Team and Tools

Dirk Riehle, FAU Erlangen

AMOS B01

Licensed under CC BY 4.0 International

Agenda

Team formation

- a. Student roles
- b. Team contract
- c. Team logo and T-shirt

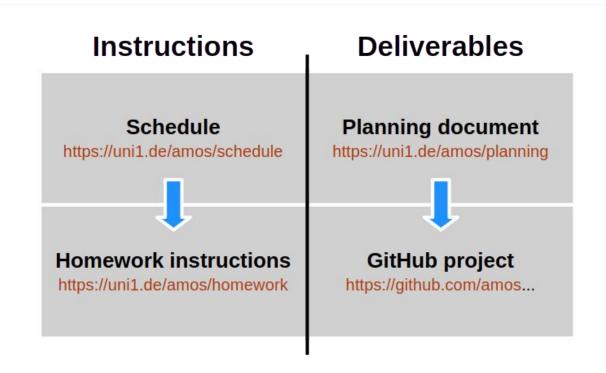
2. Project tools

- a. Planning document
- b. Feature board
- c. Code repository
- d. Imp-squared backlog

3. Team coordination

- a. Stand-up emails
- b. Happiness index

Instructions and Deliverables



1. Student Roles

The AMOS Scrum Team [1]



DE

Scrum Roles

Product owner

Holds overall responsibility that the software is valuable (to customers)

Software developer

Holds overall responsibility for design and implementation of the software

Scrum Master

Holds overall responsibility for continuous process improvement

Additional Roles

Release manager

Holds overall responsibility that a demo is ready for review

There is no Project Manager

Scrum teams are self-organizing

First Industry Partner Meeting

You should have a first meeting with your industry partner as soon as possible

- Team members in general participate to learn about project
- Product owners participate to gather requirements
- Software developers participate to ask technical questions

After the first team meeting you should still try to regularly meet the industry partner

As many team members should participate as is sensible and possible

2. Team Contract

Team Contract

A team contract is

• A contract between team members on how to conduct the project

Its main components are

- Goals and rewards
- Norms and sanctions

Goals and Rewards

Goals are what the team hopes to achieve; there are several categories:

- Learning objectives, for example,
 - o To ensure that all team members understand the course material
- Interpersonal relationship objectives, for example,
 - To foster an atmosphere of mutual respect and learning
- Functional objectives, for example,
 - To have efficient team meetings

Rewards are how you celebrate intermediate or final achievements

Let's have cake!

Norms and Sanctions

Norms are rules for expected behavior; there are several categories:

- Meeting norms, for example,
 - o Is being late to a team meeting acceptable?
- Working norms, for example,
 - O How will we make decisions?
- Coordination norms, for example,
 - Who keeps meetings on track?
- Communication norms, for example,
 - How to communicate outside of team meetings?

Sanctions are what to do if norms are violated

Sing a song to the team or do ten push-ups

One-Time Deliverable: Team Contract

Please discuss and agree on a team contract (in planning document)

Try to finish this during the first team meeting

3. Team Logo and T-shirt

Team Logo

A team logo is just that, a logo that

Represents your team and project

The logo will be used in different places

- On your team T-shirt
- In the GitHub documentation
- In your final demo and report
- Wherever you see fit



Team Logo Design

Get started with collaboratively designing the team logo during the team meeting

Continue on your self-chosen communication channels

Final execution (graphics) may be delegated to one person

Team T-shirt

Create a team T-shirt design using your logo

You can add your logo and one text

Submit your team T-shirt preferences

• Color, size, form



One-Time Deliverable: Team Logo / Team T-Shirt

Please create the team logo and T-shirt design and submit your preferences

4. Planning Document

The Planning Document

The planning document serves to

- Document basic project information
- Collect all materials that don't go easily into the repository
- Coordinate work on this less frequent information

