Bern University
of Applied Sciences

# URL-Archiver

SCRUM report

| | |
|---|---|
| Course of study | Project 1 |
| Author | Nicolin Dora, Abidin Vejseli & Kilian Wampfler |
| Advisor | Dr. Simon Kramer |
| Co-advisor | Frank Helbling |

Version 1.0 of January 4, 2024

▶ School of Engineering and Computer Science
▶ Computer Science

# Contents

# List of Tables

# List of Figures

# 1 Implementation

## 1.1 Processes

### 1.1.1 Allocation of roles

In this chapter, the Scrum roles (Product Owner, Scrum Master, Developer) and additional roles such as Customer, Stakeholder, etc. are defined.

### 1.1.2 Scrum roles

We have decided to structure our Scrum team in the following manner:

| Role | Person |
|---|---|
| Product Owner | Nicolin Dora |
| Scrum Master | Abidin Vejseli |
| Developer | Nicolin Dora, Abidin Vejseli, Kilian Wampfler |

Table 1.1: Scrum Roles

Nicolin took on the role of Product Owner as he had concrete ideas and visions for the product at the start of the project. Additionally, he took on this role because he wanted to deal with the subjects surrounding the product backlog.

Abidin took on the role of the Scrum Master as he has the most experience with the agile way of working. He has already had the opportunity to perform this role professionally on several smaller projects in the past.

Kilian took on the role of a Developer, as he is an active programmer in his job and has already gained some experience with Scrum. Therefore, self-organization is not a foreign concept to him.

Besides Kilian, all the other members of the group were also assigned the role of Developer, as otherwise the project would not have been feasible in the given time. This is due to the fact that we all work alongside the university.

### 1.1.3 Additional roles

In addition to the Scrum roles, we have assigned the following roles to our specialist lecturer and PM-coach.

| Role | Person |
|---|---|
| Stakeholder | Dr. Simon Kramer |
| Customer | Dr. Simon Kramer |
| PM-Advisor | Frank Helbling |

Table 1.2: Additional Scrum Roles

### 1.1.4 Sprint Goals

We have defined the goals of our past and current sprints in the best possible way according to the SMART[1] criteria. The goals of our sprints are listed below:

Sprint 1 Implement input handler for files (any unicode file e.g. .bib, .txt, .html and .pdf) and basic user guidance (Menu, Error messages).

Sprint 2 Implement a function to scan a provided text in order to identify and extract any URLs contained within it and upgrade our current console-based interface to enable users to easily open any extracted URL using their default web browser.

Sprint 3 Develop and implement a fully automated URL submission system that integrates with the Wayback Machine and Archive Today to ensure at least a 98

Sprint 4 Enhance the system's stability and usability by resolving identified Selenium bugs across Linux, macOS, and Edge browsers, documenting the sprint process and licenses, conducting a thorough code review, and establishing a new configuration management file, aiming for zero critical bugs at sprint closure and readying the system for seamless URL archiving integration in subsequent sprints.

Sprint 5 Complete application refactoring for asynchronous archiving and .BIB file URL integration, ensuring no critical bugs and preparing for seamless future enhancements.

[1]Atlassian blogpost: "How to write smart goals"

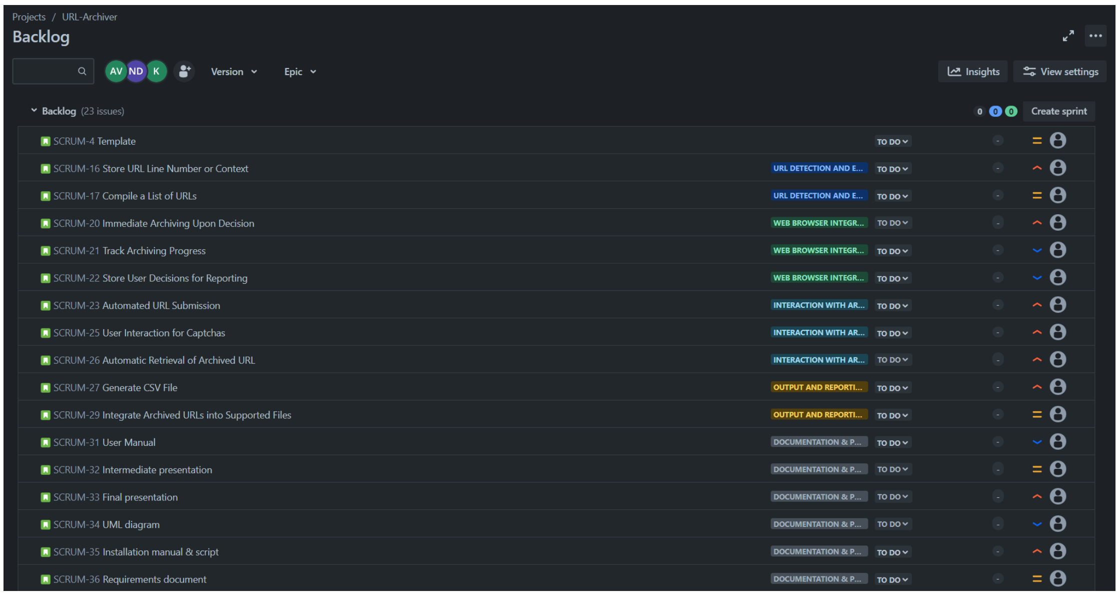
Sprint 6 Deliver a finalized application design, improved code quality, and complete documentation, with all components ready for review.

### 1.1.5 Requirements

In this chapter, we present our product and sprint backlogs, structured according to Scrum.

#### Product Backlog

Our product backlog consists of user stories and epics created by our Product Owner. The user stories are prioritised and represent a set of initial requirements that must be met to achieve our product goal. The product backlog is maintained in Jira.


Projects / URL-Archiver

Backlog

Backlog (23 issues) 0 0 0

| | | | | |
|---|---|---|---|---|
| SCRUM-4 Template | | TO DO | - | = |
| SCRUM-16 Store URL Line Number or Context | URL DETECTION AND E... | TO DO | - | ^ |
| SCRUM-17 Compile a List of URLs | URL DETECTION AND E... | TO DO | - | = |
| SCRUM-20 Immediate Archiving Upon Decision | WEB BROWSER INTEGR... | TO DO | - | ^ |
| SCRUM-21 Track Archiving Progress | WEB BROWSER INTEGR... | TO DO | - | ⌄ |
| SCRUM-22 Store User Decisions for Reporting | WEB BROWSER INTEGR... | TO DO | - | ⌄ |
| SCRUM-23 Automated URL Submission | INTERACTION WITH AR... | TO DO | - | ^ |
| SCRUM-25 User Interaction for Captchas | INTERACTION WITH AR... | TO DO | - | ^ |
| SCRUM-26 Automatic Retrieval of Archived URL | INTERACTION WITH AR... | TO DO | - | ^ |
| SCRUM-27 Generate CSV File | OUTPUT AND REPORTI... | TO DO | - | ^ |
| SCRUM-29 Integrate Archived URLs into Supported Files | OUTPUT AND REPORTI... | TO DO | - | = |
| SCRUM-31 User Manual | DOCUMENTATION & P... | TO DO | - | ⌄ |
| SCRUM-32 Intermediate presentation | DOCUMENTATION & P... | TO DO | - | = |
| SCRUM-33 Final presentation | DOCUMENTATION & P... | TO DO | - | ^ |
| SCRUM-34 UML diagram | DOCUMENTATION & P... | TO DO | - | ⌄ |
| SCRUM-35 Installation manual & script | DOCUMENTATION & P... | TO DO | - | ^ |
| SCRUM-36 Requirements document | DOCUMENTATION & P... | TO DO | - | = |

Figure 1.1: Product Backlog

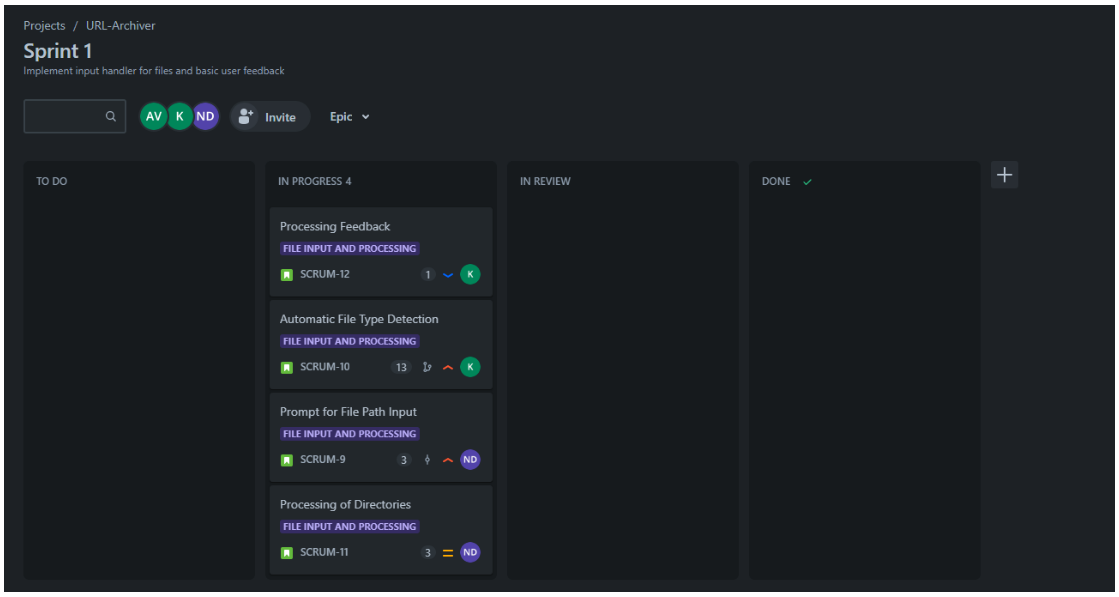
The prioritisation of user stories in the backlog is based on the business value field, which has a value between one and ten. The business value is a vague estimate of how much value the individual user story has to the business, or in our case, to our stakeholders. We endeavour to estimate the business value based on the expected importance of the function to the stakeholder. The following priorities are possible:

- Highest
- High
- Medium
- Low
- Lowest

### Sprint Backlogs

Below we describe our recent and current sprints. During the sprint planning we fill the respective sprint backlog with user stories that serve the sprint goal. A user story must satisfy our Definition of Ready before it can be included in the sprint. In addition, the stories must be estimated and the total number of story points must not exceed our defined velocity.

