

Software Engineering II: Project Organization and Management

Exercise 02 - Agile Methods

Stephan Krusche
08 May 2015

Outline

→ Agile Methods: Scrum

- Artifacts
- Meetings
- Roles
- **Exercise 1:** Creation of the Product Backlog: User Stories and Scenarios
- JIRA Agile
 - **Exercise 2:** Creation of the Product Backlog in JIRA
 - **Exercise 3:** Sprint Planning Part I - Creation of the Sprint Backlog in JIRA
 - **Exercise 4:** Sprint Planning Part II - Refinement of the Sprint Backlog in JIRA
 - **Exercise 5:** Working on the Taskboard during the Sprint
 - **Exercise 6:** Sprint Review

Controlling Software Development with a Process

- How do we control software development? Two opinions:
- Through **organizational maturity** (Humphrey 1989)
 - Repeatable process, Capability Maturity Model Integration (CMMI)
- Through **agility** (Schwaber 2001):
 - Large parts of software development is empirical in nature; they cannot be modeled with a defined process
 - There is a difference between a defined and an empirical process
 - How can software development better be described?
 - with a **defined process** control model or
 - with an **empirical process** control model?

more in Lecture 11
CMMI on June 24

Defined vs. Empirical Process



Defined Process

Planned

Follows strict rules

Avoids deviations



Empirical Process

Not entirely planned

inspect and adapt

Defined Process Control Model

- Requires that every piece of work is completely understood
- Deviations are seen as errors that need to be corrected
- Given a well-defined set of inputs, the same outputs are generated every time
- Precondition to apply this model:
 - All the activities and tasks are well defined to provide repeatability and predictability
- If the preconditions are not satisfied:
 - Lot of surprises, loss of control, incomplete or wrong work products.

Empirical Process Control Model

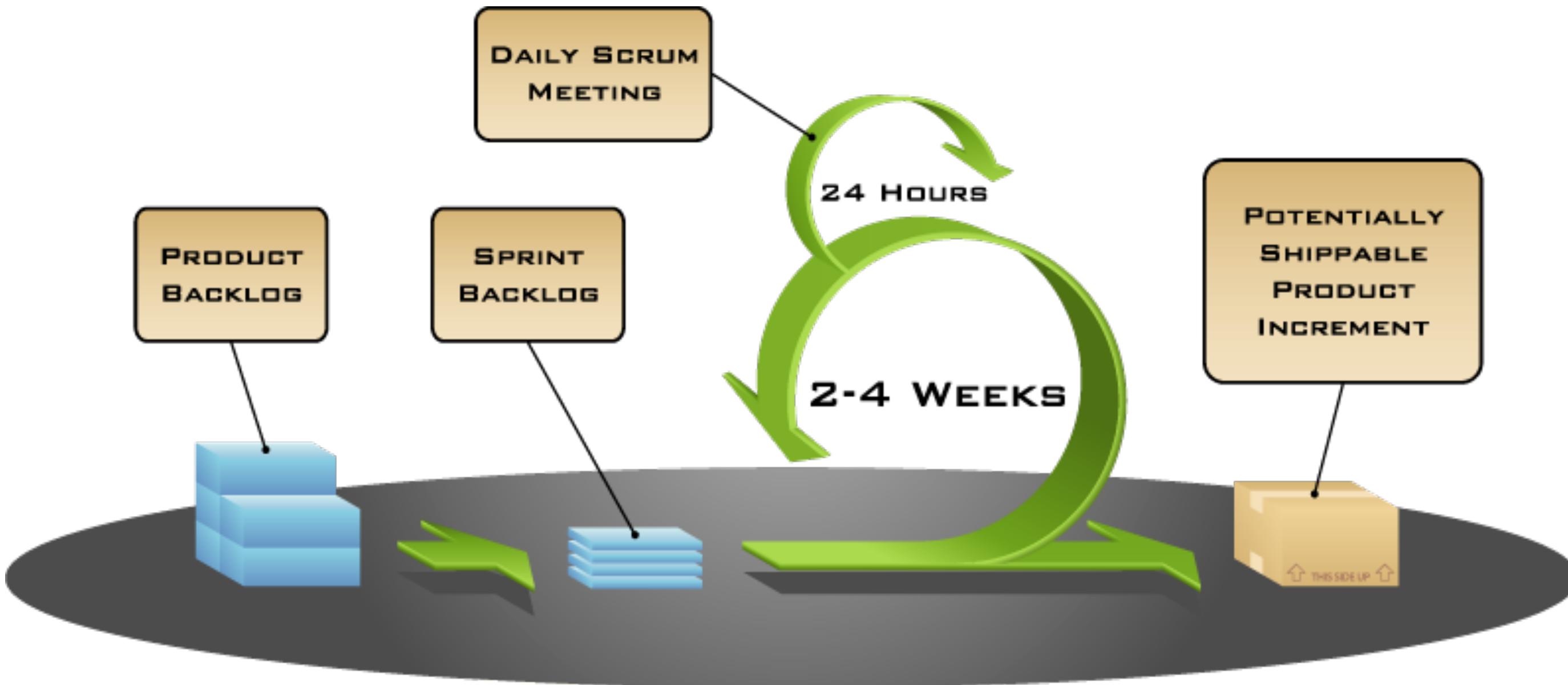
- **Empirical process:** An imperfectly defined process, not all pieces of work are completely understood
- Deviations, errors and failures are seen as opportunities that need to be investigated
 - The empirical process “expects the unexpected”
- Control and risk management is exercised through frequent inspection
- Condition when to apply this model:
 - Change is frequent and cannot be ignored
 - Change of requirements, change of technology, change in the organization, people change too.

Scrum is based on an Empirical Process Control Model

- Original definition (from Rugby): A Scrum is a way **to restart the game after an interruption**
 - The forwards of each side come together in a tight formation and try to get the ball when it is tossed in among them
- Definition in agile processes: Scrum is a technique that deals with interruptions (change)
 - Manages and controls software and product development with rapidly changing requirements
 - Improves risk management by improved communication and cooperation and the delivery of product increments.



Example of an Empirical Process: Scrum



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Scrum Artifacts

- **Product Backlog**: List of requirements for the whole product
- **Sprint Backlog**: List of requirements and tasks for one iteration (“Sprint”)
- **Potentially Shippable Product Increment**: Release to the Product Owner that contains all results of the current Sprint

Scrum Meetings

- **Project Kickoff Meeting** (Start of the Project): Create and prioritize Product Backlog
- **Sprint Planning Meeting** (Start of each Sprint): Create Sprint Backlog
- **Daily Scrum Meeting** (Every Morning, 15min): Standup meeting to share status, impediments and promises
- **Sprint Review Meeting** (End of each Sprint): Demonstration of realized backlog items to the Product Owner (and other stakeholders)

Scrum-Team

Development Team

Self-organizing and cross-functional
Realizes the product increment



Product Owner

Defines the product
Responsible for results



Scrum Master

Resolves impediments
Responsible for the process

Sprint - Definition

„A time-box of one month or less during which a “Done”, useable, and potentially releasable product increment is created. Sprints best have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.“

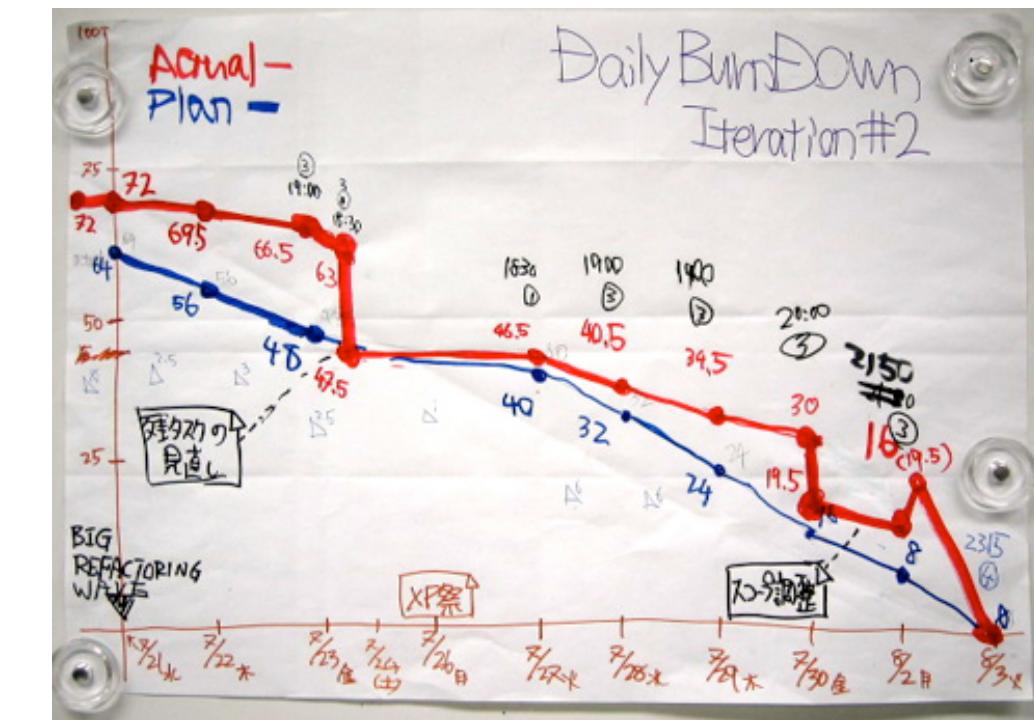
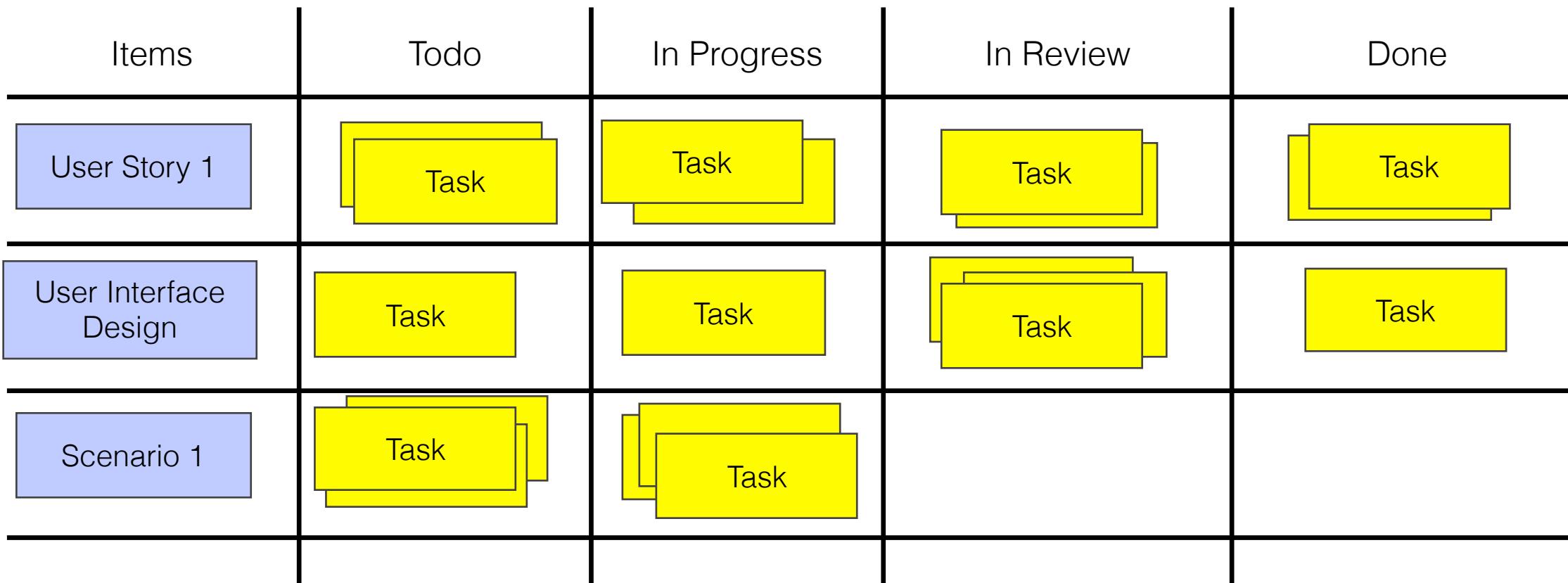
[scrumguides.org]

Sprint

- Typically 2-4 weeks long
- Starts with **Sprint Planning Meeting**
 - Create the Sprint Backlog: Selection of items to be implemented in the Sprint
 - Important: Development Team and Product Owner select the items together
- Ends with **Sprint Review Meeting**
 - Release and deliver the application (product increment)
 - Demo of the application during the meeting
 - Important: Product Owner gives feedback
- Sprint Review Meeting and Sprint Planning Meetings can be combined into a single meeting with the Product Owner
- The Scrum Team can additionally perform a Sprint Retrospective Meeting

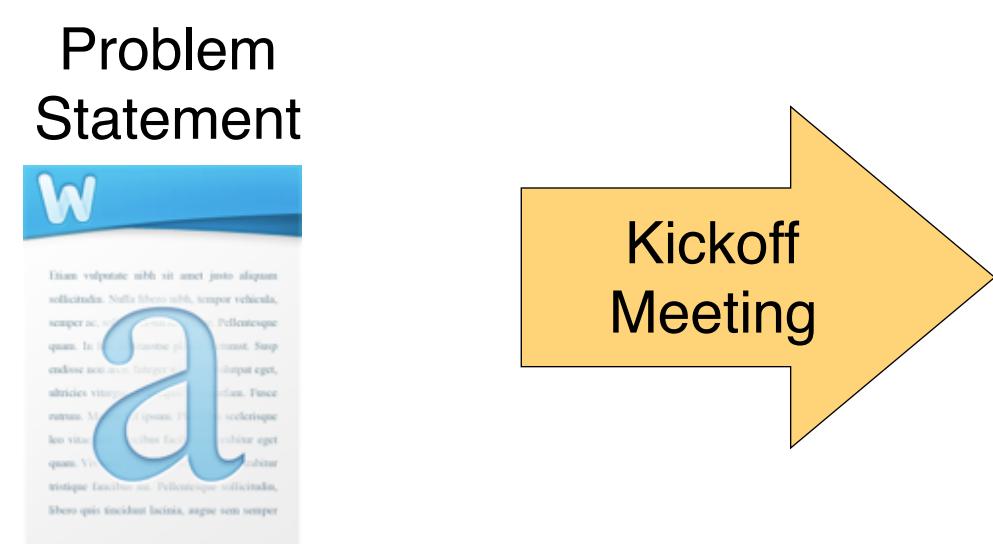
Sprint

- The Development Team
 - Works on the items in the Sprint Backlog
 - Uses e.g. a Task Board to visualize the status of these items
- The Scrum Master visualizes the progress, e.g. in a Burn Down Chart



Product Backlog

- Collection of items (e.g. user stories, scenarios, etc.) prioritized by the Product Owner
- The Product Backlog can always be changed and reprioritized during the projects
- Created on the basis of the problem statement during the Kickoff Meeting or in the phase before the actual project starts



