 0 |  |  | Estimate as unknown integration capa |
| 32 | 31 | DA | Brainstorm requiremenst and solutions | Task | 2.0 |  |  | 82132 | 2021-04-16 | 2021-04-20 | Open | 0 |  |  | Understand with PO future requireme |
| 33 | 32 | PM | Handover | Group |  |  |  | 82133 | 2021-06-18 | 2021-06-18 | Open | 0,0 |  |  |  |
| 34 | 33 | DATA | Devices, Cameras and DAQ operational | Task | 0.5 |  |  | 82134 | 2021-06-18 | 2021-06-18 | Open | 0 |  |  |  |
| 35 | 34 |  | Managing | Group |  |  |  | 82135 | 2021-02-01 | 2021-02-02 | Open | 0 |  |  |  |
| 36 | 35 | CTRL | Management | Task | 0.5 |  |  | 82136 | 2021-02-01 | 2021-02-01 | Open | 0 |  |  |  |
| 37 | 36 | CTRL | Meetings | Task | 1.5 |  |  | 82137 | 2021-02-01 | 2021-02-02 | Open | 0 |  |  | 12 x 1h meetings during project lifeti |

Fig. 3: DP021 cost-estimate's row and column expanded implementation workflow

To inject GENTEC_SLINK1 implementation workflow into Redmine using the tooling

```
python pep_tool.py youngman $chris workflow_inject \
../templates/ERD/CTRL_workflow.xlsx GENTEC_SLINK1 2021-1-11
```

creates the Redmine gantt representation of the project's workflow

Updating the spreadsheet later

```
python pep_tool.py youngman $chris workflow_update \
../templates/ERD/CTRL_workflow_GENTEC_SLINK1.xlsx
```