See also the Flowers example planning document

Table of Contents

#	Artifact name (tab in spreadsheet)	Artifact purpose
1	Project data	Provides basic project data
2	Project team	Shows project team
3	Role assignments	Tracks role assignment
4	Team contract	Shows team contract
5	Product goal	Provides product vision and project mission
6	Product glossary	Provides domain terminology of project
7	Sprint goals	Lists goals of respective sprints
8	Velocity tracking	Tracks realized story points per sprint
9	Mid-project release tracking	Tracks mid-project release sprints
10	Final project release tracking	Plans and tracks final project release sprints
11	Definition of done	Provides decision criteria for "done"
12	Documentation	Provide links to documentation of product
13	Bill of materials	Lists all third-party components

1. Project Data

Please fill in the data as needed and as you see fit

Please do not protect the online Zoom team meeting

Please do not change the Zoom link during the semester

2. Project Team

Please provide your name and GitHub id

- Almost all work takes place in the project's GitHub repository
- Please create and provide your GitHub id as soon as possible

Please use only one GitHub id during the semester

3. Role Assignments

Please fill in the roles people play in a given week (sprint)

Please finish this during the first team meeting

Project Deliverables

Please provide all deliverables (homework) in the

• **Deliverables** folder of the project's GitHub code repository

Details can be found in the homework instructions

Regular Deliverable: Planning Document

Please initialize your planning document and keep it up-to-date

Terms (The GitHub Terminology Mess)

Agile / Scrum	AMOS	GitHub	Other
Project	Project	Repository	_
Backlogs [1]	Backlogs	Columns in project [2]	-
Kanban board [3]	Feature board	Project	_
Backlog item [4]	Backlog item	Item, also issue [5]	Work item, ticket
_	Code repository	Code	_

- [1] Backlogs can be of different types: Product backlog, sprint backlog
- [2] The columns of a GitHub project represent the different backlogs + states of work
- [3] Scrum proper does not know kanban boards, but agile in general acknowledges them
- [4] Backlog items can be of different types: Feature [6], refactoring, bug fix
- [5] Terms vary throughout the GitHub user interface
- [6] Features should be presented using the user story format

5. Feature Board

Feature Board (GitHub Project)

A feature board is a slotting system used to

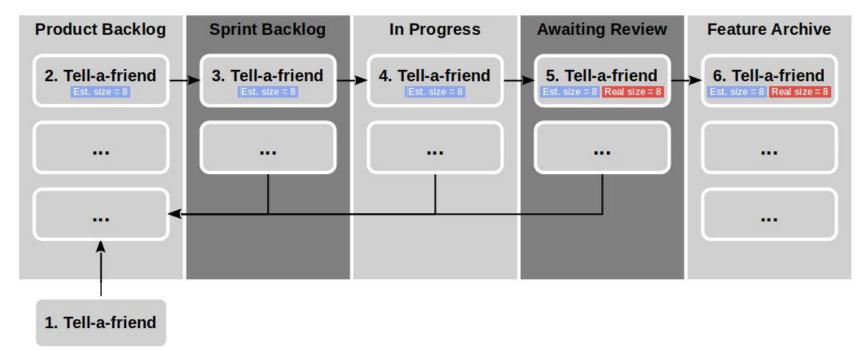
Manage the processing state of backlog items

