

# Team and Tools

---

Dirk Riehle, Univ. Erlangen

**AMOS B01**

Licensed under [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

# Agenda

---

## 1. 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

# **1. Student Roles**

---

# The AMOS Scrum Team



# Scrum Roles

---

## Product owner

- Holds overall responsibility for the product being developed

## Software developer

- Holds overall responsibility for design and implementation of product

## Scrum Master

- Holds overall responsibility for process improvement

# Additional Roles

---

## Release manager

- Holds overall responsibility that a demo is ready for review

# There is no Scrum Project Manager

---

Scrum teams are self-organizing

## **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,
  - 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,
  - Is being late to a team meeting acceptable?
- Working norms, for example,
  - 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 design 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	Mid-project release tracking	Tracks mid-project release sprints
8	Final project release tracking	Plans and tracks final project release sprints
9	Definition of done	Provides decision criteria for “done”
10	Documentation	Provide links to documentation of product
11	Bill of materials	Lists all third-party components
12	Planning poker	Provides simple tool for planning poker

# 1. Project Data

---

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

Please do not protect the online team meeting

## 2. Project Team

---

Please provide your name and GitHub id

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

## 4. GitHub IDs

---

Almost all work takes place in the project's GitHub repository

Please create and provide your GitHub id as soon as possible



# 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 Documents

---

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

# Terms (The GitHub Terminology Mess)

Scrum	AMOS	GitHub	Other
Project	<b>Project</b>	Repository	–
Backlogs [1]	<b>Backlogs</b>	Project [2]	–
Kanban board [3]	<b>Feature board</b>	Project	–
Backlog item [4]	<b>Backlog item</b>	Item, also issue [5]	Work item, ticket
–	<b>Code repository</b>	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

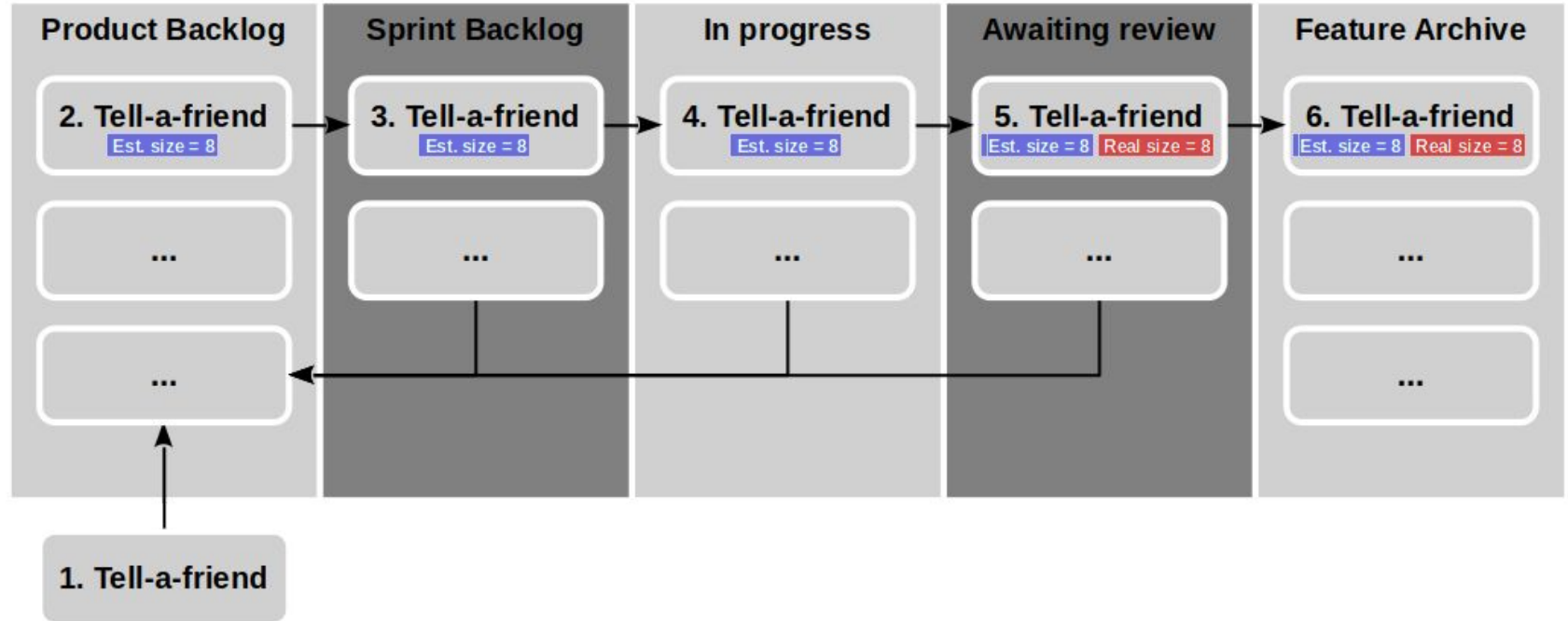
- Manage the processing state of backlog items