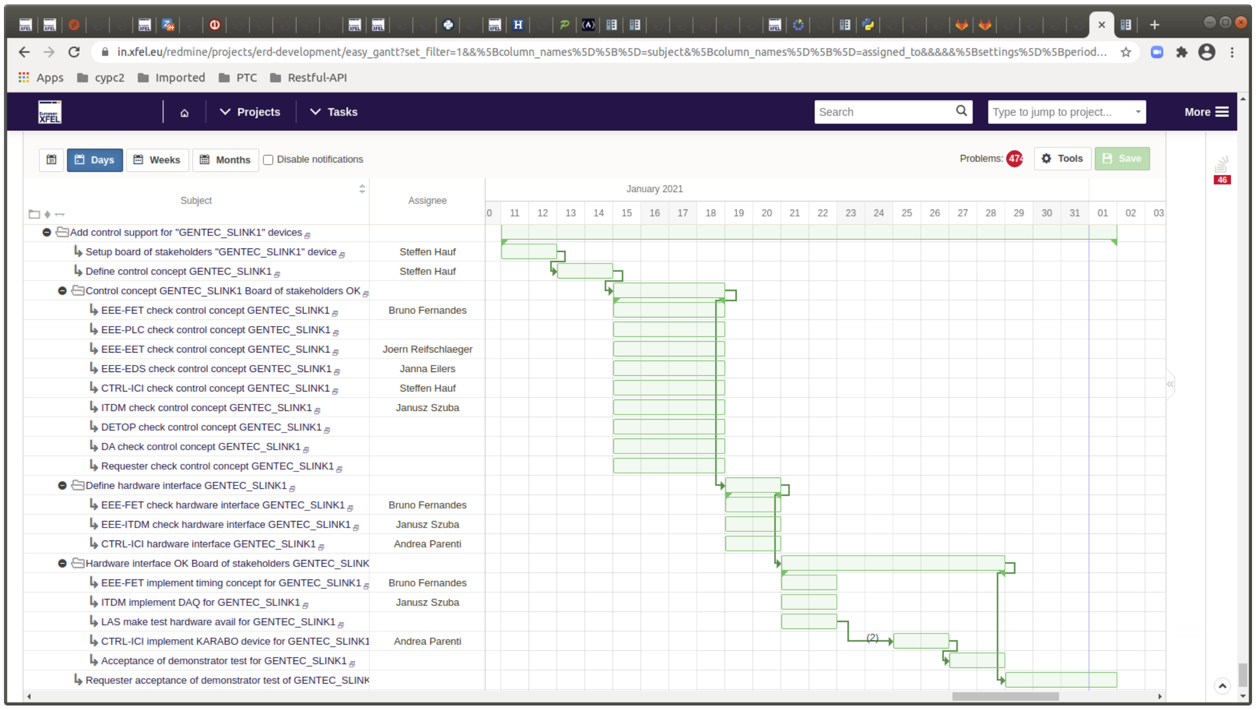
updates most information fields

This was the first workflow injection into a *live* procedure and there were a number of shortfalls:

- the '?' in the spreadsheet subject has not been replaced by GENTEC_SLINK1
- the assignment of Daniel Kane to his tasks failed silently
- it transpired that the enquiery to Redmine using the assignee name to extract the user-id fails and the tooling workround is to use a lookup dictionary.
- ActualDays spend on the task has not been filled in, we're missing time spent handling
- Comment is also empty. It should contain the task's *milestone* custom-field string which is not available in the Task *tracker* currently used by the ERD Redmine procedure

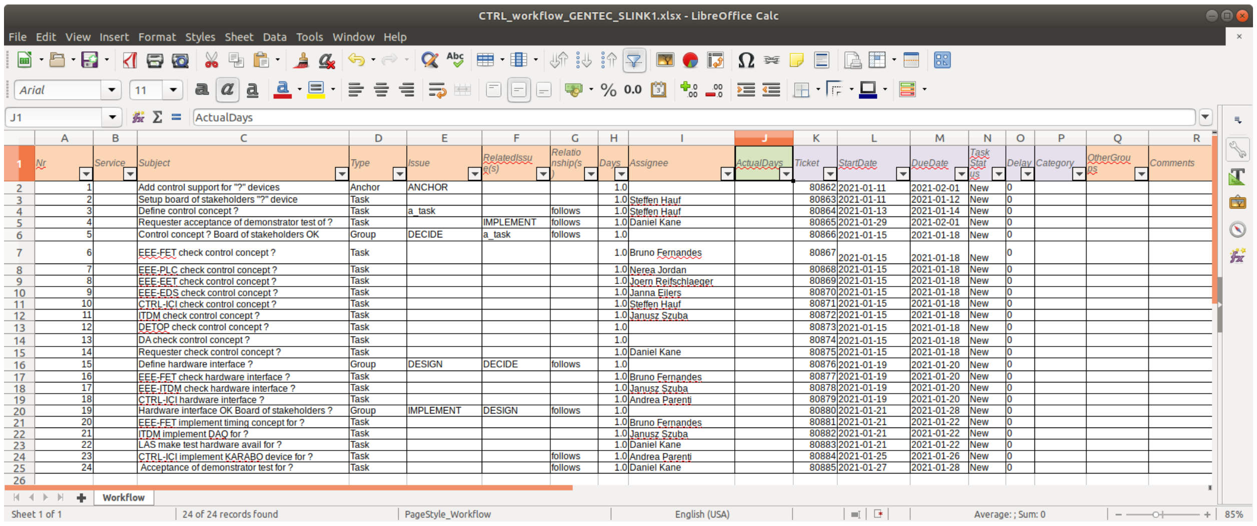
Flattening the workflow's tree graph structure into a spreadsheet table requires the the use of *Issue*, *RelatedIssue(s)*, *Relationship(s)* and *Type*.

