



Elina Kurkimäki

## **PROJECT MANAGEMENT PROCESS REFLECTED THROUGH ASSISTIVE TOOLS**

# **PROJECT MANAGEMENT PROCESS REFLECTED THROUGH ASSISTIVE TOOLS**

Elina Kurkimäki

Bachelor's thesis

May 2011

Degree Programme in Information Technology  
and Telecommunications

Oulu University of Applied Sciences

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Information Technology, Software development

---

Author(s): Elina Kurkimäki

Title of Bachelor's thesis: Project Management Process Reflected through Assistive Tools

Supervisor(s): Markku Rahikainen

Term and year of completion: Spring 2011

Number of pages: 58

---

This thesis presents a project management process of a target company with outlines the correct usage of the company's project tracking tool, Atlassian Jira. The target company uses Scrum, an agile project management framework, for its projects and combines it with some practices from Kanban. The process is documented in a business process approach using workflow charts to visualize the processes. The documented process is established as a part of the organization's quality management system that is created based on the ISO 9001 quality management system requirements. The implementation of the process description was done alongside with an empirical validation of the model in a software project. The documented project management process description includes four phases within a Scrum project: planning a project, planning a Sprint, executing a Sprint and ending a Sprint. All of the phases are described in a check list form that lines out what is to be done before that phase, who is responsible for the phase, what the procedure is and who should be present in the meeting or meetings in that phase and what should be the outcome of the meeting in question. Planning a project consists of an effort estimation and a project phasing with guides to the project and Backlog creation to Jira. Planning a Sprint describes the two Sprint planning meetings to be held before starting a Sprint and setting up the tasks in Jira as a Sprint Backlog. Carrying out a Sprint describes the daily duties the Scrum-Team has: updating the Kanban task board and having Daily Scrum meetings. Ending a Sprint contains two Scrum meetings, Sprint Review and Sprint Retrospective. The process description ends with a list of Scrum artifacts that can be used as a reference when reading the documentation. The process description in this thesis does not create a new way for project management in the target company but documents the way activities are managed for the quality management system documentation. In the process of documentation, some additions were made for unifying the project management practices within the organization.

---

Keywords: Scrum, Kanban, Project Management, Agile, Jira, Process description

## TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikka, ohjelmistojen kehitys

---

Tekijä(t): Elina Kurkimäki

Opinnäytetyön nimi: Project Management Process Reflected through Assistive Tools

Työn ohjaaja(t): Markku Rahikainen

Työn valmistuslukukausi ja -vuosi: Kevät 2011

Sivumäärä: 58

---

Työ esittelee projektinhallintaprosessin kohdeyrityksessä ohjeistaen samalla oikeaoppiseen Atlassian Jira -projektinseurantatyökalun käyttöön. Kohdeyritys käyttää projekteissaan ketterää ohjelmistokehitysmenetelmää Scrumia, johon on lisätty käytäntöjä myös Kanbanista. Prosessi on dokumentoitu käyttäen prosessikuvausmenetelmänä työnkulkukaavioita. Kyseinen prosessi luotiin osaksi kohdeorganisaation laadunhallintajärjestelmää joka rakentuu ISO 9001 -laatuvaatimusten pohjalta. Prosessikuvauksen toteuttamisen ohessa tehtiin empiirinen vahvistus mallin toimivuudesta ohjelmistoprojektin muodossa. Dokumentoitu prosessikuvaus sisältää neljä Scrum-projektin vaihetta: projektin suunnittelu, Sprintin suunnittelu, Sprintin suorittaminen ja Sprintin päättäminen. Kaikki vaiheet ovat kuvattu muistilistamuodossa, jotka erittelevät mitä tulee tehdä ennen kyseistä vaihetta, kuka on vaiheesta vastuussa, mikä on vaiheen kokouksen tai kokousten etenemismalli ja keiden tulee olla läsnä sekä mitä lopputulemia vaiheesta odotetaan. Projektin suunnitteluvaihe sisältää työmääräarvion ja projektin vaiheistuksen kuvaukset sekä ohjeet projektin ja Product Backlogin luomiseksi Jirassa. Sprintin suunnittelu kuvaa kaksi vaadittavaa Sprintin suunnittelukokousta ja Sprint Backlogin luomisen Jiraan. Sprinting suorittamisvaiheessa kuvataan Scrum-ryhmän päivittäiset työtehtävät: Kanban tehtävätaulun päivittäminen ja Daily Scrum -kokousten pitäminen. Sprintin päättämisvaiheessa on kaksi kokousta: Sprint Review ja Sprint Retrospective. Prosessikuvaus päättyy listaan Scrum-tuotteista, joita dokumentaatioissa esiintyy. Työn tuloksena syntynyt prosessikuvaus ei luo uutta projektinhallinta menetelmää kohdeyritykselle, vaan dokumentoi organisaation nykyisen toimintatavan. Dokumentointiprosessin aikana joitain lisäyksiä projektinhallintaohjeistukseen kuitenkin tehtiin projektinhallintakäytäntöjen yhtenäistämiseksi.

---

Asiasanat: Scrum, Kanban, Projektinhallinta, Ketterä, Jira, Prosessikuvaus

## **PREFACE**

I want to thank my employer, Codemate Ltd. for this opportunity to study such an interesting subject for my thesis. I also want to thank my colleagues from school and work for peer support during this process. It was big help to have friends work on their studies over the same period of time as I was.

A special thank you goes to my family for understanding my absence and giving me the space and time I needed to have this work completed.

I am surprised how fast time has passed, it seems as though I just started this school and here I am writing my last words to my thesis. It feels great.

Oulu 16 May 2011

Elina Kurkimäki

## **SYMBOLS AND ABBREVIATIONS**

ISO	International Standard Organization
WIP	Work in progress
PDF	Portable Document Format
HTML	Hypertext Markup Language
CRM	Customer relations management
CMS	Content management system

# TABLE OF CONTENTS

1 INTRODUCTION .....	9
2 AGILE SOFTWARE DEVELOPMENT .....	10
2.1 Overview .....	10
2.2 Scrum .....	11
2.2.1 Scrum-Team .....	11
2.2.2. Scrum Roles .....	12
2.2.3. Iterations .....	12
2.2.4 Activities.....	13
2.3 Kanban .....	13
3 PROCESSES.....	15
3.1 Business process .....	15
3.2 Symbols and roles in workflow charts .....	15
3.2.1 Symbols.....	15
3.2.2 Roles.....	16
3.3 ISO 9001 .....	17
4 IMPLEMENTATION .....	18
4.1 Information Gathering.....	18
4.2 Empirical Validation .....	18
4.3 Tools.....	18
4.3.1 Atlassian Jira .....	19
4.3.2 reStructuredText .....	19
5 PROJECT MANAGEMENT PROCESS .....	21
5.1 How to use this document.....	21
5.2 General.....	21
5.3 Before starting a project .....	24
5.3.1 Define project members.....	24
5.3.2 Product Backlog.....	24
5.3.3 Estimate the effort.....	25
5.3.4 Set up a project in Jira .....	27
5.4 Planning a Sprint.....	30

5.4.1 Sprint planning meeting #1 .....	30
5.4.3 Sprint planning meeting #2 .....	33
5.4.4 Setting up the Sprint Backlog in Jira .....	35
5.5 Executing a Sprint .....	38
5.5.1 Basics .....	40
5.5.2 Updating task board in Jira .....	40
5.5.3 Daily Scrum .....	44
5.6 Ending a Sprint.....	45
5.6.1 Sprint review meeting .....	46
5.6.2 Sprint retrospective meeting .....	48
5.6.3 Making a release in Jira .....	49
5.7 Artefacts .....	51
5.7.1 Impediment Backlog .....	51
5.7.2 Product Backlog.....	51
5.7.3 Selected Product Backlog.....	51
5.7.4 Potentially Shippable Product Increment .....	52
5.7.5 Sprint Backlog.....	52
5.7.6 Burndown Chart.....	52
6 CONCLUSIONS AND FUTURE WORK .....	55
7 REFERENCES .....	56



# 1 INTRODUCTION

The purpose of this thesis was to document a project management process to a target company. This project management process describes the different steps to be taken within a software project, excluding project startup and closure procedures with a customer. This process was created to list the responsibilities and out coming artifacts of different phases. This information is mostly portrayed in a check list form for readers to easily find the aspects they need to take into consideration. Outlining the correct usage of the project management tools was added to this particular process as well. The process is put together combining influences from the agile development methods Scrum and Kanban.

The target company is Codemate Ltd, a relatively small international software company that focuses on providing web-based back end solutions. Codemate's headquarters are located in Oulu, Finland. Other offices in Finland are situated in Vuokatti and Helsinki, and a greater off-shore unit is located in Dhaka, Bangladesh. Codemate has had a steady growth in the personnel and project count since starting its business in 2005, and the growth is yet continuing. At the moment, Codemate employs approximately 30 people in Finland and 70 internationally. Due to the growth of the company, there is a need for an overall framework for the project management process. The framework is created to unify the project management manners within Codemate projects. (1.)

The process is established to be a part of the company's quality management system that is being put together for applying an ISO (the International Organization for Standardization) 9001 certification.

The implementation of the framework was done alongside with an empirical study. A small project was managed by the guidelines of the framework and it is used as an example in the process.

## 2 AGILE SOFTWARE DEVELOPMENT

This chapter briefly introduces the agile software development. The target company has used Scrum in its projects before, but with this process description, Kanban is introduced as an additional tool for project management.

### 2.1 Overview

Agile software development refers to a group of software development methodologies that are based on iterative and incremental development. Requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. (2.) Agile methods are attempting to offer an answer to the business community asking for a lighter weight along with faster software development processes (3).

Agile software development has had a considerable impact on how software is developed worldwide since the Agile Manifesto was released in 2001 at a summit of seventeen independent-minded practitioners of several programming methodologies (4, 2, 5).

#### **The 12 basic principles behind the Agile Manifesto:**

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness a change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. (6.)

## **2.2 Scrum**

Scrum is an agile framework, rather than a methodology, for completing complex projects (7). Instead of providing complete descriptions of how everything should be done on the project, much is left to the Scrum-Team to decide. This is because the Scrum-Team knows best how to solve the problem they are presented. (8. What is Scrum?)

### **2.2.1 Scrum-Team**

The core of Scrum comes down to self-organizing, cross-functional teams. Self-organizing means that there is no overall team leader who assigns tasks or decides on resolutions. Those are questions that are decided by the team as a whole. Cross-functionality involves everyone necessary in implementing a feature. (8. What is Scrum?)

## 2.2.2. Scrum Roles

Scrum development is supported by two specific individuals: a Scrum Master and a Product Owner. The Scrum Master's job is to help team members use the Scrum framework to perform at their highest level. The Product Owner's role is to guide the team toward building the right product by representing the business, customers or users. (8. What is Scrum?)

## 2.2.3. Iterations

Scrum projects make progress in a series of sprints, which are time boxed iterations that last 2-4 weeks. In the beginning of a sprint, Team members commit to delivering a number of features that were listed on the project's product backlog. At the end of the sprint, these features are done and a sprint review is conducted. During the review the team demonstrates the new functionality to the Product Owner and other interested stakeholders who provide feedback that could influence the next sprint. During a sprint, the Scrum-Team holds a Daily Scrum to keep the whole team updated on the progress of the sprint. (8. What is Scrum?) The iteration flow with preparations, artifacts and members are portrayed in figure 1.