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



14

Product backlog

1

Show next photo

#107 opened by dirkriehle

Est Size = 8

1

Rate photo and proceed

#106 opened by dirkriehle

Est Size = 8

1

Show or hide photo caption

#108 opened by dirkriehle

Est Size = 8

1

Create default admin

#110 opened by dirkriehle

Est Size = 5

1

Startup and shutdown scripts

#122 opened by dirkriehle

Est Size = 3

1

Reboot and shtudown UI

#123 opened by dirkriehle

Est Size = 5

1

Cold backup and restore

#128 opened by dirkriehle

Est Size = 8

1

Flag photo

#124 opened by dirkriehle

Est Size = 8

1

View flagged photo queue

#125 opened by dirkriehle

Est Size = 8

1

Review flagged photo

#126 opened by dirkriehle

Est Size = 5

1

Track repeat offender

#127 opened by dirkriehle

1

Sprint backlog

1

Delete photo

#121 opened by dirkriehle

Est Size = 5

2

In progress

1

Select photo

#119 opened by dirkriehle

Est Size = 5

1

Change photo data

#120 opened by dirkriehle

Est Size = 3

1

Awaiting review

1

Browse photo portfolio

#109 opened by dirkriehle

Est Size = 8

Real Size = 8

8

Feature archive

1

Upload photo

#118 opened by dirkriehle

Est Size = 8

Real Size = 8

1

Change password

#117 opened by dirkriehle

Est Size = 3

Real Size = 5

1

Change basic profile

#116 opened by dirkriehle

Est Size = 5

Real Size = 5

1

Prompt basic profile

#115 opened by dirkriehle

Est Size = 5

Real Size = 5

1

Reset password

#114 opened by dirkriehle

Est Size = 5

Real Size = 5

1

Logout

#113 opened by dirkriehle

Est Size = 3

Real Size = 3

1

Login

#112 opened by dirkriehle

Est Size = 5

Real Size = 5

1

Register

#111 opened by dirkriehle

Est Size = 8

Real Size = 8

# 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)
- **Bugs 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



Tell a friend #129

dirkriehle opened this issue 2 days ago · 0 comments



dirkriehle commented 2 days ago • edited ▾

Owner



## User story

1. As a guest
2. I can email a friend about a photo
3. So that I can share my passion

## Acceptance criteria

- After hitting OK, an email is sent
- After hitting Cancel, no email is sent
- The next page is the main page

## Definition of done

To be defined week 5 onwards.




# 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 labels for features sizes
  - **Est. size = X** for estimated sizes
    - Set during sprint planning
  - **Real size = Y** for actual size
    - Set during sprint review
  - Use escalating colors

Feel free to add other templates

**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:

# 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

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

Add “Co-authored-by:” to commit message and using **the correct email address**

Please 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>  
> Co-authored-by: Dirk Riehle <dirk@riehle.org>"
```

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

Please find [more information on GitHub](https://github.com)

# 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>
```

[1] See <https://reuse.software/>

# 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

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[dirkriehle](#) / [wahlzeit](#)[Unwatch](#)

4

[Star](#)

5

[Fork](#)

289

[Code](#) [Issues 34](#) [Pull requests 1](#) [Actions](#) [Projects 1](#) [Security](#) [Insights](#) [Settings](#)

## Wahlzeit Backlogs

Updated on Mar 6

### 14 Product backlog



#### Show next photo



#107 opened by dirkriehle

Est Size = 8

#### Rate photo and proceed



#106 opened by dirkriehle

Est Size = 8

#### Show or hide photo caption



#108 opened by dirkriehle

Est Size = 8

#### Create default admin



#110 opened by dirkriehle

Est Size = 5

Automated as To do

[Manage](#)

### 1 Sprint backlog



#### Delete photo



#121 opened by dirkriehle

Est Size = 5



### 2 In progress

#### Select photo

#119 opened by dirkriehle

Est Size = 5

#### Change photo data

#120 opened by dirkriehle

Est Size = 3

Automated as In progress

### Delete photo #121



Opened in dirkriehle/wahlzeit



dirkriehle

commented on Mar 4

As a user, I can select any photo from my portfolio and have it deleted

After deleting the photo, it will not be shown any longer to any user

0

Assignees



dirkriehle



georg-schwarz

Labels



[Go to issue for full details](#)



Close issue

## **7. Imp-Squared Backlog**

---

# Imp-Squared (Impediments x Improvements) 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

## Impediments Backlog

Updated 8 minutes ago

[+ Add cards](#)[Fullscreen](#)[Menu](#)

### 2 To do + ...

#### Cards ...

Cards can be added to your board to track the progress of issues and pull requests. You can also add note cards, like this one!