Prioritized Product Backlog

ID	Name	Priority
1	<input type="checkbox"/> Backlog Item 1	Critical
2	<input type="checkbox"/> Backlog Item 2	Critical
3	<input type="checkbox"/> Backlog Item 3	Major
4	<input type="checkbox"/> Backlog Item 4	Minor
5	<input type="checkbox"/> Backlog Item 5	Minor

Priority

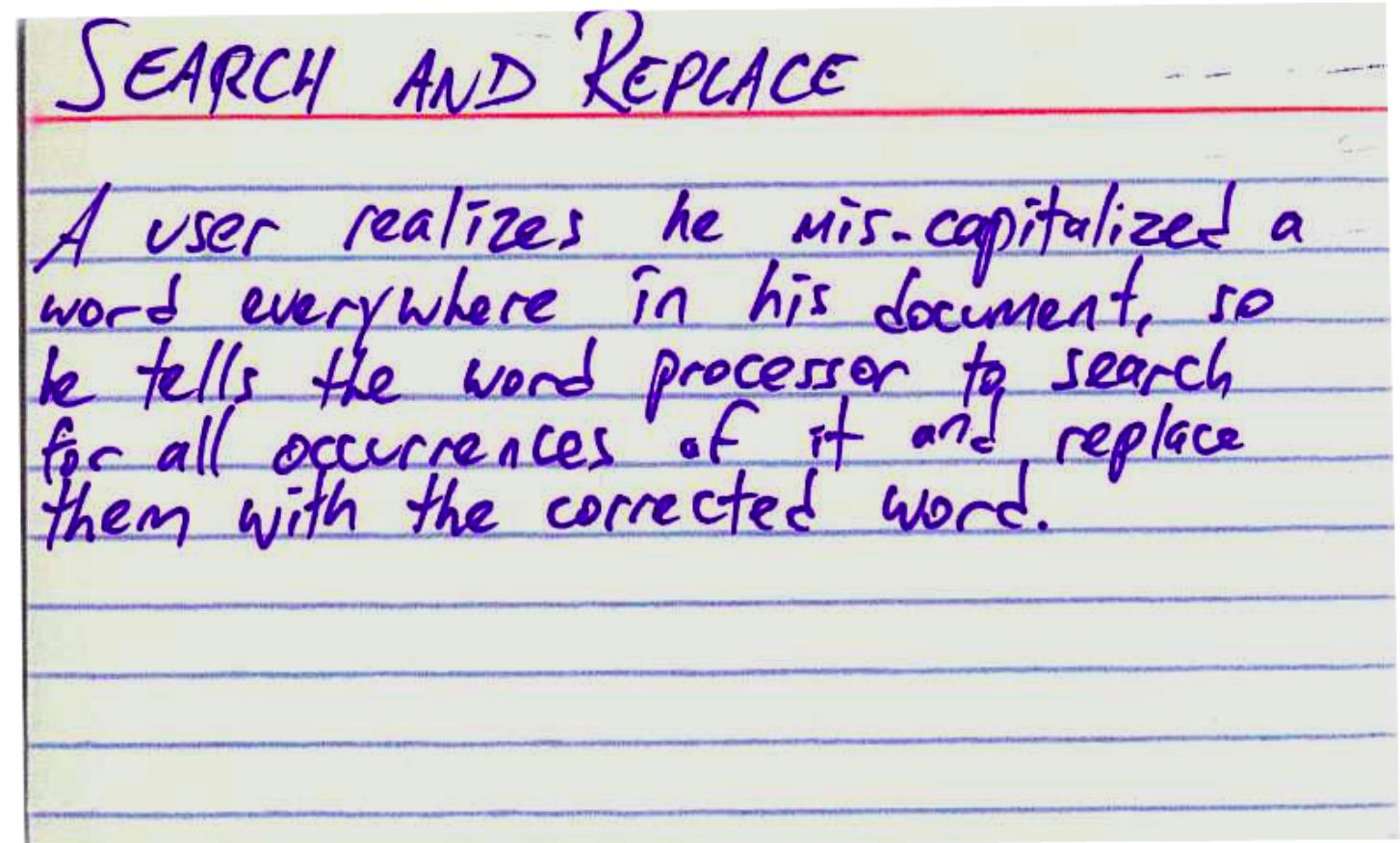
- The priority describes the importance of the requirement for the software to be developed
- If the requirement has a high value for the product (e.g. because it is the core functionality of an application), it has a high priority
- Examples for a priority scheme (that we also use in JIRA)
 - Prio 1 = Critical (Candidates for the first development sprints, must be part of the first product increment)
 - Prio 2 = Major (Must be realized within in the project, can be realized in one of the following product increments)
 - Prio 3 = Minor (Desirable, if there is enough time)
 - Prio 4 = Not Important (Might not be realized)
- ➔ Priorities of requirements can change during the project in Scrum
- ➔ **Prioritization is done by the Product Owner**

User Stories

A promise for conversation

Example for backlog items: User Stories

- A User Story includes a sentence that describes **what the user does or needs**
- Often written on a card
- Users (**roles**) are the actual **users** of the system



Properties of a good User Story: INVEST

- **Independent** - avoid overlapping User Stories
- **Negotiable** - A User Story is not a contract, but a basis for a discussion between Development Team and the Product Owner
- **Valuable** - for the user and the business. And **Vertical**: we plan and develop features and not layers
- **Estimable** - the stories on the Product Backlog represent the basis of our project plan
- **Small** - too large user stories should be partitioned into smaller ones, because the complexity increases over-proportional otherwise
- **Testable** - if a User Story is not testable, it might not be a real value for the product, this also implies **realizability**

User Story Template

As <role>,

I want to <feature>

so that <reason>

Main advantages of the template:

- Using the “I” phrasing, developer can identify with the given user role
- Defines a structure and simplifies the prioritization process
- Facilitates categorization according to user roles

Example of a User Story

As	customer
I want	a flexible possibility to store data in my database
so that	That's clear

Independent	
Negotiable	!
Valuable	!
Estimable	!
Small	!
Testable	!

→ bad example

Example of a User Story (2)

As sales representative
I want to have a possibility to determine the scale of discount of a customer
so that I am able to tell him a concrete offer with the correct price

Independent	
Negotiable	
Valuable	
Estimable	
Small	
Testable	

→ good example

Example of a User Story (3)

As	student
I want	to find my grades online
so that	I don't have to wait until the next day to know whether I passed the exam

Independent	
Negotiable	
Valuable	
Estimable	
Small	
Testable	

→ good example

Acceptance Criteria

- Conditions that a software product must satisfy to be accepted by a user, customer or other stakeholder
- Pre-established standards or requirements a product or project must meet
- Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify requirements applicable at the current stage of project integration
 - Functional (e.g. minimal marketable functionality)
 - Non-functional (e.g. minimal quality)
- These requirements represent “conditions of satisfaction.” There is no partial acceptance: either a criterion is met or it is not.

Definition of Done

- The Definition of Done (DoD) is usually a clear and concise list of requirements that a software Increment must adhere to for the team to call it complete
- Until this list is satisfied, a product Increment is not done
- ➔ Before a piece of functionality is in a potentially releasable state, it must adhere to a common understanding of completion by those working on the Increment
- Examples:
 - End-User Documentation is updated
 - Unit testing is complete and passed
 - Developer has peer reviewed code
 - Integration Testing is complete and passed
 - Configuration Management has added the functionality to the build and captured release notes
 - Product Owner has reviewed the demo and approved the User Story
 - User Acceptance Testing is complete and passed

Scenarios

Instance of a use case. A scenario represents a concrete sequence of interactions between one or more actors and the system.

Scenario Template

Scenario name	<i>Name</i>
Participating actors	Initiated by <i>actor</i>
Flow of events	1) <i>Actor step description</i>
	2) <i>System step description</i>
	3) <i>Actor step description</i>
	4) <i>System step description</i>
	...
Entry conditions	<i>Description</i>
Exit conditions	<i>Description</i>
Invariants	<i>Description</i>
Quality requirements	<i>Description</i>

Quality requirements
are also called non-
functional requirements

Scenario Example

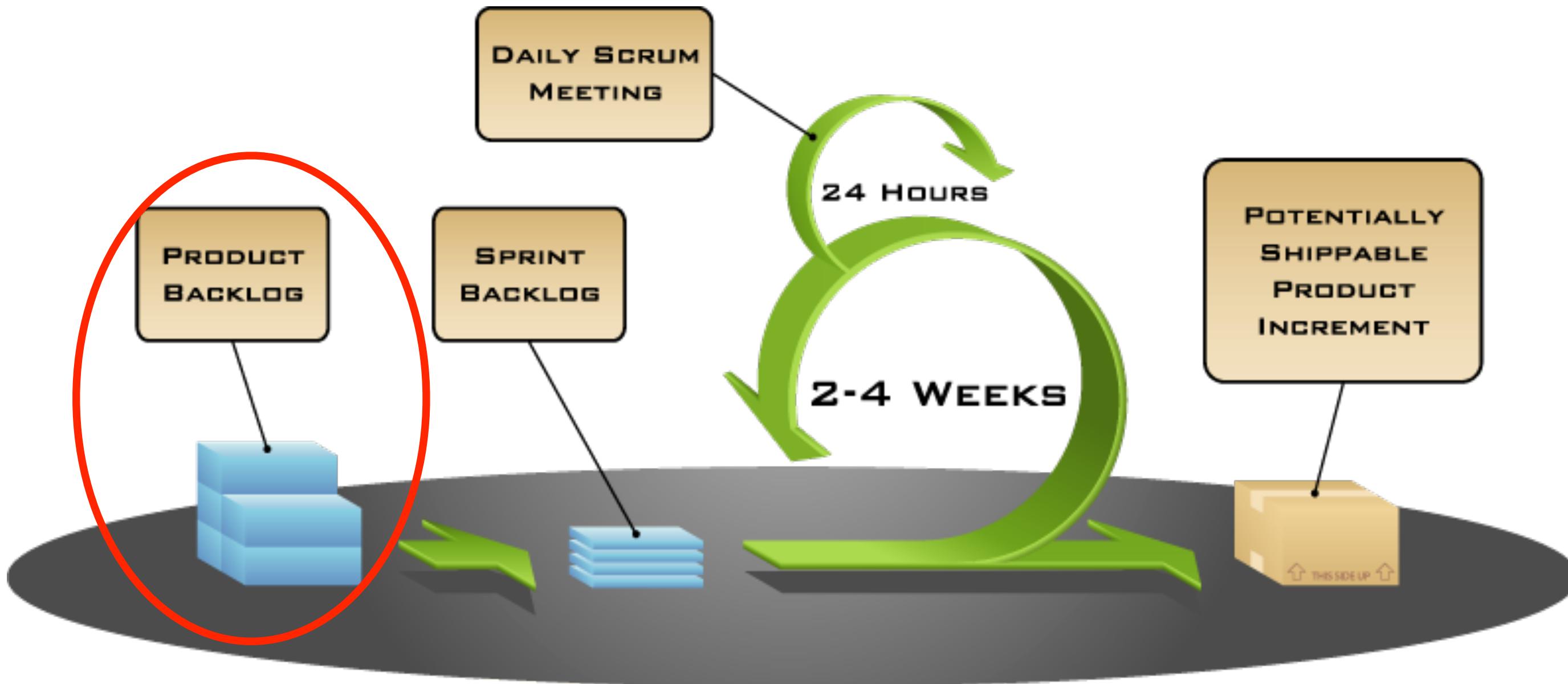
Scenario name	<i>Select Pedelec</i>
Participating actors	Initiated by <i>Marie, employee</i>
Flow of events	<ol style="list-style-type: none">1) <i>Marie opens the Pedelec App</i>
	<ol style="list-style-type: none">2) <i>The system shows a map with Marie's current location in the center of the screen and 3 Pedelecs near to Marie</i>
	<ol style="list-style-type: none">3) <i>Marie chooses the Pedelec that is closest to her</i>
	<ol style="list-style-type: none">4) <i>The app shows the range (working radius) of the chosen Pedelec as a dashed circle with the Pedelec as center</i>
Entry conditions	<i>3 Pedelecs are near to Marie</i>
Exit conditions	<i>The Pedelec is selected</i>
Invariants	-
Quality requirements	<i>Marie can select the Pedelec in 1 click</i>

Relation between User Stories, Use Cases and Scenarios

- Scenarios are typically created during Analysis
- Use Cases and Scenarios typically cover a larger scope and are more formal than User Stories
 - Scenarios and Use Cases have participating actors, preconditions, event flow, post condition and special quality requirements
 - Scenarios are instantiated use cases
 - More details about scenarios and use cases in chapter 4 of Object Oriented Software Engineering using UML by Bruegge and Dutoit
- User Stories are usually created during requirements elicitation

Source: Bernd Bruegge, Allen H. Dutoit, Object Oriented Software Engineering using UML, Design Patterns and Java

Exercise 1: Creation of the Product Backlog



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

The Pedelec App will be our running example problem statement for the exercises. **Notice:** You cannot choose it for the team project!



Time:
10 min

Exercise 1: Creation of the Product Backlog: User Stories and Scenarios

Task 1: Read the [Pedelec App Problem Statement](#) and use it for the following tasks

Task 2: Create 3 User Stories and write them into a Text Editor

Task 3: Create 1 small Scenario and write it into a Text Editor

Hint: Write user stories and scenarios for the functional requirements

Task 4: Prioritize these Product Backlog Items

Product Backlog: Example Solution - User Stories

ID	Summary	User Story	Priority
1	<input type="checkbox"/> Search available Pedelecs	As employee, I can search Pedelecs on a map so that I know where free Pedelecs are located	Critical
2	<input type="checkbox"/> Check working radius	As employee, I can check the working radius of a Pedelec so that I know whether I can reach my destination with it	Critical
3	<input type="checkbox"/> Reserve available Pedelec	As employee, I can reserve an available Pedelec for a certain time-slot so that I am sure that I can use it during this time-slot	Critical
4	<input type="checkbox"/> Return Pedelec	As employee, I can return a Pedelec after my trip so that others can again reserve it	Major
5	<input type="checkbox"/> Contact colleague	As employee, I can contact a colleague to obtain the reservation of his bike so that he knows that I need the bike urgently	Minor
6	<input type="checkbox"/> Pass reservation	As employee, I can pass my reservation of a Pedelec to my colleague so that he can obtain it	Minor
7	<input type="checkbox"/> Report damage	As employee, I can report the damage of a Pedelec so that the Pedelec can be repaired	Minor
8	<input type="checkbox"/> Unlock Pedelec	As employee, I can see a PIN so that I can unlock my reserved Pedelec	Major

Product Backlog: Example Solution - Scenarios (1)

Scenario name	<i>Search available Pedelec</i>
Participating actors	Initiated by <i>Marie, employee</i>
Flow of events	<p>1) <i>Marie opens the Pedelec App</i></p> <p>2) <i>The system shows a map with Marie's current location (Marienplatz) in the center of the screen and 3 Pedelecs near to Marie</i></p> <p>3) <i>Marie browses the map by zooming and moving the map</i></p> <p>4) <i>The app zooms and moves the map. The map shows the Pedelecs that are located on the map</i></p>
Entry conditions	-
Exit conditions	-
Invariants	-
Quality requirements	<i>Marie can zoom the map by pinching</i>

Product Backlog: Example Solution - Scenarios (2)