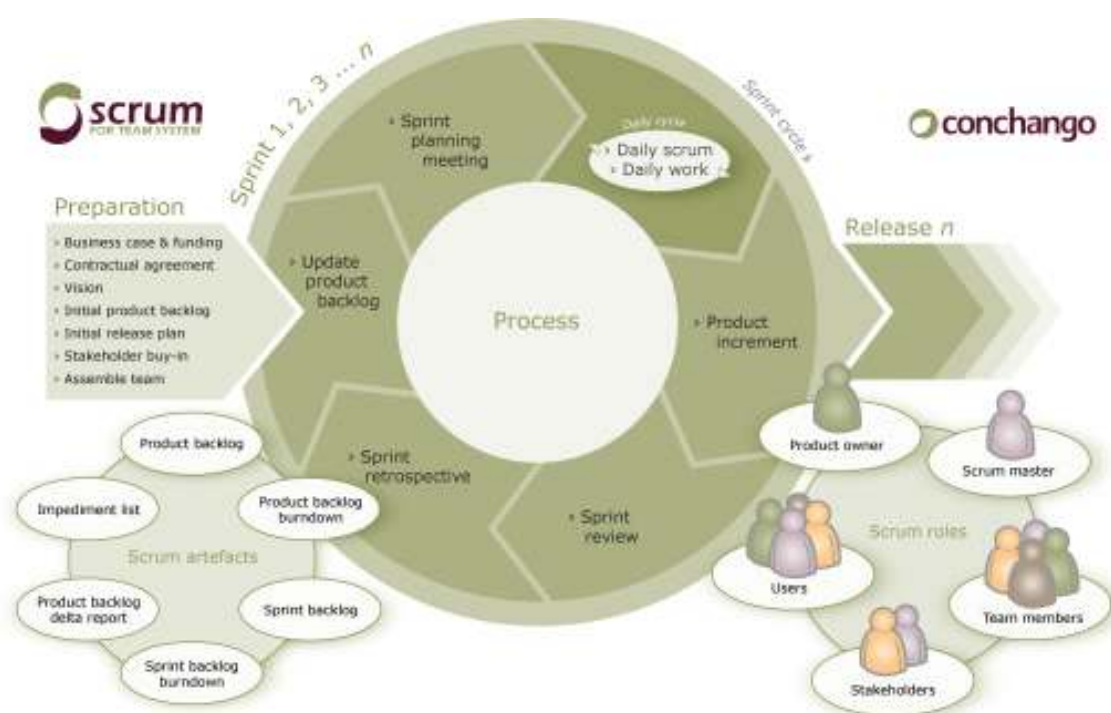


FIGURE 1. Scrum

## 2.2.4 Activities

The product itself is the primary artifact of a Scrum project. The team is expected to produce a potentially shippable version of the product at the end of each sprint. (8. What are the main activities in Scrum?)

The Product Backlog is a list functionality that is yet to be added to the product. It is prioritized by the Product Owner in the ways that the team always works on the most valuable features first. A successful way to create a product backlog is to populate it with user stories, short descriptions of functionality described from the perspective of a user or customer. (8. What are the main activities in Scrum?)

The Scrum-Team members create the sprint backlog on the first day of a sprint and during the sprint planning meeting. Where the Product Backlog is a list of features to be built, the Sprint Backlog is the list of tasks the team has to perform in order to deliver the functionality they committed to deliver during the sprint. (8. What are the main activities in Scrum?)

Sprint Burndown Chart and release Burndown Chart are other primary artifacts. They visualize the remaining amount of work. Burndown Charts are an effective tool for quickly determining if a sprint or release is on. (8. What are the main activities in Scrum?)

## 2.3 Kanban

Kanban is a lean approach to agile software development. Along with Scrum, Kanban can be part of the process toolkit of any agile company. (9.)

The core of Kanban can be summarized in the following three main ideas:

- **Visualizing the workflow**
  - Splitting work into pieces, writing each item on a card and posting them on a wall
  - Using named columns to illustrate where each item is in the workflow. This is portrayed in figure 2.

- **Limit Work In Progress (WIP)**
  - Assign limits to how many items may be in progress at each workflow state
- **Measuring the lead time**
  - Lead time is an average time to complete one item
  - Optimizing the process to make the lead time as small and predictable as possible. (9, 10.)

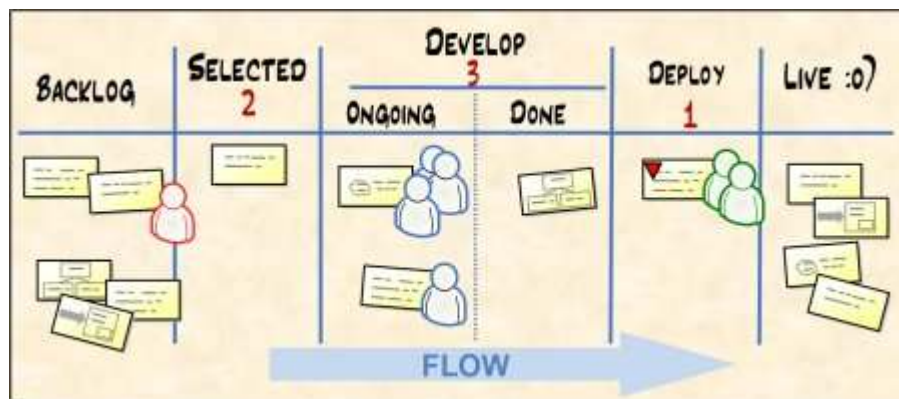


FIGURE 2. Kanban workflow

When applying Kanban, bottlenecks in the project become visible. This makes it easier for people to collaborate and that leads to optimizing the whole value chain. (9.)

## **3 PROCESSES**

Process descriptions of are the main focus of this thesis. This chapter explains the concept of business processes and the meanings of the symbols used in process workflow charts.

### **3.1 Business process**

A business process is a collection of activities or tasks that serve a particular goal. Business processes can be visualized with a flowchart as a sequence of activities where they begin with mission objectives and end with achievements of the business objective. The outcome of a well designed business process is an increase in efficiency. In other words, processes are designed to add value for the customer. (11.)

A business process can be divided into several sub-processes, which contribute to achieving the goal of the super-process at the same time having their own attributes. This is the case with practically all the project step processes described in this thesis, as they all are sub-processes for the project management process. (11.)

Business processes can be modeled in many ways, for example as a business processes workflow (11). This is the technique used in this particular process description.






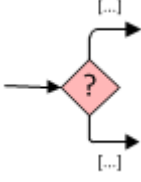
### **3.2 Symbols and roles in workflow charts**

This chapter describes the different kinds of symbols, actions and roles in process flowcharts. The flowcharts are used to visualize the project management process.

#### **3.2.1 Symbols**

The different symbols used in the flowcharts of this process are portrayed in table 1.

TABLE 1. The symbols in process workflow charts

	<p><b>Starting Point</b></p>
	<p><b>Norman Completion</b></p>
	<p><b>Process</b> Used to show process that contains sub-processes and / or activities to achieve its goal</p>
	<p><b>Activity</b> Shows discrete logical processes within a process. An activity cannot be further broken down.</p>
	<p><b>Input / Output Data</b> The information the process expects to receive and deliver</p>
	<p><b>Decision</b> Processing will split based on a decision. The given condition met will be shown on the outgoing flow line. Decisions yield two results only.</p>

### 3.2.2 Roles

Process swimlines, introduced in figure 3, are titled by roles. A line titled after a role means that the role is responsible for the actions on the line. Roles used in this software project management process are:

- Product Owner
- Scrum Master
- The Scrum-Team





*FIGURE 3. A process workflow swimlane*

### **3.3 ISO 9001**

The project management process was written to be a part of the target company's quality management system that seeks the ISO 9001, version 2008, certification. ISO is a worldwide federation of national standards bodies.

The International Standard promotes the adaptation of a process approach when developing, implementing and improving the effectiveness of a quality management system. The application of a system of processes within an organization, together with the identification and interactions of these processes, and their management to produce the desired outcome, can be referred as the process approach. (12.)

## **4 IMPLEMENTATION**

This chapter describes the creation and implementation of the project management process. The information gathering was done by using an example project as an empirical validation of the project management process.

### **4.1 Information Gathering**

The information gathering on the target company's project management processes was mostly done by interviewing the project management personnel in the company. Even other published Scrum checklists were studied for reference.

### **4.2 Empirical Validation**

The project management process was created alongside with an empirical validation of the process. A small software project was carried out in the target company and it was managed by following the process framework. The examples of Jira usage in the process documentation are from the project in question.

Two trainees were assigned to find ways to integrate Sugar CRM with Liferay CMS in as many ways as possible. The most important goal was to be able to view users added to Sugar from the Liferay environment. The project timeframe was 200 working hours for both trainees and it was split into four Sprints during a period of 10 weeks. The project finished in time and was able to meet the goals that had been set for it.

### **4.3 Tools**

The tools used in the creation of this thesis are described in this chapter. The issue tracking software of the target company is Atlassian Jira. The documentation for the target company has been written in reStructuredText format.

### 4.3.1 Atlassian Jira

Lining out the proper usage of the project management tools was a key point of this thesis. The target company's issue tracking tool of choice is Atlassian Jira, a project management system that can be accessed from a web browser.

A project is created into Jira to be a collection of issues. Issues can be tasks, user stories, test cases along with other labels. All issues in Jira have a workflow, which means that issues can have different statuses during their lifecycles, i.e. *open*, *in progress* and *resolved*. Issue information also includes issue priority, resolution of issue and logged work. (13.)

Jira offers plug-ins and add-ons for customizing the environment to meet the needs of the organization. The target company has taken Jira's agile tools into use to support their development methods.

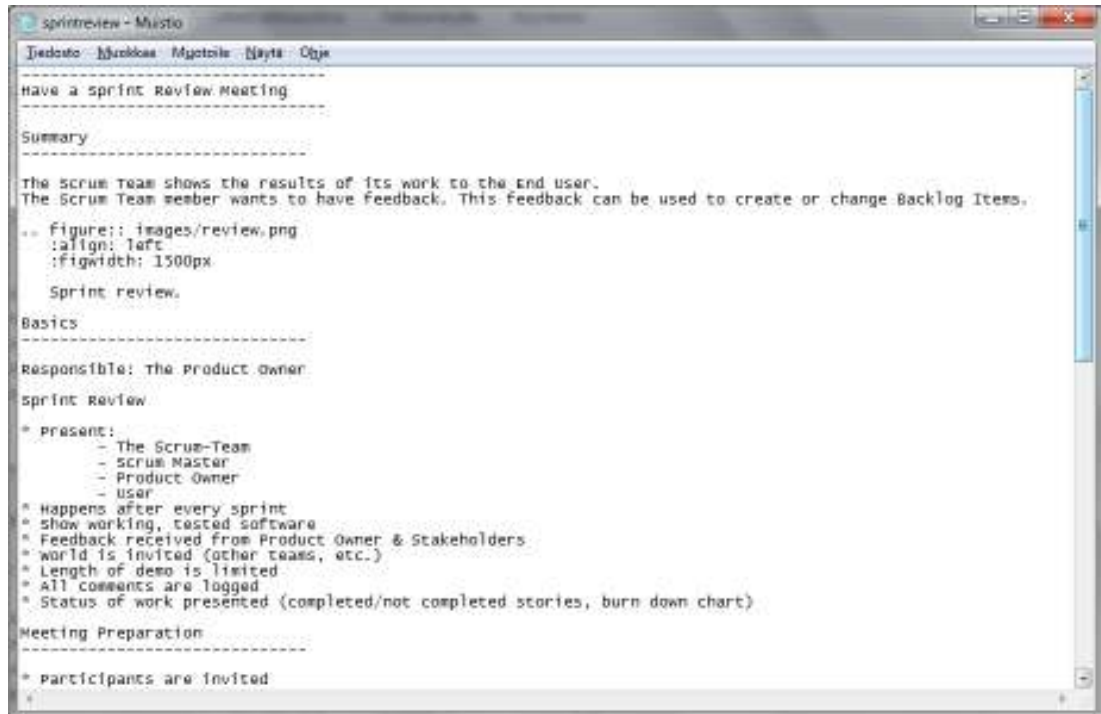
### 4.3.2 reStructuredText

The documentation for the target company has been written in reStructuredText -markup syntax for document usability. The target company has a documentation framework, where committing a reStructuredText document to the framework's version control outputs other formats of the document, such as html and pdf. This enables an easy way of keeping all documents up-to-date and easily modifiable. Writing restructuredText allows the author to focus on the content rather than the document layout.

reStructuredText is a plaintext markup syntax, as figure 4 shows. The primary goal of reStructuredText is to define a markup syntax to be used in documentation domains such as Python docstrings. It was created to be readable and simple, yet powerful enough for a non-trivial use.

reStructuredText is a revision and reinterpretation of the StructuredText and Setext lightweight markup systems. The conversion of reStructuredText

documents into useful structured data formats is the intended purpose of the markup. (14.)



```
sprintreview - Misto
-----
Have a sprint Review Meeting
-----
Summary
-----
The Scrum Team shows the results of its work to the end user.
The Scrum Team member wants to have feedback. This feedback can be used to create or change Backlog Items.
.. figure:: images/review.png
   :align: left
   :figwidth: 1500px
   Sprint review.
-----
Basics
-----
Responsible: The Product Owner
sprint Review
* Present:
  - The Scrum-Team
  - Scrum Master
  - Product Owner
  - user
* Happens after every sprint
* show working, tested software
* Feedback received from Product Owner & Stakeholders
* world is invited (other teams, etc.)
* Length of demo is limited
* All comments are logged
* Status of work presented (completed/not completed stories, burn down chart)
-----
Meeting Preparation
-----
* Participants are invited
```

FIGURE 4. Writing reStructuredText

## **5 PROJECT MANAGEMENT PROCESS**

This chapter explains the process for managing Codemate software projects and works as a guide for the proper usage of Jira issue tracking tool. The described process focuses on managing an existing project, excluding the project startup and closing procedures with the customer.