The Excel definitions used by PMO cost-estimates and the workflows of non-PMO procedures are currently identical.



| Subject | Assignee |
|---|---|
| Add control support for "GENTEC_SLINK1" devices | |
| Setup board of stakeholders "GENTEC_SLINK1" device | Steffen Hauf |
| Define control concept GENTEC_SLINK1 | Steffen Hauf |
| Control concept GENTEC_SLINK1 Board of stakeholders OK | |
| EEE-FET check control concept GENTEC_SLINK1 | Bruno Fernandes |
| EEE-PLC check control concept GENTEC_SLINK1 | |
| EEE-EET check control concept GENTEC_SLINK1 | Joern Reifschlaeger |
| EEE-EDS check control concept GENTEC_SLINK1 | Janna Eilers |
| CTRL-ICI check control concept GENTEC_SLINK1 | Steffen Hauf |
| ITDM check control concept GENTEC_SLINK1 | Janusz Szuba |
| DETOP check control concept GENTEC_SLINK1 | |
| DA check control concept GENTEC_SLINK1 | |
| Requester check control concept GENTEC_SLINK1 | |
| Define hardware interface GENTEC_SLINK1 | |
| EEE-FET check hardware interface GENTEC_SLINK1 | Bruno Fernandes |
| EEE-ITDM check hardware interface GENTEC_SLINK1 | Janusz Szuba |
| CTRL-ICI hardware interface GENTEC_SLINK1 | Andrea Parenti |
| Hardware interface OK Board of stakeholders GENTEC_SLINK | |
| EEE-FET implement timing concept for GENTEC_SLINK1 | Bruno Fernandes |
| ITDM implement DAQ for GENTEC_SLINK1 | Janusz Szuba |
| LAS make test hardware avail for GENTEC_SLINK1 | |
| CTRL-ICI implement KARABO device for GENTEC_SLINK1 | Andrea Parenti |
| Acceptance of demonstrator test for GENTEC_SLINK1 | |
| Requester acceptance of demonstrator test of GENTEC_SLINK | |

Fig. 4: GENTEC_SLINK1 Gantt chart view in Redmine.



| Nr | Service | Subject | Type | Issue | RelatedIssue(s) | Relationship(s) | Days | Assignee | ActualDays | Ticket | StartDate | DueDate | TaskStatus | Delay | Category | OtherGroups | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Add control support for "?" devices | Anchor | ANCHOR | | | 1.0 | | | 80862 | 2021-01-11 | 2021-02-01 | New | 0 | | | |
| 2 | | Setup board of stakeholders "?" device | Task | | | | 1.0 | Steffen Hauf | | 80863 | 2021-01-11 | 2021-01-12 | New | 0 | | | |
| 3 | | Define control concept ? | Task | a_task | | follows | 1.0 | Steffen Hauf | | 80864 | 2021-01-13 | 2021-01-14 | New | 0 | | | |
| 4 | | Requester acceptance of demonstrator test of ? | Task | | IMPLEMENT | follows | 1.0 | Daniel Kane | | 80865 | 2021-01-29 | 2021-02-01 | New | 0 | | | |
| 5 | | Control concept ? Board of stakeholders OK | Group | DECIDE | a_task | follows | 1.0 | | | 80866 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 6 | | EEE-FET check control concept ? | Task | | | | 1.0 | Bruno Fernandes | | 80867 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 7 | | EEE-PLC check control concept ? | Task | | | | 1.0 | Nerea Jordan | | 80868 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 8 | | EEE-EET check control concept ? | Task | | | | 1.0 | Joern Reifschlaeger | | 80869 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 9 | | EEE-EDS check control concept ? | Task | | | | 1.0 | Janna Eilers | | 80870 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 10 | | CTRL-ICI check control concept ? | Task | | | | 1.0 | Steffen Hauf | | 80871 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 11 | | ITDM check control concept ? | Task | | | | 1.0 | Janusz Szuba | | 80872 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 12 | | DETOP check control concept ? | Task | | | | 1.0 | | | 80873 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 13 | | DA check control concept ? | Task | | | | 1.0 | | | 80874 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 14 | | Requester check control concept ? | Task | | | | 1.0 | Daniel Kane | | 80875 | 2021-01-15 | 2021-01-18 | New | 0 | | | |
| 15 | | Define hardware interface ? | Group | DESIGN | DECIDE | follows | 1.0 | | | 80876 | 2021-01-19 | 2021-01-20 | New | 0 | | | |
| 16 | | EEE-FET check hardware interface ? | Task | | | | 1.0 | Bruno Fernandes | | 80877 | 2021-01-19 | 2021-01-20 | New | 0 | | | |
| 17 | | EEE-ITDM check hardware interface ? | Task | | | | 1.0 | Janusz Szuba | | 80878 | 2021-01-19 | 2021-01-20 | New | 0 | | | |
| 18 | | CTRL-ICI hardware interface ? | Task | | | | 1.0 | Andrea Parenti | | 80879 | 2021-01-19 | 2021-01-20 | New | 0 | | | |
| 19 | | Hardware interface OK Board of stakeholders ? | Group | IMPLEMENT | DESIGN | follows | 1.0 | | | 80880 | 2021-01-21 | 2021-01-28 | New | 0 | | | |
| 20 | | EEE-FET implement timing concept for ? | Task | | | | 1.0 | Bruno Fernandes | | 80881 | 2021-01-21 | 2021-01-22 | New | 0 | | | |
| 21 | | ITDM implement DAQ for ? | Task | | | | 1.0 | Janusz Szuba | | 80882 | 2021-01-21 | 2021-01-22 | New | 0 | | | |
| 22 | | LAS make test hardware avail for ? | Task | | | | 1.0 | Daniel Kane | | 80883 | 2021-01-21 | 2021-01-22 | New | 0 | | | |
| 23 | | CTRL-ICI implement KARABO device for ? | Task | | | follows | 1.0 | Andrea Parenti | | 80884 | 2021-01-25 | 2021-01-26 | New | 0 | | | |
| 24 | | Acceptance of demonstrator test for ? | Task | | | follows | 1.0 | Daniel Kane | | 80885 | 2021-01-27 | 2021-01-28 | New | 0 | | | |