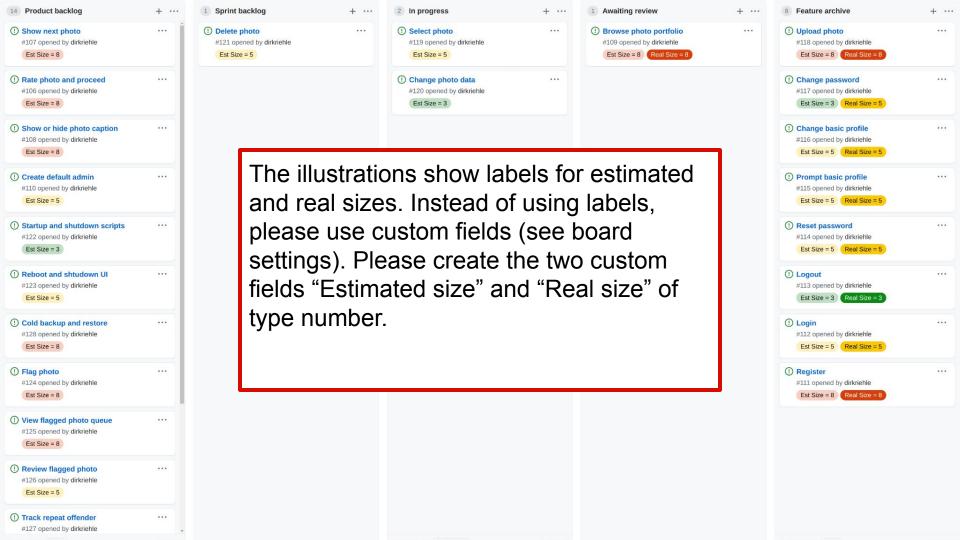
Types of slots (work process states, represented by columns)

- Product backlog (needs doing)
- Sprint backlog (ready to be done)
- In progress (being worked on)
- Awaiting review (needs sign-off)
- Feature archive (finished and archived)

Please do not change the feature board's setup

Example Feature Board and Backlog Items





Backlog Items (Also: Issue, Work Item, Ticket)

A backlog item is an

Ideally semantically closed task that needs doing

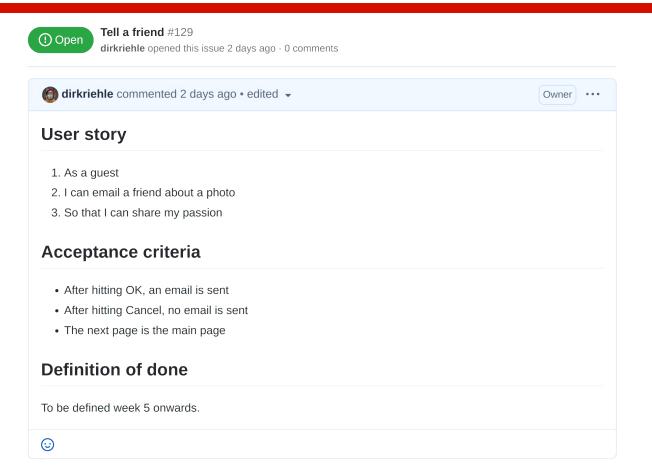
Types of backlog items

- Features (functional and non-functional user requirements)
- Refactorings (behavior-preserving code improvements)
- Bug fixes (fixes to malfunctioning code)

We use GitHub issues to represent backlog items

- Either by directly entering them into the GitHub project (the feature board)
- Or by entering them as an issue, also assigning them to the feature board