Scenario name	<i>Check working radius</i>	
Participating actors	Initiated by <i>Marie, employee</i>	
Flow of events	<ol style="list-style-type: none">1) <i>Marie selects one available Pedelec</i>	
		<ol style="list-style-type: none">2) <i>The system shows a dashed 20km radius around the Pedelec as center of the circle. It also shows a reserve button at the bottom of the page</i>
Entry conditions	<i>The app is already open and Pedelecs are visible on the map</i>	
Exit conditions	<i>The Pedelec is selected and the range of the Pedelec is shown</i>	
Invariants	<i>-</i>	
Quality requirements	<i>Marie can select a Pedelec with one click</i>	

Product Backlog: Example Solution - Scenarios (3)

Scenario name	<i>Reserve available Pedelec</i>
Participating actors	Initiated by <i>Marie, employee</i>
Flow of events	<p>1) <i>Marie selects the reserve button at the bottom of the page</i></p>
	<p>2) <i>The system shows a dialog for the start and end time of the reservation</i></p>
	<p>3) <i>Marie chooses 14:30 as start and 16:30 as end time and selects the confirm button</i></p>
	<p>4) <i>The app confirms the reservation for this time-slot</i></p>
Entry conditions	<i>The app is already open and one available Pedelec is selected</i>
Exit conditions	<i>The Pedelec is reserved for the chosen timeslot</i>
Invariants	-
Quality requirements	<i>Marie can reserve a Pedelec with four clicks</i>

Product Backlog: Example Solution - Scenarios (4)

Scenario name	<i>Return Pedelec</i>
Participating actors	Initiated by <i>Marie, employee</i>
Flow of events	<p>1) <i>Marie parks the Pedelec in the parking slot and locks it. Then she opens the app.</i></p> <p>2) <i>The app automatically displays the Pedelec that is currently in use (based on Marie's reservation time-slot)</i></p> <p>3) <i>Marie chooses the return button</i></p> <p>4) <i>The app confirms that Marie return the Pedelec and that it is now available for other reservations</i></p>
Entry conditions	<i>The Pedelec is reserved and in use</i>
Exit conditions	<i>The Pedelec is free for reservations again</i>
Invariants	-
Quality requirements	<i>Marie can return the Pedelec with one click</i>

Product Backlog: Example Solution (Summary)

Prioritized Product Backlog

ID	Summary	Priority
1	<input type="checkbox"/> Search available Pedelecs	Critical
2	<input type="checkbox"/> Check working radius	Critical
3	<input type="checkbox"/> Reserve available Pedelec	Critical
4	<input type="checkbox"/> Return Pedelec	Major
5	<input type="checkbox"/> Contact colleague	Minor
6	<input type="checkbox"/> Pass reservation	Minor
7	<input type="checkbox"/> Report damage	Minor
8	<input type="checkbox"/> Unlock Pedelec	Major

Outline

- Agile Methods: Scrum

- Artifacts
- Meetings
- Roles
- **Exercise 1:** Creation of the Product Backlog: User Stories and Scenarios



JIRA Agile

- **Exercise 2:** Creation of the Product Backlog in JIRA
- **Exercise 3:** Sprint Planning Part I - Creation of the Sprint Backlog in JIRA
- **Exercise 4:** Sprint Planning Part II - Refinement of the Sprint Backlog in JIRA
- **Exercise 5:** Working on the Taskboard during the Sprint
- **Exercise 6:** Sprint Review

JIRA Agile

→ Create and Manage the Product Backlog

- Create Backlog Items (e.g. Scenario, User Story)
- Estimate the difficulty
- Define sub-tasks for the **unit of work** during development
- Prioritization of backlog items and tasks
- Create and Manage Sprint Backlogs
 - Plan Sprints
 - Track progress in active Sprints
 - View reports about finished Sprints

JIRA Dashboard

All issues currently assigned to you.

Assigned to Me			
T	Key ↓	P	Summary
POM15EXAMPLE-4	POM15EXAMPLE-4		Create New Meeting
LS1ADMIN-1869	LS1ADMIN-1869		Einmal das Komplett paket
ECPP2P-83	ECPP2P-83		Gruppierung mit sofortiger Ausführung der Operationen ermöglichen
ECPP2P-82	ECPP2P-82		Exception bei Umlauten in ECPBinaryReader
ECPP2P-80	ECPP2P-80		Löschen einer Bibliothek wird nicht übertragen

1–5 of 27 1 2 3 4 5 6 ►

Issues in progress			
T	Key	P	Summary
ECPP2P-61	ECPP2P-61		ECPP2P-51 / Share Project gegen EMFStore implementieren und TestCase dafür erstellen

Activity Stream

POM 2015

Today

Christoph Winter edited Team List
(view change)
11 minutes ago Comment

Ljubomir Dshevlekov edited Team List
(view change)
20 minutes ago Comment

Niklas Scholz edited Team List
(view change)
21 minutes ago Comment

Praveer Rai edited Team List
(view change)
25 minutes ago Comment

Create Issue Screen

Create Issue Configure Fields

Project* POM 2015 Example Project

Issue Type* Task Some issue types are unavailable due to incompatible field configuration and/or workflow associations.

Assignee Automatic

Due Date Assign to me

Priority Major

Difficulty M The size of the issue, describing the amount of work to resolve it.

Description Component/s None

Create another Create Cancel

The difficulty of the issue describes the amount of work estimated to resolve it. It is measured with t-shirt sizes: S, M, L, XL

Configuration of the fields visible on your screen

Issue Type (e.g. Task, Bug, User Story, Scenario)

The Person responsible for the resolution of the issue

'Create' button

The difficulty of the issue describes the amount of work estimated to resolve it. It is measured with t-shirt sizes: S, M, L, XL

Level of Difficulty

- In our projects, T-Shirt sizes represent the rough difficulty of the realization of the issue, i.e. how much effort / time the Development Team needs to resolve the issue
 - Small (**S**)
 - Medium (**M**)
 - Large (**L**)
 - Extra Large (**XL**) —> too big, split it into smaller issues!
- In the Sprint Planning Meeting the team decides on the difficulty of the backlog items by estimating Story Points (1/2, 1, 2, 3, 5, 8, 13, 21, 100) using e.g.:
 - Poker Planning
 - Team Estimation Game
- ➔ Both are agile estimation techniques: we introduce more techniques in the *Estimation and Scheduling* lecture and exercise

Lecture on June 03
Exercise on June 05

You can edit
the issue

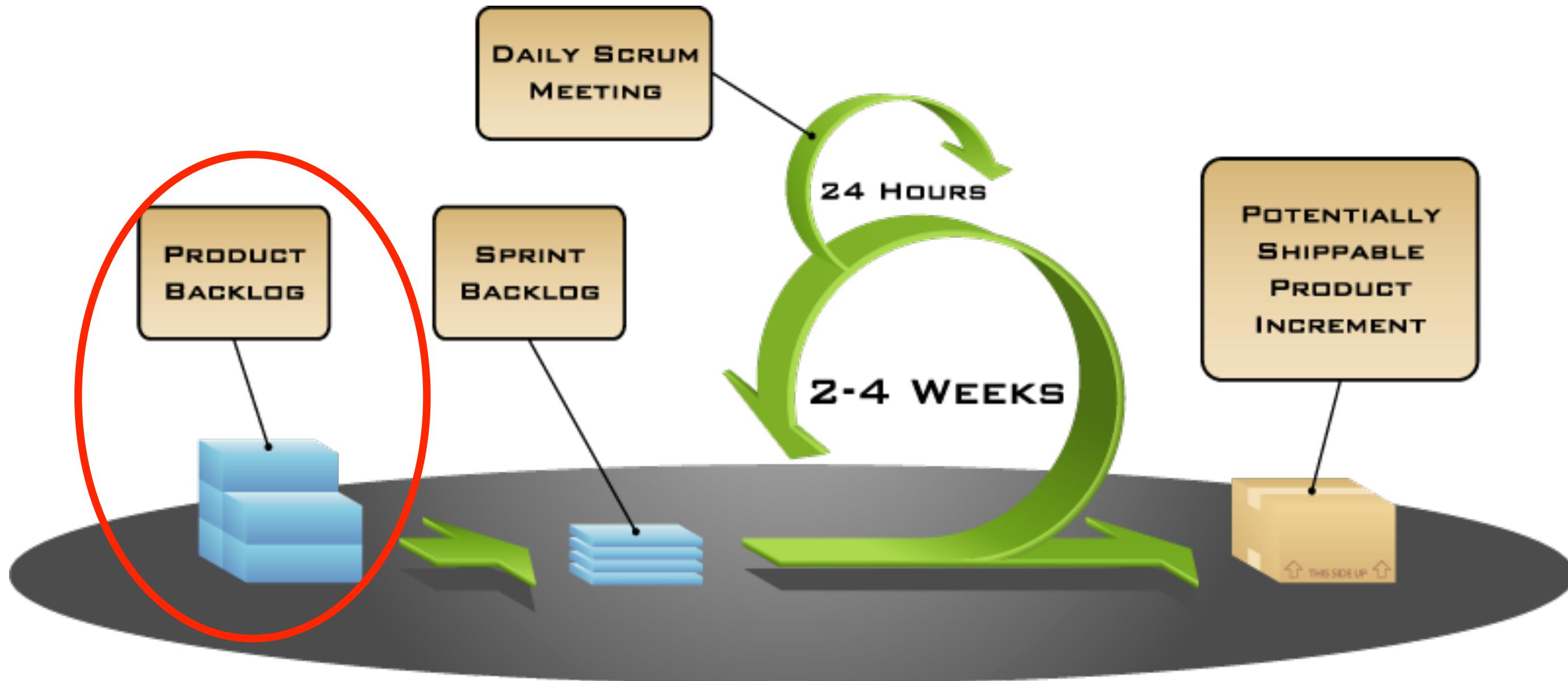
Issue Browser

The screenshot shows the LS1 JIRA Issue Browser interface. At the top, there's a navigation bar with links for LS1 JIRA, Dashboards, Projects, Issues, Agile, Create, Search, Help, Settings, and User Profile. The main content area displays an issue titled "[Stephan Krusche] Pedelec Selection" under the POM 2015 Example Project / POM15EXAMPLE-6. The issue details are as follows:

Type:	User Story	Status:	OPEN (View Workflow)
Priority:	Critical	Assignee:	Stephan Krusche (POM)
Difficulty:	M	Reporter:	Stephan Krusche (POM)
User Story Role:	User	Votes:	0
User Story Feature:	select a Pedelec	Watchers:	1 Stop watching this issue
User Story Reason:	I can see the range on the map	HipChat discussions	
Acceptance Criteria:	<input type="checkbox"/> The Range is displayed as a circle <input type="checkbox"/> The Pedelec is highlighted when it is selected	Dedicated room: Create a room Choose a room	

Below the details, there's a "Description" section with the placeholder text "Click to add description". A yellow callout bubble points to the "Edit" button in the toolbar with the text "You can edit the issue". Another yellow callout bubble points to the "Start Progress" and "Close" buttons with the text "You can change the issue's status". A third yellow callout bubble points to the "Votes" and "Watchers" sections with the text "You can vote and watch issues".

Exercise 2: Creation of the Product Backlog in JIRA



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



Time:
8 min

Exercise 2: Creation of the Product Backlog in JIRA

Task 5: Login to JIRA and open the POM 2015 Example Project

Task 6: Create 3 User Stories of **Exercise 1** in JIRA and **assign it to yourself**

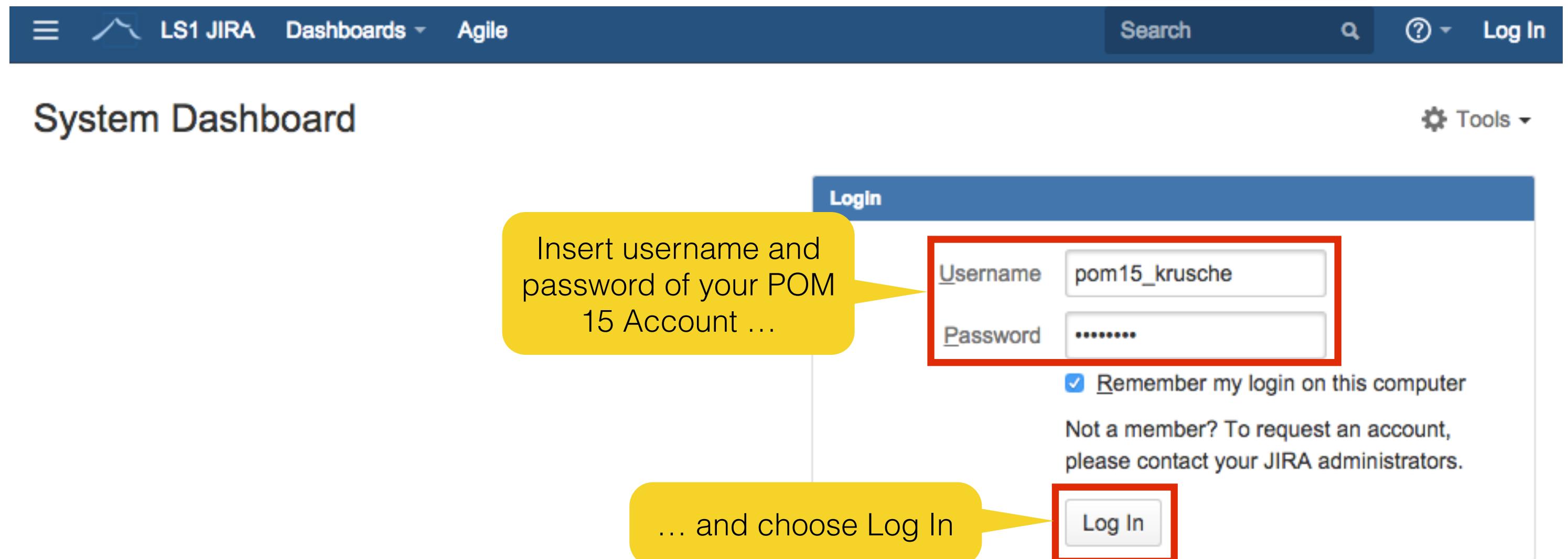
Task 7: Create 1 Scenario of **Exercise 1** in JIRA and **assign it to yourself**

Hint: Use the Prefix [Your Full Name] to easily find the issue later (just for exercise purposes)

Task 8: Open and inspect the Product Backlog in JIRA Agile

Task 4: Login into JIRA and open the POM 2015 Example Project

<https://jirabruegge.in.tum.de>



Task 4: Login into JIRA and open the POM 2015 Example Project

What's New in JIRA

1. Spend time where it counts

2. Drive more informed decisions

3. Create issues in the moment

Take a quick tour of the new features »

Subscribe to get JIRA tips, tricks, and product updates in your inbox:

Email

Don't show again. You will still be able to access this dialog through the 'Help' Menu in the application header.

Close What's New in JIRA

Task 4: Login into JIRA and open the POM 2015 Example Project

The screenshot shows the LS1 JIRA System Dashboard. At the top, there is a navigation bar with links for 'Dashboards', 'Projects', 'Issues', 'Agile', 'Create', 'Search', and user profile. A yellow callout bubble labeled 'Choose Projects' points to the 'Projects' button. Below the navigation bar, the dashboard displays sections for 'Assigned to Me' (no issues assigned), 'Issues in progress' (no matching issues found), and 'Favorite Filters' (no favorite filters). A yellow callout bubble labeled 'Choose POM 2015 Example Project' points to the 'POM 2015 Example Project (POM15E...)' link in the 'CURRENT PROJECT' section, which is highlighted with a red box.