Fig. 5: GENTEC_SLINK1 implementation workflow sheet (post injection update)

## 9.14 Workflow Excel background information

Excel *direct names* are used to identify column usage. The following table lists the name and there content type (Number1 is a number with 1 decimal precision). Content alignment is always Horizontal: General, Vertical: Center, and Indent: 0.

| Column title | Column direct name | Content |
|---|---|---|
| Nr | Nr | General |
| Service | Service | General |
| Subject | Subject | General |
| Type | Type | General |
| Issue | LinkName | General |
| RelatedIssue(s) | TargetLinkname | General |
| Relationship(s) | Relationship | General |
| HoldingTicket(s) | HoldingTickets | General |
| HoldingRelationship(s) | HoldingRelationships | General |
| FTEdays | Days | Number1 |
| Assignee | Assignee | General |
| Actual FTEdays | ActualDays | Number1 |
| Ticket | Ticket | Number0 |
| StartDate | StartDate | General |
| DueDate | DueDate | General |
| TaskStatus | TaskStatus | General |
| Delay | Delay | Text |
| Category | Category | General |
| Other groups involved | OtherGroups | General |
| Comments | Comments | General |

**Note:** Managers should never have to create workflow Excel spreadsheets, they should copy templates. If you're creating a template you're probably doing something wrong.

Where manager edit Excel spreadsheets (to add/modify assignees, etc.) is a matter of preference. I prefer to do this on Windows (Office 2019), but often modify on Ubuntu (LibreOffice 6.0.7.3 00m0(Build:3)) and save changes with Microsoft Excel 2007-2013 XML Format. Openpyxl also gets it right on both platforms.

CHAPTER 10

# PTC - Project Scheduling

## 10.1 Introduction

The alignment of short and long procedure Redmine project containers described in the proceeding chapters allows all project implementation workflows requiring Data Dept resources to be managed and operated on in the same way.

In this chapter the possibilities of scheduling implementation workflows are evaluated. Both long and short procedures have two scheduling stages for a project, a procedure review and approval stage followed by an implementation stage for approved projects.

The review stage is most clearly seen in the ERD process, albeit without being complete as request injection (by email) has to be included. Additionally the PMO review stage is not implemented as it is not a DATA dept responsibility and currently the DATA cost-estimate stage is not organised with Redmine.

