# Meetings

Notes on meetings. Each meeting is documented under a section title of the form: `<month>-<day> <duration_in_minutes>m <event title>`

## 10-07 60m First skype with PO (teamPy)

Participants: team + Pyry

Agenda:

- introduction to project
- general discussion

## 10-09 120m Meeting Jelena

Participants: team + Jelena

Agenda:

- introduction to research group
- learning about research practices
- discussion about project

## 10-14 120m Drafts meeting

Participants: team - TuIiJu

Agenda:

- initial plans
- current drafts
- distribution of work

## 10-22 60m Quick SM-PO chat

Participants: Samuel + Pyry

Agenda:

- status overview
- schedule
- practical issues
- technical research methods

### 10-23 210m Artifacts workskype

Participants: Samuel, Joona, Matias

Agenda:

- Process Overview – Matias
- Product Vision – Joona
- Definition of Done – Joona

### 10-24 120m Artifacts workskype

Participants: Samuel, Teemu, Tuomo

Agenda:

- Product Backlog
- Technical Overview

### 10-25 90m Tech workskype

Participants: Samuel, Matias, Teemu, Tuomo, Iiro

Agenda:

- Discussed and studied the state of the art

### 10-25 120m Triton workskype

Participants: Samuel, Tuomo, Iiro, (Matias)

Agenda:

- Studied Triton and SLURM

### 10-28 180m Wedshop

Participants: team - Juho

Agenda:

- general status check
- discussing artifacts
- researching about the state of the art

## 10-30 300m Frifury

Participants: team - Juho

Agenda:

- Agilefant (time tracking)
- Technical overview
- Product vision
- Other artifacts

## 10-30 120m First meeting with PO

Notes by Joona & Teemu

Questions we discussed

- importance of queue management functions
- technical details

Things we clarified

- requirements by the PO
- the current workflow

Decisions made

- Command prompt applications, no web UI
- Two different applications: Server and client
- Client application contacts server. The client application doesn't have to be running all the time.
- Server application contacts the clusters to run the experiments, manages the experiment queues and collects information on running experiments. The server also saves the information and notifies the client if the experiments go wrong. In an ideal situation the server application is always running.
- Queue management can use either jobman or slurm
- Server makes a .csv document about the past experiments and their outputs and the client can download it.
- Server can tell the information on the available GPU:s etc.
- Config:
  - Defines the ID of the experiment (author, subject, name, group name, git commit ID)
  - Define the variables that must be extracted and sent to the server

  - The Preconditions (minimum available disk space, expected max time, minimum RAM)
  - The files that must be sent to the cluster

Things yet to research/decide

- Simo
- Possibly in the future a web interface and user login

Roadmap

- TBD

## 11-03 90m Lunch with Simo

Participants: Samuel + Simo Tuomisto

Agenda:

- introduction
- general design approach
- practicalities

Hi,

I had a small chat with Samuel during lunch, but I'll gladly accompany you for an another talk.

I briefly outlined that my recommendation would be to utilize a master-slave structure where the client software would ssh to the cluster (triton, gpu machine etc), start a master deamon process and run desired number of slave jobs using applicable shell wrapper (sbatch, bash).

This master could then listen runtime communication from slaves and stash them for the client to collect. Client could then collect them via ssh.

The main reason for this recommendation is that piping ssh through multiple layers of gateways is inefficient and troublesome to configure. Typical network structure looks like this:

```
user <-> user network gateway / shell server <-> internet <-> cluster
network gateway / frontend <-> cluster node
```

Now running job in cluster node requires ssh to frontend, but if a response would be initiated by the cluster node, it would require ssh to the user shell server and from there to the user machine. As other ports besides ssh ports are not visible to the internet (for security reasons) this makes two-way communication between node and user at the least complicated.

Thus a work flow where run information gathering is initiated by the user is much more straightforward and it requires only one ssh connection. This also means that passphrase protected private ssh keys can be stashed to the user machine instead of moving them all the way to the nodes.

This scheme can be run as is on other machines. There instead of sbatch, some other shell would be used and communication would be done to localhost instead of the login node.