**Sprint 1** Below is a screenshot of our board from the first sprint with the corresponding sprint goal.



Sprint 1

Implement input handler for files and basic user feedback

TO DO

IN PROGRESS 4

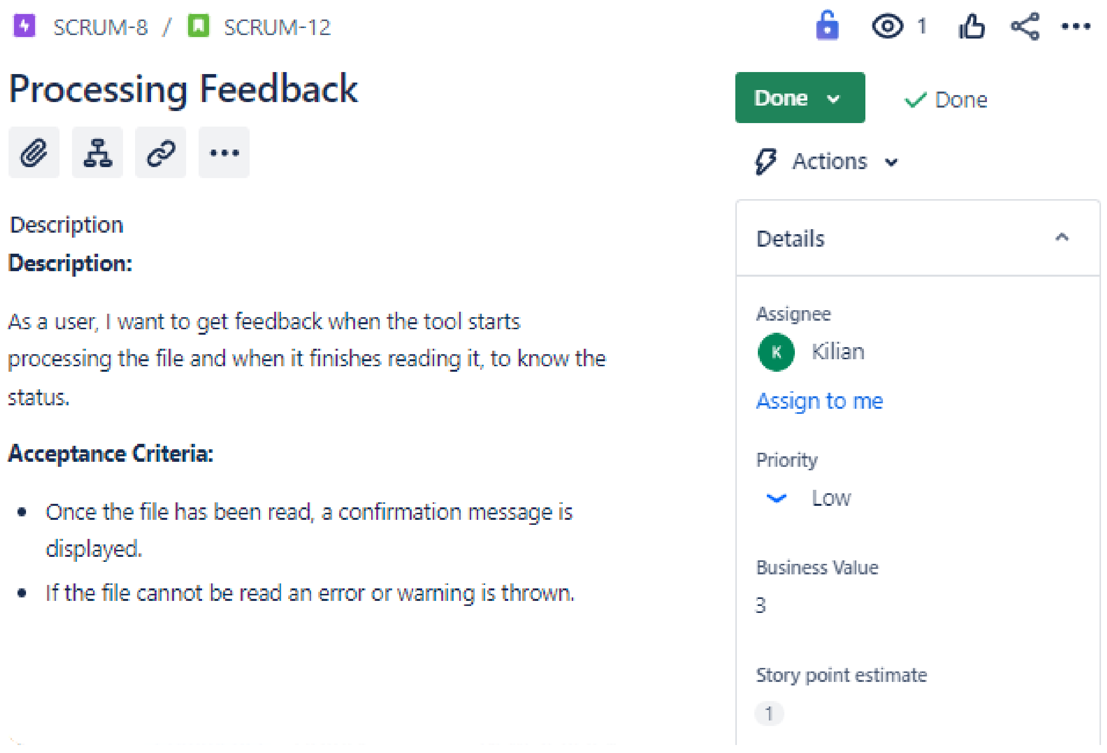
- Processing Feedback — FILE INPUT AND PROCESSING — SCRUM-12 — 1 — K
- Automatic File Type Detection — FILE INPUT AND PROCESSING — SCRUM-10 — 13 — K
- Prompt for File Path Input — FILE INPUT AND PROCESSING — SCRUM-9 — 3 — ND
- Processing of Directories — FILE INPUT AND PROCESSING — SCRUM-11 — 3 — ND

IN REVIEW

DONE ✓

Figure 1.2: Sprint 1 Backlog

The user stories from the first sprint are shown below. The stories have been estimated and prioritised.



Processing Feedback

Description

**Description:**

As a user, I want to get feedback when the tool starts processing the file and when it finishes reading it, to know the status.

**Acceptance Criteria:**

- Once the file has been read, a confirmation message is displayed.
- If the file cannot be read an error or warning is thrown.

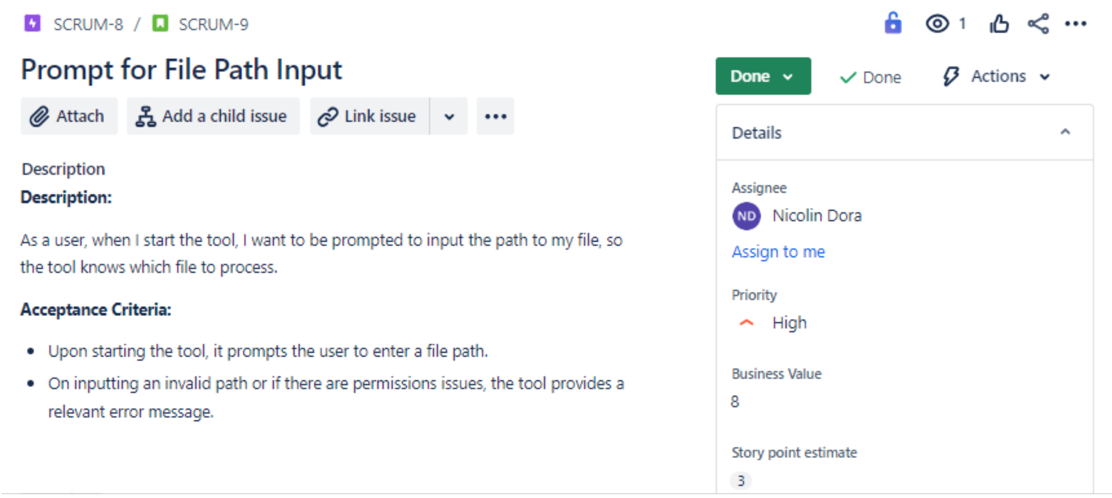
Assignee
Kilian
Assign to me

Priority
Low

Business Value
3

Story point estimate
1

Figure 1.3: User Story Detail for "Processing Feedback"



Prompt for File Path Input

Description

**Description:**

As a user, when I start the tool, I want to be prompted to input the path to my file, so the tool knows which file to process.

**Acceptance Criteria:**

- Upon starting the tool, it prompts the user to enter a file path.
- On inputting an invalid path or if there are permissions issues, the tool provides a relevant error message.

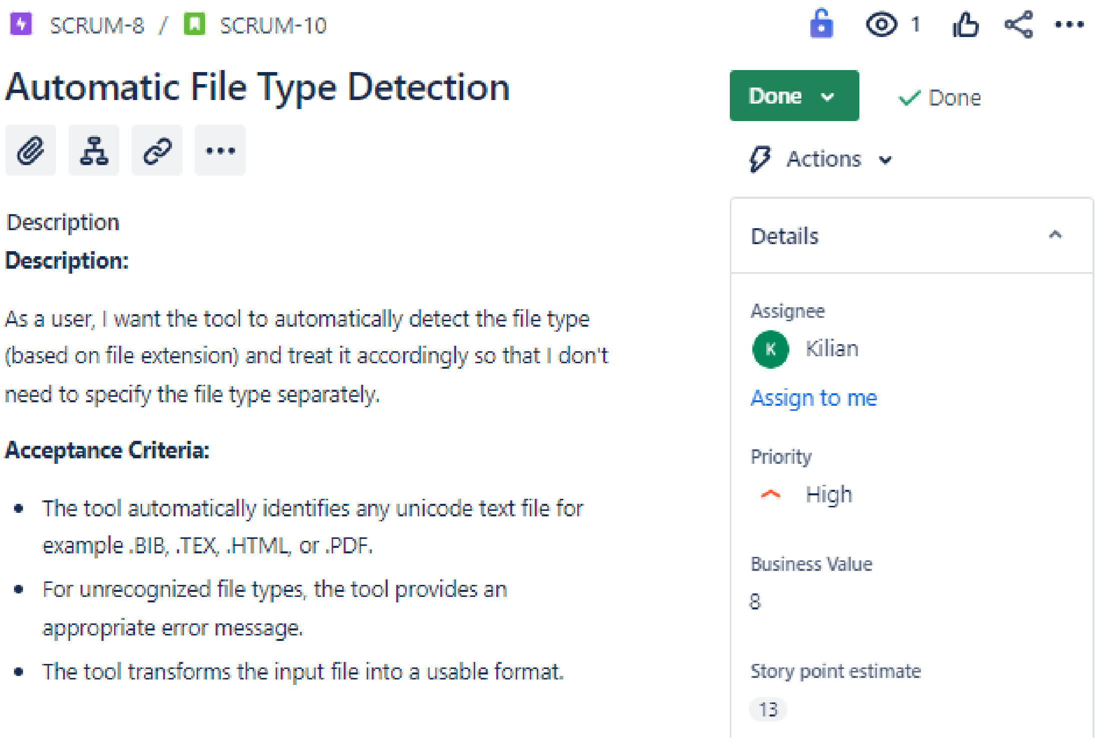
Assignee
Nicolin Dora
Assign to me

Priority
High

Business Value
8

Story point estimate
3

Figure 1.4: User Story Detail for "Prompt for file path input"


SCRUM-8 / SCRUM-10

Automatic File Type Detection

Description

**Description:**

As a user, I want the tool to automatically detect the file type (based on file extension) and treat it accordingly so that I don't need to specify the file type separately.

**Acceptance Criteria:**

- The tool automatically identifies any unicode text file for example .BIB, .TEX, .HTML, or .PDF.
- For unrecognized file types, the tool provides an appropriate error message.
- The tool transforms the input file into a usable format.

Details

Assignee: Kilian

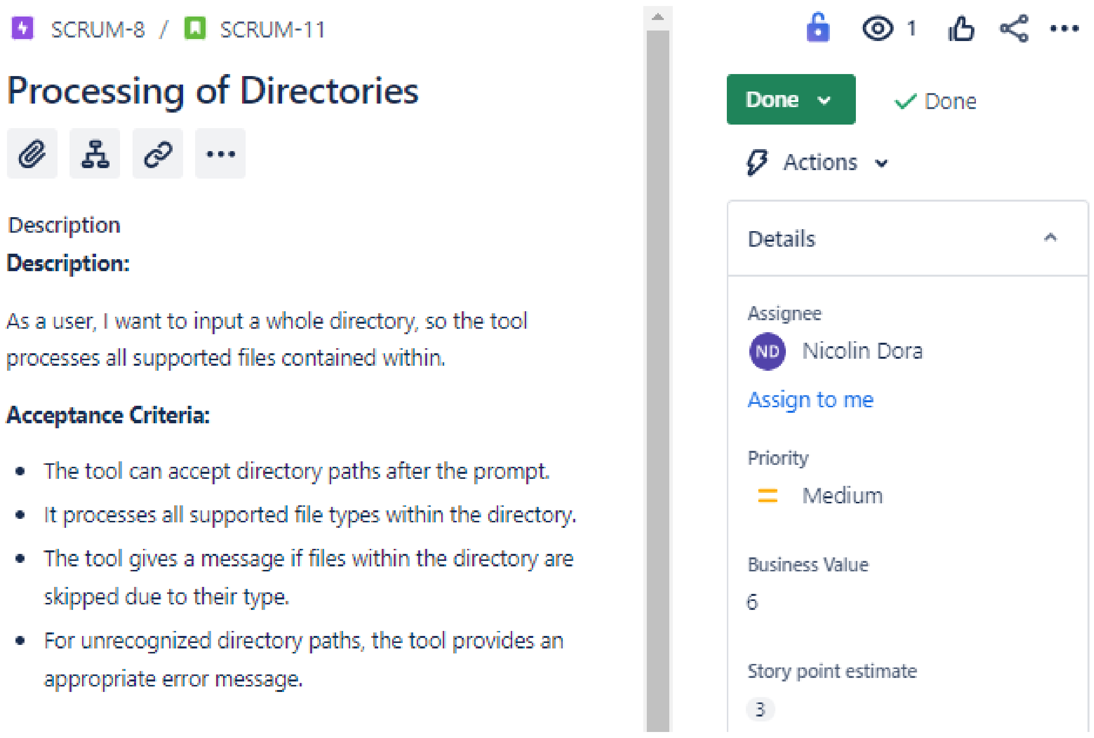
Assign to me

Priority: High

Business Value: 8

Story point estimate: 13

Figure 1.5: User Story Detail for "Automatic File Type Detection"


SCRUM-8 / SCRUM-11

Processing of Directories

Description

**Description:**

As a user, I want to input a whole directory, so the tool processes all supported files contained within.