### **5.1 How to use this document**

The chapters in this document are in a chronological order for project management. Each chapter contains descriptive process charts and a few notes and guidelines. This information is mostly portrayed in a check list form for readers to easily find the aspects they need to take into consideration. Instructions on how to manage a project with Jira issue tracking tool follows. The documentation starts with some basic guidelines and rules, continues through the different steps of a project and ends with a summary of different artifacts involved in projects.

### **5.2 General**

Codemate's projects are completed using the methods of Scrum and Kanban. The actions and processes are described with workflow charts, as can be seen in figure 5 for example. This documentation focuses on four different phases of the project management: planning the project, planning a sprint, executing a sprint and ending a sprint. See figure 6.

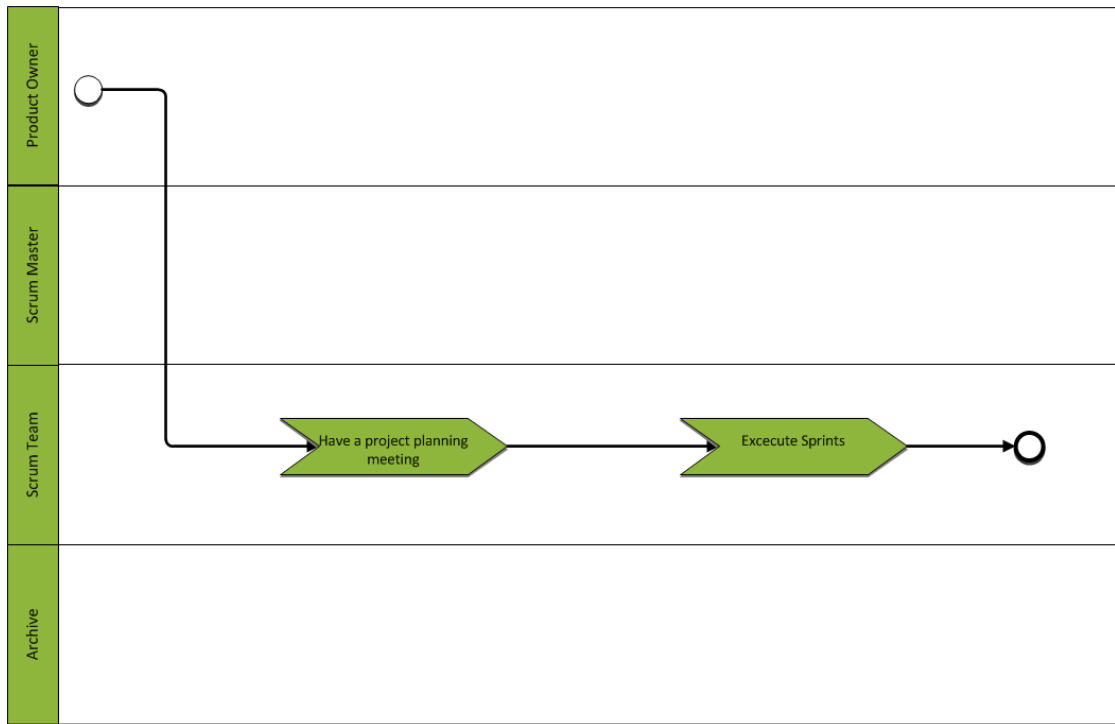


FIGURE 5. Project overview

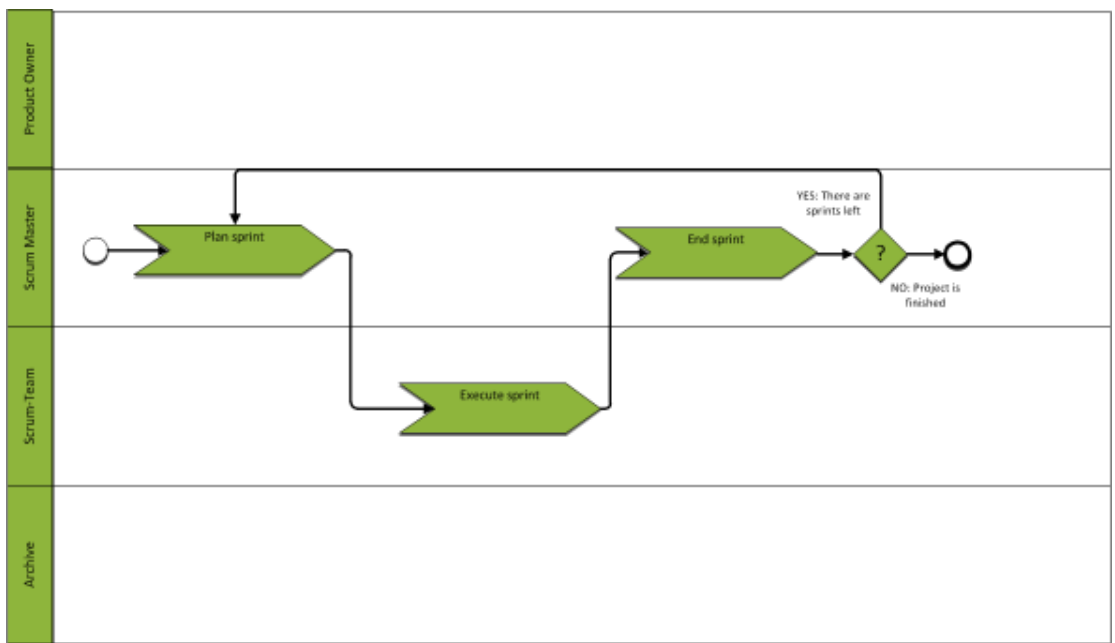


FIGURE 6. Overview of executing Sprints.

Below is a list of some basic practices and rules that should be followed. They line out the core of Scrum.

## 10 Basic Practices:

- Have a clear vision.
- Keep Product Backlog maintained
- Sort Product Backlog based on business value
- The Scrum-Team evaluates the size of Backlog Items
- Have Daily Scrums
- Follow Burndown Charts
- The Sprint should not be disturbed by management and/or customers
- The software the Scrum-Team delivers meets the Definition of Done
- Have a collaborate Sprint Review
- Have Sprint Retrospective meetings.

## Basics meeting rules:

- Prepare every meeting. See figure 7 for the steps to take.
- Start a meeting with presenting the meeting goal and agenda (figure 8).
- Every meeting starts on time and ends on time.
- Every meeting is an open meeting.
- Every meeting is time-boxed



*FIGURE 7. Meeting preparation, Product Owner as the responsible*



*FIGURE 8. Starting a meeting*

## **5.3 Before starting a project**

This chapter points out a few aspects to be taken care of before starting a project, such as defining project members, stocking up the Product Backlog, the process of effort estimation and creating a project in Jira.

### **5.3.1 Define project members**

This chapter explains the different members to be included in the Scrum-Team. There are some guidelines to be followed for the team size and the distribution of roles.

The project members consist of at least three parties:

- Product Owner
  - In this document, referring to Product Owner means either Codemate's internal customer representative or an external representative from the customer.
- Scrum Master
- The Scrum-Team

Have a clearly defined Product Owner, who:

- is empowered to prioritize
- has the knowledge to prioritize
- has a direct contact to the Scrum-Team

The Scrum-Team

- has no more than seven members at the most
- is located in the same workspace

### **5.3.2 Product Backlog**

There has to be a Product Backlog to start working on a Scrum project.

The Product Owner stocks up the initial Product Backlog mostly on his/her own, possibly consulting with the specialists available. After the initial



gathering, the product backlog is reformed with the Scrum-Team's estimations.

Responsible: Product Owner

Product Backlog requirements are as follows:

- Top items are prioritized by business value
- Top items are estimated
- Estimation is done by the Scrum-Team
- Top items small enough to fit in a sprint
- Product Owner has to understand the purpose of each item in the product backlog

Use User Stories to describe the requirements for the software. User stories describe functional requirements from the End User's point of view. See below for a template for a User Story:

As <a specific user> I want <something> so I can achieve <specific value>

### **5.3.3 Estimate the effort**

For strategic planning one needs to know the size of the Backlog Items.

It is a necessary input for planning a release and gaining an understanding of how much the Scrum-Team is able to deliver in a Sprint. It enables the Scrum-Team members to see what will happen in the future phases of the project. The process of effort estimation is portrayed in figure 9.

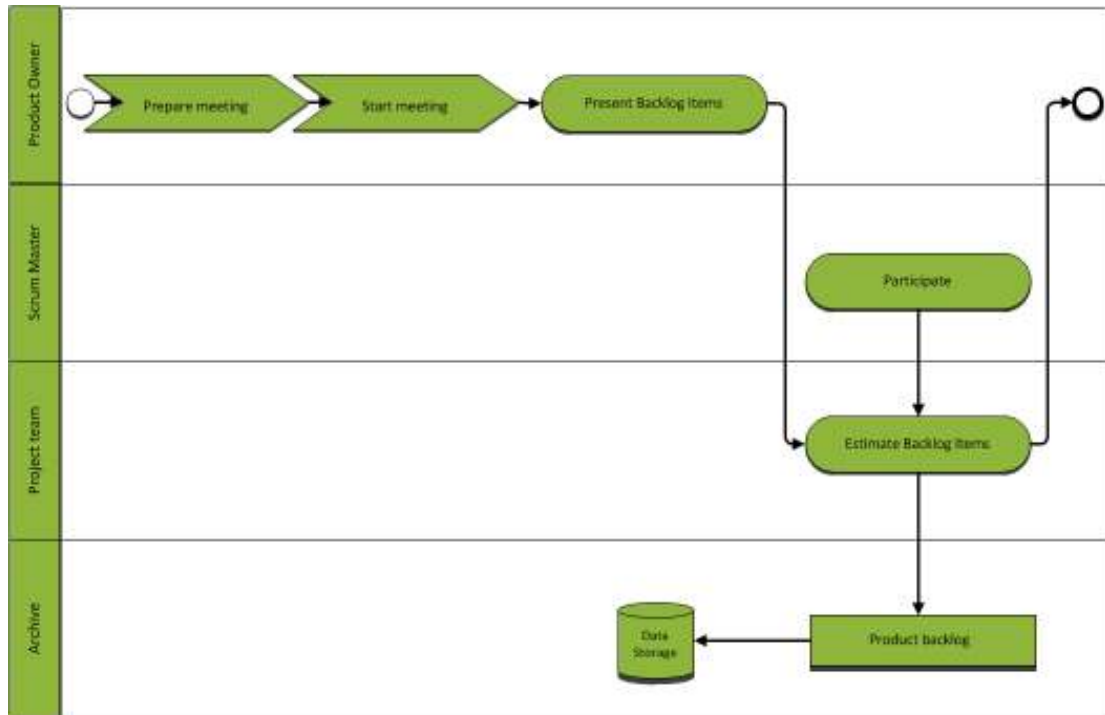


FIGURE 9. Effort is estimated by going through each User Story in the Product Backlog.

Below is a checklist for an effort estimation meeting. It includes meeting basics, preparation and moderation guidelines.

## Basics

Responsible: The Scrum-Team

- Present:
  - Product Owner
  - Scrum Master
  - All Scrum-Team members
- Gain better understanding of Backlog Items by decomposing them if appropriate.
- Only the Scrum-Team estimates. A Product Owner has to be present to help to decide when a story can be broken into smaller stories.

## Sprints

- Should last from two to four weeks
- Each Sprint produces a potentially shippable product increment.

## User stories

- Should be estimated by Story Points rather than time.

## Meeting preparation

- Product Backlog is prioritized
- Product Backlog is visible and accessible to everyone in the meeting.
- A set of Planning Poker cards for each Scrum-Team member is available.

## Procedure

- The Product Owner presents the Product Backlog Items s/he wants to have estimated.
- The Scrum-Team plays Planning Poker for selected Backlog Items.
- In case the Backlog Item is large but has to go into the next or subsequent Sprints, the Scrum-Team splits the Backlog into smaller portions of work. Re-sized items are estimated again on their own.
- Identify Backlog Items that need clarification by the time of the next Estimation Meeting.

### 5.3.4 Set up a project in Jira

- Create a project
- Add project members
  - Scrum Master is the project lead
  - Scrum-Team

Navigate to current projects from the Jira menubar shown in figure 10.

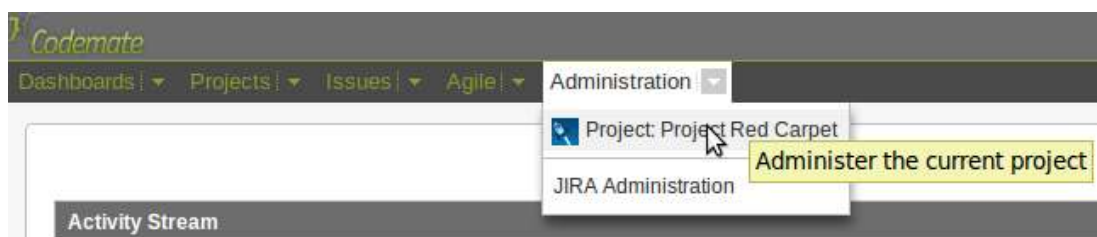


FIGURE 10. View current projects through Administration.