I also recommend that the communication between the master and slaves is done using some simple network protocol instead of by files. This would lessen the load on shared file systems.

I'm checking on giving you access to Triton some way or another.

Regards, Simo Tuomisto CS/Triton Admin

## 11-04 180m Wedshop

Participants: team - Juho

Agenda:

- Aalto-student contract
- Time tracking & Agilefant
- Evaluation criteria
- Technical overview (Simo's input)
- EES participation

## 11-04 105m EES 01: Adopting scrum

Participants: Samuel

PO and requirements:

- Interviews for requirements gathering and/or researching feasibility of potential technical solutions -Samuel
- How to modify customer requirements into user stories into the backlog
- How to specify requirements if different persons from client company have different opinions about a topic

- What is the customers role as a product owner when the team has done the service design

Teamwork:

- Scrum, scrum master and team leadership. Should a Scrum Master take the role of a Team Leader?
- Leadership in self-organizing teams. Eliciting intrinsic motivation and self-direction and initiative in team members. -Samuel
- Teamwork power structure, work organization, effort distribution

Software quality:

- How to make your sprint board reflect your DoD
- Quality assurance and automated deployment (continuous integration) with Github, CircleCI and Heroku

### 11-06 90m Expert meeting (Simo Tuomisto)

Notes:

- Neroman could run on a web browser or be a QT application
- Excel or application shows results (multiple options)
- parameter combinations can be numerous
- SSH can be slow + file system unreliable + can crash unexpectedly
- http could be a better protocol
- SSH-Stdout-payload-scp
- Refresh/real time data
- Neroman could be http server application
- Possible implementing order nerokid-neromum-neroman
- Job scheduling
- hdf5 could be used by neromum to save the experiment output data

### 11-06 210m Frifury

Participants: team

Agenda:

- 15m Scrum
- 20m General discussion
- 40m Agilefant

- 20m Schedule
- 100m Market research (matute)
- 100m Prototype design (iijo)

**Scrum:**

Tuomo:

- Has studied Jobman
- Thought it is important to show the technical overview to the PO

Iiro:

- Has studied iPython
- We should continue researching user requirements

Matias:

- Has thought about teamwork practices and Agilefant
- We should communicate more with the end users and then improve the accuracy of the software design

Teemu:

- Thinks it is very important to have good plan before starting to code
- Feels the current plan is pretty reasonable

Joona:

- Has thought about teamwork practices and Agilefant
- Thinks we need to define classes and make class diagrams
- Need input how to specify experiments

Samuel:

- Has been doing techincal overview to prepare to our interviews
- Has been interviewing simo
- Redone agilefant
- Participated in EES

**Notes on existing tools**

Blocks:

- saving and resuming training
- monitoring, analyzing values
- Theano operations
- algorithms to optimize a model

Ipython:

- manages job distibution
- manages parallelism
- seemingly quite similar to Slurm

Ask the custom whether they have used Blocks/Ipython/Sacred etc

## 11-06 180m Satscrum in Hangout

Participants: team

Agenda:

- 10m Scrum
- 30m Discussing results
- 120m Prototype development (iitutejo)
- 80m Prototype development (ma)
- 20m Discussing results

Tasks:

- iitu by 17:00: come up with a design in how the Python package and version management should be dealt in our project. Google and check existing projects.
- jote by 17:00: start sketching the prototype.

Homework:

- Everyone installs and tests a Python 3.5 environment

**Homework**

On Ubuntu 14.04:

```
sudo add-apt-repository ppa:fkrull/deadsnakes
sudo apt-get update
sudo apt-get install python3.5

python3.5 -m venv projects/neronet/.venv
```

Resources:

- PEP-0405
- venv

## 11-09 90m Review of product vision

Participants: team + PO

Agenda:

- 20:00: 10m Discussing agenda
- 20:10: 10m Summary of past weeks
- 20:20: 20m Discussing existing tools
- 20:40: 20m Updating the product vision
- 21:00: 20m Discussing backlog management
- 21:20: 10m Discussing contacting other researchers

Links:

- https://floobits.com/smarisa/neronet/file/doc/notes_on_meetings.md:362
- https://floobits.com/smarisa/neronet/file/doc/project_information.md:35
- https://floobits.com/smarisa/neronet/file/doc/project_information.md:393
- https://github.com/smarisa/neronet/blob/sprint/doc/product_vision.pdf
- https://floobits.com/smarisa/neronet/file/doc/notes_on_tools.md:115

Notes:

- Keep studying LadderNet's similar functionalities and sacred. Could these be utilized with our project? Possibly also check Blocks.
- Query requirements and other ideas after prototype is finished

    - Targets: Simo, Blocks lead developer at GitHub and Torch developers, Google group, Reddit
    - Example: Hey, we've been thinking a tool like ... would be great. What do you think?
- Later when the product has taken some shape and we've had some user testing, possibly start marketing more clearly

### Backlog management

*See notes on tools.*

Agilefant

- As a user, I want to specify experiments so that they are easy to manage and run.
    - As a user, I want to alter the experiments data programmatically so that I can manage it more effectively.
    - As a user, I want to batch import experiments data so that it is easy to migrate.
    - As a user, I want to specify experiments by name, files and parameters to distinguish them.
    - As a user, I want to specify experiment collections to help manage them.
    - As a user, I want to edit existing experiments and collections so that I can update them.
- As a user, I want configurable views to my experiments and collections so that I can review them effectively.
    - As a user, I want a compact CLI summary view so that I can review them remotely.
    - As a user, I want a compact CLI collections view so that I can review them remotely.
    - As a user, I want a compact CLI experiment view so that I can review experiment details remotely.
    - As a user, I want an interactive and configurable GUI view to review my experiments.
- As a user, I want to manage the running of experiments so that my resources are best used.
    - As a user, I want to configure clusters to best utilize my resources.
        - As a user, I want to configure clusters by address and type to specify my computing resources.
            - As a user, I want to configure unmanaged nodes to utilize simple clusters.

        - As a user, I want to configure Slurm cluster gateways to utilize Slurm clusters.
    - As a user, I want to submit individual experiments to be run so that I can later analyse them.
    - As a user, I want to submit batches to be run so that I can later analyse them.
    - As a user, I want to specify experiment batches and queues to best utilize my resources.

- As a user, I want to monitor ongoing experiments so that I know what to do next.
    - As a user, I want to monitor batch and experiment run status so that I know what is going on.
- As a user, I want to access the data of my past experiments so that I can analyse them.
    - As a user, I want to backup and extract the experiments data files manually so that I remain in full control.
    - As a user, I want to batch export experiments data so that I remain in full control.

## 11-11 360m Wedshop

Participants: team

Agenda:

- 13:00: 15m Scrum (Python 3.5 environment setup homework)
- 13:15: 45m Team spirit recap
- 14:00: 15m Updating process & vision artifacts (jote)
- 14:00: 15m Updating technical & dod artifacts (tuma)
- 14:35: 185m Backlog item planning poker
- 17:40: 50m Neroman development (iite)
- 17:40: 50m Nerokid and Neromum development (tuma)
- 18:30: 20m Discussing results and making decisions
- 18:50: 10m Updating agilefant
- 19:00: Going home!

### Team spirit recap

- Mission: Why we exist
    - Create useful software for Pyry (and others)

  - We are doing this project to learn: software development, requirements engineering, architecture, project management, quality assurance, Scrum, communication with client
  - We want grade five, quality award

- Values: What we believe in and how we will behave
  - Superior quality
  - Self-development
  - Respect
  - TBD, Achievement? – We work together to deliver superior results and achieve understanding.

- Vision: What we want to be
  - We want to see ourselves as the best of the course teams
  - We want to win the Quality award!
  - We want to get grade 5+.
  - We want to get an awesome reference (GitHub repo) that we can market on our future job applications.
  - We want our tool to serve people in such a way that a community of users develops around it and continues it's development. We want to launch a successful open source project, which we can speak proudly of even years from now.