**Acceptance Criteria:**

- The tool can accept directory paths after the prompt.
- It processes all supported file types within the directory.
- The tool gives a message if files within the directory are skipped due to their type.
- For unrecognized directory paths, the tool provides an appropriate error message.

Assignee: Nicolin Dora

Assign to me

Priority: Medium

Business Value: 6

Story point estimate: 3

Figure 1.6: User Story Detail for "Processing of Directories"

In the first sprint, the burn down chart looks like this:

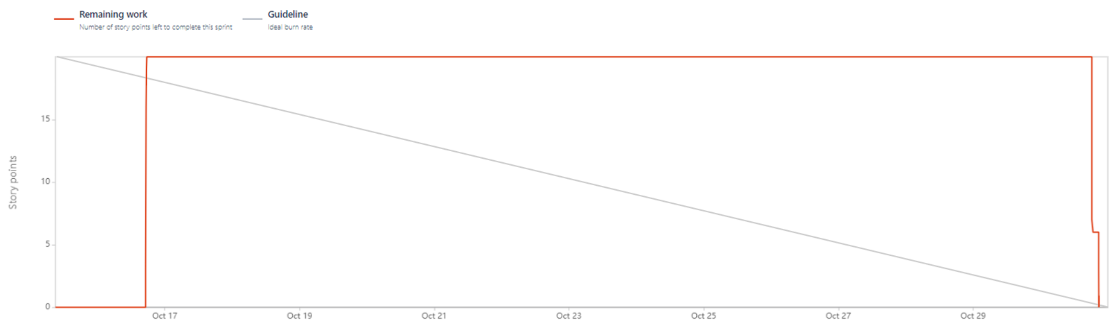
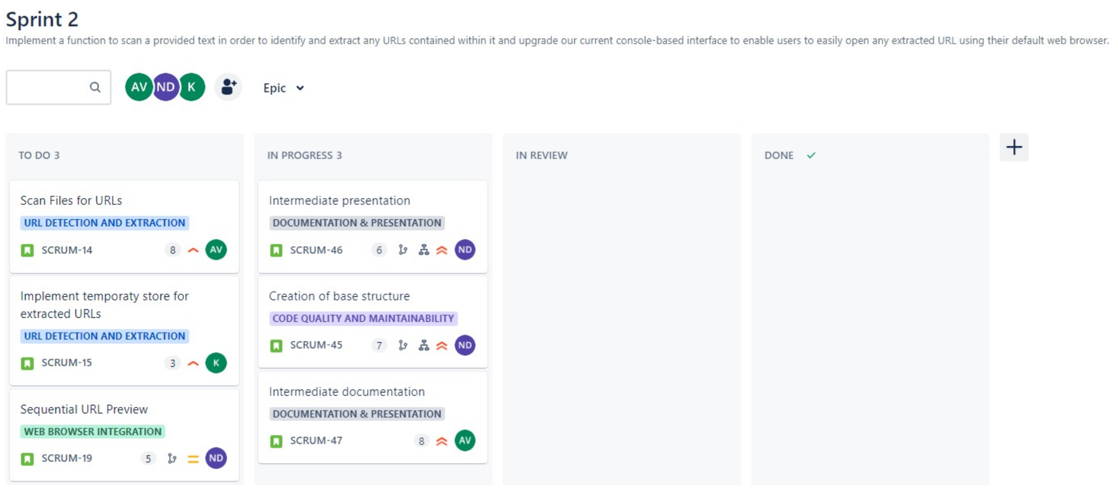


Figure 1.7: Sprint 1 Burn Up Chart

The reason for this is that we did not create tasks for our user stories as they were small enough. Furthermore, a code review for corresponding user stories could only be conducted towards the end of the sprint, resulting in the finalisation of user stories at that point.

**Sprint 2** Below is a screenshot of our board from the second sprint with the corresponding sprint goal.



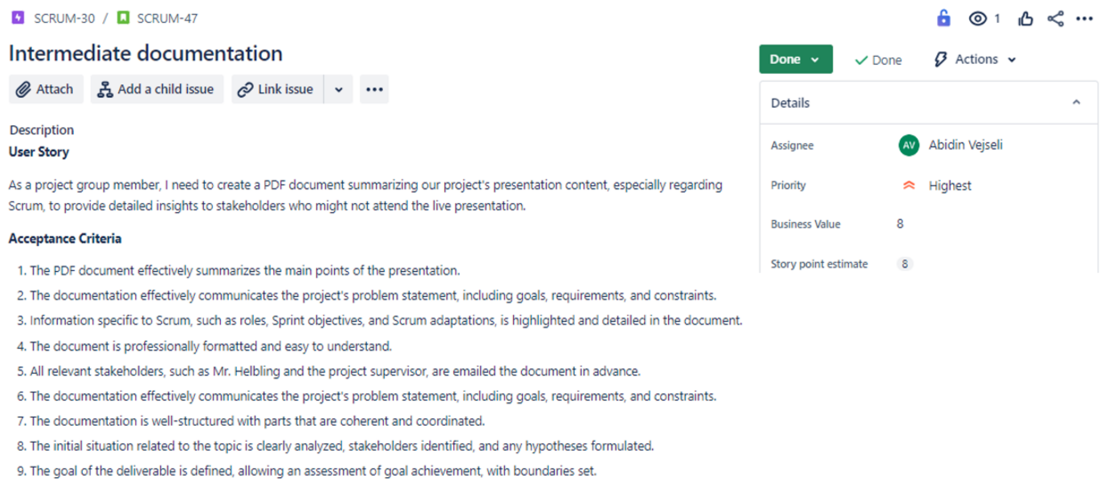
Sprint 2

Implement a function to scan a provided text in order to identify and extract any URLs contained within it and upgrade our current console-based interface to enable users to easily open any extracted URL using their default web browser.

| TO DO 3 | IN PROGRESS 3 | IN REVIEW | DONE ✓ |
|---|---|---|---|
| Scan Files for URLs — URL DETECTION AND EXTRACTION — SCRUM-14 — 8 — AV | Intermediate presentation — DOCUMENTATION & PRESENTATION — SCRUM-46 — 6 — ND | | |
| Implement temporaty store for extracted URLs — URL DETECTION AND EXTRACTION — SCRUM-15 — 3 — K | Creation of base structure — CODE QUALITY AND MAINTAINABILITY — SCRUM-45 — 7 — ND | | |
| Sequential URL Preview — WEB BROWSER INTEGRATION — SCRUM-19 — 5 — ND | Intermediate documentation — DOCUMENTATION & PRESENTATION — SCRUM-47 — 8 — AV | | |

Figure 1.8: Sprint 2 Backlog

The user stories from the second sprint are shown below. The stories have been estimated and prioritised.



SCRUM-30 / SCRUM-47

Intermediate documentation

Description

**User Story**

As a project group member, I need to create a PDF document summarizing our project's presentation content, especially regarding Scrum, to provide detailed insights to stakeholders who might not attend the live presentation.

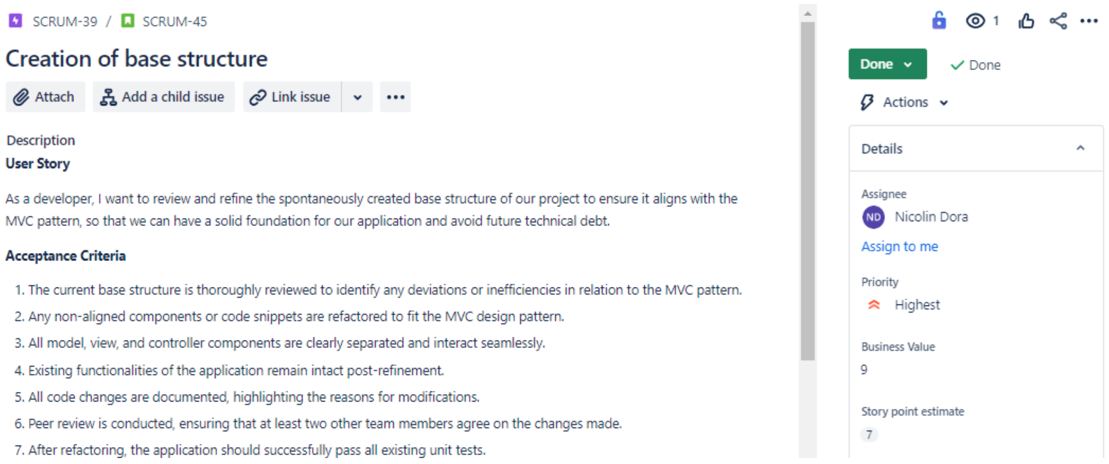
**Acceptance Criteria**

1. The PDF document effectively summarizes the main points of the presentation.
2. The documentation effectively communicates the project's problem statement, including goals, requirements, and constraints.
3. Information specific to Scrum, such as roles, Sprint objectives, and Scrum adaptations, is highlighted and detailed in the document.
4. The document is professionally formatted and easy to understand.
5. All relevant stakeholders, such as Mr. Helbling and the project supervisor, are emailed the document in advance.
6. The documentation effectively communicates the project's problem statement, including goals, requirements, and constraints.
7. The documentation is well-structured with parts that are coherent and coordinated.
8. The initial situation related to the topic is clearly analyzed, stakeholders identified, and any hypotheses formulated.
9. The goal of the deliverable is defined, allowing an assessment of goal achievement, with boundaries set.

Details

| | |
|---|---|
| Assignee | AV Abidin Vejseli |
| Priority | Highest |
| Business Value | 8 |
| Story point estimate | 8 |

Figure 1.9: User Story Detail for "Intermediate Documentation"



**Creation of base structure**

Description

**User Story**

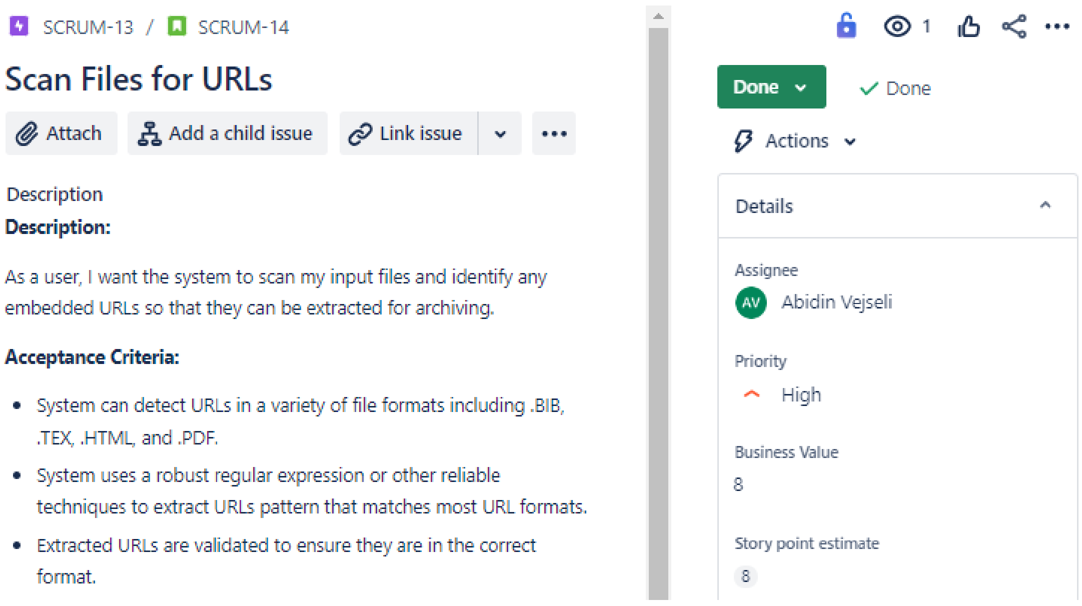
As a developer, I want to review and refine the spontaneously created base structure of our project to ensure it aligns with the MVC pattern, so that we can have a solid foundation for our application and avoid future technical debt.