Editing the project members (see figure 12) is opened through the *View Members* link in the project information box. An example is displayed in figure 11.



FIGURE 11. View project members.



FIGURE 12. Edit project members

Based on the Scrum-Team's estimations and items in the Product Backlog, split the development period into sprints and add matching versions to the Jira project.

Manage a current project earlier in the process. New versions to a project can be added via the *Manage versions* link in the project versions box illustrated in Figure 13. A Version is set by giving it a title, description (optional) and a release date (see figure 14).

Versions		
<input type="checkbox"/> Manage versions (displayed in the order of newest first)		
Sprint 4 (week 11)		25/Mar/11
Sprint 3 (week 9)		11/Mar/11
Sprint 2 (week 7)		25/Feb/11
Sprint 1 (week 5)		11/Feb/11
Sprint 0 (week 4)		28/Jan/11

FIGURE 13. Select Manage versions

**Manage Versions**

On this page you can manage the versions for the Project Red Carpet project.  
The newest first displayed in the order of newest first.

**Add Version**

Version Name:

Description:

Previous Code:

Schedule:

Name	Description	Release Date	Schedule	Operations
Sprint 4 (week 11)		25/Mar/11		<a href="#">Edit Details</a>   <a href="#">Merge</a>   <a href="#">Release</a>   <a href="#">Archive</a>   <a href="#">Delete</a>
Sprint 3 (week 9)		11/Mar/11		<a href="#">Edit Details</a>   <a href="#">Merge</a>   <a href="#">Release</a>   <a href="#">Archive</a>   <a href="#">Delete</a>
Sprint 2 (week 7)		25/Feb/11		<a href="#">Edit Details</a>   <a href="#">Merge</a>   <a href="#">Release</a>   <a href="#">Archive</a>   <a href="#">Delete</a>
Sprint 1 (week 5)		11/Feb/11		<a href="#">Edit Details</a>   <a href="#">Merge</a>   <a href="#">Release</a>   <a href="#">Archive</a>   <a href="#">Delete</a>
Sprint 0 (week 4)		28/Jan/11		<a href="#">Edit Details</a>   <a href="#">Merge</a>   <a href="#">Release</a>   <a href="#">Archive</a>   <a href="#">Delete</a>

FIGURE 14. Add a version

Add items in the Product Backlog to Jira as issue cards which represent user stories / tasks. The unscheduled items represent the Product Backlog in Jira. The *Create issue* link is located in the top right corner of the Jira environment, as illustrated in figure 15.

Elena Kurkimaki |

Project:

Issue Type:

FIGURE 15: Creating a new user story / functional requirement can be accessed from the Create Issue -link

Fill the issue name as the issue summary. The issues can be given a great deal of information, such as priorities and versions. Fix Versions refers to the Sprint where it is to be completed. At this point all other fields besides the title are left blank or marked Unknown. (See figure 16.)

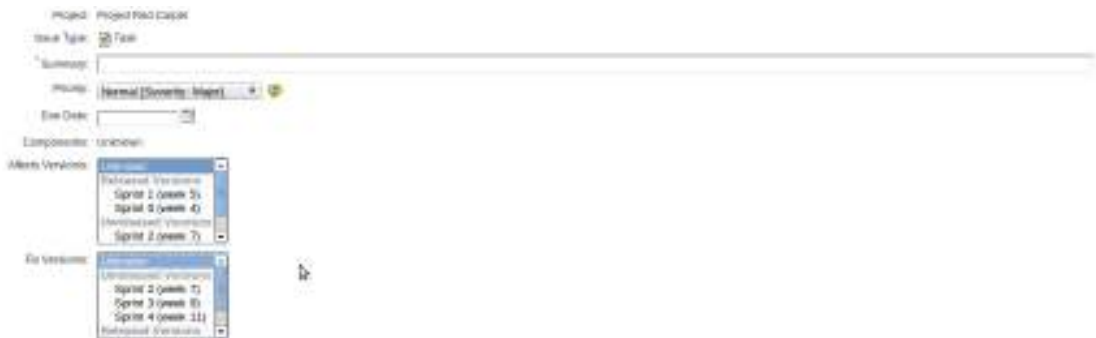


FIGURE 16. Issue card template

Initially the issue card is created without assigning it to anyone or giving it an estimate. See figure 17 for an example.



FIGURE 17. Assigning of the tasks and time evaluation are both left blank at this point.

## 5.4 Planning a Sprint

This chapter explains the two Sprint Planning Meetings. Also creating a Sprint Backlog to Jira is described.

### 5.4.1 Sprint planning meeting #1

The goal of this meeting is to understand what the End User wants. This meeting provides the Scrum-Team with a clear vision of what is needed by

the End User. At the end of this meeting the Scrum-Team will be able to decide what they are able to deliver. The process flowchart is shown in figure 18.

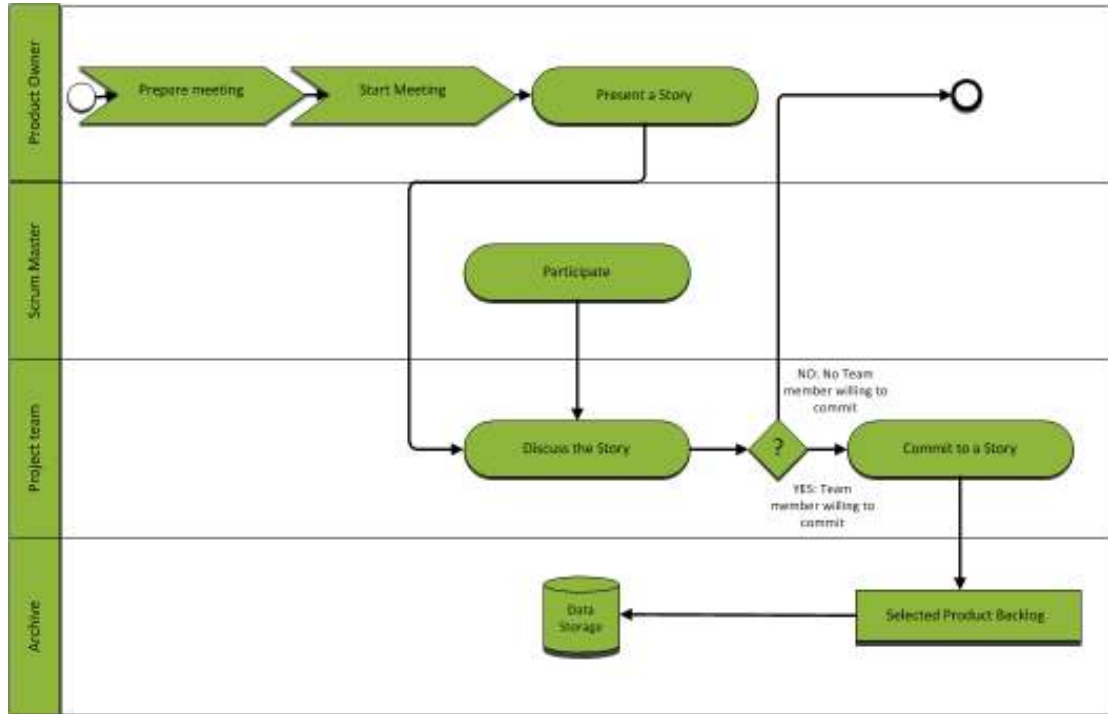


FIGURE 18: Sprint planning meeting #1

Below is a checklist for the first Sprint planning meeting. It includes meeting basics, preparation and moderation guidelines.

## Basics

Responsible: The Product Owner

- Present:
  - Product Owner
  - Scrum Master
  - The whole Scrum-Team
- Only the Scrum-Team members decide how many Backlog Items they can deliver in the Sprint.
- Have an up-to-date Product Backlog
- Results in a Selected Product Backlog
- The whole team believes the plan is achievable.
- Product Owner is satisfied with the priorities.

## **Meeting preparation**

- Participants are invited.
- Product Backlog is prioritized.
- Backlog Items are estimated.
- Product Backlog is visible and accessible to everyone attending the meeting.
- The meeting room is suitable for team discussions.
- The results of the Sprint Review and the Retrospective are available.
- The Sprint Schedule is defined in the following way:
  - Appointment for the Sprint Planning 1
  - Appointment for the Sprint Planning 2
  - The first day of the Sprint
  - The last day of the Sprint
  - Appointment for the Daily Scrum
  - Appointment for the Sprint Review Meeting
  - Appointment for the Sprint Retrospective

## **Procedure**

- Review the basic parameters: start and end date, time and location of the Sprint review meeting, the Scrum-Team availability and Definition of Done.
- Present and discuss each story. A time box for each story may be useful to make the whole meeting on follow the plan. Holding a reserve at the end of this section for difficult stories often makes it easier to continue if the discussion gets stuck on one story.
- Commit to the stories. Go through the list, one at a time and in order of priority. Get the Scrum-Team or a team member to commit each one until no one commits to any more.
- Agreement. Confirm the selected product backlog of the committed stories with the Product Owner.



### Duration / Location

The ideal duration and frequency of the meeting is 60 minutes per every week of the Sprint. Run this meeting in the morning - to be able to have the Sprint Planning Meeting #2 on the same day.

### Output

- Selected Product Backlog
- Requirements for each Backlog Item.

### 5.4.3 Sprint planning meeting #2

In this meeting, the Scrum-Team has the chance to create the design for the solution they want to implement. At the end of this meeting, the Scrum-Team knows how to build the functionality they are going to deliver in that Sprint. The meeting process chart is presented in figure 19.

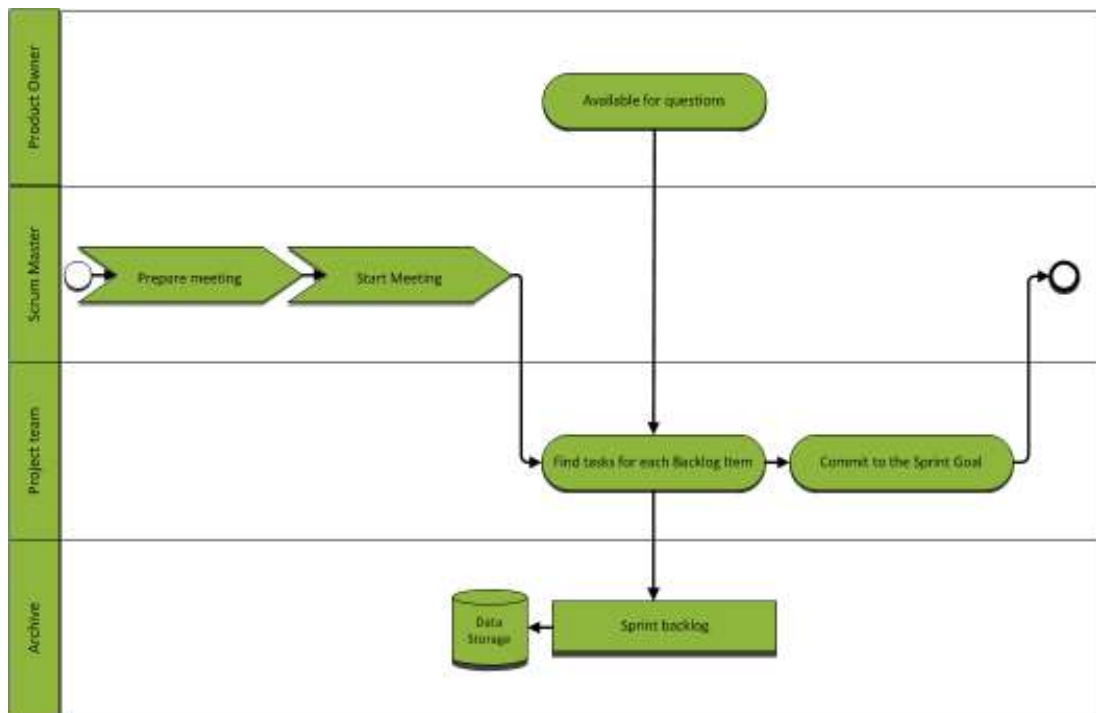


FIGURE 19. Sprint planning meeting #2

Below is a checklist for the second Sprint planning meeting. It includes meeting basics, preparation and moderation guidelines.