- Strategy: What our competitive game plan will be
  - Objective: Ace the course and develop a very useful and popular tool
  - Scope: See product vision
  - Advantage: We have high motivation, we meet in person every week, active and responsible Scrum Master

- Balanced scorecard: TBD

#### Neroman development

- Database daemon + CLI client + GUI client

### 11-13 60m Demo exercise

Participants: team

Agenda:

- 13:00 10m Agree demo presentation tasks
- 13:10 40m Practice presentation
- 13:55 5m Pause

## 11-13 60m Sprint retrospective

Participants: team

Agenda:

- 14:00 10m Sketch retrospective structure
- 14:10 30m Discussion
- 14:40 10m Review
- 14:50 5m Get everyone's aalto account id (username), send them to Simo
- 14:50 10m Pause

### Evaluate and rank teamwork practices

Sprint planning:

- backlog items must be clear and simple -teemu
- backlog items have been unclear, but the user guide probably helps -joona
- it would have been better if the PO had created the stories from scratch -matias, tuomo
- the PO should give input when developing the user guide -joona
- we should do it on monday -tuomo, joona, teemu
- we should make sure we reserve enough time for the actual story selection on Monday -matias

Daily scrums:

- we have mostly been doing teamwork, so there has been little new info in the scrums -Matias -Joona -Teemu
- they have been overly long and they have extended due to inexperience.
- people are late.

Teamwork sessions:

- sessions are too long and sometimes people get hungry.
- generally someone has to leave early or comes late
- we balanced the session lengths(wed 6h fri 5h).

Tools:

- flowdock is good x6
- for remote work we have been using google hangout and skype. Skype has proven to be the most stable.

- for faster communication we are using whatsapp.
- agilefant has a steep learning curve. -Iiro
- people tend to forget to log their time at agilefant.
- hope to use more github during sprints
- floobits ain't very good. Doesn't seem to work in its intended purpose.

### Top 3 tools

1. GitHub
2. Flowdock
3. Agilefant

### Worst 3 tools

1. Floobits
2. Six tactics
3. Agilefant

### How teamwork could be improved

- People should be more on time.
- hard to think on improvements on sprint 0

### Aalto usernames

smarisa, perat1, marjakj1, pitkanl5, blomqvt1, tahkai1

## 11-13 120m Sprint 0 demo

Participants: team + coach

Agenda:

- 15:00 10m Discussing agenda -Samuel
- 15:05 20m Process overview -Joona
- 15:25 20m Retrospective results -Teemu
- 15:45 10m Product vision -Matias
- 15:55 15m Technical overview -Tuomo
- 16:10 10m Definition of done -Iiro
- 16:20 10m Backlogs -Iiro
- 16:30 15m Prototype demo -Samuel
- 16:45 15m Post review discussion

## 11-16 120m Sprint 1 planning

Participants: team + PO

Agenda:

- 11:00: Agenda
- 11:10: Review of sprint 0 results
- 11:20: Sprint team leader's word
- 11:30: Discussing the budget
- 11:40: Discussing the product backlog
- 11:50: Definition of the sprint goal
- 12:10: Selection of backlog items (BIs)
- 12:30: Clarification of BIs (user guide)
- 12:50: Committing to the sprint goals

What is sprint planning: *A time-boxed event of 1 day, or less, to start a Sprint. It serves for the Scrum Team to inspect the work from the Product Backlog that's most valuable to be done next and design that work into Sprint backlog.*

### Sprint team leader

Leaders:

- S0: Samuel
- S1: Joona
- S2: Iiro
- S3: Tuomo
- S4: Matias
- S5: Teemu
- S6: Samuel

Tasks:

- Scheduling & communication with all parties
- Planning and preparing meeting agendas
- Keeping track of progress and ensuring achievement of sprint goals
- Making sure the *big picture* progresses optimally
- Making sure the sprint review & demo is excellent
- Maintaining and boosting team spirit and motivation in collaboration with the scrum master

Note:

- The scrum master will coach and support the sprint leader.
- Constructive feedback is given to the leader by all team members at the end of the sprint.