**Acceptance Criteria**

1. The current base structure is thoroughly reviewed to identify any deviations or inefficiencies in relation to the MVC pattern.
2. Any non-aligned components or code snippets are refactored to fit the MVC design pattern.
3. All model, view, and controller components are clearly separated and interact seamlessly.
4. Existing functionalities of the application remain intact post-refinement.
5. All code changes are documented, highlighting the reasons for modifications.
6. Peer review is conducted, ensuring that at least two other team members agree on the changes made.
7. After refactoring, the application should successfully pass all existing unit tests.

Assignee: Nicolin Dora
Assign to me
Priority: Highest
Business Value: 9
Story point estimate: 7

Figure 1.10: User Story Detail for "Creation of Base Structure"



**Scan Files for URLs**

Description

**Description:**

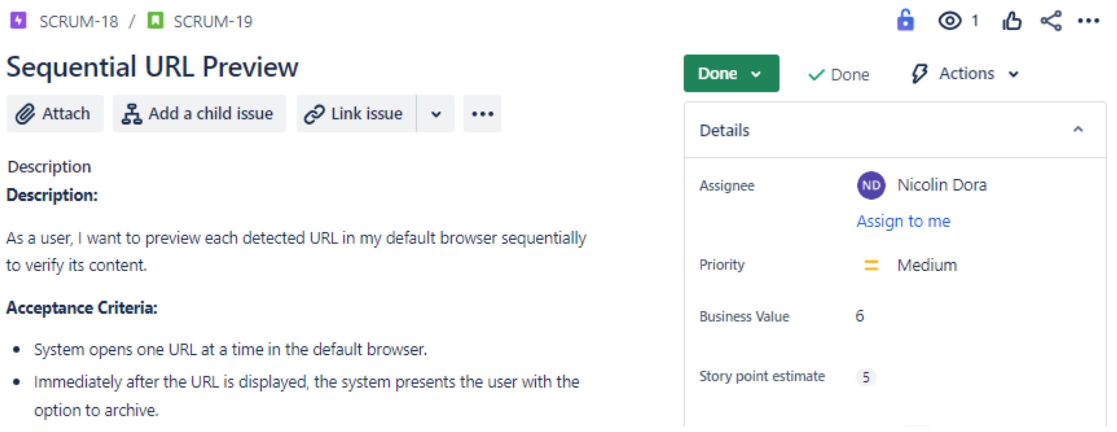
As a user, I want the system to scan my input files and identify any embedded URLs so that they can be extracted for archiving.

**Acceptance Criteria:**

- System can detect URLs in a variety of file formats including .BIB, .TEX, .HTML, and .PDF.
- System uses a robust regular expression or other reliable techniques to extract URLs pattern that matches most URL formats.
- Extracted URLs are validated to ensure they are in the correct format.

Assignee: Abidin Vejseli
Priority: High
Business Value: 8
Story point estimate: 8

Figure 1.11: User Story Detail for "Scan Files for URLs"


SCRUM-18 / SCRUM-19

Sequential URL Preview

Description

**Description:**

As a user, I want to preview each detected URL in my default browser sequentially to verify its content.

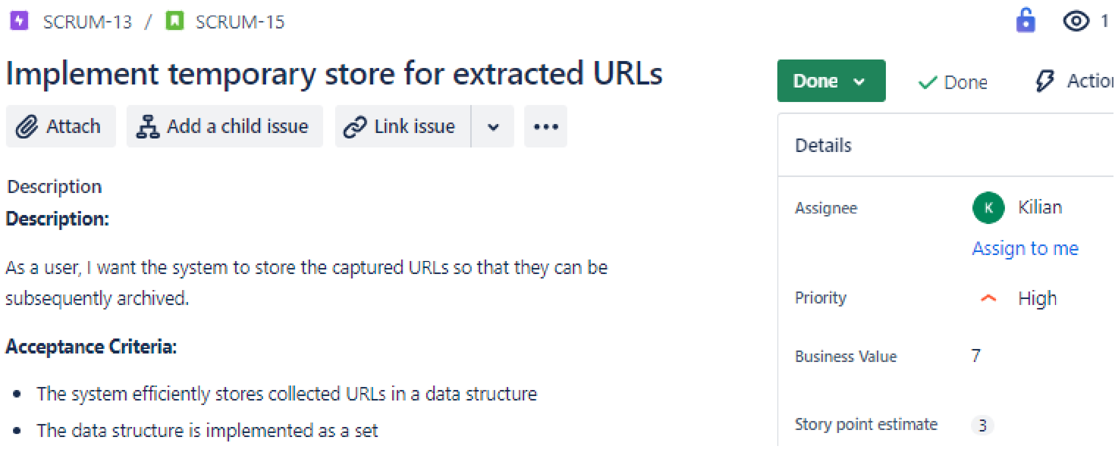
**Acceptance Criteria:**

- System opens one URL at a time in the default browser.
- Immediately after the URL is displayed, the system presents the user with the option to archive.

Done ✓ Done

| Details | |
|---|---|
| Assignee | Nicolin Dora |
| Priority | Medium |
| Business Value | 6 |
| Story point estimate | 5 |

Figure 1.12: User Story Detail for "Sequential URL Preview"


SCRUM-13 / SCRUM-15

Implement temporary store for extracted URLs

Description

**Description:**

As a user, I want the system to store the captured URLs so that they can be subsequently archived.

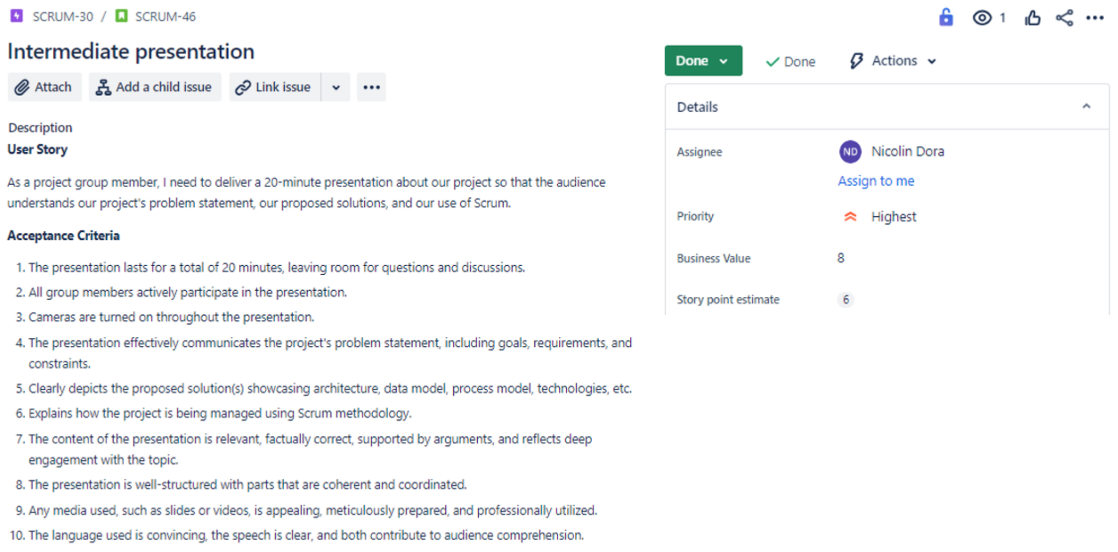
**Acceptance Criteria:**

- The system efficiently stores collected URLs in a data structure
- The data structure is implemented as a set

Done ✓ Done

| Details | |
|---|---|
| Assignee | Kilian |
| Priority | High |
| Business Value | 7 |
| Story point estimate | 3 |

Figure 1.13: User Story Detail for "Implement temporary store for extracted URLs"


SCRUM-30 / SCRUM-46

Intermediate presentation

Description

**User Story**

As a project group member, I need to deliver a 20-minute presentation about our project so that the audience understands our project's problem statement, our proposed solutions, and our use of Scrum.

**Acceptance Criteria**

1. The presentation lasts for a total of 20 minutes, leaving room for questions and discussions.
2. All group members actively participate in the presentation.
3. Cameras are turned on throughout the presentation.
4. The presentation effectively communicates the project's problem statement, including goals, requirements, and constraints.
5. Clearly depicts the proposed solution(s) showcasing architecture, data model, process model, technologies, etc.
6. Explains how the project is being managed using Scrum methodology.
7. The content of the presentation is relevant, factually correct, supported by arguments, and reflects deep engagement with the topic.
8. The presentation is well-structured with parts that are coherent and coordinated.
9. Any media used, such as slides or videos, is appealing, meticulously prepared, and professionally utilized.
10. The language used is convincing, the speech is clear, and both contribute to audience comprehension.

| | |
|---|---|
| Assignee | Nicolin Dora<br>Assign to me |
| Priority | Highest |
| Business Value | 8 |
| Story point estimate | 6 |

Figure 1.14: User Story Detail for "Intermediate presentation"

In the second sprint, the burn down chart looks like this:

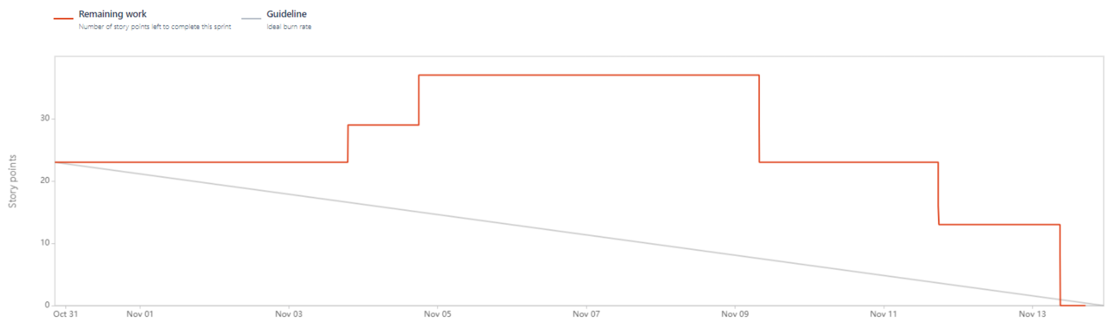
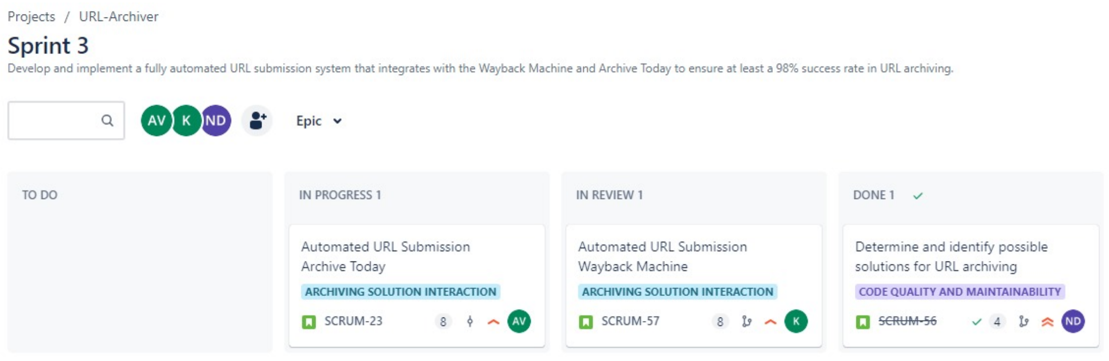


Figure 1.15: Sprint 2 Burn Down Chart

Compared to the initial sprint, our workload significantly increased, and we introduced new user stories after the sprint began. Despite these additions, we maintained a strong pace and seamlessly managed the extra tasks that arose from the intermediate presentation and documentation requirements.