## 10.2 Project scheduling constraints

Constraints arise due to missing resources, which can be:

- a suitably experienced DATA FTE is not available. Specialization should be avoided but task completion can be accelerated by assigning a person who has dealt with the issue previously, a decision which amplifies specialization.
- a time issue exists concerning when a given task must be performed. This can be the on-loan availability period of a piece of h/w, beamtime at an instrument, the availability of requester, etc.

## 10.3 Scheduling a project in a procedure

Solving a simplified version of a problem often leads to understanding the bigger problem and its solution. Therefore we start by understaning how to schedule a single project in a small set of projects of a single procedure.

For ease of reading *project implementation* is dropped and *workflow* used instead of *project implementation workflow* in the remainder of the chapter.

### 10.3.1 Scheduling strategies

Two extremes exists:

- No scheduling, where work is allocated according to what can be done on the day. This strategy has light management requirements but is likely to delay project rollout due to a Dept resource being unavailable or occupied, unless workload is low or resource availability is high.
- Perfect scheduling, where all tasks of all workflows to be scheduled are completely understood in terms of duration and resources required. This strategy is also not delay free as some tasks may require a specific assignee or an unexpected time issues arises.

Scheduling reality will presumably lie somewhere between the extremes, but minimizing delay summed over all workflows should provide a useful indicator for establishing whatever the stategy is.

### 10.3.2 Project injection into Redmine

A PMO workflow is chosen which simplifies the discussion further as no request stage handling is required and the cost-estimate FTE task definitions, agreed to with the PO, are injected into Redmine with:

- an initial workflow *start_date* set,
- all workflow tasks defined and grouped under sub-tasks when required,
- task duration days set,
- and assignees not assigned.

Injection does not require PO prerequisite or Financial activities, e.g providing purchased items required, to be addressed. Instead missing items are notified by developers setting the task (Redmine) status to *Feedback* and flagging the condition in the task (Redmine) Category when the deficit is reached.

Workflow injection is performed by:

```
# inject a comples workflow
python pep_tool.py youngman $chris workflow_inject ./complex_test.xlsx
Complex 2021-02-01
...
# get the anchor ID
python pep_tool.py youngman $chris slims_anchor | grep Complex
82371,,[82404,...],Elvis,Open,Normal,PTC-PMO test,PMO project Complex,
,139,-1,,10
```

Further tooling operations use the anchor task ID (here 82371) to identify the workflow. In the above example the ID was extracted by greping for Complex in the list of anchor tasks present, but the workflow ID can also be read from the xlsx Ticket column of the first row following injection.

### 10.3.3 Shifting an entire workflow

One aim of using Redmine is to minimize manager workload. A likely initial manager task is visually (lining up Gantt chart entries) checking (as yet no scheduling tooling is implemented to recommend and perform a correction) the injected workflow *start_date* for conflicts with previously injected workflows.

Shifting an entire workflow is performed by:

```
# move the workflow forward
python pep_tool.py youngman $chris workflow_shift 82371 '2021-3-8'
...
# move the workflow backward
python pep_tool.py youngman $chris workflow_shift 82371 '2021-2-1'
...
```

Shifts are performed by creating ordered task lists, where forward (backward) shifting requires the *due_date* (*start_date*) to be ordered with latest (earliest) *due_date* (*start_date*) listed first. The tasks in the list are then stepped through with forward (backward) shifts executed by updating the *due_date* then the *start_date* (*start_date* then *due_date*). The target date of a shift can be any date and the present (today) date has no significance.

The shift implementation described above ensures that a task *start_date* cannot be later than its *due_date*. This and the task chaining relationships used in procedure workflows shoul guarantee that shifts can be performed.

### 10.3.4 Changing a single task duration withing a workflow

**Note:** If the aim is to modify a single task duration, then do this via the browser interface. The single target task ID methods described in this section can fail to produce the desired result if preceding and follower task settings are ignored. The latter are not ignored when the Excel or browser interface is used for the change.

