

Project Setup

Dirk Riehle, Univ. Erlangen

AMOS A02

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

Scrum Roles

Product owner

- Holds overall responsibility for the product being developed
- Provides product vision and product requirements
- Plans and helps plan development and tracks progress

Software developer

- Holds overall responsibility for design and implementation of product
- Estimate complexity (“size”) of product features to be implemented
- Organizes and allocates design and implementation tasks

Scrum master

- Holds overall responsibility for following Scrum properly
- Is responsible for tracking and resolving impediments
- Is responsible for team and process improvement

Tools Overview

The planning document (Google Sheets)

- Used by everyone

The feature board (GitHub project)

- Used predominantly by the product owner

The code repository (GitHub code repository)

- Use predominantly by the software developer

The impediments backlog (GitHub project)

- Used by the Scrum Master

The Flowers Project Example

Wahlzeit

[[show](#) | [tell](#)] — [[signup](#) | [login](#) | [configure](#)]

About


This website is to show, discuss, and praise photos!

Photo Filter

[Click to toggle filter!](#)

Community

[It is Wahlzeit!](#)



Who/what/where is that?
[Click to show/hide description!](#)

Praise it!

☐ 10
☐ 9
☐ 8
☐ 7
☐ 6
☐ 5
☐ 4
☐ 3
☐ 2
☐ 1

Or [skip](#) it.

a friend about this photo: <http://localhost:8585/x1ac6.html> — Send to the owner of this photo!

Please help keep this community site clean! photo as inappropriate if necessary.

This website is to show the best in photos!

[[blog](#)] — [[about](#) | [contact](#) / [imprint](#) | [terms](#)] — [language: en | [de](#)] — [photo size: [XS](#) | [S](#) | M | [L](#) | [XL](#)] — [debug: [reset](#)]

[processing time: 0.001 seconds]

1. Project Management

There is no Scrum Project Manager

Scrum teams are self-organizing

In (AMOS) practice, product owners also often play project managers

The Planning Document

The planning document serves

- To document and coordinate basic structural information
- To collect all materials that don't go easily into the repository

See also the [Flowers example planning document](#)

Table of Contents

#	Artifact Name	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)

We strongly encourage to do this once at the start of the project

The release manager is responsible for ensuring that the software can be demoed

The GitHub Repository

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

2. Product Owner / Feature Board

Backlog Entries (Issue, Work Items, Tickets)

A backlog entry is an

- ideally semantically closed
- **task that needs doing**

Types of backlog entries

- **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 entries

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 week 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:

Feature Board (GitHub Project)

A feature board is a slotting system used to

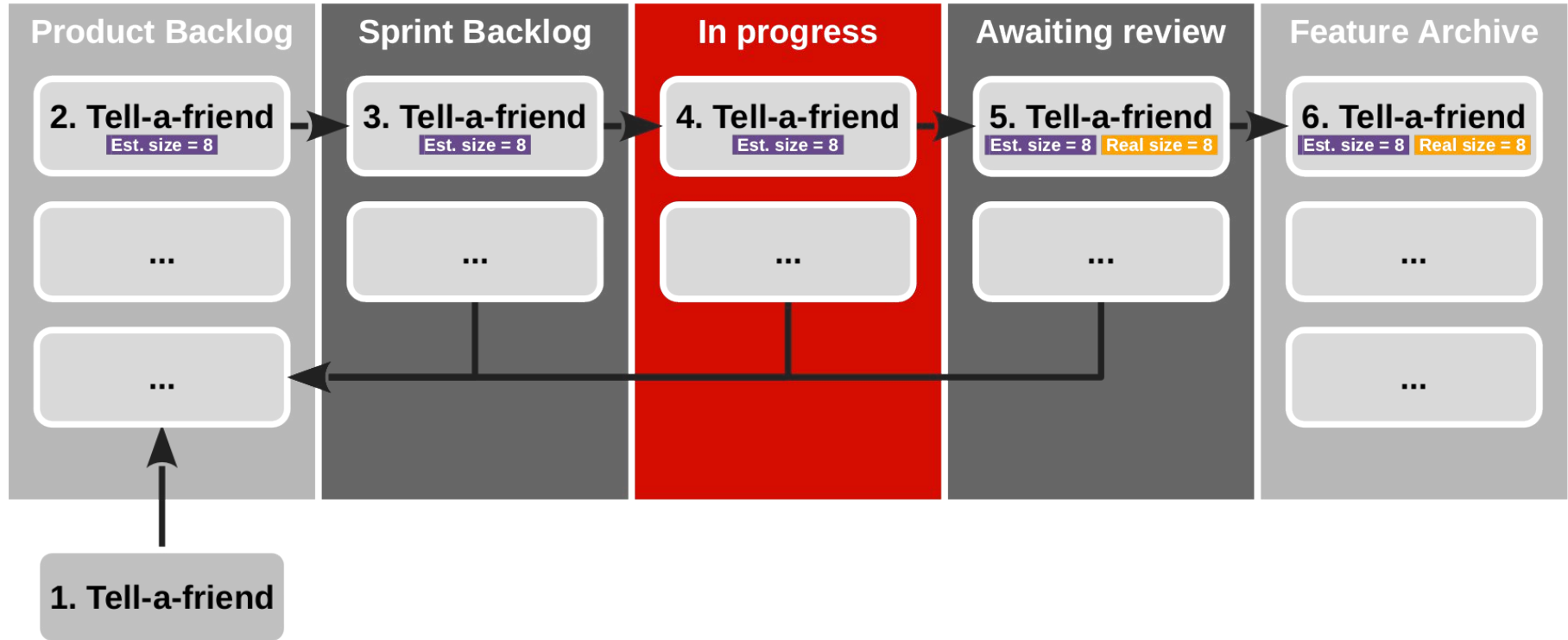
- Manage the processing state of backlog entries

Types of slots (work process states)

- 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 Entries



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

Synonyms

Scrum	AMOS	GitHub	Other
Backlog entry	Backlog entry	Issue	Work item, ticket
Backlog entry	Feature	Issue	
Backlog entry	Refactoring	Issue	
Backlog entry	Bug fix	Issue	
User story	User story		
Product backlog	Product backlog		
Sprint backlog	Sprint backlog		

2. Software Developer / 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, make sure you document this in your commits

Add “Co-authored-by:” to commit message and 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>"
```

Please find [more information on GitHub](https://profröhle.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

- May make later desired license change difficult, including proprietary use
- Check with teaching team whether library is OK

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

4. Scrum Master / Impediments Backlog

Impediments Backlog (GitHub Project)

The impediment backlog is a slotting system used to

- Manage the processing state of project impediments

Impediments are non-technical problems that are

- Holding the team and project back

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

Thank you! Any questions?

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

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 2022 Dirk Riehle, some rights reserved