**Sprint 3** Below is a screenshot of our board from the third sprint with the corresponding sprint goal.



Projects / URL-Archiver

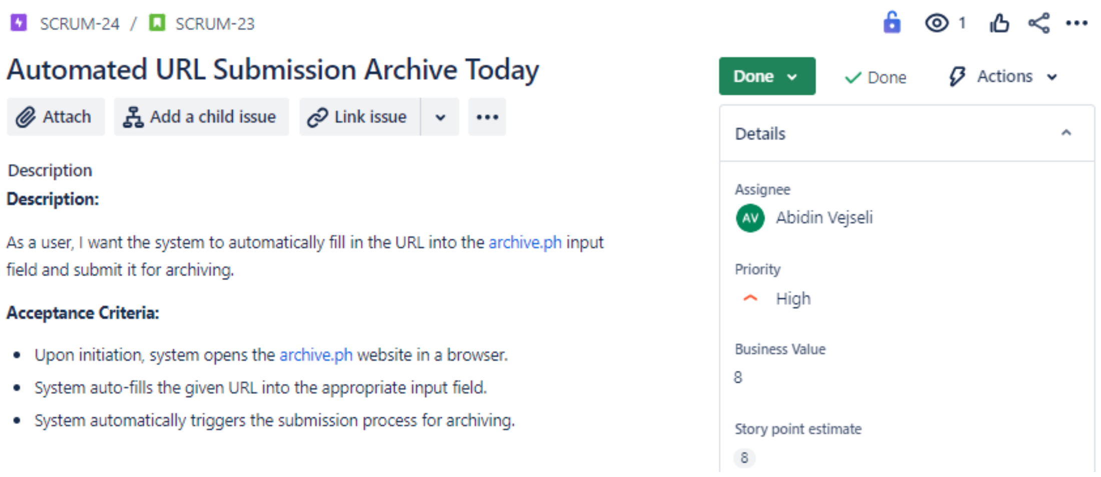
Sprint 3

Develop and implement a fully automated URL submission system that integrates with the Wayback Machine and Archive Today to ensure at least a 98% success rate in URL archiving.

AV
K
ND
Epic
TO DO
IN PROGRESS 1
Automated URL Submission Archive Today
ARCHIVING SOLUTION INTERACTION
SCRUM-23
8
AV
IN REVIEW 1
Automated URL Submission Wayback Machine
ARCHIVING SOLUTION INTERACTION
SCRUM-57
8
K
DONE 1
Determine and identify possible solutions for URL archiving
CODE QUALITY AND MAINTAINABILITY
~~SCRUM-56~~
4
ND

Figure 1.16: Sprint 3 Backlog

The user stories from the third sprint are shown below. The stories have been estimated and prioritised.



Automated URL Submission Archive Today

Description

**Description:**

As a user, I want the system to automatically fill in the URL into the archive.ph input field and submit it for archiving.

**Acceptance Criteria:**

- Upon initiation, system opens the archive.ph website in a browser.
- System auto-fills the given URL into the appropriate input field.
- System automatically triggers the submission process for archiving.

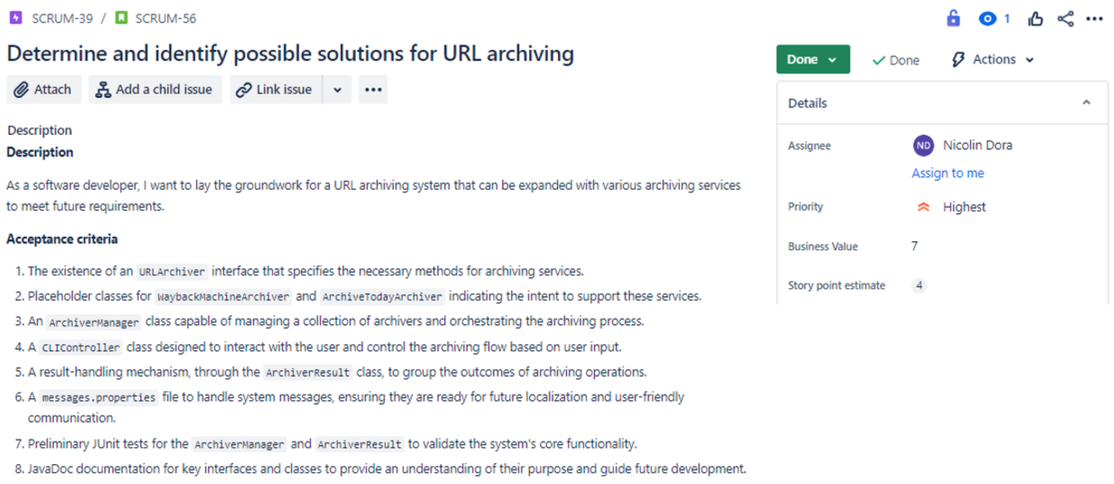
Assignee: Abidin Vejseli

Priority: High

Business Value: 8

Story point estimate: 8

Figure 1.17: User Story Detail for "Intermediate Documentation"



Determine and identify possible solutions for URL archiving

Description

**Description**

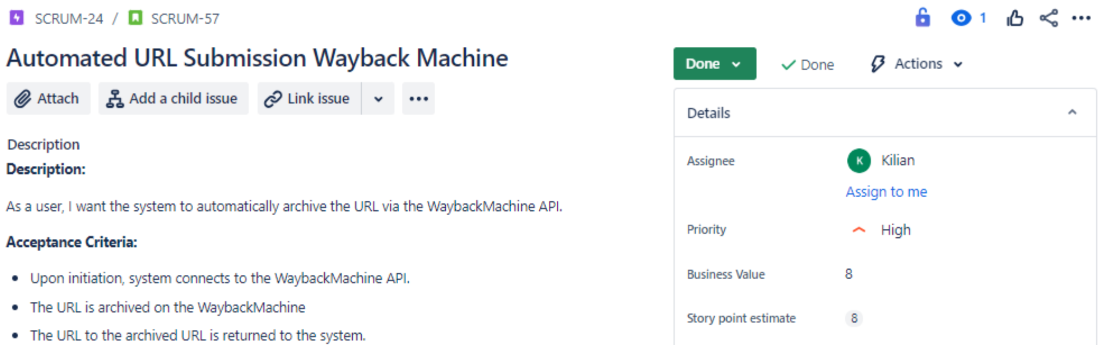
As a software developer, I want to lay the groundwork for a URL archiving system that can be expanded with various archiving services to meet future requirements.

**Acceptance criteria**

1. The existence of an `URLArchiver` interface that specifies the necessary methods for archiving services.
2. Placeholder classes for `WaybackMachineArchiver` and `ArchiveTodayArchiver` indicating the intent to support these services.
3. An `ArchiverManager` class capable of managing a collection of archivers and orchestrating the archiving process.
4. A `CLIController` class designed to interact with the user and control the archiving flow based on user input.
5. A result-handling mechanism, through the `ArchiverResult` class, to group the outcomes of archiving operations.
6. A `messages.properties` file to handle system messages, ensuring they are ready for future localization and user-friendly communication.
7. Preliminary JUnit tests for the `ArchiverManager` and `ArchiverResult` to validate the system's core functionality.
8. JavaDoc documentation for key interfaces and classes to provide an understanding of their purpose and guide future development.

| | |
|---|---|
| Assignee | Nicolin Dora<br>Assign to me |
| Priority | Highest |
| Business Value | 7 |
| Story point estimate | 4 |

Figure 1.18: User Story Detail for "Creation of Base Structure"



Automated URL Submission Wayback Machine

Description

**Description:**

As a user, I want the system to automatically archive the URL via the WaybackMachine API.

**Acceptance Criteria:**

- Upon initiation, system connects to the WaybackMachine API.
- The URL is archived on the WaybackMachine
- The URL to the archived URL is returned to the system.

| | |
|---|---|
| Assignee | Kilian<br>Assign to me |
| Priority | High |
| Business Value | 8 |
| Story point estimate | 8 |

Figure 1.19: User Story Detail for "Scan Files for URLs"

In the third sprint, the burn down chart looks like this:

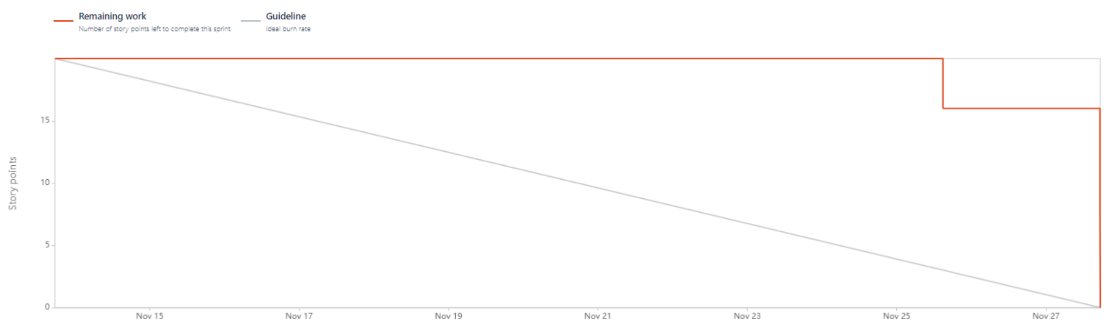
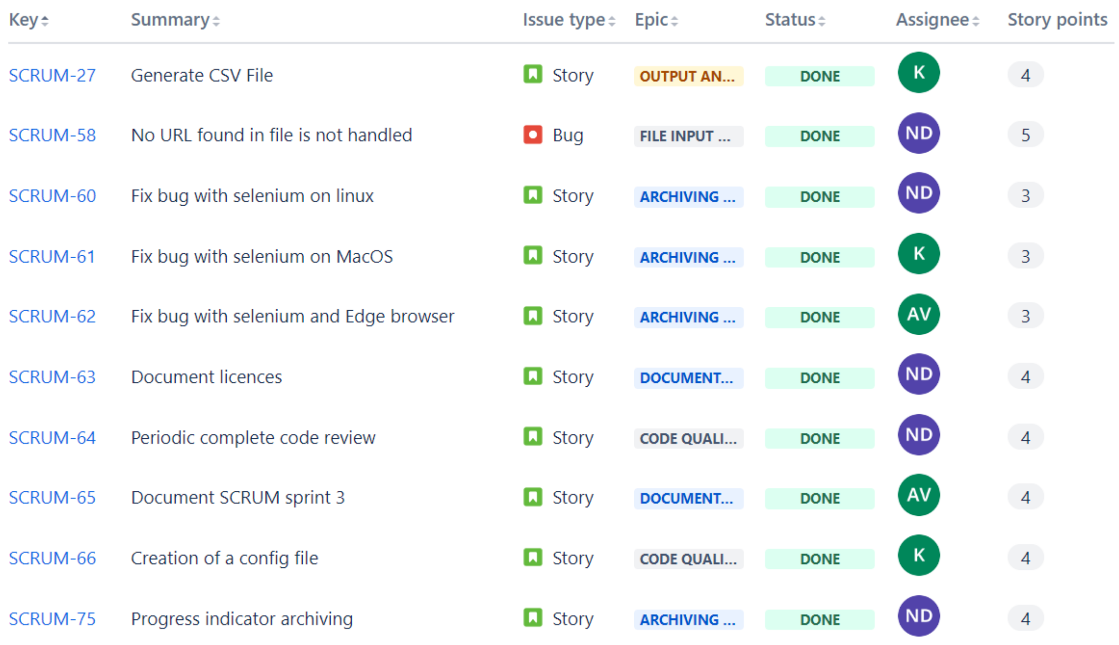


Figure 1.20: Sprint 3 Burn Down Chart

During the third sprint, our progress on user stories was limited to the completion of only three due to the ’special week 3’. This meant that these stories had to be completed in the second half of the sprint due to the heavy workload of this special week. The impact of this adaptation to our sprint schedule due to ’special week 3’ is reflected in the trends seen in our burndown chart.