Changing a task duration, *start_date* to *due_date*, can resolve a resource conflict with another in-progress project. To change a task duration the difference between the task's *start_date* and *due_date* must be changed whilst keeping *start_date* earlier than *due_date* and the *start_date* not before the *end_date* of the preceding task.

Observed Redmine start_date and due_date change behaviour (recheck needed):

- setting due_date after the start_date of a follower task auto-aligns follower task(s) start_date(s) and end_date(s). Start_dates are aligned to Mon-Friday and due_dates to Mon-Sun.
- setting due_date earlier auto-align start_date of a follower but leaves its due_date unchanged.
- start_date can be set to Mon-Sun.
- due_date can be set to Mon-Sun.
- start_date of a task cannot be set before the due_date of the preceding task (exception.ValidationError).
- setting a task's due_date earlier auto-aligns follower task(s) start_date(s) but leaves due_date(s) unchanged.

An example of changing *due_date* is:

```
# find target's date settings
python pep_tool.py youngman $chris peek 82325
...
'start_date': '2021-04-01', 'due_date': '2021-04-03'
...
# set due date later
python pep_tool.py youngman $chris set_due_date 82325 '2021-04-05'
python pep_tool.py youngman $chris peek 82325
...
'start_date': '2021-04-01', 'due_date': '2021-04-05'
...
```

Similarly *start_date* can be changed with set_start_date. However, limitations exist: the *start_date* cannot be moved before a related task's *end_date*, etc. and the Python-Redmine API will throw a informative error message if an illegal change attempt is made.

### 10.3.5 Changing many task durations within a workflow

**Note:** Try to avoid changing many task durations by estimating and setting realistic task durations on workflow injection. Later updates are more probably single tasks activities.

As a first offering task *due_date* moves to later dates are supported, which levers Redmine's automatic alignment of following tasks. This should satisfy the use case where the workflow was initially injected with short placeholder task (Days) durations. A complete solution allowing *start_date* (earlier of later) and *due_date* (earlier) changes is left to later.

**Note:** Group (Type value) tasks are not operated on as their duration is set by the associated sub-tasks.

An example of changing task durations follows.

Always update the spreadsheet of the workflow:

```
python pep_tool.py youngman $chris workflow_update
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
```

Increase the *Days* for those Tasks (Type columnn value) where an duration increase is required and validate to confirm what you want is there!

**Warning:** Validation is common and an important step in most Excel procedures as it checks actual task attribute setting in the Redmine project, and not only their Excel values. It is however limited as not all change requests can necessarily be identified as incorrect.

```
python pep_tool.py youngman $chris workflow_validate
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
row 24 (due - start) work days  1 < 10 required days
... Subject CTRL-ICI implement KARABO device for DUE_DATES
device for DUE_DATES
...
```

Push the changes to Redmine. Note that the pushed use_dates are calculated using workdays which excludes week-ends.

```
python pep_tool.py youngman $chris workflow_push
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
82908 set due_date 2021-03-25
...
```

Pull the changes to Excel and validate the result

```
python pep_tool.py youngman $chris workflow_update
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
python pep_tool.py youngman $chris workflow_validate
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
```

The validation should print no due_date-start_date differences.

### 10.3.6 Changing many task statuses within a workflow

Why is setting task duration straightforward using the Excel interface? It's because working days is a simple concept, there is no needed to work with dates, weekends and days in the year. Status changing requires knowing what states exist and the allowed transitions from one to another, the state transition diagram for the Redmine tracker used must be known. For Managers the Elvis' allowed next state transitions are:

| Current State | Next open States | Next closed States |
|---|---|---|
| Open | Under Review | Closed, Cancelled |
| Under Review | New, Open | Cancelled, Rejected |
| Cancelled | Open, Under Review | |
| Rejected | Open, Under Review | |
| New | In Progress | Closed, Cancelled |
| In Progress | New, Feedback, Blocked, Resolved | Cancelled |
| Feedback | Blocked, In Progress | |
| Blocked | Feedback, In Progress | |
| Resolved | In Progress | Closed |
| Closed | Resolved | |

The manager's choice of final State should not be limited to next states only which requires that the path on the way to the target state is reached silently.