## **Basics**

Responsible: The Scrum Master

- Present:
  - Scrum Master
  - The whole Scrum-Team
  - Product Owner reachable for questions

## **Meeting preparation**

- Participants are invited
- The Selected Backlog is visible and accessible for the task planning.

## **Procedure**

- The Scrum-Team members find tasks for each Selected Backlog Item.
- The effort of one task should not be greater than one day.
- The Scrum-Team commits to the Sprint Goal.
- If the Scrum-Team believes that the Sprint Backlog is too small:
  - move the most important Backlog Items from the Product Backlog to the Sprint Backlog together with the Product Owner.
- If the Scrum-Team believes that the Sprint Backlog is too large:
  - remove Backlog Items together with the Product Owner.

## **Duration / Location**

The ideal duration and frequency of the meeting is 60 minutes per every week of the Sprint. Run this meeting right after the Sprint Planning Meeting #1.

## **Output**

- Sprint Backlog
- Design of your application
- Architecture diagrams, charts, drawings

- Ensure that the Scrum-Team has a clear understanding of how to do it.

#### 5.4.4 Setting up the Sprint Backlog in Jira

The Sprint Backlog is created into Jira by assigning the previously created story cards for the Sprint in question and dividing them in to smaller portions of work by adding sub-tasks to them. Take the unscheduled cards (see figure 20) created in 6.3 and edit them with the following information:

- Version: Assign the task to a sprint
- Original estimate: Estimate as precisely as possible the effort put into completing that one task.



*FIGURE 20: Edit issue*

Assigning the task to a Sprint is done by setting the Fix version to the Sprint in question as shown in figure 21. The original estimate for the story can be set here as well (see figure 22). If the story is large and therefore is to be divided into sub-tasks, their estimated effort is added to the main story automatically, and in that case, the effort here can be left blank.

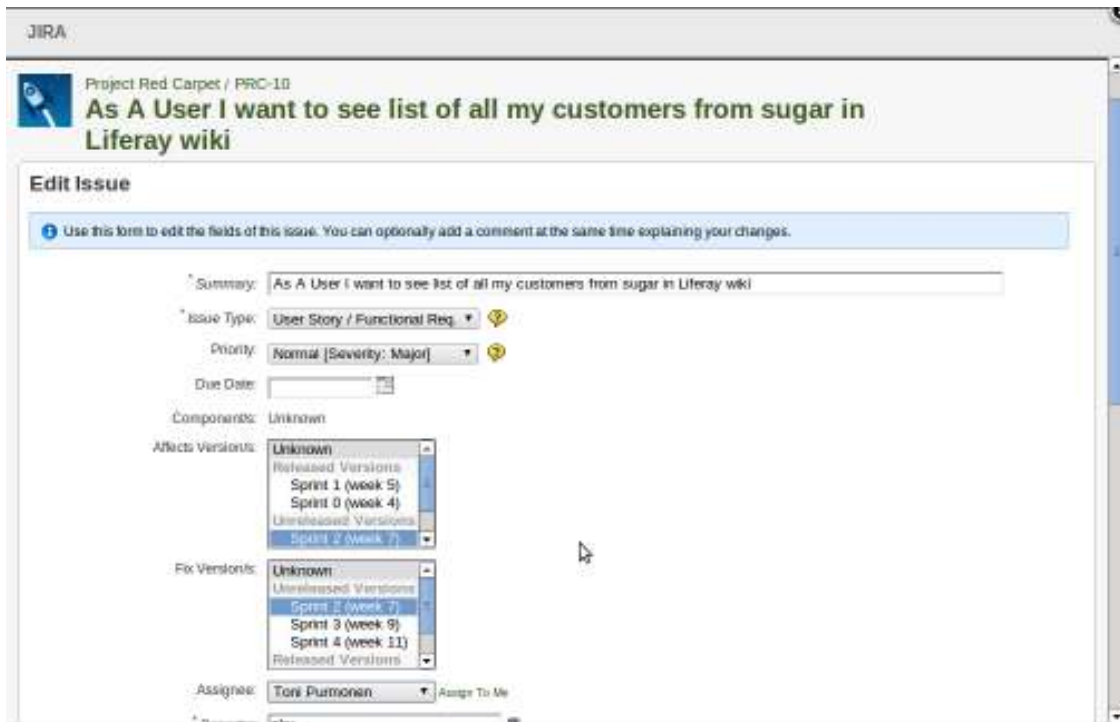


FIGURE 21: Issue information ready for editing

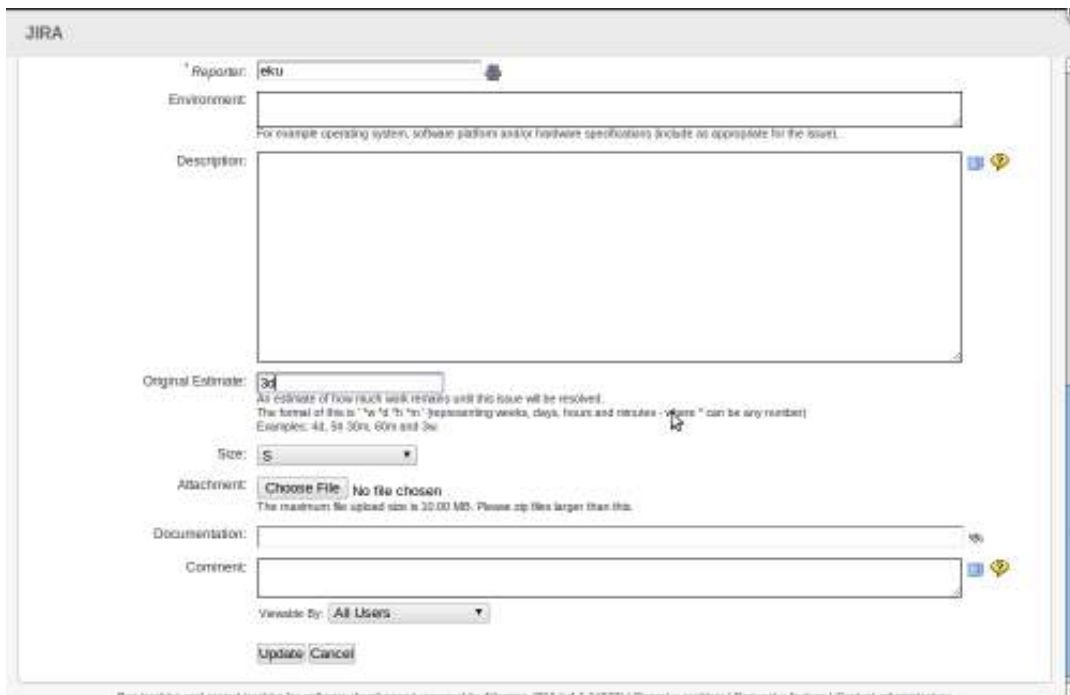


FIGURE 22: Setting up original estimate

User Stories can be split into smaller tasks by adding sub-tasks to them. The ideal is that if an issue takes longer than a day to be completed, it should be divided into smaller portions of work. To add sub-task to stories, first go to the planning board from the Agile menu, as shown in figure 23.

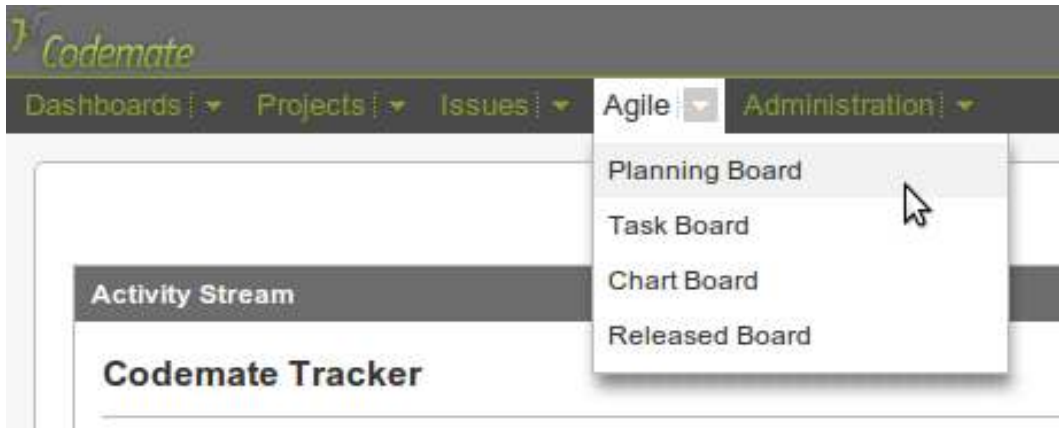


FIGURE 23. Planning board is opened from the Agile drop down menu

The User Stories are portrayed as cards on the Planning Board. One can add a sub-task to a story by clicking the plus sign on the card shown in figure 24. The information of the sub-task is much alike with its parent story's (figure 25).



FIGURE 24. Adding a sub-task to a User Story

**New Sub-task**

**PRC-1**  
As A User I want to see list of all my customers from sugar in Liferay wiki

Sub-task Normal [Severity: Major]

\* Summary:  
asdfas  
Characters left: 249

Version: • Sprint 2 (week 7)

Component: Unknown

Original estimate: 2d

Assignee: Toni Purmonen

Reporter: Elina Kurkimaki

Create Create and Close Cancel

*FIGURE 25: Fill in the sub-task information: type, severity, summary, fixed sprint, estimate and assignee*

## 5.5 Executing a Sprint

This chapter describes the tasks involved in executing a Sprint. Figure 26 illustrates the lifecycle of a Sprint.

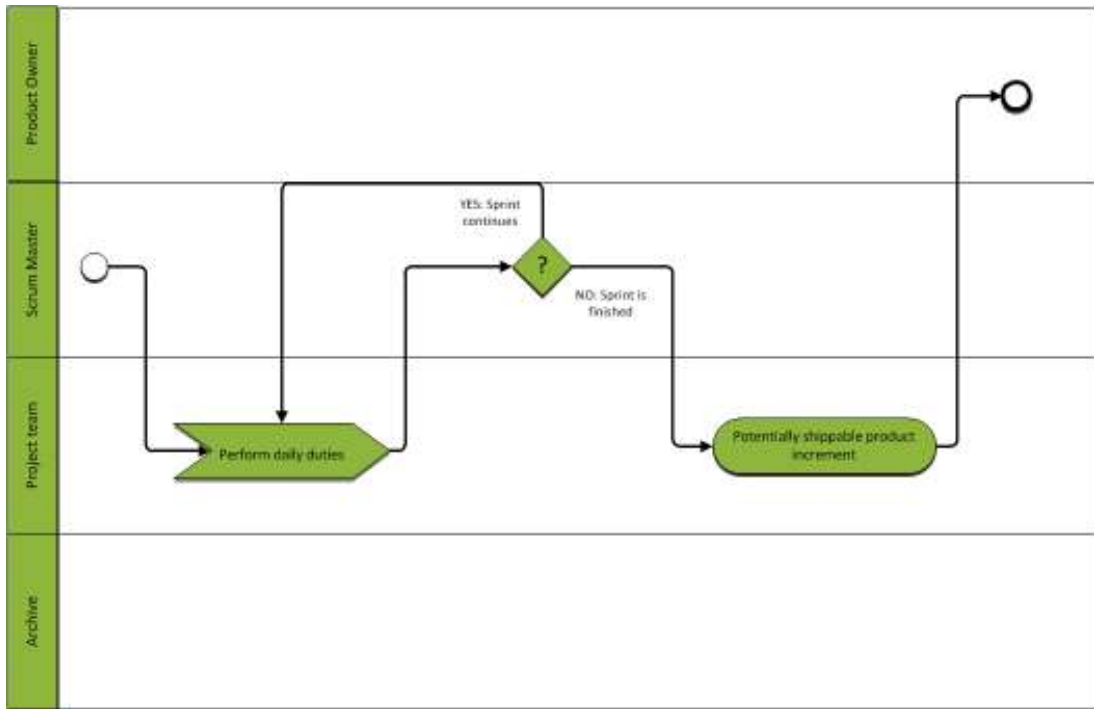


FIGURE 26. Sprint on a daily basis

Figure 27 illustrates the daily duties in a Sprint. Updating the Kanban task board in Jira is described first in this chapter. Guidelines to having Daily Scrum are explained after that.