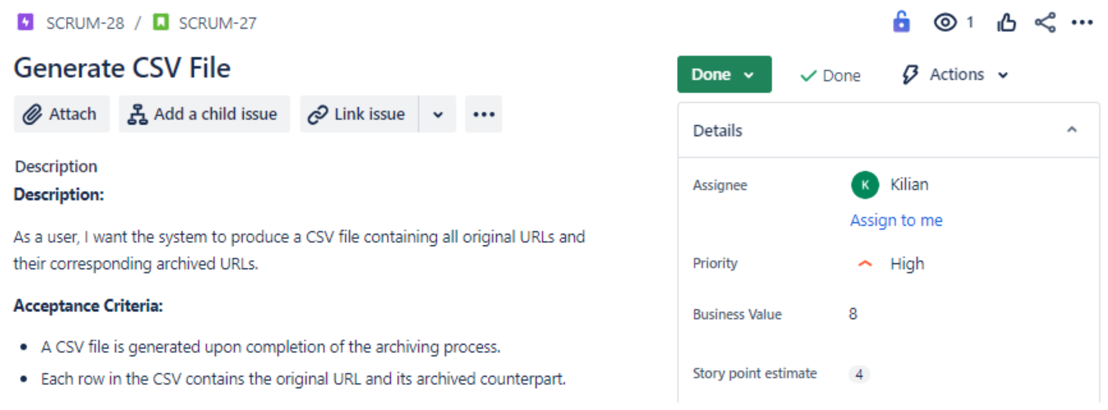
**Sprint 4** Below is a screenshot of our board from the forth sprint.



| Key | Summary | Issue type | Epic | Status | Assignee | Story points |
|---|---|---|---|---|---|---|
| SCRUM-27 | Generate CSV File | Story | OUTPUT AN... | DONE | K | 4 |
| SCRUM-58 | No URL found in file is not handled | Bug | FILE INPUT ... | DONE | ND | 5 |
| SCRUM-60 | Fix bug with selenium on linux | Story | ARCHIVING ... | DONE | ND | 3 |
| SCRUM-61 | Fix bug with selenium on MacOS | Story | ARCHIVING ... | DONE | K | 3 |
| SCRUM-62 | Fix bug with selenium and Edge browser | Story | ARCHIVING ... | DONE | AV | 3 |
| SCRUM-63 | Document licences | Story | DOCUMENT... | DONE | ND | 4 |
| SCRUM-64 | Periodic complete code review | Story | CODE QUALI... | DONE | ND | 4 |
| SCRUM-65 | Document SCRUM sprint 3 | Story | DOCUMENT... | DONE | AV | 4 |
| SCRUM-66 | Creation of a config file | Story | CODE QUALI... | DONE | K | 4 |
| SCRUM-75 | Progress indicator archiving | Story | ARCHIVING ... | DONE | ND | 4 |

Figure 1.21: Sprint 4 Backlog

The user stories from the forth sprint are shown below. The stories have been estimated and prioritised.



## Generate CSV File

Description

**Description:**

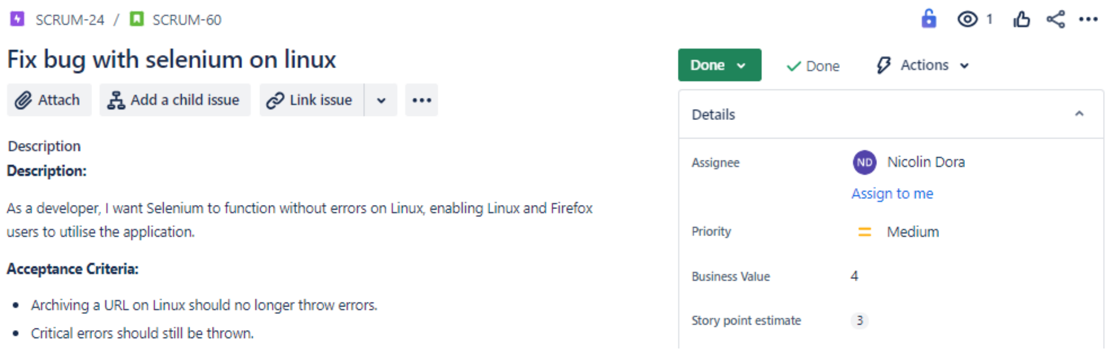
As a user, I want the system to produce a CSV file containing all original URLs and their corresponding archived URLs.

**Acceptance Criteria:**

- A CSV file is generated upon completion of the archiving process.
- Each row in the CSV contains the original URL and its archived counterpart.

| Details | |
|---|---|
| Assignee | Kilian |
| Priority | High |
| Business Value | 8 |
| Story point estimate | 4 |

Figure 1.22: User Story Detail for "Generate CSV File"


SCRUM-24 / SCRUM-60

Fix bug with selenium on linux

Description

**Description:**

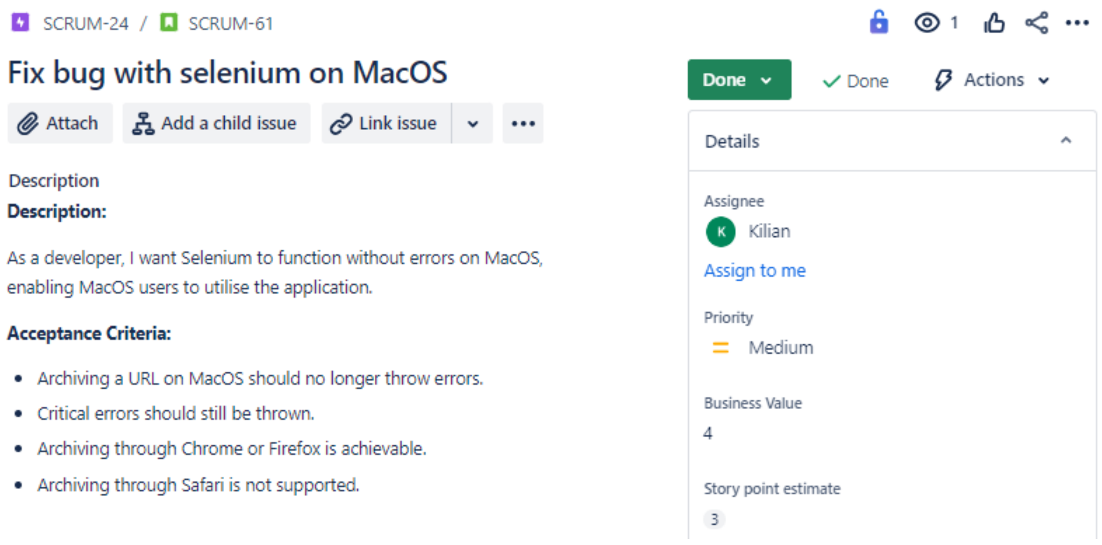
As a developer, I want Selenium to function without errors on Linux, enabling Linux and Firefox users to utilise the application.

**Acceptance Criteria:**

- Archiving a URL on Linux should no longer throw errors.
- Critical errors should still be thrown.

| | |
|---|---|
| Assignee | Nicolin Dora |
| | Assign to me |
| Priority | Medium |
| Business Value | 4 |
| Story point estimate | 3 |

Figure 1.23: User Story Detail for "Fix bug with selenium on linux"


SCRUM-24 / SCRUM-61

Fix bug with selenium on MacOS

Description

**Description:**

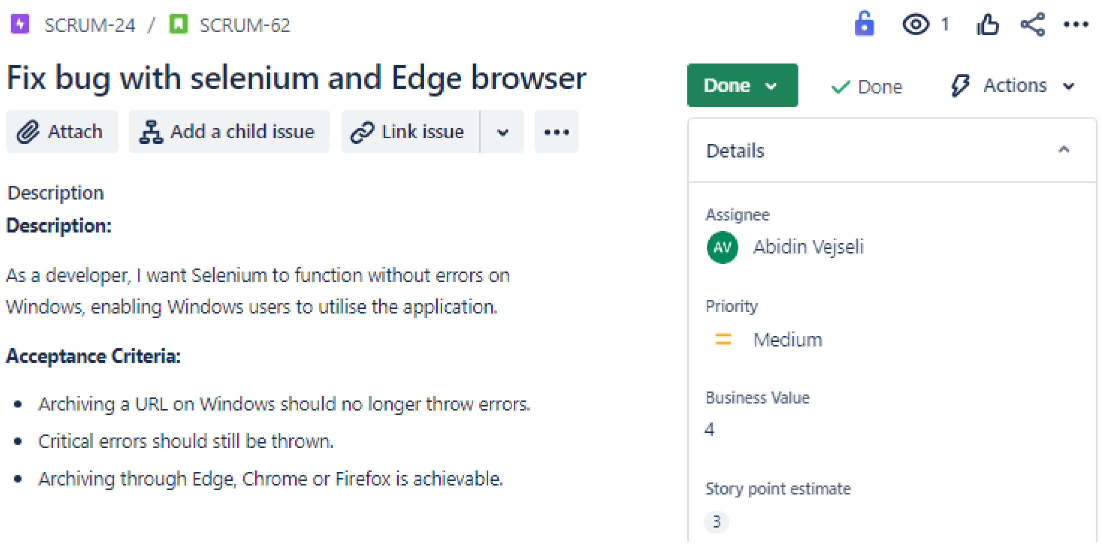
As a developer, I want Selenium to function without errors on MacOS, enabling MacOS users to utilise the application.

**Acceptance Criteria:**

- Archiving a URL on MacOS should no longer throw errors.
- Critical errors should still be thrown.
- Archiving through Chrome or Firefox is achievable.
- Archiving through Safari is not supported.

| | |
|---|---|
| Assignee | Kilian |
| | Assign to me |
| Priority | Medium |
| Business Value | 4 |
| Story point estimate | 3 |

Figure 1.24: User Story Detail for "Fix bug with selenium on MacOS"


SCRUM-24 / SCRUM-62

Fix bug with selenium and Edge browser

Description
**Description:**

As a developer, I want Selenium to function without errors on Windows, enabling Windows users to utilise the application.