Task 5: Create at least 2 User Stories in JIRA and assign it to yourself

<https://jirabruuegge.in.tum.de/browse/POM15EXAMPLE>

The screenshot shows the JIRA interface for the 'POM 2015 Example Project'. The top navigation bar includes links for JIRA + Agile, Confluence, Bamboo, Stash, HockeyApp, HipChat, and Balsamiq. Below the navigation bar, there are links for LS1 JIRA, Dashboards, Projects, Issues, Agile, and a prominent 'Create' button, which is highlighted with a red box and a yellow callout containing the text 'Click on Create'. The project details below the navigation bar show the key 'POM15EXAMPLE', lead 'Stephan Krusche', and category 'POM 2015'. The main content area features a summary section with the text 'Example Project for the POM 2015 Lecture'.

Task 5: Create at least 2 User Stories in JIRA and assign it to yourself

The screenshot shows the JIRA 'Create Issue' interface. At the top, there are buttons for 'Create Issue' and 'Configure Fields'. Below that, the 'Project' dropdown is set to 'POM 2015 Example Project'. The 'Issue Type' dropdown is highlighted with a red box and set to 'User Story'. A yellow callout points to this field with the text 'Choose User Story as issue type'. The 'Summary' field contains '[Stephan Krusche] Pedelec Selection'. The 'Assignee' field is populated with 'Stephan Krusche (POM)' and has a link 'Assign to me' below it. A yellow callout points to this field with the text 'Assign to yourself'. The 'Priority' is set to 'Critical'. The 'Due Date' and 'Difficulty' fields are present but not filled. A yellow callout points to the entire 'User Story' section with the text 'Fill out all fields to your needs (remember INVEST)'. The 'User Story Role' dropdown is set to 'User'. The 'User Story Feature' field contains 'select a Pedelec'. The 'User Story Reason' field contains 'I can see the range on the map'. Both of these fields are enclosed in a large red box. A yellow callout points to this box with the text 'Create appropriate Acceptance Criteria'. At the bottom, there is an 'Acceptance Criteria' section with a checkbox for 'The Range is displayed as a circle' and a link 'Add or modify issue items'. A yellow callout points to this section with the text 'Finally, click on the 'Create' button.' At the very bottom, there are buttons for 'Create another', 'Create', and 'Cancel'.

Create Issue

Configure Fields

Project * POM 2015 Example Project

Issue Type * User Story

Some issue types are unavailable due to incompatible field configuration and/or workflow associations.

User Story Infos

Summary * [Stephan Krusche] Pedelec Selection

Assignee Stephan Krusche (POM)

Assign to me

Priority Critical

Due Date

Difficulty M

User Story Role User

Begin typing to find and create labels or press down to select a suggested label.

The Role in a User Story: As a [role], I can [feature] so that [reason].

User Story Feature select a Pedelec

The feature of a user story: As a [role], I can [feature] so that [reason].

User Story Reason I can see the range on the map

The reason in a User Story: As a [role], I can [feature] so that [reason].

Acceptance Criteria The Range is displayed as a circle

Add or modify issue items

Create another Create Cancel

Choose User Story as issue type

Assign to yourself

Fill out all fields to your needs (remember INVEST).

Create appropriate Acceptance Criteria

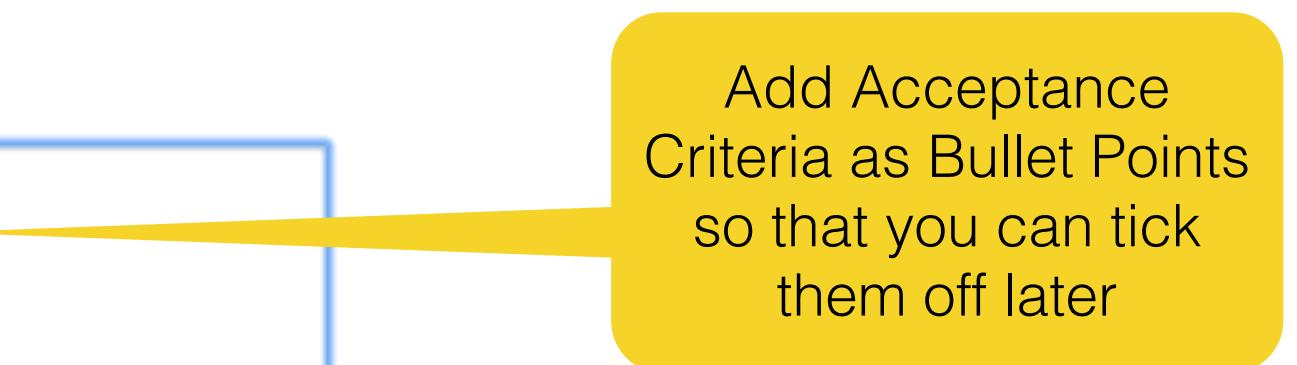
Finally, click on the 'Create' button.

Task 5: Create at least 2 User Stories in JIRA and assign it to yourself

Acceptance Criteria

- * The Range is displayed as a circle
- * The Pedelec is highlighted when it is selected

 [Return to Checkbox form](#)  [help](#)



Add Acceptance Criteria as Bullet Points so that you can tick them off later

Task 5: Create at least 2 User Stories in JIRA and assign it to yourself



Task 5: Create at least 2 User Stories in JIRA and assign it to yourself

The screenshot shows a JIRA issue page for a User Story. The page has a header with navigation links like LS1 JIRA, Dashboards, Projects, Issues, Agile, Create, Search, and a user profile. The main content area displays the issue details:

Issue Type: User Story

Details

- Type: User Story
- Status: OPEN (View Workflow)
- Priority: Critical
- Difficulty: M
- User Story Role: User
- User Story Feature: select a Pedelec
- User Story Reason: I can see the range on the map
- Acceptance Criteria:
 - The Range is displayed as a circle
 - The Pedelec is highlighted when it is selected

Description

Click to add description

Activity

All Comments History Activity Transitions

There are no comments yet on this issue.

People

Assignee: Stephan Krusche (POM)
Reporter: Stephan Krusche (POM)
Votes: 0
Watchers: 1 Stop watching this issue

HipChat discussions

Dedicated room:
Create a room Choose a room

Other rooms:
Issue mentioned in 0 rooms

Dates

Created: 1 minute ago
Updated: 1 minute ago

Task 7: Create at least 1 Scenario in JIRA and assign it to yourself

The screenshot shows the 'Create Issue' page in JIRA. The 'Project' dropdown is set to 'POM 2015 Example Project'. The 'Issue Type' dropdown is set to 'Scenario', which is highlighted with a red box and has a yellow callout pointing to it with the text 'Choose Scenario as issue type'. The 'Summary' field contains 'Return Pedelec'. The 'Assignee' dropdown is set to 'Stephan Krusche (POM)', which is also highlighted with a red box and has a yellow callout pointing to it with the text 'Assign to yourself'. The 'Priority' dropdown is set to 'Major'. The 'Due Date' field is empty. Below these fields is a large red box highlighting the 'Scenario' section. This section includes 'Participating Actors' (Marie, Employee), 'Scenario Flow of Events' (list of checkboxes for parking, locking, opening app, choosing return button, confirmation), 'Scenario Entry Conditions' (list of checkboxes for reservation status), 'Scenario Exit Conditions' (list of checkboxes for availability), and 'Scenario Quality Requirements' (list of checkboxes for one-click return). A yellow callout pointing to this section with the text 'Fill out all fields of a scenario'. At the bottom right of the page is a yellow callout pointing to the 'Create' button with the text 'Finally, click on the 'Create' button. Choose 'Create another' to save time when adding multiple issues'. The 'Create' button is highlighted with a red box.

Create Issue

Configure Fields

Project * POM 2015 Example Project

Issue Type * Scenario

Some issue types are unavailable due to incompatible field configuration and/or workflow associations.

Scenario Infos

Summary * Return Pedelec

Assignee Stephan Krusche (POM)

Priority Major

Due Date

Scenario

Participating Actors Marie Employee

Begin typing to find and create labels or press down to select a suggested label.

the participating actors of this scenario

Scenario Flow of Events

- Marie parks the Pedelec in the parking slot and locks it. Then she opens the app.
- The app automatically displays the Pedelec that is currently in use (based on Marie's reservation time-slot)
- Marie chooses the return button
- The app confirms that Marie return the Pedelec and that it is now available for other reservations

Add or modify issue items

the event flow of the scenario with user and system steps

Scenario Entry Conditions

- The Pedelec is reserved and in use

Add or modify issue items

Scenario Exit Conditions

- The Pedelec is free for reservations again

Add or modify issue items

Scenario Quality Requirements

- Marie can return the Pedelec with one click

Add or modify issue items

Create another Create Cancel

Choose Scenario as issue type

Assign to yourself

Fill out all fields of a scenario

Finally, click on the 'Create' button. Choose 'Create another' to save time when adding multiple issues

Task 7: Create at least 1 Scenario in JIRA and assign it to yourself

Scenario Flow of Events

- * Marie parks the Pedelec in the parking slot and locks it. Then she opens the app.
- * The app automatically displays the Pedelec that is currently in use (based on Marie's reservation time-slot)
- * Marie chooses the return button
- * The app confirms that Marie return the Pedelec and that it is now available for other reservations

 [Return to Checkbox form](#)  [help](#)

the event flow of the scenario with user and system steps

Scenario Entry Conditions

- * The Pedelec is reserved and in use

Use the Bullet Point notation (with stars) for Flow of Events, Entry Conditions, Exit Conditions and Quality Requirements

Note: JIRA does not support Scenarios natively and the current screen is a workaround to support it :-)

Task 8: Open and inspect the Product Backlog in JIRA Agile

The screenshot shows the JIRA Agile interface. At the top, there is a navigation bar with links for LS1 JIRA, Dashboards, Projects, Issues, Agile (which is currently selected and has a red box around it), Create, Search, Help, Settings, and User Profile. Below the navigation bar, the main content area displays a project summary for 'POM 2015 Example Project / POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection'. This summary includes buttons for Edit, Comment, Assign, More, and Start Progress. To the right of the summary is a 'Details' section with fields for Type (User Story), Priority (Critical), Status (Open), and Assignee (Stephan Krusche (POM)). A yellow speech bubble points to the 'More' button with the text: 'If you see "more" instead of "View All Boards", click on "more"'. A yellow arrow points from this bubble to the 'More' button. A yellow speech bubble also points to the 'View All Boards' option in the 'RECENT BOARDS' dropdown menu with the text: 'Click on Agile'.

If you see “more” instead of “View All Boards”, click on “more”

Click on Agile

Click on View All Boards

Task 8: Open and inspect the Product Backlog in JIRA Agile

The screenshot shows the JIRA Agile Boards page. At the top, there is a navigation bar with links for LS1 JIRA, Dashboards, Projects, Issues, Agile, Create, Search, Help, Settings, and User Profile. Below the navigation bar, a yellow callout bubble says: "Search the board name: In this exercise the board includes your full name". On the left, there is a sidebar with a "Boards" section and a "Create board" button. The main area is titled "All boards" and has a "Board Type: All" dropdown set to "Stephan Krusche", which is highlighted with a red box. A yellow callout bubble points to this dropdown with the text: "Click on the Board to open it". Below the dropdown, a table lists a single board entry:

Board name	Board type	Administrators	Saved Filter	Visibility	...
Pom15 Agile Board Stephan Krusche	Scrum	Stephan Krusche	Filter for Pom15 Agile Board Stephan Krusche 2	RESTRICTED	...

Task 8: Open and inspect the Product Backlog in JIRA Agile

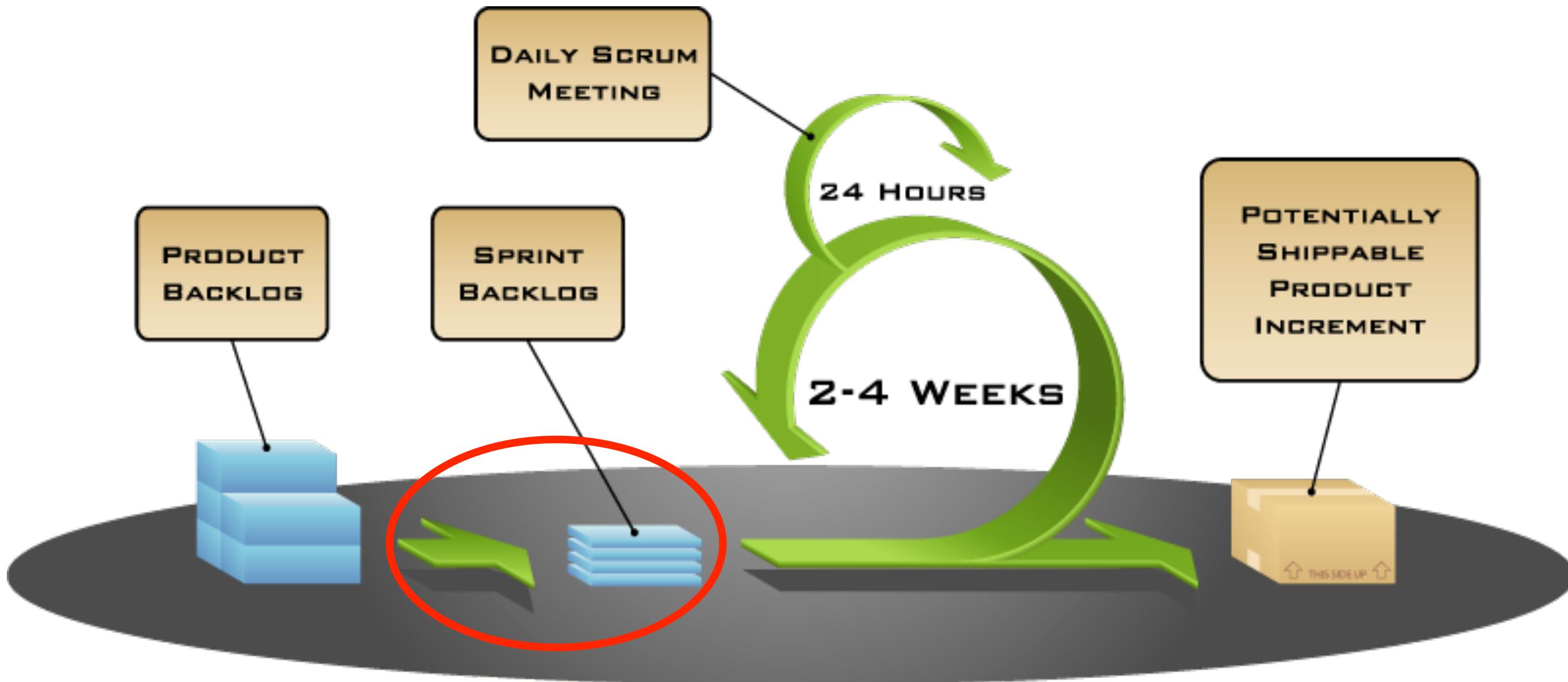
The screenshot shows the JIRA Agile interface for the 'Pom15 Agile Board Stephan Krusche'. The top navigation bar includes 'LS1 JIRA', 'Dashboards', 'Projects', 'Issues', 'Agile', 'Create', 'Search', and user profile icons. A yellow callout bubble points to the 'Choose Backlog' button in the top right. Below the navigation, a secondary menu bar has tabs for 'Backlog' (which is highlighted with a red box), 'Active sprints', 'Reports', 'Board', and a collapse icon. A 'QUICK FILTERS' section contains a search icon, a 'Only My Issues' button (also highlighted with a red box), and a 'Recently Updated' link. On the left, a vertical sidebar lists 'VERSIONS' and 'EPICS'. The main content area displays the 'Backlog' with 1 of 5 issues visible. An issue titled 'POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection' is listed, with its title also highlighted with a red box. A yellow callout bubble points to this issue with the text 'Click on one issue in the Product Backlog'. A 'Create Sprint' button is located in the top right corner of the backlog view.