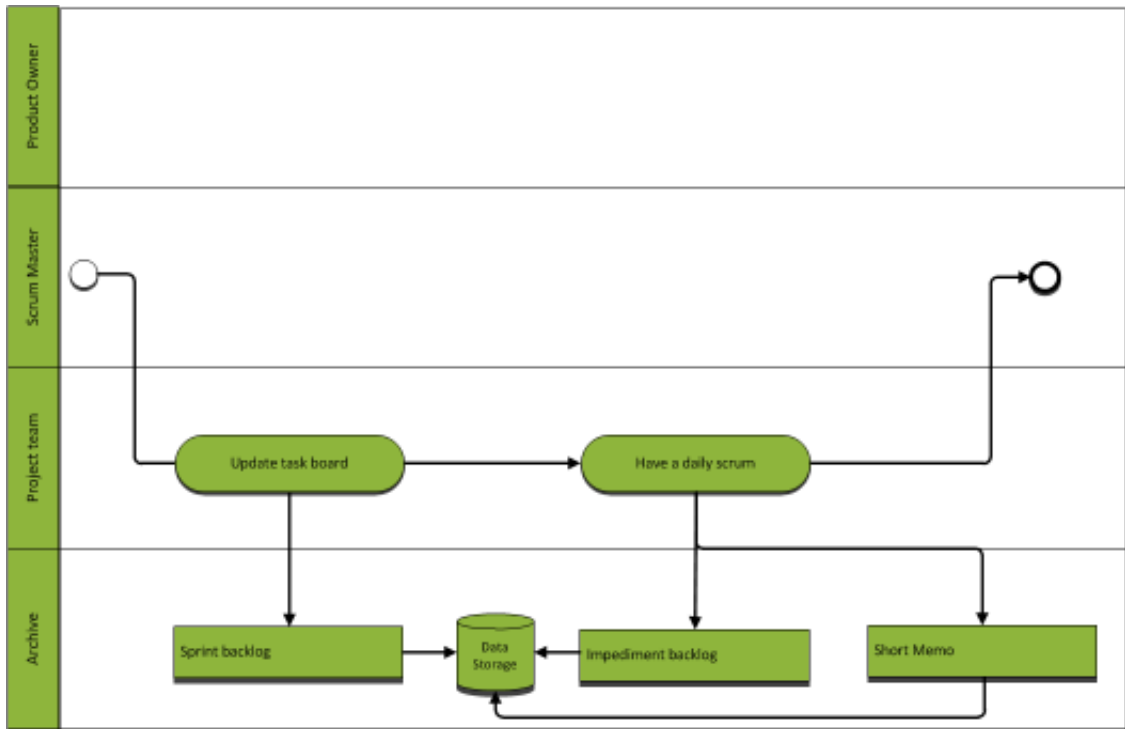


FIGURE 27. Daily duties of the Scrum-Team in the Sprint execution process

### **5.5.1 Basics**

Below is a checklist for a Sprint. It includes sprint basics, Sprint Backlog requirements, daily responsibilities and guidelines for assigning tasks.

#### **Some basic Sprint ideas:**

- The length is two to four weeks or less.
- Always end on time.
- The Scrum-Team must not be disrupted or controlled by outsiders.
- Definition of Done is known by all team members and is achievable in one iteration.

#### **Sprint Backlog**

- is highly visible.
- is updated daily.
- is owned exclusively by the Scrum-Team.

#### **Daily Responsibilities**

- The Sprint Backlog is updated daily through the Kanban task board.
- The task board is maintained by the Scrum-Team only.

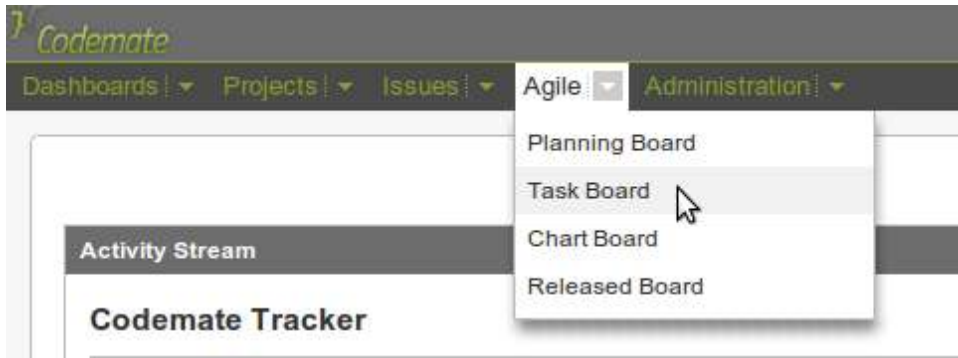
#### **Task assignment and division**

- There should not be more than two issues assigned to a team member at a time.
- Not all of the Scrum-Team members should have two issues assigned to them at once.

### **5.5.2 Updating task board in Jira**

The issue cards assigned to a specific sprint is the Sprint Backlog. The task board, shown in Figure 29, is accessed from the *Agile* menu illustrated in figure 28.





**FIGURE 28.** Open Task Board



**FIGURE 29.** The Task Board consists of six columns representing the six different statuses an issue card can have

When a Scrum-Team member takes an issue under development, they assign the task to themselves. The card information is opened as shown in figure 30.



**FIGURE 30.** To start working on an issue, open the issue for viewing

For assigning of an issue to someone, the issue card has to be opened for editing. This is shown in figure 31. Once under editing, an issue can be assigned for a drop down list or selecting *Assign to me* (figure 32).

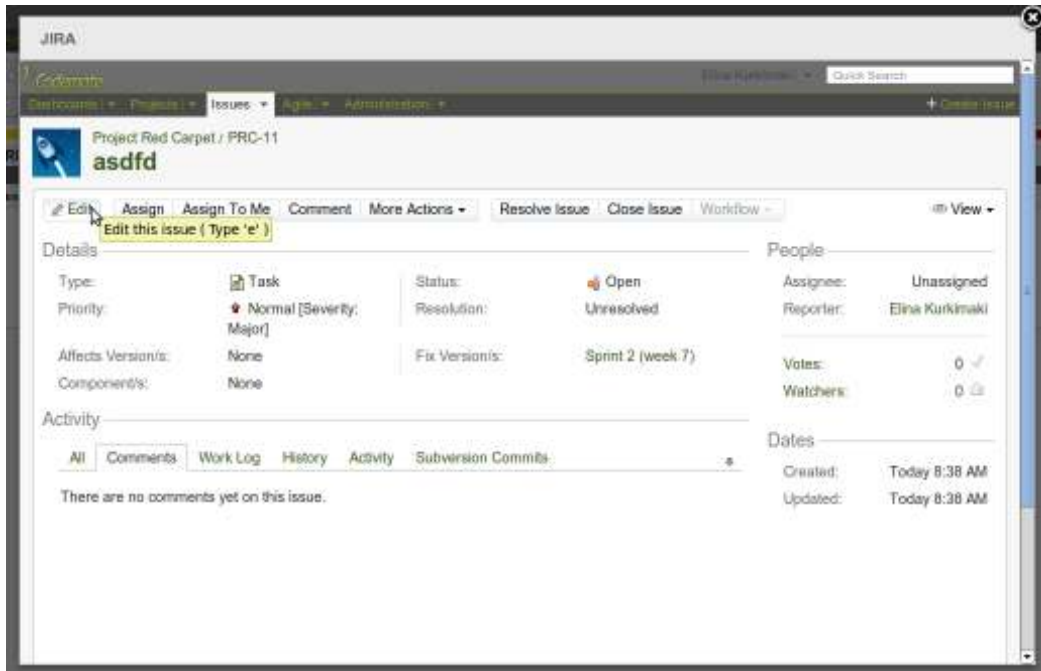


FIGURE 31. Issue has to be opened for editing to be assigned

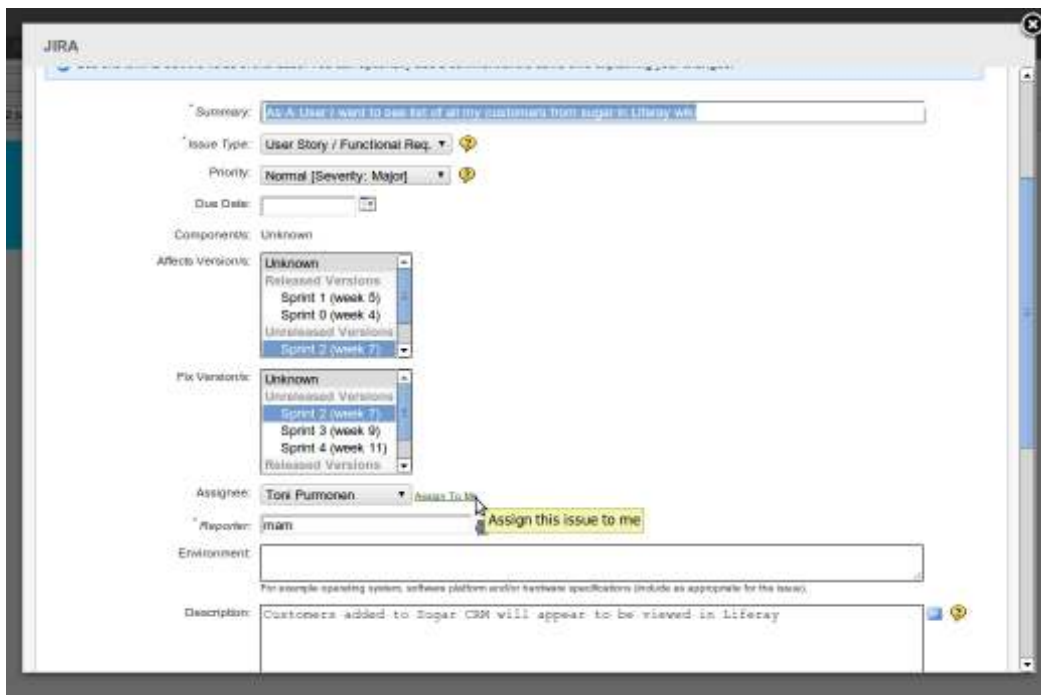


FIGURE 32: By clicking "Assign to me", the issue is assigned to the current Jira user

The Scrum-Team members keep track of cards assigned to them by moving them along the task board illustrated in figure 33. In the Sprint planning phase, the cards are added under *TODO* and *PRIORITY TODO*.

When a task is taken under development, the assignee drags the card to *IN PROGRESS*. Once the task has reached the definition of done, it is moved to *READY FOR VERIFICATION*, and the team member can pick a new task from the *PRIORITY TODO*-list. The person responsible for verifying the outcome comes in and drags the card from *READY FOR VERIFICATION* to *IN VERIFICATION* for examination. Once the issue has been verified, the responsible moves the card to *VERIFIED*, and the issue is considered closed. See table 2 for column meanings.

**TABLE 2. Task board columns and their references to Sprint artefacts**

TO-DO	Sprint Backlog
PRIORITY TO-DO	Sprint Backlog top prioritized items
IN PROGRESS	In progress
READY FOR VERIFICATION	Resolved, not closed
IN VERIFICATION	In verification
VERIFIED	Closed



**FIGURE 33. Items are dragged into places by assignees**

Logging work is an important part of keeping track of the project progress compared to the estimated effort. Each of the Scrum-Team members update daily their effort to issues assigned to them. The work can be logged by opening up the issue information as presented before, or straight from the Kanban board. The *Log Work* link is shown in figure 34. A work log consists of a description of work and time spent on it. The logged effort amount automatically adjusts the remaining effort to meet the current situation (figure 35).



FIGURE 34. Work is logged to tasks daily

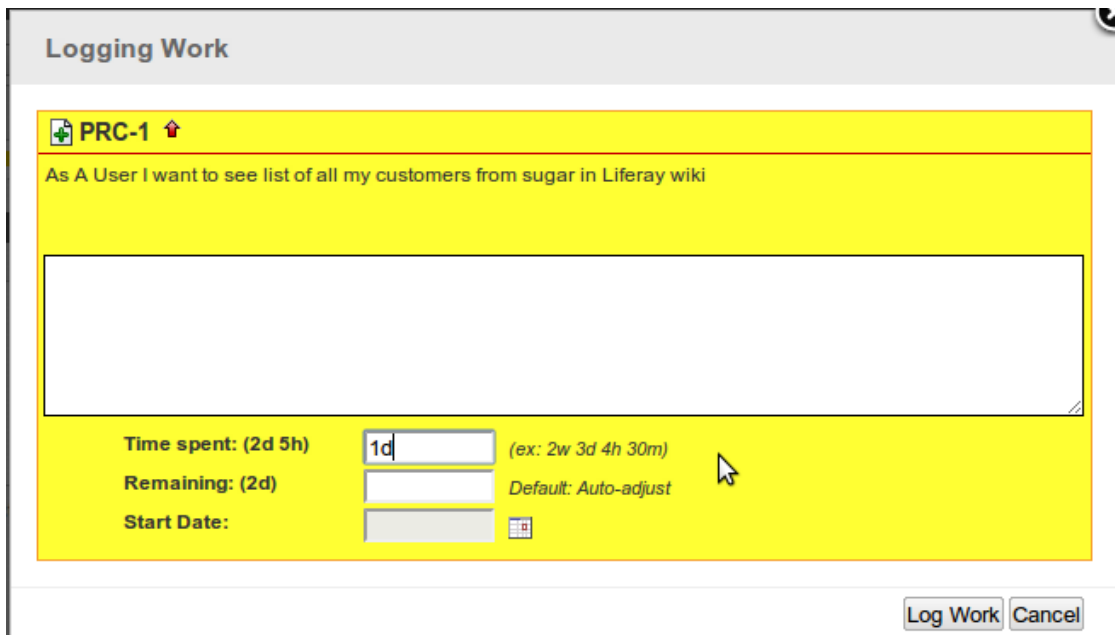


FIGURE 35. The remaining estimate is auto adjusted based on the logged work

### 5.5.3 Daily Scrum

This meeting is the time for the Scrum-Team to plan and coordinate its daily activities and discuss the impediments. The Task Board helps the Scrum-Team to on its daily activities.