**Acceptance Criteria:**

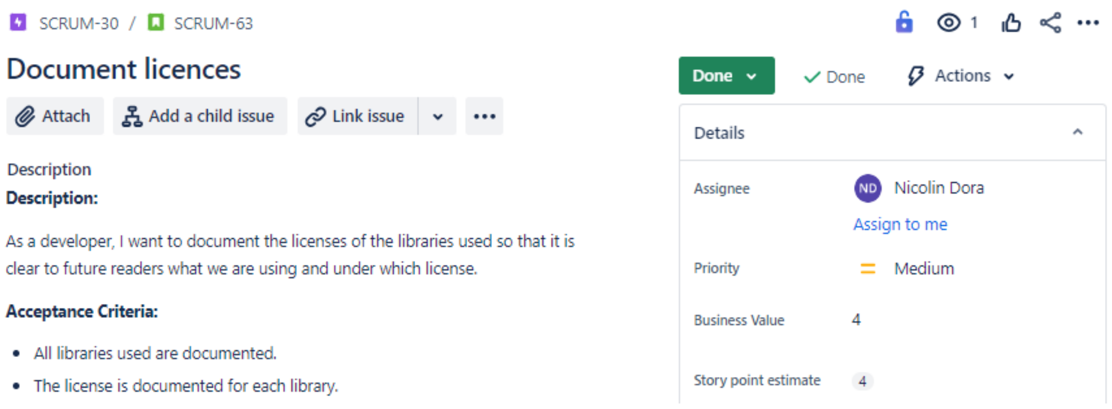
- Archiving a URL on Windows should no longer throw errors.
- Critical errors should still be thrown.
- Archiving through Edge, Chrome or Firefox is achievable.

Done

Details

| | |
|---|---|
| Assignee | Abidin Vejseli |
| Priority | Medium |
| Business Value | 4 |
| Story point estimate | 3 |

Figure 1.25: User Story Detail for "Fix bug With selenium and Edge browser"


SCRUM-30 / SCRUM-63

Document licences

Description
**Description:**

As a developer, I want to document the licenses of the libraries used so that it is clear to future readers what we are using and under which license.

**Acceptance Criteria:**

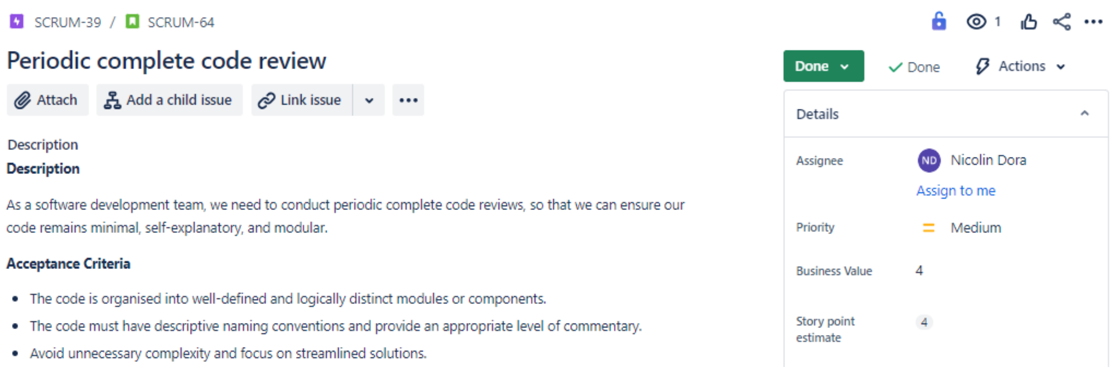
- All libraries used are documented.
- The license is documented for each library.

Done

Details

| | |
|---|---|
| Assignee | Nicolin Dora |
| Priority | Medium |
| Business Value | 4 |
| Story point estimate | 4 |

Figure 1.26: User Story Detail for "Document licences"


SCRUM-39 / SCRUM-64

Periodic complete code review

Description
**Description**

As a software development team, we need to conduct periodic complete code reviews, so that we can ensure our code remains minimal, self-explanatory, and modular.

**Acceptance Criteria**

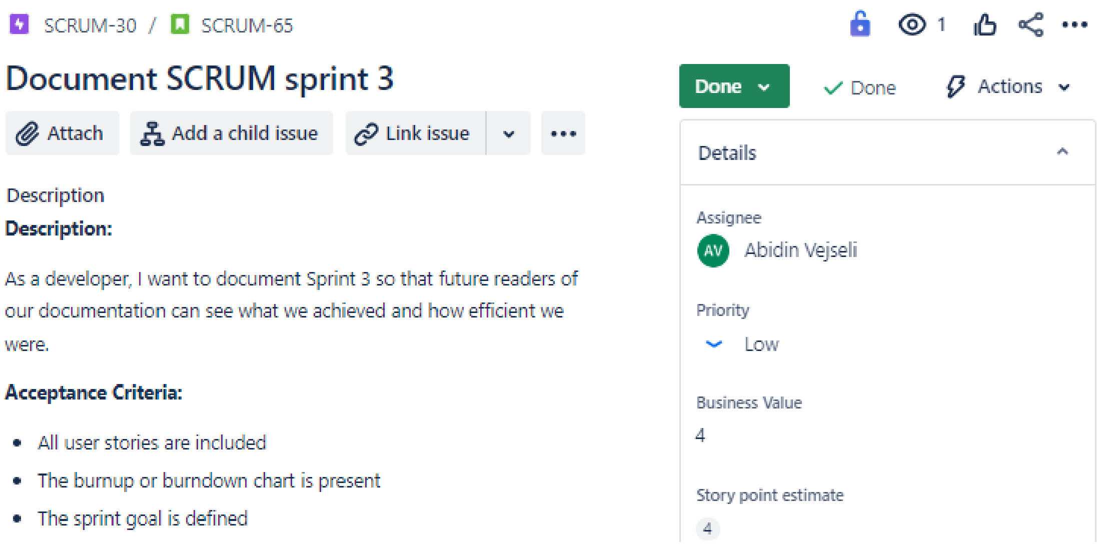
- The code is organised into well-defined and logically distinct modules or components.
- The code must have descriptive naming conventions and provide an appropriate level of commentary.
- Avoid unnecessary complexity and focus on streamlined solutions.

Done

Details

| | |
|---|---|
| Assignee | Nicolin Dora |
| Priority | Medium |
| Business Value | 4 |
| Story point estimate | 4 |

Figure 1.27: User Story Detail for "Periodic complete code review"


SCRUM-30 / SCRUM-65

Document SCRUM sprint 3

Description

**Description:**

As a developer, I want to document Sprint 3 so that future readers of our documentation can see what we achieved and how efficient we were.

**Acceptance Criteria:**

- All user stories are included
- The burnup or burndown chart is present
- The sprint goal is defined

Details

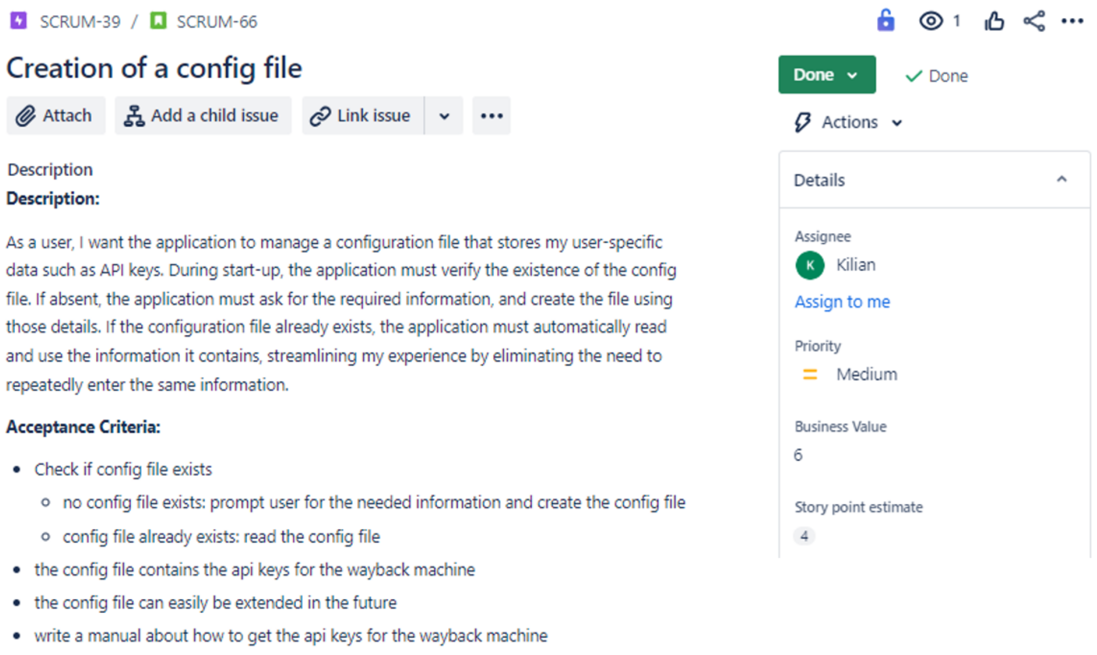
Assignee: AV Abidin Vejseli

Priority: Low

Business Value: 4

Story point estimate: 4

Figure 1.28: User Story Detail for "Document SCRUM sprint 3"


SCRUM-39 / SCRUM-66

Creation of a config file

Description

**Description:**

As a user, I want the application to manage a configuration file that stores my user-specific data such as API keys. During start-up, the application must verify the existence of the config file. If absent, the application must ask for the required information, and create the file using those details. If the configuration file already exists, the application must automatically read and use the information it contains, streamlining my experience by eliminating the need to repeatedly enter the same information.

**Acceptance Criteria:**

- Check if config file exists
  - no config file exists: prompt user for the needed information and create the config file
  - config file already exists: read the config file
- the config file contains the api keys for the wayback machine
- the config file can easily be extended in the future
- write a manual about how to get the api keys for the wayback machine

Details

Assignee: K Kilian

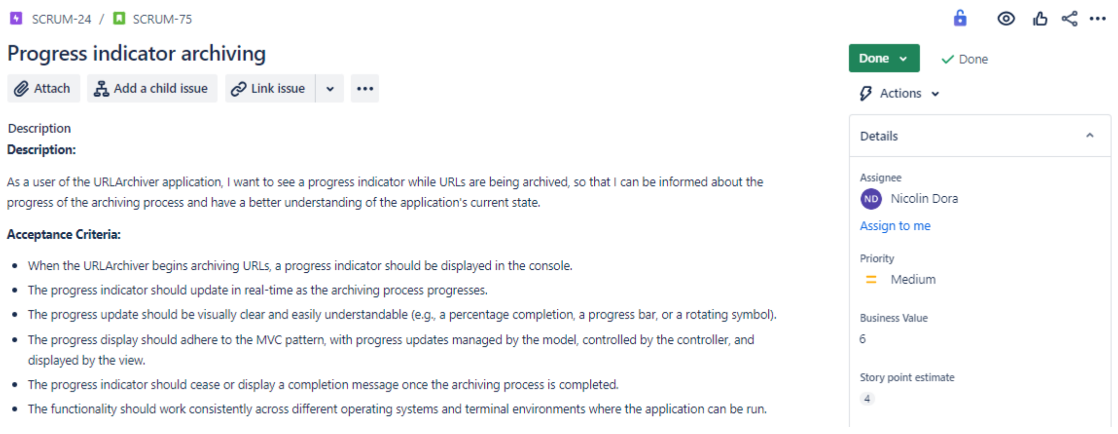
Assign to me

Priority: Medium

Business Value: 6

Story point estimate: 4

Figure 1.29: User Story Detail for "Creation of a config file"



Progress indicator archiving

Description

**Description:**

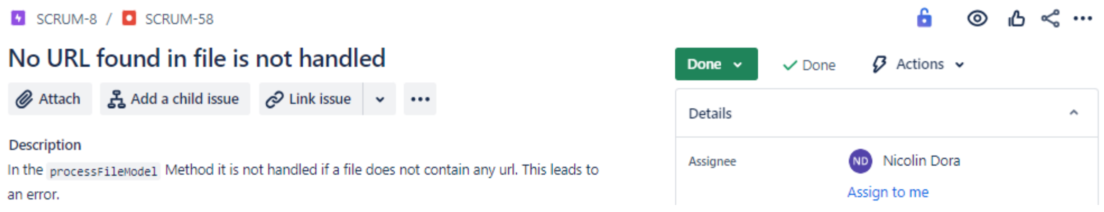
As a user of the URLArchiver application, I want to see a progress indicator while URLs are being archived, so that I can be informed about the progress of the archiving process and have a better understanding of the application's current state.