Task 8: Open and inspect the Product Backlog in JIRA Agile

The screenshot shows the JIRA Agile interface for the 'Pom15 Agile Board Stephan Krusche' project. The left sidebar has 'VERSIONS' and 'EPICS' sections. The main area shows the 'Backlog' with 5 of 9 issues visible. A yellow callout labeled 'Product Backlog' points to the backlog list, which is highlighted with a red border. Another yellow callout labeled 'Issue Detail View' points to the details of the first issue in the backlog, which is also highlighted with a red border. The issue details include:

POM 2015 Example Project / POM15EXAMPLE-7	
[Stephan Krusche]	Return Pedelec
Details	
Status:	OPEN
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None
Epic:	None
People	
Reporter:	Stephan Krusche (POM)
Assignee:	Stephan Krusche (POM)

Exercise 3: Sprint Planning Part I - Creation of the Sprint Backlog in JIRA



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Sprint Planning Meeting

- The Development Team estimates the difficulty for the most important items in the Product Backlog
- Development Team and Product Owner select Product Backlog items that can be realized in the Sprint
 - The Development Team negotiates with the Product Owner how many items it can realize in the Sprint
 - The Product Owner defines when an item is accepted (e.g. using acceptance criteria)
 - **Important:** The Sprint Backlog cannot be changed by the Product Owner within the Sprint to protect the team from too many changes

Prioritized and Estimated Product Backlog

ID	Name	Priority	Difficulty
1	<input type="checkbox"/> Search available Pedelecs	Critical	Medium
2	<input type="checkbox"/> Check working radius	Critical	Large
3	<input type="checkbox"/> Reserve available Pedelec	Critical	Small
4	<input type="checkbox"/> Return Pedelec	Major	Small
5	<input type="checkbox"/> Contact colleague	Minor	Medium
6	<input type="checkbox"/> Pass reservation	Minor	Medium
7	<input type="checkbox"/> Report damage	Minor	Large
8	<input type="checkbox"/> Unlock Pedelec	Major	Small



Sprint 1 Backlog

ID	Name	Difficulty
1	<input type="checkbox"/> Search available Pedelecs	Medium
2	<input type="checkbox"/> Check working radius	Large



Time:
3 min

Exercise 3: Sprint Planning Part I - Creation of the Sprint Backlog in JIRA

Task 9: Create a new Sprint

Task 10: Move 2 Backlog Items to the new Sprint Backlog

Task 11: Start the Sprint

Task 9: Create a new Sprint

The screenshot shows the JIRA interface for the 'Pom15 Agile Board Stephan Krusche' project. The top navigation bar includes 'LS1 JIRA', 'Dashboards', 'Projects', 'Create', 'Search', and user profile options. A yellow callout bubble points to the 'Create' button with the text 'Click on Create Sprint'. Below the header, there are filters for 'QUICK FILTERS: Only My Issues' and 'Recently Updated'. The left sidebar has tabs for 'VERSIONS' and 'EPICS'. The main area shows the 'Backlog' with 5 of 9 issues visible. One issue is selected, shown in a detailed view on the right:

POM 2015 Example Project / POM15EXAMPLE-6
[Stephan Krusche] Pedelec Selection
Estimate: *Unestimated*
Details
Status: OPEN
Component/s: None

A yellow callout bubble on the right side notes: 'Note: Independent of the difficulty, JIRA displays "Unestimated" because we did not use Story Points yet'.

Issue Key	Description	Assignee	Status
POM15EXAMPLE-6	Pedelec Selection	Stephan Krusche	Open
POM15EXAMPLE-7	Return Pedelec	Stephan Krusche	Open
POM15EXAMPLE-8	Search available Pedelecs	Stephan Krusche	Open
POM15EXAMPLE-9	Check working radius	Stephan Krusche	Open
POM15EXAMPLE-10	Reserve available Pedelec	Stephan Krusche	Open

Note: You
might need to
rename the
Sprint

Task 9: Create a new Sprint

Sprint
Backlog
(empty)

Product
Backlog

Pom15 Agile Board Stephan Krusche

QUICK FILTERS: Only My Issues Recently Updated

Sprint 1 0 issues Start Sprint ... Linked pages

PLAN YOUR SPRINT
This is a sprint. Plan a sprint by dragging issues here.

+ Create issue 0 issues Estimate 0

Backlog 5 of 9 issues visible Clear all filters Create Sprint

- POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection
- POM15EXAMPLE-7 [Stephan Krusche] Return Pedelec
- POM15EXAMPLE-8 [Stephan Krusche] Search available Pedelecs
- POM15EXAMPLE-9 [Stephan Krusche] Check working radius
- POM15EXAMPLE-10 [Stephan Krusche] Reserve available Pedelec

+ Create issue

POM 2015 Example Project / POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection Estimate: Unestimated Details Status: OPEN Component/s: None Affects Version/s: None Fix Version/s: None Epic: None People Reporter: Stephan Krusche (POM) Assignee: Stephan Krusche (POM) Dates Created: 21.04.2015 19:28 Updated: 21.04.2015 19:28

Task 10: Move 2 Backlog Items to the new Sprint Backlog

Move the backlog items from the Product Backlog into the Sprint 1 Backlog

Pom15 Agile Board Stephan Krusche

QUICK FILTERS: Only My Issues Recently Updated

Sprint 1 0 issues Start Sprint ... Linked pages

VERSIONS EPICS

PLAN YOUR SPRINT
This is a sprint. Plan a sprint by dragging issues here.

+ Create issue 0 issues Estimate 0

Backlog 5 of 9 issues visible Clear all filters Create Sprint

POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection
POM15EXAMPLE-7 [Stephan Krusche] Return Pedelec
POM15EXAMPLE-8 [Stephan Krusche] Search available Pedelecs
POM15EXAMPLE-9 [Stephan Krusche] Check working radius
POM15EXAMPLE-10 [Stephan Krusche] Reserve available Pedelec

+ Create issue

POM 2015 Example Project / POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection Estimate: Unestimated Details Status: OPEN Component/s: None Affects Version/s: None Fix Version/s: None Epic: None People Reporter: Stephan Krusche (POM) Assignee: Stephan Krusche (POM) Dates Created: 21.04.2015 19:28 Updated: 21.04.2015 19:28

The screenshot shows a Jira Agile board interface. On the left, a yellow callout box contains the instruction: "Move the backlog items from the Product Backlog into the Sprint 1 Backlog". Two red arrows point from the backlog items listed under "Backlog" to the "Sprint 1" section. The "Sprint 1" section has a placeholder message: "This is a sprint. Plan a sprint by dragging issues here." To the right of the board, a detailed view of an issue titled "POM15EXAMPLE-6" is shown, including its status, components, and dates.

Task 10: Move 2 Backlog Items to the new Sprint Backlog

The screenshot shows a JIRA Agile board titled "Pom15 Agile Board Stephan Krusche". The board has two main sections: "Sprint 1" and "Backlog". A yellow callout bubble labeled "Sprint 1 Backlog" points to the top section, which contains two issues: "POM15EXAMPLE-8 [Stephan Krusche] Search available Pedelecs" and "POM15EXAMPLE-9 [Stephan Krusche] Check working radius". A yellow callout bubble labeled "Product Backlog" points to the bottom section, which contains three issues: "POM15EXAMPLE-6 [Stephan Krusche] Pedelec Selection", "POM15EXAMPLE-7 [Stephan Krusche] Return Pedelec", and "POM15EXAMPLE-10 [Stephan Krusche] Reserve available Pedelec". To the right of the board, a detailed view of issue "POM15EXAMPLE-9" is shown, including its title, description, status, and various metadata fields.

Pom15 Agile Board Stephan Krusche

QUICK FILTERS: Only My Issues Recently Updated

Sprint 1 2 issues

Start Sprint ... Linked pages

EPICS

Sprint 1 Backlog

Product Backlog

Backlog 3 of 7 issues visible Clear all filters Create Sprint

POM 2015 Example Project / POM15EXAMPLE-9

[Stephan Krusche] Check working radius

Estimate: Unestimated

Details

Status: OPEN

Component/s: None

Affects Version/s: None

Fix Version/s: None

Epic: None

People

Reporter: Stephan Krusche (POM)

Assignee: Stephan Krusche (POM)

Task 11: Start the Sprint

LS1 JIRA Dashboards Projects Issues Agile Create Search ?

Pom15 Agile Board Stephan Krusche

Backlog Active sprints Reports Board

QUICK FILTERS: Only My Issues Recently Updated

VERSIONS EPICS

Sprint 1 2 issues

Start Sprint Linked pages

Click on Start Sprint

POM 2015 Example Project / POM15EXAMPLE-9 [Stephan Krusche] Check working radius Estimate: Unestimated

POM15EXAMPLE-8 [Stephan Krusche] Search available Pedelecs

POM15EXAMPLE-9 [Stephan Krusche] Check working radius

+ Create issue

2 issues Estimate 0

Details Status: OPEN Component/s: None Affects Version/s: None

Task 11: Start the Sprint

Ignore the warning because we do not yet use Story Points

Start Sprint

⚠ Issues POM15EXAMPLE-8 and POM15EXAMPLE-9 do not have a value for the 'Estimate' field. Values entered after the start of the sprint will be treated as scope change.

2 issues will be included in this sprint.

Make sure to use Sprint 1 as name. Leave the default duration of 2 weeks

Sprint Name: * Sprint 1

Start Date: * 21.04.15 23:40

End Date: * 5.05.15 23:40

Click Start

Task 11: Start the Sprint

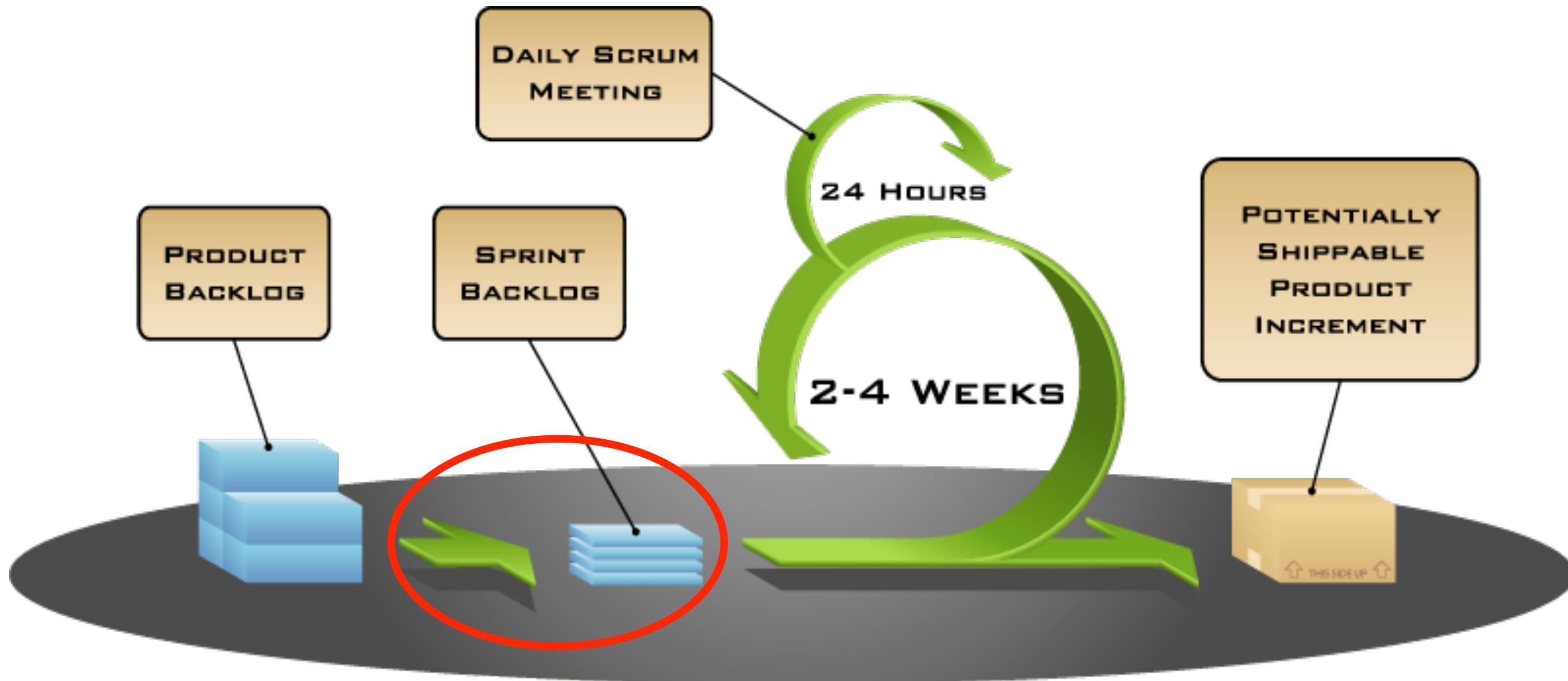
JIRA Agile automatically activates the page Active sprints

The screenshot shows a JIRA Agile board titled "Pom15 Agile Board Stephan Krusche". The top navigation bar includes "LS1 JIRA", "Dashboards", "Projects", "Issues", "Agile", "Create", "Search", and user profile icons. Below the title, there are filters for "SPRINT: Sprint 1", "QUICK FILTERS: Only My Issues", and "Recently Updated". The board has three columns: "To Do", "In Progress", and "Done". The "To Do" column contains two tasks:

- POM15EXAMPLE-8: [Stephan Krusche] Search available Pedelecs
- POM15EXAMPLE-9: [Stephan Krusche] Check working radius

The "Active sprints" button in the top right is highlighted with a red box. A yellow speech bubble points to this button with the text "JIRA Agile automatically activates the page Active sprints".