#### Basics

Responsible: The Scrum Master

- Present:
  - Scrum Master
  - The Scrum-Team
- This 15-minute meeting is held daily at the same time in the same location.

## **Meeting preparation**

- Participants are invited.
- Sprint backlog is updated.
- Task board is updated; each Scrum-Team member updates the state of the tasks assigned to them.
- Tasks on the Sprint Backlog are editable, addable, removable and sortable.

## **Procedure**

- Each Scrum-Team member answers the following questions:
  - What have you done since the last Daily Scrum?
  - What are you going to do before the next Daily Scrum?
  - Are there any impediments that might prevent you from achieving the goals of the current sprint?
- Scrum Master is the one obliged to solve any impediments that might occur.
- Daily scrum is to be kept short, avoid sidetalks.

## **Output**

- A short memo
- Clear insight into who does what
- Input for the Impediment Backlog.

## **5.6 Ending a Sprint**

This chapter explains the actions that take place when a Sprint is finished: Sprint Review Meeting, Sprint Retrospective Meeting and making a release in Jira.

### **Basics**

- Each iteration should end on time.
- Definition of Done is respected by the whole team.

## 5.6.1 Sprint review meeting

The Scrum-Team shows the results of its work to the End User. The Scrum-Team members want to have feedback. This feedback can be used to create or change Backlog Items. See the process chart in figure 36.

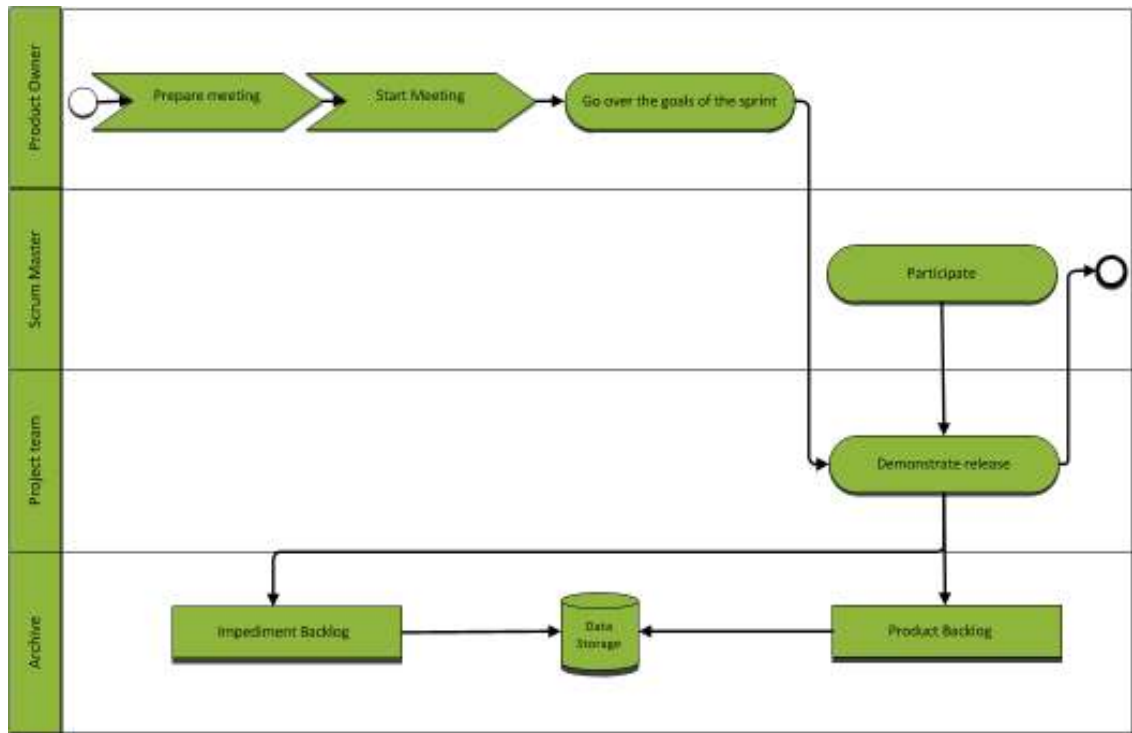


FIGURE 36. The process of a Sprint review meeting

Below is a checklist for the Sprint review meeting. It includes meeting basics, preparation and moderation guidelines.

### Basics

Responsible: The Product Owner

- Present:
  - The Scrum-Team
  - Scrum Master
  - Product Owner
  - User
- Takes place after every sprint
- Shows the working, tested software
- Feedback is received from Product Owner & Stakeholders.

- Everyone is invited, even other teams, for example.
- The length of the demo is limited.
- All comments are logged.
- The status of the work is presented.

### **Meeting preparation**

- Participants are invited.
- The Sprint Goal is visible to everyone.
- The Selected Product Backlog is accessible and visible to everyone.
- The team has prepared workstations, devices and everything else needed to demonstrate the new functionality.

### **Procedure**

- The Product Owner welcomes the people to the Sprint Review.
- The Product Owner reminds everybody in the room what the purpose of the Sprint has been.
- The product development team demonstrates the new functionality and lets the End User play with the new functionality.
- The Scrum Master facilitates the session.
- The Feedback of the End User will be documented by the Product Owner and/or Scrum Master.
- If the Product Owner wants to change a feature:
  - add a new Backlog Item to the Product Backlog.
- If the Scrum-Team reports an impediment which is not solved yet:
  - add the impediment to the Impediment Backlog.

### **Duration / Location**

The ideal duration and frequency of the meeting is 90 minutes at the end of the Sprint.

### **Output**

- Feedback from the End User
- Input for the Impediment Backlog

- Feedback from the Scrum-Team generates the inputs to the Product Backlog.

## 5.6.2 Sprint retrospective meeting

This meeting is for finding out what has to be improved by discussing the Sprint implementation. It is about improving the productivity of the Team by learning from the past for the future. The meeting process is illustrated below in figure 37.

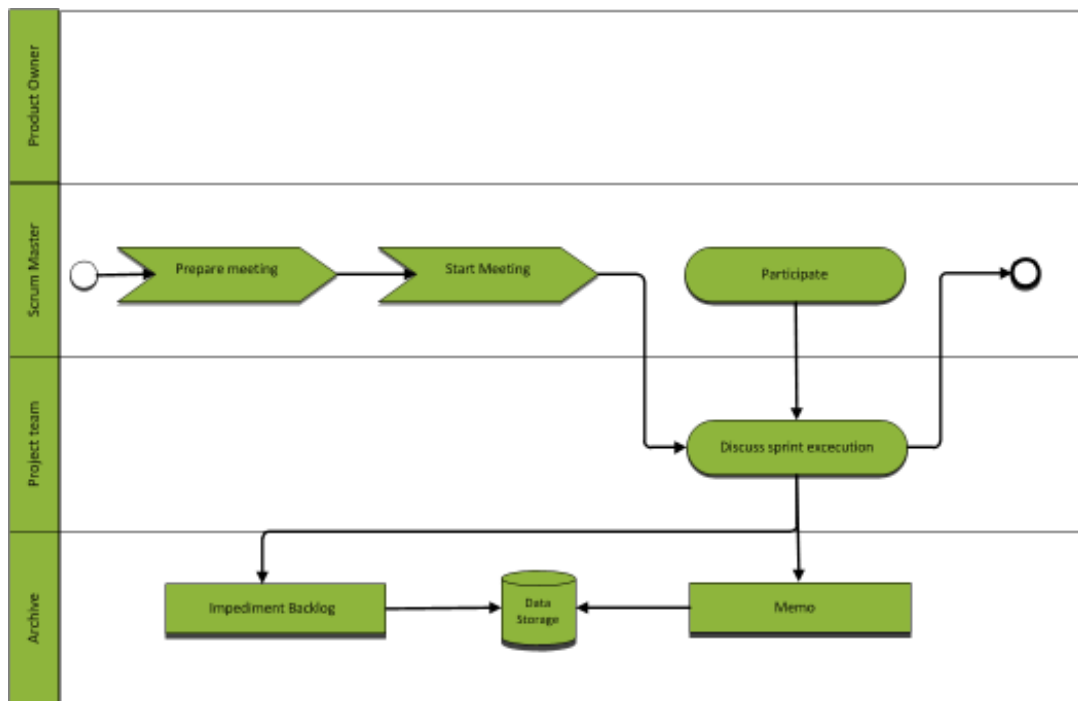


FIGURE 37. The Sprint retrospective process

Below is a checklist for the Sprint retrospective meeting. It includes meeting basics, preparation and moderation guidelines.

### Basics

Responsible: The Scrum Master

- Present:
  - The Scrum-Team
  - Scrum Master



### **Meeting preparation**

- Participants are invited.

### **Procedure**

- Present the goal of the meeting.
- Have the Scrum-Team answer the following questions:
  - What went well?
  - What could be improved?
- Results in concrete improvement proposals
- Some proposals are implemented.
- The whole team participates,

### **Duration / Location**

The ideal duration and frequency of the meeting is 90 minutes, held some minutes after the review.

### **Output**

- Memo
- Input for the Impediment Backlog

### **5.6.3 Making a release in Jira**

Make a release after completing iteration.

Go to planning board like described earlier in the documentation. Each version has their own Sprint box on the right side of the board, where a link to making a release can be found. See figure 38 for an example.



FIGURE 38. Making a release is done by clicking the release button in the Sprint box.

When releasing, Jira saves snapshots of the iteration Burndown Charts. If not all issues are completed, the future of undone cards can be decided here as well. Figure 39 is an example of such a release case.

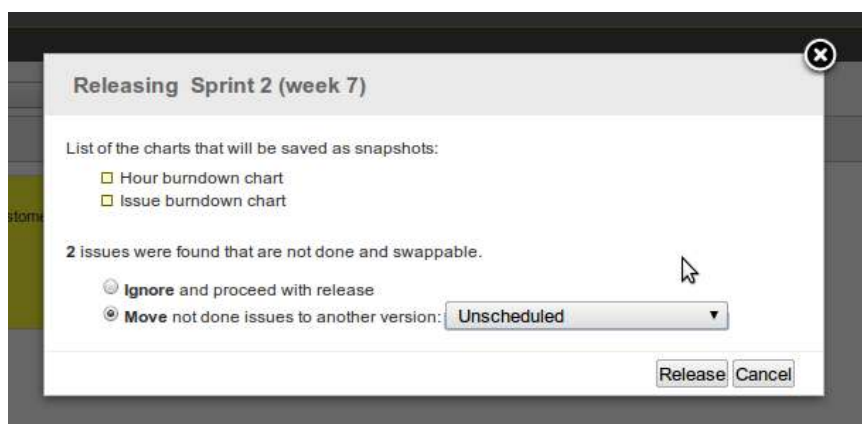


FIGURE 39. Release asks to determine the future of unfinished issues of a Sprint in a Release

## **5.7 Artefacts**

This is a reference list of different artefacts involved in projects. Their reference in Jira is also explained.

### **5.7.1 Impediment Backlog**

The impediment backlog is the ultimate risk action board. A Scrum Master uses this list to visualize the impediments to the productivity of the Scrum-Team. It also reflects his/hers action items as one needs to remove these blocking issues as fast as possible.

Responsible: The Scrum Master.

### **5.7.2 Product Backlog**

The Product Backlog is a list. This list contains the so called Product Backlog Items (words, stories, requirements, features, functionality etc.) a Scrum-Team wants to deliver in the future. All the Product Backlog Items in this list are sorted based on the business value.

Responsible: Product Owner

#### **Jira**

The Product Backlog is the list of the unscheduled story cards viewable in the Planning Board and Task Board.

### **5.7.3 Selected Product Backlog**

The selected Product Backlog is a sorted list of the Product Backlog Items the Scrum-Team wants to build till the end of a Sprint. The difference to Sprint Backlog is that the stories are not divided into smaller tasks yet.