**Warning:** Excel state change updates are only allowed on Task (Type column value) rows. Group state changes may automatically update or need extra handling. The first offering is additionally limited to open to closed state transition paths for all rows as a trail.

An example of changing task states follows.

Always update the spreadsheet of the workflow:

```
python pep_tool.py youngman $chris workflow_update
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
```

Change the *TaskStatus* for those Tasks (Type columnn value) where a state change is required and validate to confirm what you want is there!

```
python pep_tool.py youngman $chris workflow_validate
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
row 8 Error no path In Progress -> Open
row 17 Error unknown status: BADname
row 19 Warning taking 1st path
Feedback -> In Progress -> New -> Closed
Feedback -> In Progress -> Resolved -> Closed
Feedback -> Blocked -> In Progress -> New -> Closed
Feedback -> Blocked -> In Progress -> Resolved -> Closed
...
```

There should be no *TaskStatus* associated warnings.

Push the changes to Redmine. Note that the pushed use_dates are calculated using workdays which excludes weekends.

```
python pep_tool.py youngman $chris workflow_push
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
TBD
...
```

Pull the changes to Excel and, as in the Task duration section, validate the result.

### 10.3.7 Changing many task assignees within a workflow

You're getting bored of the same procedure all the time. Setting a name in the assignee is handled as task Duration and Status:

- upload,
- change names,
- validate,
- and if no corrections push

To overcome the difficulty of extracting the Redmine user ID, needed in the push, by querying with the name which fails if Group access is not allowed for the manager. The set of *known* ID's is stored in the config file and missing names, Elon below, need their IDs adding.

```
python pep_tool.py youngman $chris workflow_validate
../templates/ERD/CTRL_workflow_due_dates_later.xlsx
...
row 5 unknown Assignee name Elon Musk
...
```

### 10.3.8 Initial tooling scheduling requirements

The above discussion indicates an initial functionality set for schedule tooling.

- to identify an unavailable FTE resource the procedure PEPs must provide a list of developer absences in a format (Excel, json, yaml. . . ) which can be interfaced to the tooling.
- to automate FTE resource checks and rescheduling an additional Redmine value is required to hold a csv list of candidate assignees for a task.
- workflow inject (already implemented)
- workflow modify into Redmine task attributes (start_date, due_date, assignee)
- workflow update from redmine of task attributes, sub-tasking and relationships (attribute only update implemented)
- workflow shift of its start date (entire workflow shifts implemented)
- workflow multi task duration changes (end_date extension implemented)
- workflow multi task assignee changes
- workflow multi task status changees
- workflow check for identifying resource conflicts with other projects
- workflow reschedule to reschedule projects according to rules applied.
- workflow compare for testing the above functionalities

- workflow extract to generate the workflow spreadsheet from Redmine definitions, which eases workflow evolution, check pointing, etc. This functionality is most useful for workflows where all information content is in Redmine (e.g. ERD or CRDs), but not PMO where additional information is in the spreadsheet. However, some loss including Comments, Service, and OtherGroups will occur.
- code a similation of Redmine, or is a Redmine differed change mode provided (rollback or check pointing), to allow complex multiple start_date and due_date changes to be evaluated as possible before updating Redmine.
- import python config values into Excel for use with pulldown menues such as assignees

The functionality set should not include

- workflow clear to reset Ticket and similar update characters is not needed as a copy of the template should be used. Resetting copies of working workflow spreadsheets is likely to produce the wrong result, e.g. the name insert parameter has no injection point.

### 10.3.9 Adding in other procedures

The above discussion and associated tooling tests identified the set of requirements for single procedure scheduling. It also provided sufficient insight to suggest the requirements for integrating workflow information from othe procedures. The suggestions are:

- procedure precedence (PMO (or XO) > CRD > ERD) defines how procedure workflow information is to be included.
- lower precedence workflow task information should be loosely coupled using the *blocked by* relationship rather than sub-tasking. This approach requires additional tooling to flag blocking situations.
- workflow check functionality should allow identification of FTE conflicts in lower precedence workflows, but it may not be required to apply rescheduling to them

# CHAPTER 11

# PTC - Project visibility

TBD - Who sees what, who can do what, etc.