Exercise 4: Sprint Planning Part II - Refinement of the Sprint Backlog in JIRA



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Refinement of the Sprint Backlog

- The Development Team needs to discuss the concrete implementation to work on a backlog item
- The Development Team creates tasks to realize the backlog items (e.g. to match the acceptance criteria defined by the product owner)

Backlog Item

Sprint 1 Backlog

ID	Name	Difficulty
1	<input type="checkbox"/> Search available Pedelecs <input type="checkbox"/> Display Pedelec on Map <input type="checkbox"/> Display user location on Map	Medium
2	<input type="checkbox"/> Check working radius <input type="checkbox"/> Let the user select a Pedelec <input type="checkbox"/> Show the range as radius on the map <input type="checkbox"/> Show a reservation button	Large

Task



Time:
5 min

Exercise 4: Sprint Planning Part II - Refinement of the Sprint Backlog in JIRA

Task 12: Create at least 2 Sub-Tasks for the 2 Sprint Backlog Items and assign it to yourself

Task 12: Create Sub-Tasks for Sprint 1 Backlog Items and assign it to yourself

The screenshot shows a JIRA Agile board for the project "Pom15 Agile Board Stephan Krusche". The board has three columns: To Do, In Progress, and Done. A backlog item titled "POM15EXAMPLE-9" is selected and highlighted with a red border. A yellow callout bubble points to this item with the text "Select one Backlog Item". Above the board, a yellow callout bubble points to the "Active sprints" tab, which is highlighted with a red border. On the right side, a detailed view of the selected backlog item "POM15EXAMPLE-9" is shown. This view includes fields for title, assignee, estimate, and sub-tasks. A yellow callout bubble points to the "Create Sub-Task" button, which is also highlighted with a red border. Another yellow callout bubble points to a checkbox labeled "0" with the text "Choose Sub-Tasks".

Pom15 Agile Board Stephan Krusche

SPRINT: Sprint 1 QUICK FILTERS: Only My Issues Recently Up

To Do In Progress Done

Switch to Active Sprints tab

Select one Backlog Item

Choose Sub-Tasks

Backlog Active sprints Reports Board

POM 2015 Example Project / POM15EXAMPLE-9 [Stephan Krusche] Check working radius Estimate: Unestimated Sub-Tasks Create Sub-Task There are no sub-tasks Development Create branch

Click on Create Sub-Task

Task 12: Create Sub-Tasks for Sprint 1 Backlog Items and assign it to yourself

Create Subtask : POM15EXAMPLE-8 Configure Fields ▾

Issue Type * Sub-Task ?

Some issue types are unavailable due to incompatible field configuration and/or workflow associations.

Field Tab Info

Summary * [Stephan Krusche] Display user location on Map

Assignee Stephan Krusche (POM) Assign to me

Due Date 29.04.15 Calendar icon

Priority Major ?

Difficulty M

The size of the issue, describing the amount of work to resolve it.

Description

Create another Create Cancel

Issue Type Sub-Task is preselected

Choose a summary that starts with [Your Full Name] and assign to yourself

Click on Create

TUM © 2015 Bernd Bruegge, Stephan Krusche

POM SS15 - Ex. 02 - Agile Methods

74

Task 12: Create Sub-Tasks for Sprint 1 Backlog Items and assign it to yourself

The screenshot shows an Agile board in LS1 JIRA. The board has three columns: To Do, In Progress, and Done. There are two main backlog items visible:

- POM15EXAMPLE-8** 2 sub-tasks [Stephan Krusche] Search available Pedelecs
 - POM15EXAMPLE-11** [Stephan Krusche] Display Pedelec on Map
 - POM15EXAMPLE-12** [Stephan Krusche] Display user location on Map
- POM15EXAMPLE-9** 3 sub-tasks [Stephan Krusche] Check working radius
 - POM15EXAMPLE-13** [Stephan Krusche] Let the user select a Pedelec
 - POM15EXAMPLE-14** [Stephan Krusche] Show the range as radius on the map
 - POM15EXAMPLE-15** [Stephan Krusche] Show a reservation button

Two yellow callout boxes with arrows point to the backlog items:

- A yellow callout box points to the first backlog item (**POM15EXAMPLE-8**) with the text "Backlog Item with 2 Sub-Tasks".
- A yellow callout box points to the second backlog item (**POM15EXAMPLE-9**) with the text "Backlog Item with 3 Sub-Tasks".

Task 12: Create Sub-Tasks for Sprint 1 Backlog Items and assign it to yourself

The screenshot shows a JIRA Taskboard for the 'Pom15 Agile Board Stephan Krusche' under 'Sprint 1'. The board has columns for 'To Do', 'In Progress', and 'Done'. In the 'To Do' column, two issues are visible: 'POM15EXAMPLE-8' and 'POM15EXAMPLE-9'. 'POM15EXAMPLE-8' has two sub-tasks: 'POM15EXAMPLE-11' and 'POM15EXAMPLE-12'. 'POM15EXAMPLE-9' has three sub-tasks: 'POM15EXAMPLE-13', 'POM15EXAMPLE-14', and 'POM15EXAMPLE-15'. A yellow callout bubble points from the text 'Sub-Task also appear in the Issue Detail View in the Taskboard' to the sub-task list for 'POM15EXAMPLE-8'. To the right of the board is the detailed view for 'POM15EXAMPLE-8', showing its summary, estimate (Unestimated), and sub-tasks.

To Do In Progress Done

Sub-Task also appear in the Issue Detail View in the Taskboard

POM 2015 Example Project / POM15EXAMPLE-8
[Stephan Krusche] Search available Pedelecs
Estimate: Unestimated
Sub-Tasks
Create Sub-Task

Issue Key	Summary	Status
POM15EXAMPLE-11	[Stephan Krusche] Display Pedelec on Map	OPE
POM15EXAMPLE-12	[Stephan Krusche] Display user location on Map	OPE

Development
Create branch

Task 12: Create Sub-Tasks for Sprint 1 Backlog Items and assign it to yourself

The screenshot shows a JIRA issue page for a User Story titled "[Stephan Krusche] Search available Pedelecs". The User Story is categorized as a "User Story" with a priority of "Critical". It has a status of "OPEN" and is assigned to Stephan Krusche (POM). The User Story is described as "search Pedelecs on a map" and "I know where free Pedelecs are located". Acceptance criteria include "The user sees all Pedelecs on a map" and "The user can see his own location". A yellow callout bubble points from the "Sub-Tasks" section to the text "Sub-Task also appear in the Issue Browser".

POM 2015 Example Project / POM15EXAMPLE-8

[Stephan Krusche] Search available Pedelecs

Edit Comment Assign More Start Progress Close Export

Details

Type: User Story Status: OPEN (View Workflow)

Priority: Critical

Difficulty: M

User Story Role: Employee

User Story Feature: search Pedelecs on a map

User Story Reason: I know where free Pedelecs are located

Acceptance Criteria:

- The user sees all Pedelecs on a map
- The user can see his own location

Description

Click to add description

Sub-Tasks

Progress

1. [Stephan Krusche] Display Pedelec on Map
2. [Stephan Krusche] Display user location on Map

People

Assignee: Stephan Krusche (POM)

Reporter: Stephan Krusche (POM)

Votes: 0

Watchers: 1 Stop watching this issue

HipChat discussions

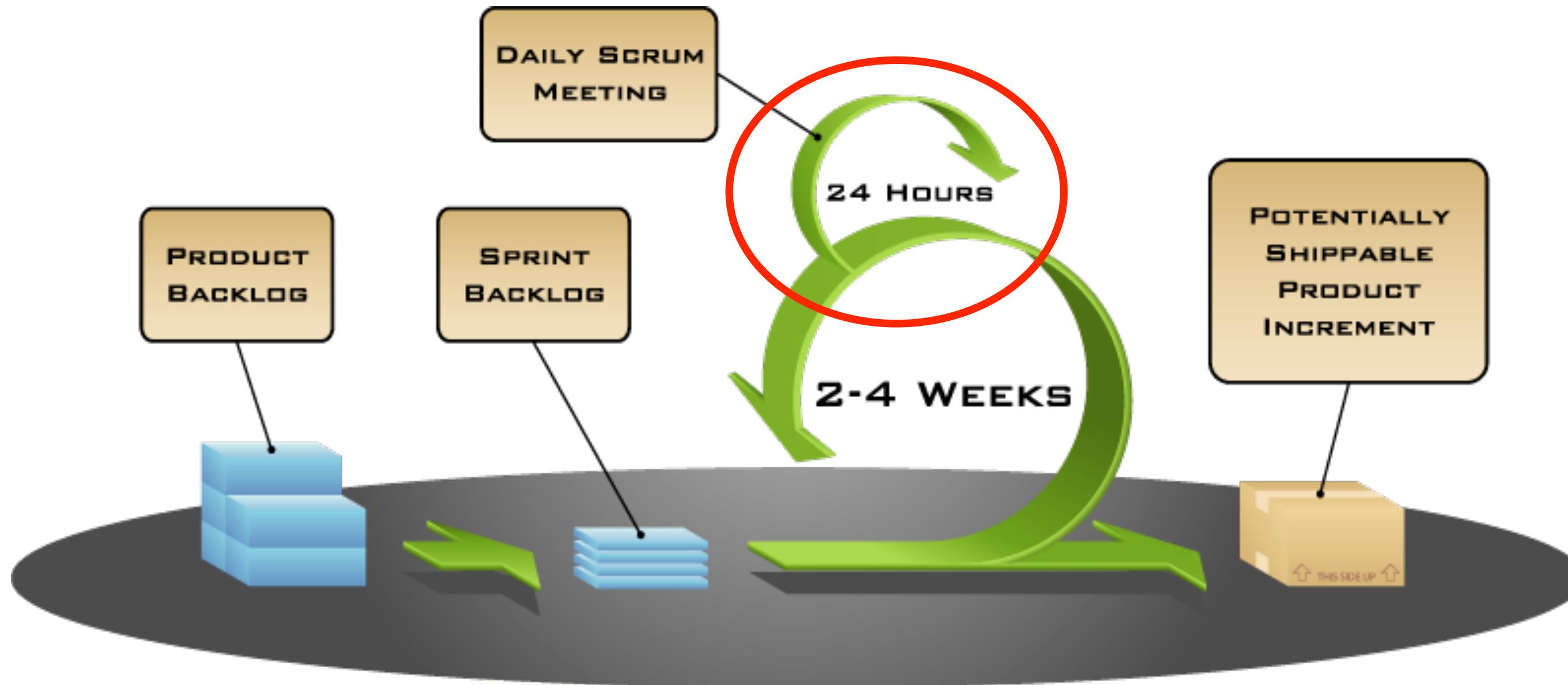
Dedicated room: Create a room Choose a room

Other rooms: Issue mentioned in 0 rooms

Dates

Created:

Exercise 5: Working on the Taskboard during the Sprint



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Daily Scrum Meeting

- Main purpose: Risk reduction by early information sharing and discussion
- 15 min **standup meeting** every day
- Every developer answers the following three questions:
 - Status: What did you do since the last meeting?
 - Impediments: Are there any impediments in your way? (also called blockers)
 - Promises: What do you promise to resolve until the next meeting?

What would you
do if you cannot
meet daily?



Time:
5 min

Exercise 5: Working on the Taskboard during the Sprint

Task 13: Open the Taskboard and start/resolve Sub-Tasks

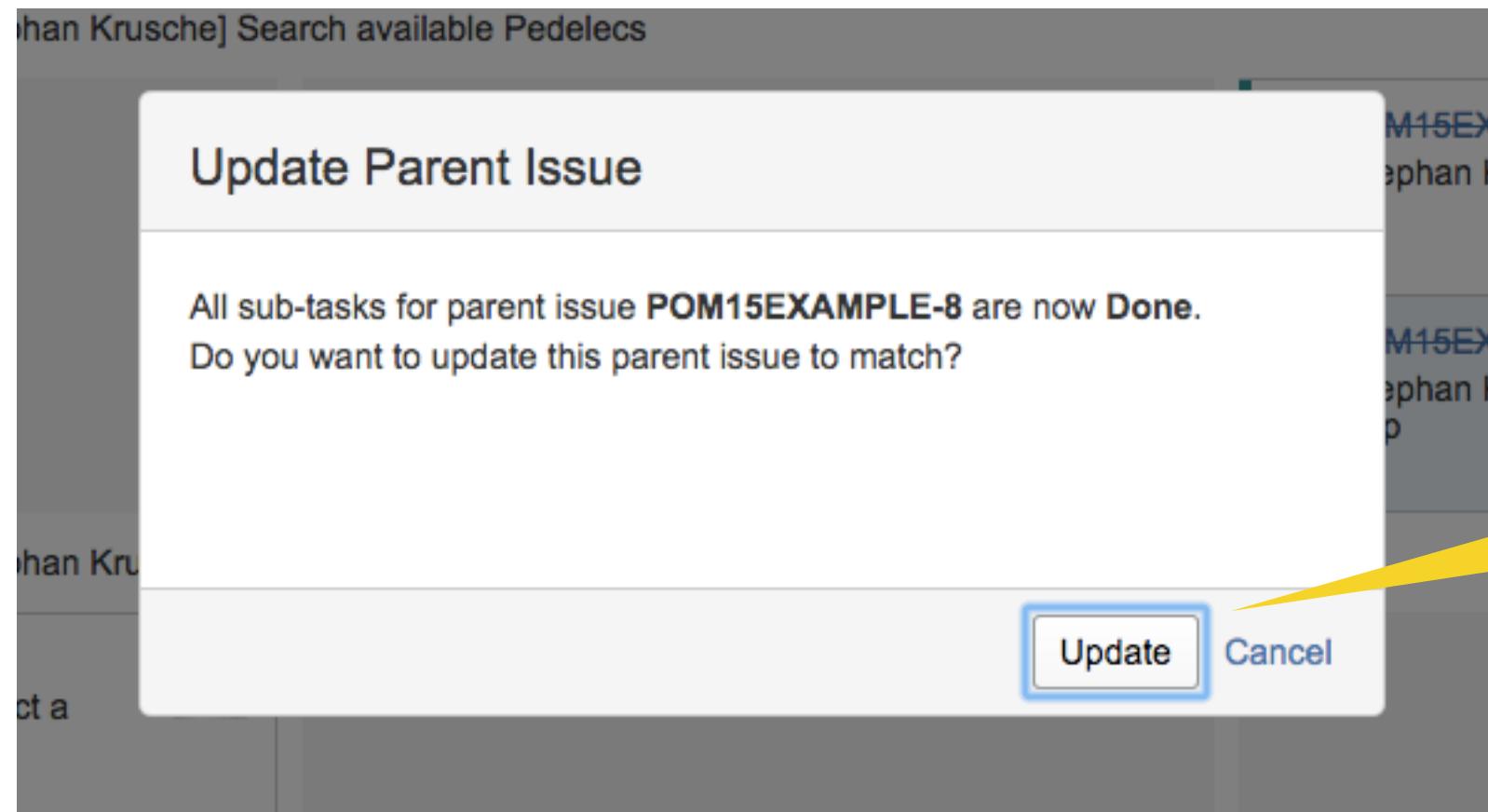
Task 13: Open the Taskboard and start/resolve Sub-Tasks

The screenshot shows a JIRA Agile Taskboard titled "Pom15 Agile Board Stephan Krusche". The board has three columns: "To Do", "In Progress", and "Done". The "To Do" column contains several tasks under two main issues: "POM15EXAMPLE-8" and "POM15EXAMPLE-9". Each task has a red arrow pointing from the "To Do" column to the "Done" column, indicating they have been completed. Yellow callout boxes on the right side provide instructions for moving tasks:

- Move 2 tasks from To Do to Done (for POM15EXAMPLE-8 tasks)
- Move 2 tasks from To Do to Done (for POM15EXAMPLE-9 tasks)
- Move 1 task from To Do to In Progress (for POM15EXAMPLE-15)

Issue	Sub-task	Description	Status
POM15EXAMPLE-8	POM15EXAMPLE-11	[Stephan Krusche] Display Pedelec on Map	To Do
	POM15EXAMPLE-12	[Stephan Krusche] Display user location on Map	To Do
POM15EXAMPLE-9	POM15EXAMPLE-13	[Stephan Krusche] Let the user select a Pedelec	To Do
	POM15EXAMPLE-14	[Stephan Krusche] Show the range as radius on the map	To Do
	POM15EXAMPLE-15	[Stephan Krusche] Show a reservation button	To Do

Task 13: Open the Taskboard and move subtasks



Choose update to also close the User Story

Task 13: Open the Taskboard and move subtasks

The screenshot shows the LS1 JIRA Agile Board interface. The top navigation bar includes LS1 JIRA, Dashboards, Projects, Issues, Agile, Create, Search, Help, Settings, and User Profile. The board title is "Pom15 Agile Board Stephan Krusche". Filter options include SPRINT: Sprint 1, QUICK FILTERS: Only My Issues (selected), and Recently Updated.