Example Issue Using User Story Format



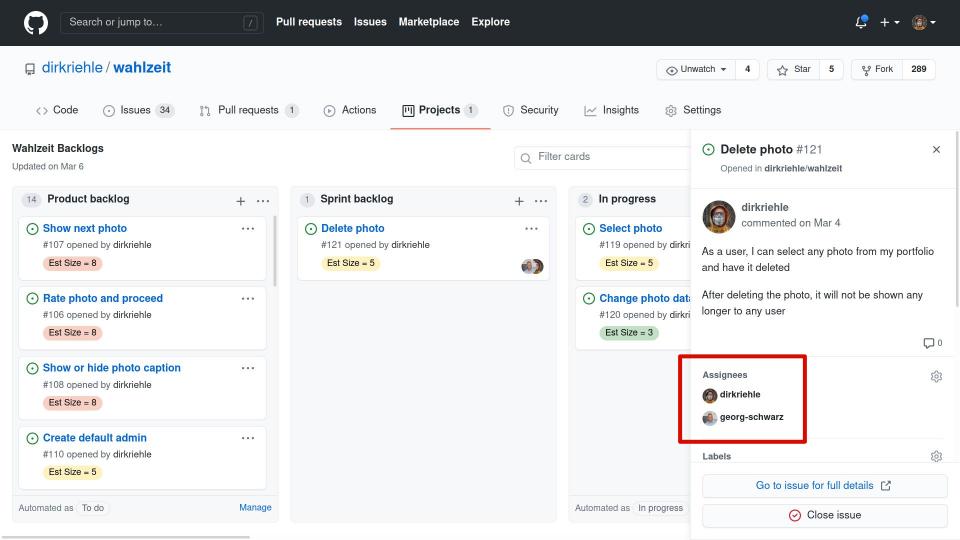
Issue Templates

Please create a template for feature requests

- The template should contain fields for
 - Short name (already preset as title field)
 - Short description (using user story)
 - Acceptance criteria (to test for fulfillment)
 - Definition of done (from sprint 5 on)
- Create custom fields for features sizes
 - Estimated size = X for estimated sizes
 - Set during sprint planning
 - Real size = Y for actual size
 - Set during sprint review
 - Use escalating colors

Issue: Feature Request / Suggest new feature (using user story) User story 1. As a {}. 2. I want / need {} 3. So that {} Acceptance criteria Criterion 1 Criterion 2 **Definition of done** · Added after week 5 · Always the same Optional additional items Issue default title: Assignees: Labels:

Feel free to add other templates



Regular Deliverable: Feature Board

Please initialize your feature board and keep it up-to-date

For the initial content, meet with your industry partner asap

Commit a screenshot of your feature board

6. Code Repository

Git User Setup

Please use only one account and one email address and stick to it

Please configure your name and email address for your local repository

Please make sure this is the same information as on github.com

```
dirk@host$ git config --local user.name "Dirk Riehle"
dirk@host$ git config --local user.email "dirk@riehle.org"
```

Git Commit Sign-off

Please sign-off on your commits as your work using -signoff

This will add your name and email address to the commit message

```
dirk@host$ git commit -m "Fixed issue #123" --signoff
```

Pair Programming / Co-Authoring

If you are pair programming, please make sure you document this in your commits

- Add "Co-authored-by:" to commit message using the correct email address
- Double-check the syntax (otherwise co-authorship will not be recognized)

```
dirk@host$ git commit -a -m "Fixed problem
>
Co-authored-by: Stefan Buchner < stefan.buchner@fau.de > "
```

Each co-authored-by needs to be on its own line to be recognized

Declaring collaboration in the feature board is not enough

Please find more detailed information on GitHub

Making Your Work Count

We use git, not GitHub [1], to look at your work

If you want your work to count,

- Do not squash your commits
- Do not delete branches with relevant work

The AMOS Project Licenses

For source code, we use the MIT license

See https://opensource.org/licenses/MIT

For other data, we use the CC BY 4.0 license

See https://creativecommons.org/licenses/by/4.0/

License and Copyright Declaration in Files

Please use the REUSE SOFTWARE [1] format to declare license and copyright

```
// SPDX-License-Identifier: MIT
// SPDX-FileCopyrightText: 2010-2021 Dirk Riehle <dirk@riehle.org>
// SPDX-FileCopyrightText: 2019 Georg Schwarz <georg.schwarz@fau.de>
```

Alternatively, just track license and contributors in MIT license file

Open Source Governance

Do not add copyleft-licensed libraries to your project

Follow these rules of thumb on license choice

- OK: Permissive licenses (MIT, BSD, Apache)
- May be OK: Weakly protective (a.k.a. "weak copyleft")
- Usually not OK: Strongly protective (a.k.a. "reciprocal" or "copyleft")
- Never OK: Non-software licenses, no license

Professionals (i.e. companies) use code scanners to check

7. Imp-Squared Backlog

Impediments and Improvements (Imp-Squared) Backlog

The imp-squared backlog is a slotting system used to

Manage the processing state of project impediments and improvements
 Impediments are non-technical problems that are