### Budget

Facts:

- We have at least 10h of storyless work per developer to be done during sprint 1, thus only 5x25h of developer time is left for story fulfilment work.
- Total story points in product backlog defined for non-epic stories at the end of sprint 0: 26

Assumptions:

- We have two two person teams doing pair programming.
- One story point equals 4h of (productive) pair time:
- 40min of studying for and planning the work and sketching the user guide
- 90min of developing the code and unit tests
- 30min of documenting the work and updating the user guide
- 40min of peer reviewing and testing the work of an other pair
- 40min for backlog management, time tracking and pauses

Results:

- The sprint budget is 50h of pair time.
- We can do about 12.5 story points per sprint.
- Our average weekly pair time budget is about 16h.
- Our week velocity (number of story points done per week) should be about 4.

Question:

- Are the assumptions reasonable?
- Which stories should we do in sprint 1?

### Sprint goal

References:

- Scrum.org: Scrum Glossary

- ScaledAgileFrameworks.com: Sprint goals
- RomanPichler.com: Effective sprint goals
- Luxoft.com: 7 sprint goal patterns for building great teams

Quotes:

- *Sprint Goals are a high level summary of the business and technical goals that the team and Product Owner agree to accomplish in a sprint.*
- *Sprint Goal: a short expression of the purpose of a Sprint, often a business problem that is addressed. Functionality might be adjusted during the Sprint in order to achieve the Sprint Goal.*
- *The Sprint Goal, or mission objective, is not repeated over the course of many sprints. Each sprint should have its own unique mission.*
- *Setting the Sprint Goal is the Product Owner's responsibility, but crafting it is a shared responsibility of the Scrum Team, including PO.*
- *The Sprint Goal encourages initiative in multiple areas – teamwork, technology, quality, and mindset. However, while improving those areas, make sure that you are creating a potentially releasable increment by the end of the Sprint.*

Benefits:

- Align team members to a common purpose
- Ensures that everyone moves in the same direction
- Facilitates prioritisation
- Facilitates teamwork
- Facilitates giving and analysing feedback
- Supports communication

Candidates:

- Develop a prototype that offers the most basic functionality via a CLI
- (any alternatives?)

### Sprint backlog

Sprint 1 in Agilefant

### Meeting results

- Assumptions are resonable.
- Sprint goal: Develop a prototype that offers the most basic functionality via a CLI

## 11-17 90m Team leader mentoring session

Participants: Joona + Samuel

Agenda:

- 19:00 Schedule
- 19:10 Work plan
- 19:50 Architecture, interfaces, databases, work instructions

## 11-18 360m Wedshop

Participants: team

Agenda:

- 13.00: Discussing agenda
- 13.05: Daily scrum
- 13.15: Discussing work distribution (f.ex tuma: submit experiments to unmanaged nodes (5 pts) jo: user guide (3 pts) iite: the rest (5 pts) sa: sphinx reviewers: tuma-> jo, jo->iite, iite->tuma)
- 13.20: Quick discussion of tools (sphinx, unittest, coverage.py?, PEP 8, autopep)
- 14.00: Dividing user stories to subtasks, estimating work, Updating user guide and Defining component interfaces (DBs, configs, nero daemons, individual functions) x6
- 16.00: Pair programming x4
- 16.00: EES2 josa
- 18.30: Sphinx demo, PEP 8, autoPep
- 18:50: Update agilefant (time tracking)

User guide:

- implementation 10h
- peer review 2h

Submit experiments to unmanaged nodes:

- planning 3h
- neroman 3h
- neromum 3h
- nerokid 3h
- testing 3h

- documentation 2h
- peer review 3h

Specify clusters by address and type to specify my computing resources: (4h)

- planning 30min
- neroman 120min
- testing 30min
- documentation 30min
- peer review 30min

Specify experiments by name, files and parameters and edit and delete them: (8h)

- setup 40min
- planning 110min
- neroman 150min
- testing 60min
- documentation 60min
- peer review 60min

Want an experiment status report so that I can review experiment status details: (8h)

- planning 30min
- neroman 120min
- neromum
- nerokid
- testing 90min
- documentation 90min
- peer review 90min

Daily scrum:

Iiro:

- Didn't participate

Tuomo:

- Has listened at a boring sprint planning
- No preferences as to what to do next