The board has three columns: To Do, In Progress, and Done.

- To Do:** Contains a single collapsed item: POM15EXAMPLE-8 [Stephan Krusche] Search available Pedelecs.
- In Progress:** Contains two collapsed items:
 - POM15EXAMPLE-8 [Stephan Krusche] Search available Pedelecs
 - POM15EXAMPLE-9 [Stephan Krusche] Check working radius
 - POM15EXAMPLE-15 [Stephan Krusche] Show a reservation button
 - POM15EXAMPLE-13 [Stephan Krusche] Let the user select a Pedelec
 - POM15EXAMPLE-14 [Stephan Krusche] Show the range as radius on the map
- Done:** Contains two completed items:
 - POM15EXAMPLE-11 [Stephan Krusche] Display Pedelec on Map
 - POM15EXAMPLE-12 [Stephan Krusche] Display user location on Map

A yellow callout bubble labeled "In Progress" points to the In Progress column, and another yellow callout bubble labeled "Done / Resolved" points to the Done column.

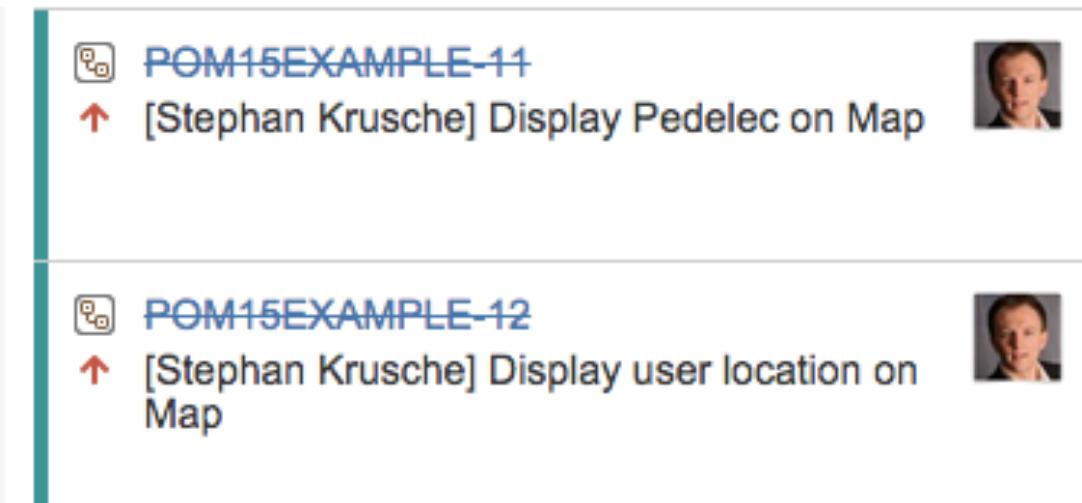
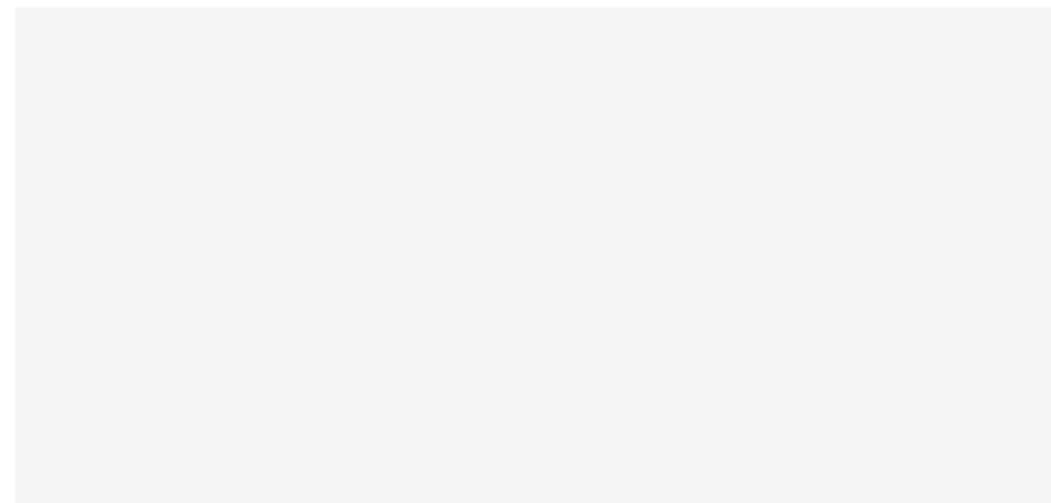
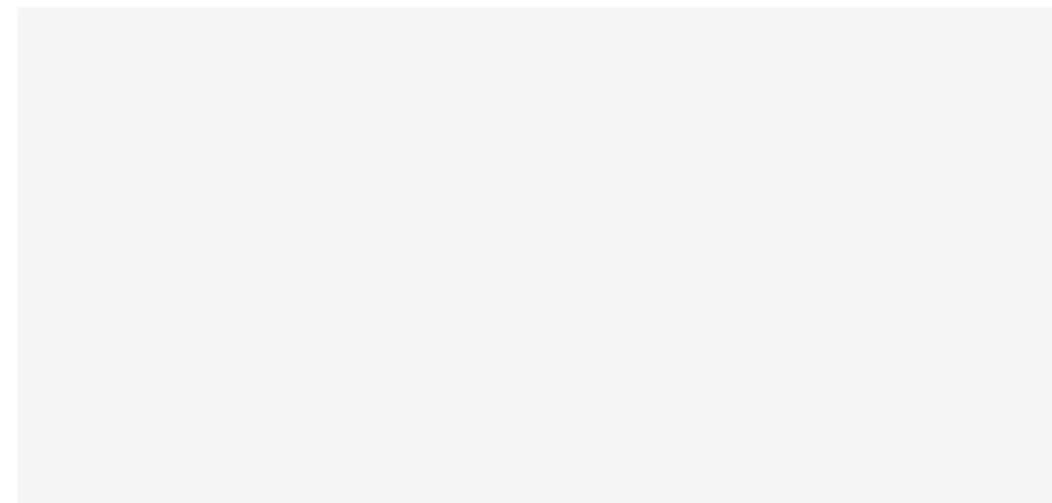
Issue Workflow in JIRA

To Do

In Progress

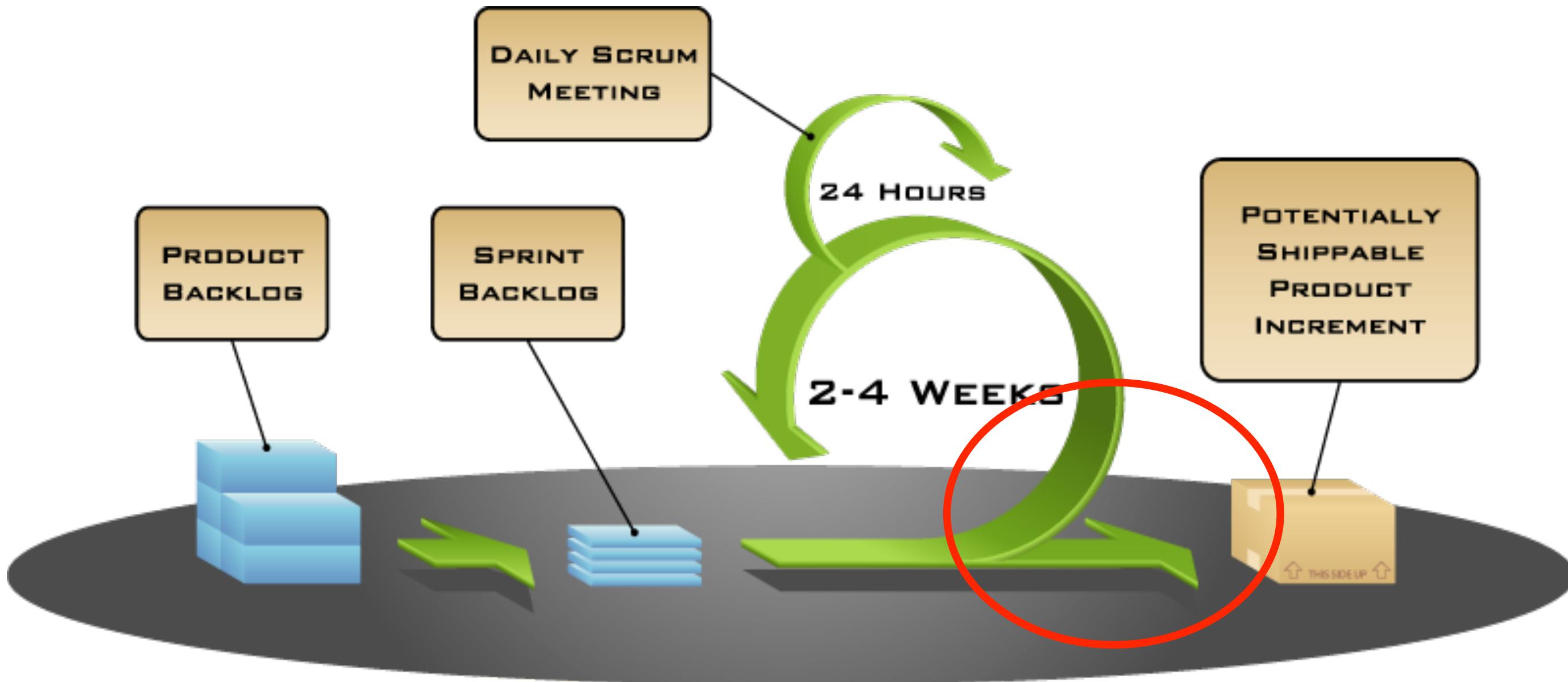
Done

✓  POM15EXAMPLE-8 2 sub-tasks [Stephan Krusche] Search available Pedelecs



- You can also setup JIRA Agile to show an additional column for the state **In Review**

Exercise 6: Sprint Review



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Sprint Review Meeting

- The Development Team delivers a Product Increment including the realized items from the Sprint Backlog
 - Deliver the application before the Review Meeting to the Product Owner
- The Development Team demonstrates the Product Increment to the **Product Owner**
- The Product Owner (or other stakeholders) gives feedback and decides whether the items are realized completely
 - Realized items are ticked off in the Sprint Backlog

Sprint 1 Backlog

ID	Name	Difficulty
1	<input checked="" type="checkbox"/> Search available Pedelecs	Medium
	<input checked="" type="checkbox"/> Display Pedelec on Map	
	<input checked="" type="checkbox"/> Display user location on Map	
2	<input type="checkbox"/> Check working radius	Large
	<input checked="" type="checkbox"/> Let the user select a Pedelec	
	<input checked="" type="checkbox"/> Show the range as radius on the map	
	<input type="checkbox"/> Show a reservation button	

Sprint Review Meeting (continued)

- Unrealized items move back to the Product Backlog
 - These are candidates for the next Sprint
 - Feedback of the Product Owner (or other stakeholders) is stored in JIRA
- The Product Owner can add new requirements to the product backlog or change existing ones
- The Review Meeting can include a Sprint Retrospective (can also be a separate meeting)

Sprint 1 Backlog

ID	Name	Difficulty
1	<input checked="" type="checkbox"/> Search available Pedelecs	Medium
	<input checked="" type="checkbox"/> Display Pedelec on Map	
	<input checked="" type="checkbox"/> Display user location on Map	
2	<input type="checkbox"/> Check working radius	Large
	<input checked="" type="checkbox"/> Let the user select a Pedelec	
	<input checked="" type="checkbox"/> Show the range as radius on the map	
	<input type="checkbox"/> Show a reservation button	