#### **Jira**

The Sorted Product Backlog is the list of the story cards assigned to a specific Sprint. They are viewable in the Planning Board and Task Board.

## **5.7.4 Potentially Shippable Product Increment**

At the end of the Sprint the Scrum-Team delivers a Potentially Shippable Product Increment. It is a piece of the product that one does not have to work on again.

## **5.7.5 Sprint Backlog**

The Sprint Backlog is a list of tasks that enables the Scrum-Team to visualize its activities. It helps to synchronize the activities of the Scrum-Team.

### **Jira**

The Sprint Backlog is the list of story and task cards assigned to a specific Sprint. It is viewable in the Planning Board and Task Board.

## **5.7.6 Burndown Chart**

Burndown Charts show the work remaining over time. The work remaining is on the Y axis and time is on the X axis. Ideally, the work remaining should go up and down and eventually trend downward. The tracking of the progress is done by the Scrum-Team.

### **Jira**

Jira has two kinds of Burndown Charts:

- Issue Burndown Chart
- Hour Burndown Chart

The Burndown Charts are updated automatically based on the logged work and issue statuses. Opening the Chart Board is shown in figure 40. Current Sprint Burndown Charts, such as the one in figure 41, can be found there.

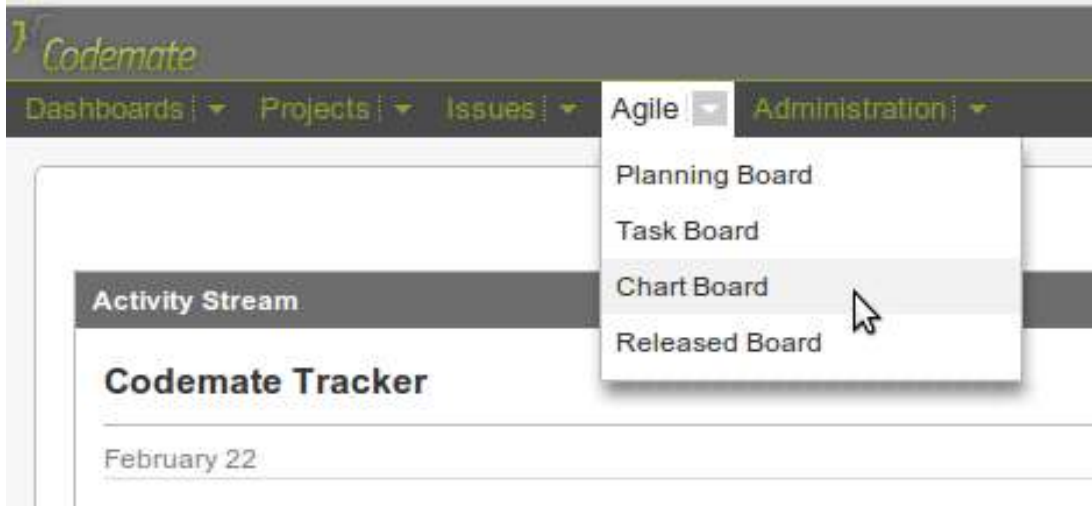


FIGURE 40. Open the chart board

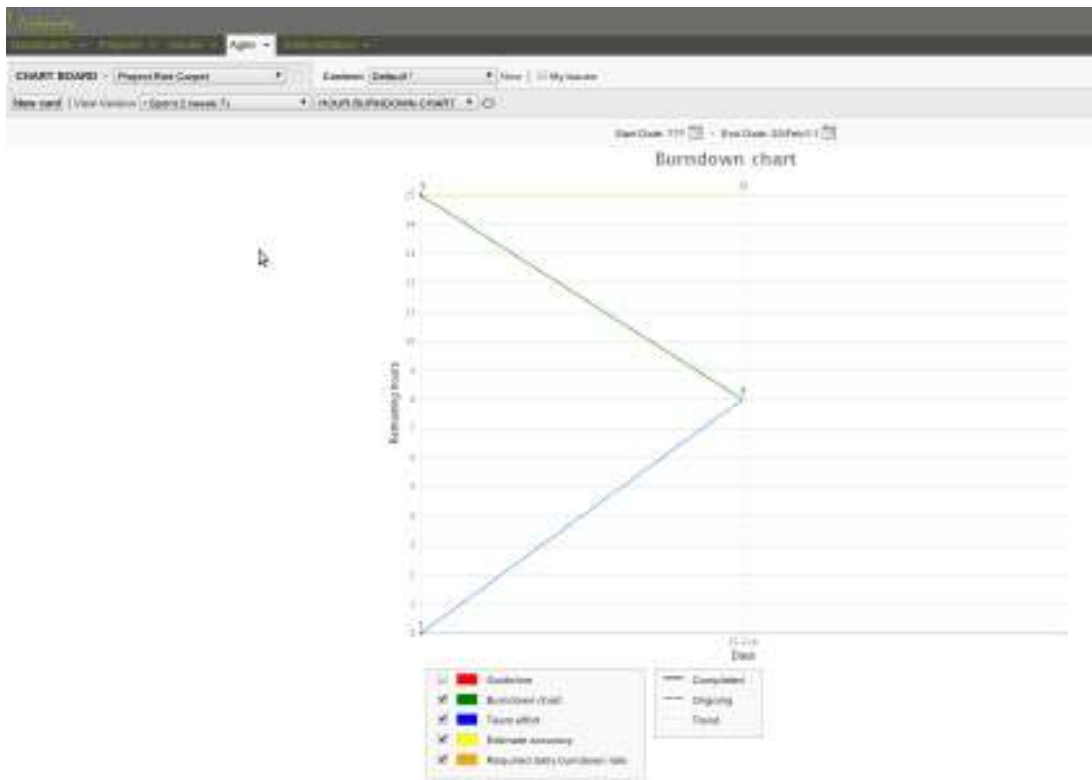


FIGURE 41. Current sprint hour Burndown Chart

Past Sprint Burndown Charts can be viewed in the Released Board. The Released Board opens as illustrated in figure 42. Figure 43 is an example of a past Sprint issue Burndown Chart.

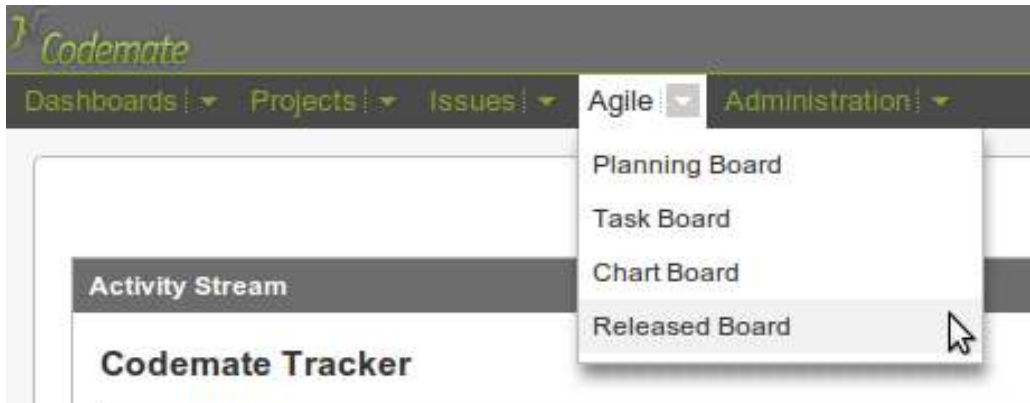


FIGURE 42. Released board show information on passed versions

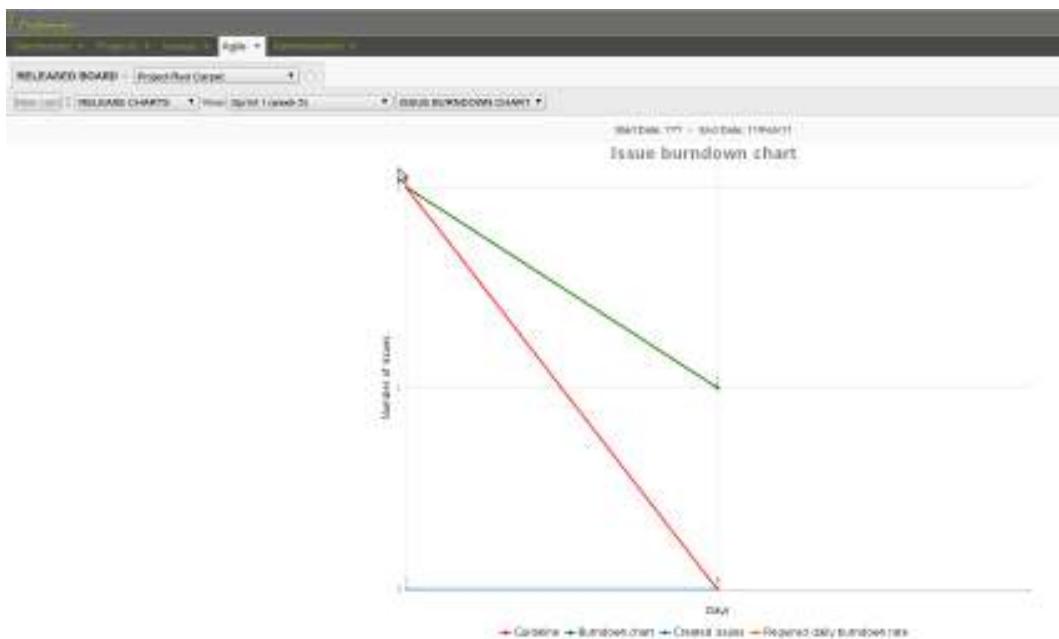


FIGURE 43. An example of a issue Burndown Chart in Jira

## **6 CONCLUSIONS AND FUTURE WORK**

The documented process was not created to establish a new project management process to the target company. It was merely meant to be documented for the company's quality management system as the process is being done in the organization at the moment.

In the planning phase of the process it was discovered that as the overall framework for a project management process was the same amongst different projects, there was some fine tuning to be done in the guidelines for steps to be taken within a project. The process presented in this thesis is a combination of presenting what was already being done in the organization and introducing some new practices from the agile development methods.

The established project management process proved to be efficient based on the empirical validation and feedback from the target company. Both Scrum and Kanban combined make it easy to tailor a project process to meet the organization's needs.

The future work for the process includes broadening the scope to include the customer-related sub-processes and exception situations. The process will be taken into practice in the target company's projects in the future, which also means including the designing and carrying out personnel training.

The formatting of this process description will most likely be reformed in the future as an overall documentation template is created for all of the processes described for the target company's quality management system. As this process is only one part of the quality management system, the work on other process descriptions for the system starts where this thesis finishes.

## 7 REFERENCES

1. About Codemate. Available at: <http://www.codemate.com/company>. Date of data acquisition: 28.4.2011
2. Wikipedia 2011. Agile software development. Available at: [http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development). Date of data acquisition: 22.4.2011
3. Abrahamsson, Pekka - Salo, Outi - Ronkainen, Jussi - Warsta, Juhani 2002. Agile software development methods - Review and analysis. Espoo: VTT Publishing.
4. Dyba, T. - Dingsoyr, T. 2009. What Do We Know about Agile Software Development? IEEE Software vol.26, issue 5. P.6.
5. The Agile Manifesto. Available at: <http://www.agilealliance.org/the-alliance/the-agile-manifesto/>. Date of data acquisition: 22.4.2011.
6. Principles behind the Agile Manifesto. Available at: <http://www.agilemanifesto.org/principles.html>. Date of data acquisition: 22.4.2011.
7. Scrum Is an Innovative Approach to Getting Work Done. Available at: [http://www.scrumalliance.org/learn\\_about\\_scrum](http://www.scrumalliance.org/learn_about_scrum). Date of data acquisition: 22.4.2011.
8. Introduction to Scrum - An Agile Process. Available at: <http://www.mountangoatsoftware.com/topics/scrum>. Date of data acquisition: 22.4.2011.
9. Kanban. Available at: <http://www.crisp.se/kanban>. Date of data acquisition: 22.4.2011.
10. Kniberg, Henrik - Skarin, Mattias 2010. Kanban and Scrum making the most of both. USA: C4Media Inc.
11. Wikipedia 2011. Business process. Available at: [http://en.wikipedia.org/wiki/Business\\_process](http://en.wikipedia.org/wiki/Business_process). Date of data acquisition: 22.4.2011.
12. Quality management systems - Requirements 2008. International Standard ISO 9001. Switzerland: ISO.



**13.**Jira issue and project tracking. Available at:  
<http://www.atlassian.com/software/jira/>. Date of data acquisition  
25.4.2011.

**14.**reStructuredText. Available at: <http://docutils.sourceforge.net/rst.html>.  
Date of data acquisition 24.4.2011.