Describe where what is in Gitlab.

How people can see where their and other projects are.

## 11.1 Gitlab repository

EuXFEL's Gitlab repository https://git.xfel.eu/gitlab/ptc/ stores all PTC role files (docs, short and long procedure implementation spreadsheet templates and their project instance versions, and tooling source code)

A Gitlab *project*, unfortunately the same word again, contains a set of files which are logically associated w.r.t. the functionality they address. This maps to PTC functionality sets like source files for Python tooling, rst files for documentation, etc.

The PTC Gitlab project structure should reflect the expected usage where who needs access to which project and which projects have dependencies on other projects. The proposed as opposed to the existing flat structure is shown below

# CHAPTER 12

# PTC - Problem corners

This chapter contains problems seen but not addressed.

## 12.1 Standardization (and Vetting)

Asking customers to use standard solutions could help reduce FTE load, for more important work, but how do customers know what DATA recommends as a standard. Additionally standards change and the method choosen to communicate standards should give insight into new development or replacements to allow forward planning. Standards are both h/w and s/w.

Vetting is not a nice job, but offers some degree of control over purchases. I'm sure that Nicola, who takes most of the brunt, would apreciate a DATA review of the procedural problems (already purchased before vetting) and how standardization recommendations are to be handled.

## 12.2 What is out there queries

When handling a vetting request or deciding what to buy, or what s/w should be use or developed it is necessary to identify whether it's a standard, is or is not already integrated, whether there's a recommended different solution, etc? DATA groups probably have very different answers to the above questions, we should review and improve if needed. Remember new people in DATA or EuXFEL cannot know the answer.

## 12.3 Hidden FTE work

Are DATA members involved in wish work of their own, for others, or their group that is not accounted in a project?

## 12.4 Ancient equipment

Integrating equipment that was last used years before, not at XFEL, has not been integrated, has an obsolete interface, may not even work, and will only be used once is a problem. But many instruments and user experiments come with such equipment. DATA access to proposals is needed to recommend a better solution or get early warning is needed.

Many cameras have been integrated, would it not be better to use in house equivalent or better than detectors. How to convince users that it's an advantage probably means making it easy for them w.r.t. use and analysis.

## 12.5 Missing h/w maintenance, replacement and upgrade strategies

Recent cost-estimates have included projects which are known to require later upgrades. Is it a good strategy not to have a path to the required solution not mapped out? This seems particularly true of instrument completion projects where very old ideas are picked up.

Smaract controllers, what is the strategy for eventual standardization? That is one example, but there many. Can there be a strategy without blocking improvements?

Opening a dusty cabinet and finding a spare with a postit saying "possibly faulty" is not a maintenace, repair or replacement strategy. Who should have a strategy and do they?

## 12.6 Difficult relations

What's the answer for personnel requesting work from DATA who consistently do something which is not aligned with DATA norms. How to turn this situation into something profitable?

# CHAPTER 13

# PTC - Glossary

**developer** a DATA member delegated a work task of a project implementation workflow.

**GL** DATA group leader.

**implementation workflow** the project specific workflow which specifies work implementation tasks and their execution order as specified by DATA as the resource owner.

**procedure workflow** the procedure specific workflow defining the steps needed as a project progresses through the procedure.

**PM** the project manager delegated by the PO as representative working with PTC. The PM can be the PO.

**PO** the project owner, i.e. the requester.

**project** the requested deliverable.

**RAL** the Rules, Automation, and Levering concept behind PTC tooling.

**TL** DATA team leader.

**services** a functionality that DATA provides to the facility, e.g. control of equipment, data acquisition, data analysis, etc.

**workflow** a graph showing the tasks, task groups and their execution order needed to complete a work (project).

# CHAPTER 14

# Indices and tables

- genindex
- modindex
- search

# Index