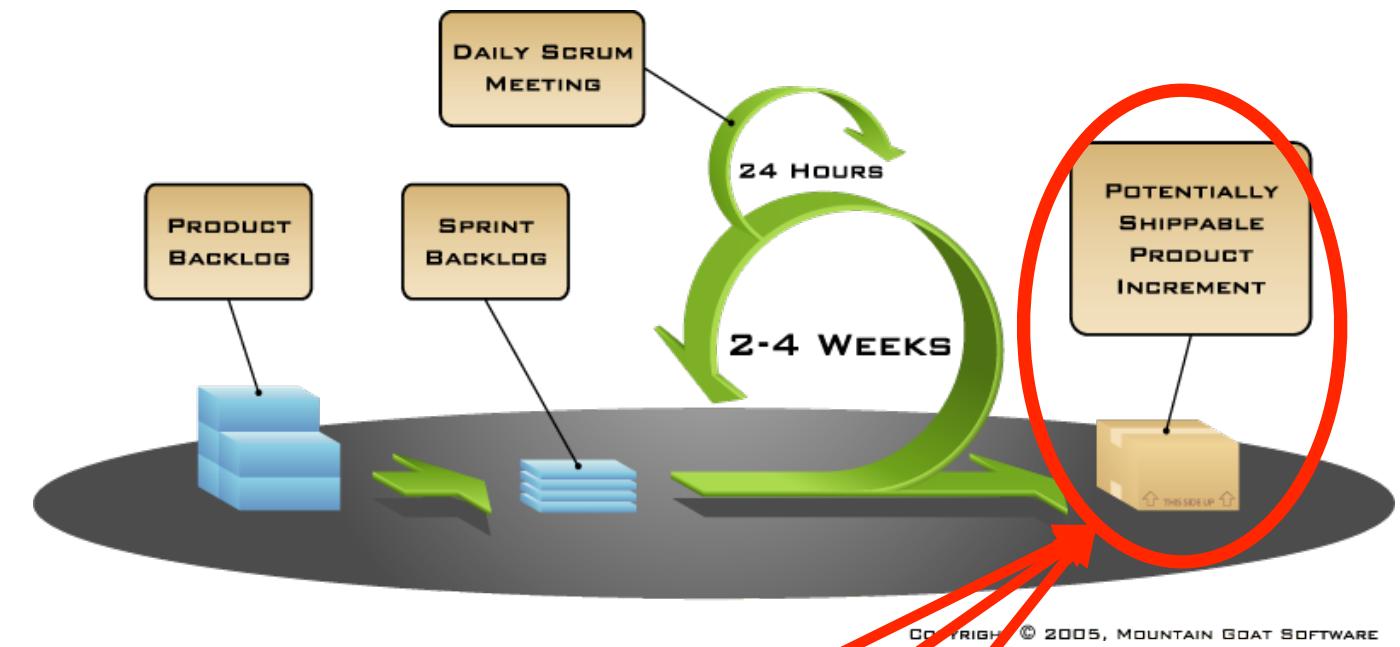


Prioritized and Estimated Product Backlog

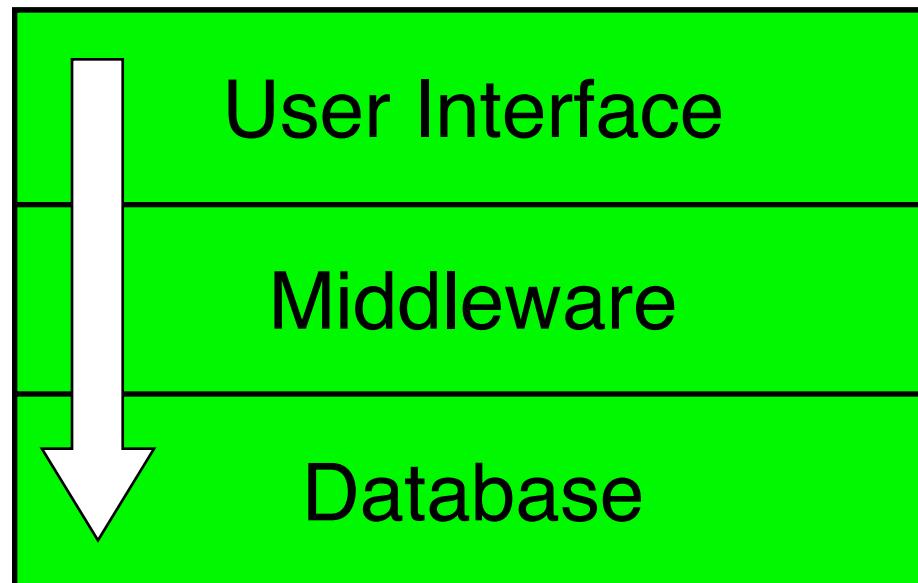
ID	Name	Priority	Difficulty
1	<input checked="" type="checkbox"/> Search available Pedelecs	Critical	Medium
2	<input type="checkbox"/> Check working radius	Critical	Large
3	<input type="checkbox"/> Reserve available Pedelec	Critical	Small
4	<input type="checkbox"/> Return Pedelec	Major	Small
5	<input type="checkbox"/> Contact colleague	Minor	Medium
6	<input type="checkbox"/> Pass reservation	Minor	Medium
7	<input type="checkbox"/> Report damage	Minor	Large
8	<input type="checkbox"/> Unlock Pedelec	Major	Small
9	<input type="checkbox"/> Notification after Repair	Minor	?

Potentially Shippable Product Increment

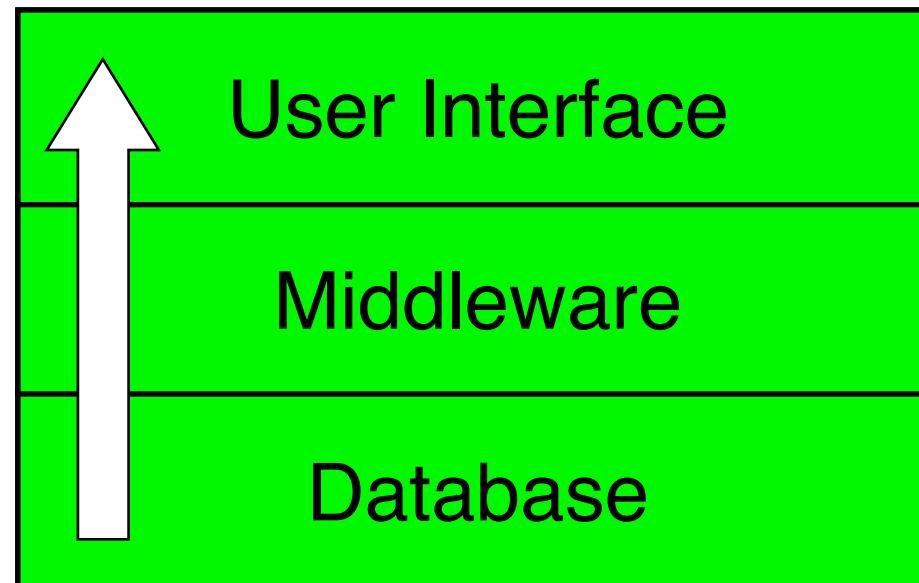
- Each sprint focuses on the incremental creation of an working system
- Can be thrown away (“falsified”) or delivered
- The product owner decides



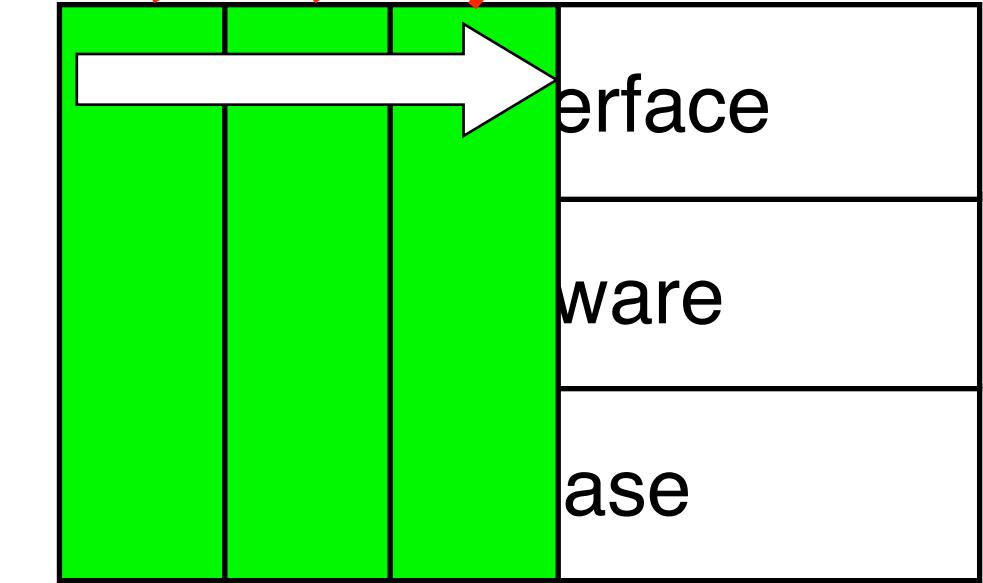
Top Down Integration



Bottom-Up Integration



Horizontal Integration



Vertical Integration



Time:
5 min

Exercise 6: Sprint Review

Task 13: Review Acceptance Criteria

Task 14: Finish the Sprint

Task 15: Inspect the Sprint Report

Task 13: Review Acceptance Criteria

POM 2015 Example Project / POM15EXAMPLE-8

[Stephan Krusche] Search available Pedelecs

[Edit](#) [Comment](#) [Assign](#) [More](#) [Reopen](#)

Details

Type: [User Story](#) Status: [CLOSED](#) (View Workflow)

Priority: [Critical](#)

Difficulty: M

User Story Role: [Employee](#)

User Story Feature: search Pedelecs on a map

User Story Reason: I know where free Pedelecs are located

Acceptance Criteria:

- The user sees all Pedelecs on a map
- The user can see his own location

People

Assignee: Stephan Krusche (POM)

Reporter: Stephan Krusche (POM)

Votes: 0

Watchers: 1 [Stop watching this issue](#)

HipChat discussions

Dedicated room: [Create a room](#) [Choose a room](#)

Other rooms: Issue mentioned in 0 rooms

Dates

Created:

Task 13: Review Acceptance Criteria

POM 2015 Example Project / POM15EXAMPLE-8
[Stephan Krusche] Search available Pedelecs

[Edit](#) [Comment](#) [Assign](#) More [Reopen](#) [Print](#) [Export](#)

Details

Type:	User Story	Status:	CLOSED (View Workflow)
Priority:	Critical		
Difficulty:	M		
User Story Role:	Employee		
User Story Feature:	search Pedelecs on a map		
User Story Reason:	I know where free Pedelecs are located		
Acceptance Criteria:	<input checked="" type="checkbox"/> The user sees all Pedelecs on a map <input checked="" type="checkbox"/> The user can see his own location		

Description

Click to add description

Sub-Tasks

1. <input checked="" type="checkbox"/> [Stephan Krusche] Display Pedelec on Map	View CLOSED Stephan Krusche (POM)
2. <input checked="" type="checkbox"/> [Stephan Krusche] Display user location on Map	View CLOSED Stephan Krusche (POM)

People

Assignee: Stephan Krusche (POM)
Reporter: Stephan Krusche (POM)
Votes: 0
Watchers: 1 Stop watching this issue

HipChat discussions

Dedicated room:
[Create a room](#) [Choose a room](#)

Other rooms:
Issue mentioned in 0 rooms

Dates

Created: Yesterday

Acceptance Criteria (Yellow Callout):
Tick off all acceptance criteria that the Product Owner accepts

Task 14: Finish the Sprint

SPRINT: Sprint 1 ▾

QUICK FILTERS: Only My Issues Recently Updated

To Do

Sprint Name: Sprint 1

Start Date: 21.04.15 23:40

End Date: 5.05.15 23:40

Complete Sprint...

Back in the Agile Board select Sprint 1

Then choose Complete Sprint...

POM15EXAMPLE-11 [Stephan Krusche] Display Pedelec on Map

POM15EXAMPLE-12 [Stephan Krusche] Display user location on Map

POM15EXAMPLE-13 [Stephan Krusche] Let the user select a Pedelec

POM15EXAMPLE-14 [Stephan Krusche] Show the range as radius on the map

POM15EXAMPLE-15 [Stephan Krusche] Show a reservation button

Task 14: Finish the Sprint

The screenshot shows a Jira Agile board for 'Stephan Krusche'. The board has columns for 'In Progress' and 'Done'. A modal dialog box is open, titled 'Complete Sprint'. Inside the dialog, it says 'Sprint 1' and '1 issue was Done.' Below that, it says '1 partially complete issue will be returned to the top of the backlog.' A note at the bottom states: 'Sub-tasks are not included in the total(s) above, and are always included in the same sprint as their parent issue.' At the bottom right of the dialog are 'Complete' and 'Cancel' buttons, with 'Complete' being highlighted by a red box. A yellow callout bubble points from the text 'Partially complete issues are moved back to the Product Backlog' to the 'Complete' button.

5 Agile Board Stephan Krusche

Sprint 1 ▾ QUICK FILTERS: Only My Issues Recently Updated

In Progress Done

M15EXAMPLE-8 2 sub-tasks [Stephan Krusche] Search available Pedelecs

Complete Sprint

Sprint 1

1 issue was Done.

1 partially complete issue will be returned to the top of the backlog.

Sub-tasks are not included in the total(s) above, and are always included in the same sprint as their parent issue.

Partially complete issues are moved back to the Product Backlog

Complete Cancel

Task 14: Inspect the Sprint Report

LS1 JIRA Dashboards Projects Issues Agile Create Search ? Reports

Pom15 Agile Board Stephan Krusche Backlog Active sprints Reports Board

REPORTS

- Burndown Chart
- Sprint Report**
- Epic Report
- Epic Burndown
- Version Report
- Release Burndown
- Velocity Chart
- Control Chart
- Cumulative Flow Diagram

② **How to read this chart**
Understand the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is excessive scope creep.
[Hide this information](#)

Sprint Report Sprint 1 ▾
Closed Sprint, ended by Stephan Krusche (POM) 21.04.15 23:40 - 22.04.15 08:37 [Linked pages](#) [View Sprint 1 in Issue Navigator](#)

In the Sprint Report we could see a Burndown Chart. As we did not estimate and completed the Sprint in 30 min, we cannot see much here

Inspect the different reports on the left

JIRA automatically switches to Reports

Retrospective time!
Create a retrospective page to review & improve your iterations!

Create Close

Sprint Retrospective Meeting

- Scrum Master and Development Team meet to discuss how the previous Sprint worked out
- The Product Owner can also participate if necessary
- There are different Retrospective techniques
- Often it is most effective to brainstorm about things that worked well and things that did not work well
- Each team member is asked to identify specific things that the team should:
 - Start doing
 - Stop doing
 - Continue doing

Scrum as Methodology

- Involvement of the customer
 - Onsite customer (“co-located”)
- Planning
 - Checklists and incremental daily plans (“Daily Scrum”)
- Reuse
 - Checklists from previous projects
- Modeling
 - Models may or may not be used
- Process
 - Iterative, incremental and adaptive process
- Control and Monitoring
 - Risk management distributed across daily meetings.

Summary

- **Scrum**
 - **3 Artifacts:** Product Backlog, Sprint Backlog, Potentially Shippable Product Increment
 - **4 Meetings:** Project Kickoff, Sprint Planning, Daily Scrum, Sprint Review
 - **3 Roles:** Scrum Master, Product Owner, Developer
- **JIRA Agile**
 - Create and manage the Product Backlog and its backlog items (e.g. User Stories / Scenarios)
 - Plan the Sprint by choosing backlog items and by adding Sub-Tasks
 - Track the status of a Sprint using a Taskboard and the progress of a Sprint with a Burn Down Chart
 - Prioritize and estimate backlog items —> more about estimation in a later exercise
 - Review the Sprint using acceptance criteria

References

- Stephan Krusche, Lukas Alperowitz, Bernd Bruegge and Martin Wagner - [Rugby: An Agile Process Model Based on Continuous Delivery](#), 1st International Workshop on Rapid Continuous Software Engineering (RCoSE'14), June 2014
- Stephan Krusche and Lukas Alperowitz - [Introduction of Continuous Delivery in Multi-Customer Project Courses](#), 36th International Conference on Software Engineering (ICSE'14), Software Engineering Education and Training (SEET), June 2014
- Ken Schwaber and Mike Beedle - Agile Software Development with Scrum, Prentice Hall, October 2001
- Mike Cohn - User Stories Applied - For Agile Software Development, Addison-Wesley, March 2004
- Mike Cohn - Agile Estimating and Planning, Prentice Hall, November 2005
- Bernd Bruegge and Allen H. Dutoit - Object Oriented Software Engineering using UML, Design Patterns and Java, 2010
- Agiles Manifesto - <http://agilemanifesto.org>
- Mike Cohn, Mountain Goat Software - <http://www.mountaingoatsoftware.com>
- Agile42, Scrum Cheat Sheet - http://media.agile42.com/content/Scrum_Cheat_Sheet_09_09.pdf
- Bill Wake - INVEST in Good Stories, and SMART Tasks - <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks>
- Meet the new gold standard in workflow automation - <http://blogs.atlassian.com/2014/08/meet-new-gold-standard-workflow-automation>
- 8 Steps to a Definition of Done in JIRA - <http://blogs.atlassian.com/2013/10/8-steps-to-a-definition-of-done-in-jira>
- Sprint Retrospective - <http://www.mountaingoatsoftware.com/agile/scrum/sprint-retrospective>