- No problems, too much time used on defining file hierarchy

Matias:

- The same as Tuomo
- Wants to install tools
- Too little time for the first sprint planning

Teemu:

- The same
- Wants to start working
- Didn't have time to plan setting clusters well enough at the sprint planning

Joona:

- Participated at the sprint planning
- Has planned the team meeting with Samuel
- Wants to start working
- Others don't know enough compared to Samuel

### Notes

- Sphinx documentation
- Google style guide

## 11-18 105m EES 02: Requirements engineering

Participants: Samuel, Joona

PO and requirements:

- Methods to elicit/gather requirements from potential users, other than interviews, maybe via web communities? -Samuel
  - stranger editable mind map of feature ideas posted for others to edit
  - quick briefing + invitation for a meeting or skype
- Developing a User guide as a kind of product prototype to boost discussion and understanding of user stories. -Samuel
  - good idea
- documentation: levels, minimalism, uptodateness, usefulness, keep the code clean and self evident

## 11-20 300m Story skype & Frifury

Participants: team + PO (90m)

Agenda:

- 14:00 Development
- 14:40 Scrum
- 15:00 Development
- 15:45 Discuss user guide (JoIiTe)
- 16:15 Discuss user guide (JoMaTu)
- 17:00 Pyry on Skype
- 18:20 Testing & documentation workshop
- 18:50 Time tracking

## 11-25 360m Wedshop

Participants: team

Agenda:

- 13:00 Scrum
- 13:20 Development
- 13:35 Discussing peer review and GUI user guide and user stories
- 13.50 Development
- 18:00 Status review
- 18:20 Estimating remaining tasks
- 18:50 Time tracking

### Scrum

Iiro & Teemu: - Trying to make tests work - Trying to figure out if functions should return exceptions (Iiro: yes) - hdf5 requires numpy - Will succeed in implementing user stories by next week - Will finish specify experiments and start specify clusters

Tuomo & Matias: - Make neromum code better - Made one-to-many connection from mum to kids - Documenting - Trying to get sphinx to work - Potential problems: parsing packages - Will make better communication between mum and kids - Will make communication happen between neromanand neromum

Joona: - Skype PO to get input to user guide - Writing user guide - No problems - Will finish CLI user guide

## 11-27 300m Frifury

Participants: team

Agenda:

- 14:00 Team programming
- 14:15 Scrum
- 14:30 Checking the user guide https://github.com/smarisa/neronet/blob/us1_user_guide/doc/user_guide.
- 15:00 Development
- 18:30 Estimating remaining tasks
- 18:50 Time tracking

### Scrum

Samuel: - Getting familiar with the current daemon library and TCP servers - Coding python-daemon - Will finish the daemon

Tuomo: - Socked connection - Thought about test cases - No problems - Will start making tests

Matias: - Flow from man to kid works - No problems - Will make tests and documentation

Teemu: - Specifying clusters - Will make experiment status

Joona: - Finished CLI user guide and fixed syntax errors - Will start making artifacts - Should be no problems

## 12-02 360m Wedshop

Participants: team + PO(60min)

Agenda:

- 13:00 Development
- 13:15 Scrum
- 13:30 Unit testing + documentation
- 14:00 Skype with PO on user guide and backlog (jo)
- 14:30 Preparing demo (iimatutejo)
- 17:00 Peer review (iimatu) + Artifacts (tejo)
- 18:00 Retrospective with Matias

Daily Scrum:

Teemu: - Made experiment status report to print the experiment's current state - Potential problems: Maybe too much to do - Will finish experiment status report

Samuel: - Planned things to do - Consulted people - Potential problems: A lot to do, still hopeful - Will test triton

Tuomo & Matias: - Made unit tests for submitting experiments - Made connection between neroman and neromum work better - Problems: Code structuring changes has slowed down development - Will make rsync work

Joona: - Started making Progress report - Planned things to do - Finished CLI user guide - Potential problems: Maybe too much work - Will skype with PO