Added by dirkriehle

#### Automation ...

[Automatically move your cards](#) to the right place based on the status and activity of your issues and pull requests.

Added by dirkriehle

### 0 In progress + ...

### 0 Done + ...

+ Add column

# 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

## **7. 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

## Standup Emails

Done


Finished implementation of login feature

Plans

Will work on UX for user management next

Challenges

Still waiting for the new icon for login!

 Send Email

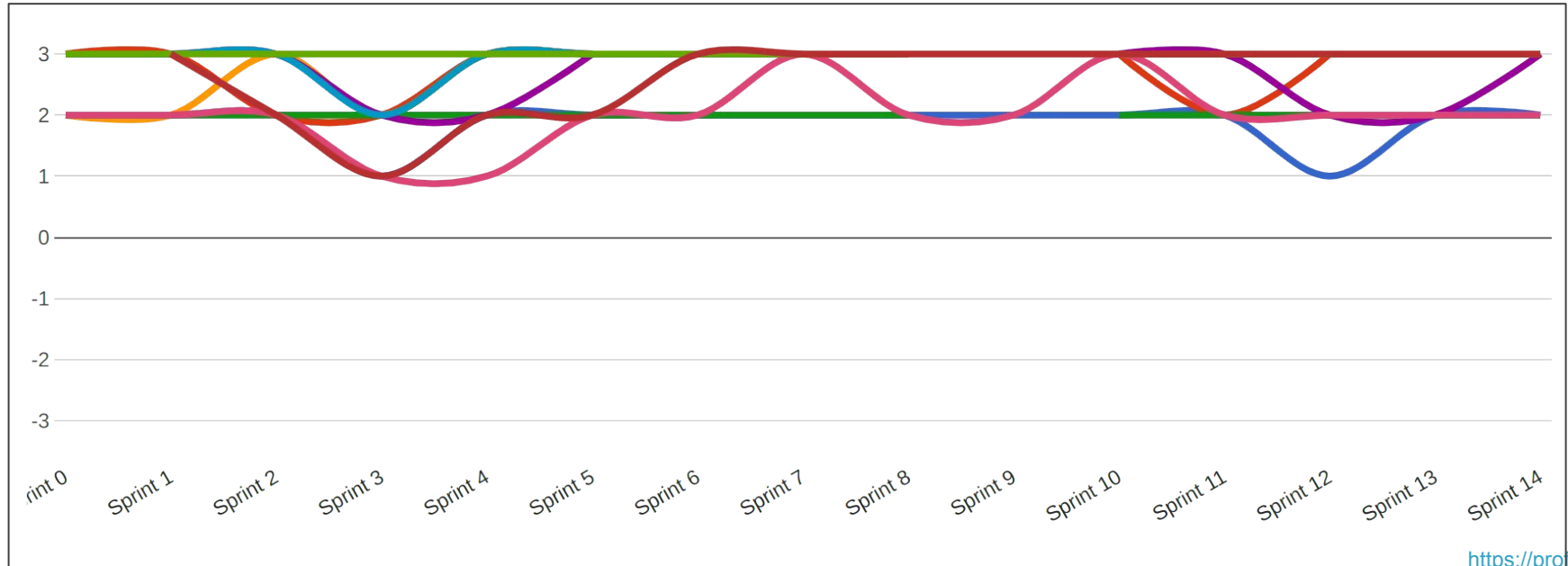
## **8. Happiness Index**

---

# Happiness Index (Originally: Emotions Seismograph)

The happiness index shows your team's happiness over time

The resulting picture indicates potential problems early on



# 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

---

## 1. 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. Impediments backlog

## 3. Team coordination

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

# Thank you! Any questions?

---

[dirk.riehle@fau.de](mailto:dirk.riehle@fau.de) – <https://oss.cs.fau.de>

[dirk@riehle.org](mailto:dirk@riehle.org) – <https://dirkriehle.com> – [@dirkriehle](#)

# Legal Notices

---

## License

- Licensed under the [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/) license

## Copyright

- © Copyright 2023 Dirk Riehle, some rights reserved