Holding the team and project back

Improvements are non-technical desires to

Improve team performance



Regular Deliverable: Imp-Squared Backlog

Please initialize your imp-squared backlog and keep it up-to-date

Commit a screenshot of your imp-squared backlog

8. Stand-up Emails

Stand-up Emails

Stand-up emails are a communication mechanism that serves to

Ensure regular updates about each other's work state / progress

When writing a stand-up email, please consider these three topics

- 1. What did you get done since you last sent a stand-up email?
- 2. What are your next steps / plans of work to do?
- 3. What challenges are you facing?

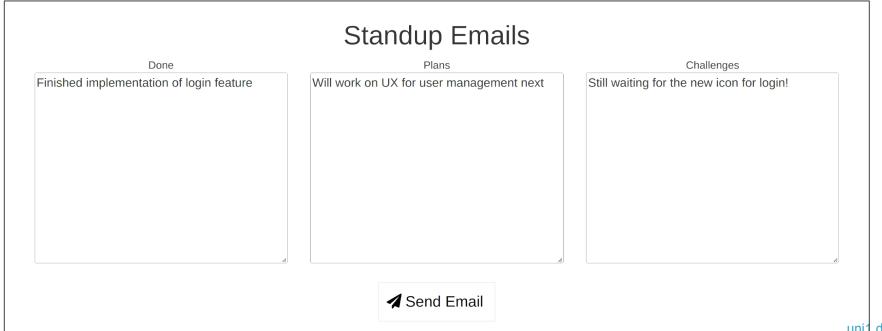
Separate from the stand-up emails, feel free to

Have your own communication channels

Regular Deliverable: Stand-up Emails

Please send stand-up emails using the tool https://happy-amos.appspot.com/

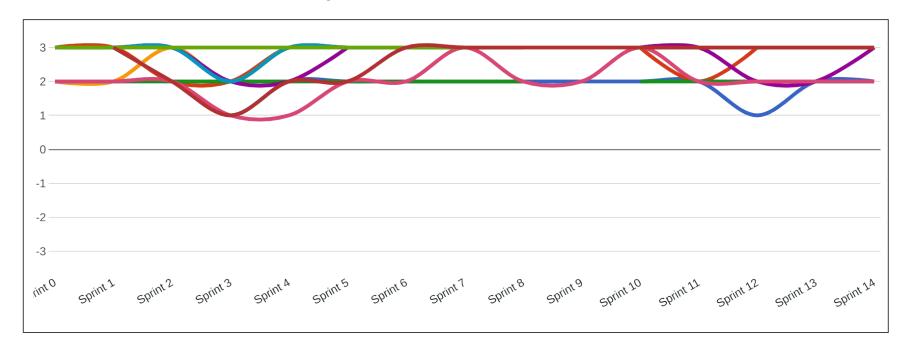
At least twice in total, on different days of the week



9. Happiness Index

Happiness Index [1]

The happiness index tracks general satisfaction to indicate potential problems



Regular Deliverable: Happiness Index

Please indicate your happiness using the tool https://happy-amos.appspot.com/

Every team meeting, including the first and last one, until end of day

Your contributions (your happiness) remains anonymous

Summary

Team formation

- a. Student roles
- b. Team contract
- c. Team logo and T-shirt

2. Project tools

- a. Planning document
- b. Feature board
- c. Code repository
- d. Imp-squared backlog

3. Team coordination

- a. Stand-up emails
- b. Happiness index

Thank you! Any questions?

<u>dirk.riehle@fau.de</u> – <u>https://oss.cs.fau.de</u>

<u>dirk@riehle.org</u> – <u>https://dirkriehle.com</u> – <u>@dirkriehle</u>

Legal Notices

License

• Licensed under the <u>CC BY 4.0 International</u> license

Copyright

© Copyright 2009, 2024 Dirk Riehle, some rights reserved