**Retrospective**

- Improvements since sprint 0
- Replaced six tactics with Team spirit recap
- Balanced our team's power structure by selecting a team leader for each sprint
- Practices
- Developing user guide first:
  - Matias: We shall see
  - Iiro: I feel we have strayed from the reason we began to implement a user guide
- Team leader per sprint
  - Matias: Great to not give samuel all responsibility
  - Iiro: Might be a bit confusing for the PO and I'm not sure how I'll manage it in the next sprint
- Pair programming
  - Matias: Works fine, a lot of time used for coordination within the pair
  - Iiro: I feel that our pair programming might have been bit inefficient in development and testing, but it helped a lot in planning and documentation
- Sprint planning
  - Matias: Doesn't remember
- Daily Scrums
  - Matias: Worked fine this time, not of much use
  - Iiro: Need to attend more of them…
- Teamwork sessions

  - Matias: times fine, people are away too often, peer reviews should be distributed evenly throughout the sprint

- Tools:
- user guide
- flowdock
  - Matias: Hasn't used much
  - Iiro: doesn't like the idea of PO reading the messages
- skype
  - Matias: Wasn't used much
- whatsapp
  - Matias: reads often
  - Iiro: I don't keep my phones internet always on so it hasn't been optimal
- agilefant
  - Matias: No complaints
  - Iiro: Still feels bit rigid
- team spirit
- sphinx
  - Matias: not used
- sharelatex
- google calendar
  - Matias: works fine
  - Iiro: Haven't used it much after we scheduled regular times
- github
  - Matias: works fine
  - Iiro: The usage still needs some working. There are some files that shouldn't have been pushed to the remote
- floobits
  - Matias: Hasn't used much
  - Iiro: Good for meetings
- Top 3 Tools Matias: not of use
- Worst 3 Tools
- Improvements to Teamwork
- Matias:
  - people are away too often
  - peer reviews should be distributed evenly throughout the sprint

    - The pace is not always the best

- Iiro:
    - Pair programming needs to be streamlined

- Planning Implementation of Improvements
- Feedback to the sprint leader
- Matias: Works fine, synchronisation problems may occur, should read and understand the code before the next sprint

### Progress review

Notes: - Product backlog updated in the previous session - Check thet the user stories follow some specified user story template - Retrospective before project progress review? On wednesday? - The coach appreciates trying new work practices and tools, and evaluating their usefulness in the Sprint retros.

The required artifacts: - Product vision (see Template) - Product backlog - Sprint goals of the current and completed Sprints - Sprint backlog of the current Sprint - Definition of Done - Test session charter(s) for peer testing (see Template) - Allocated and spent effort per person per Sprint - Process overview (see Template) - Technical overview - Progress report / Final report slides (see Template) - Learning Diaries

Learning diary: - The diary must contain a new entry with 1) at least three educational observations related to the use of Scrum or other work methods 2) a summary of one's main contributions to the project since the previous entry

The progress report slideset: - project results - realization of the 1) Sprint goals, 2) Product backlog items and 3) other results - a script and screenshots of the software demo that the team will show in the review - Project status - evaluation of software quality - spent and remaining effort per person per Sprint - results of the Sprint Retros

## 12-03 300m Thurage

Participants: team (-Matias)

Agenda:

- 12:00 Finishing user stories, artifacts and report
- 14:00 Peer reviewing all work

## 12-04 120m Frifury

Participants: team

Agenda:

- 14:00 Practice presentation of progress report
- 15:00 Estimating story points for new BIs

## 12-04 120m Sprint 1 review & sprint 2 planning

Participants: team + PO + Coach

Agenda:

- 16:00 Presentation of progress report (35 minutes)
- 16:35 Feedback and questions (10 minutes)
- 17:00 Sprint planning
- 17:05 Sprint team leader's word
- 17:10 Discussing the budget and product backlog
- 17:20 Definition of the sprint backlog and goal
- 17:50 Committing to the sprint
- 18:00 Sprint 2 backlog refinement (team)

### Presentation of progress report

- Intro -Joona & Samuel
- Results -Samuel & Joona
- Demo -Iiro & Teemu
- Quality -Matias & Tuomo
- Effort -Matias & Tuomo
- Retros -Joona & Samuel