**Acceptance Criteria:**

- When the URLArchiver begins archiving URLs, a progress indicator should be displayed in the console.
- The progress indicator should update in real-time as the archiving process progresses.
- The progress update should be visually clear and easily understandable (e.g., a percentage completion, a progress bar, or a rotating symbol).
- The progress display should adhere to the MVC pattern, with progress updates managed by the model, controlled by the controller, and displayed by the view.
- The progress indicator should cease or display a completion message once the archiving process is completed.
- The functionality should work consistently across different operating systems and terminal environments where the application can be run.

| | |
|---|---|
| Assignee | Nicolin Dora |
| | Assign to me |
| Priority | Medium |
| Business Value | 6 |
| Story point estimate | 4 |

Figure 1.30: User Story Detail for "Progress indicator archiving"



No URL found in file is not handled

Description

In the `processFileModel` Method it is not handled if a file does not contain any url. This leads to an error.

| | |
|---|---|
| Assignee | Nicolin Dora |
| | Assign to me |

Figure 1.31: Bug Detail for "No URL found in file is not handled"

In the forth sprint, the burn down chart looks like this:

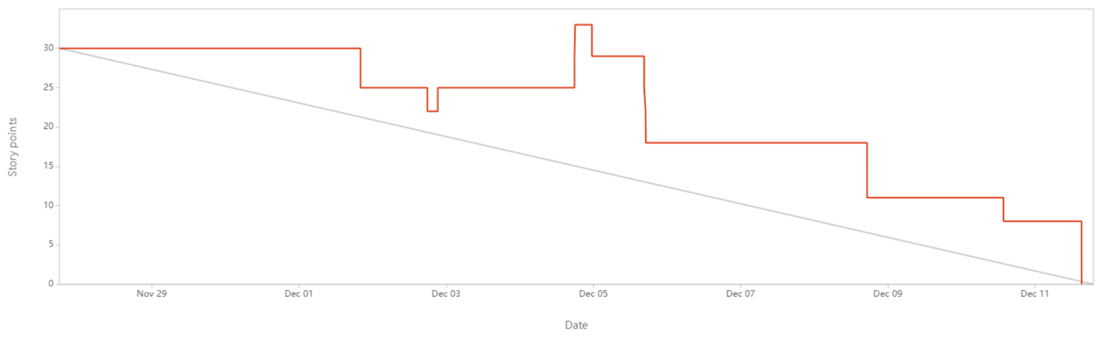
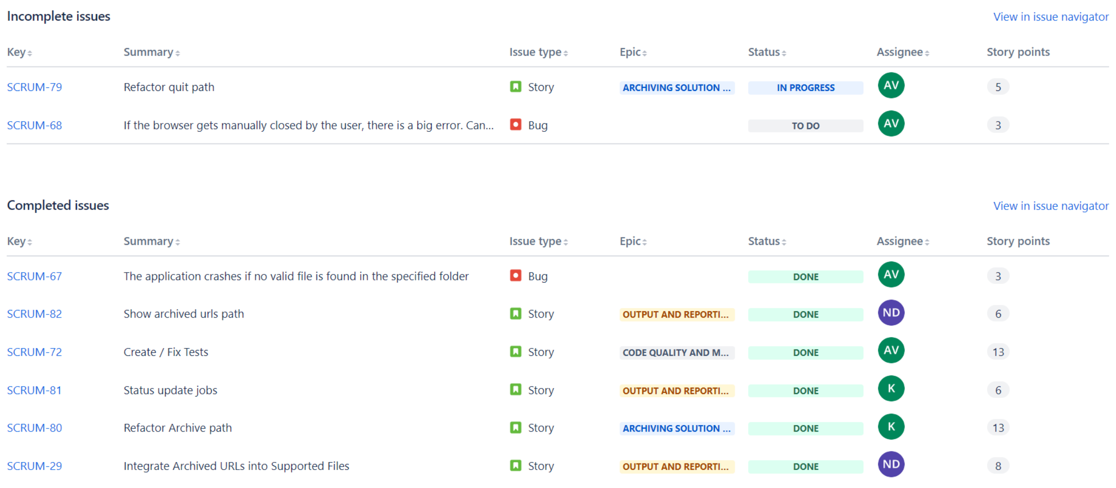


**Figure 1.32:** Sprint 4 Burn Down Chart

Compared to the previous sprint, we were more efficient and successfully accommodated additional user stories that were initiated after the sprint began. The completion of all allocated user stories within the sprint timeframe demonstrates our solid teamwork and sprint management capabilities.

**Sprint 5** Below is a screenshot of our board from the fifth sprint.



Incomplete issues — View in issue navigator

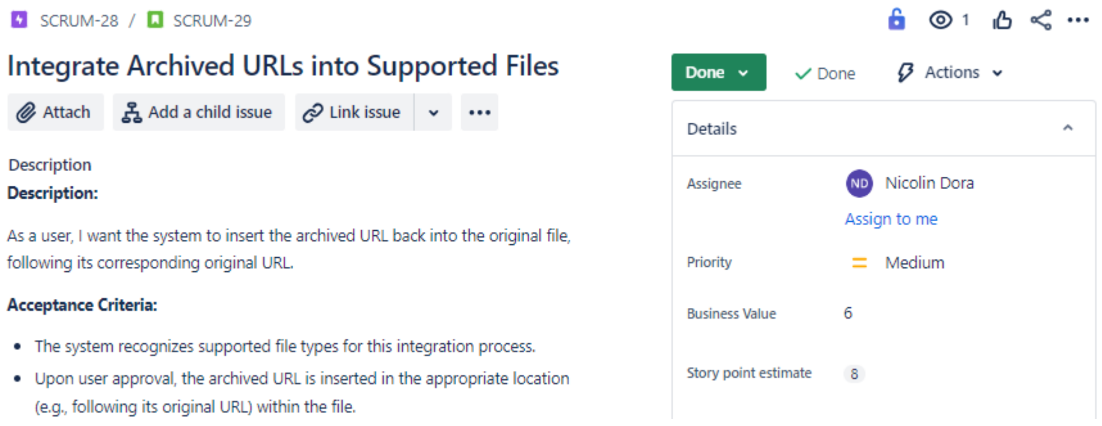
| Key | Summary | Issue type | Epic | Status | Assignee | Story points |
|---|---|---|---|---|---|---|
| SCRUM-79 | Refactor quit path | Story | ARCHIVING SOLUTION ... | IN PROGRESS | AV | 5 |
| SCRUM-68 | If the browser gets manually closed by the user, there is a big error. Can... | Bug | | TO DO | AV | 3 |

Completed issues — View in issue navigator

| Key | Summary | Issue type | Epic | Status | Assignee | Story points |
|---|---|---|---|---|---|---|
| SCRUM-67 | The application crashes if no valid file is found in the specified folder | Bug | | DONE | AV | 3 |
| SCRUM-82 | Show archived urls path | Story | OUTPUT AND REPORTI... | DONE | ND | 6 |
| SCRUM-72 | Create / Fix Tests | Story | CODE QUALITY AND M... | DONE | AV | 13 |
| SCRUM-81 | Status update jobs | Story | OUTPUT AND REPORTI... | DONE | K | 6 |
| SCRUM-80 | Refactor Archive path | Story | ARCHIVING SOLUTION ... | DONE | K | 13 |
| SCRUM-29 | Integrate Archived URLs into Supported Files | Story | OUTPUT AND REPORTI... | DONE | ND | 8 |

Figure 1.33: Sprint 5 Backlog

The user stories from the fifth sprint are shown below. The stories have been estimated and prioritised.



Integrate Archived URLs into Supported Files

Description
**Description:**

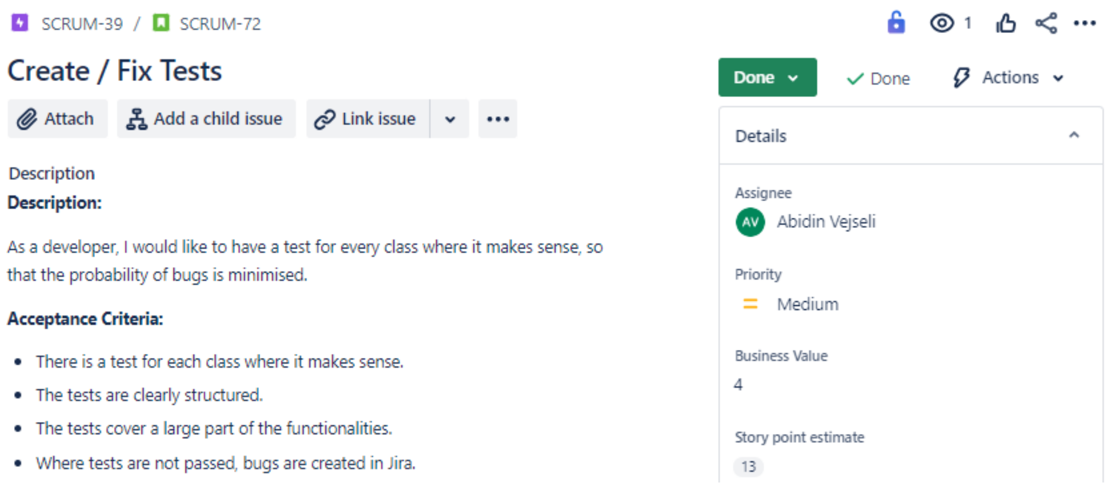
As a user, I want the system to insert the archived URL back into the original file, following its corresponding original URL.

**Acceptance Criteria:**

- The system recognizes supported file types for this integration process.
- Upon user approval, the archived URL is inserted in the appropriate location (e.g., following its original URL) within the file.

| | |
|---|---|
| Assignee | ND Nicolin Dora |
| Priority | Medium |
| Business Value | 6 |
| Story point estimate | 8 |

Figure 1.34: User Story Detail for "Integrate Archived URLs into Supported Files"


SCRUM-39 / SCRUM-72

Create / Fix Tests

Description

**Description:**

As a developer, I would like to have a test for every class where it makes sense, so that the probability of bugs is minimised.

**Acceptance Criteria:**

- There is a test for each class where it makes sense.
- The tests are clearly structured.
- The tests cover a large part of the functionalities.
- Where tests are not passed, bugs are created in Jira.

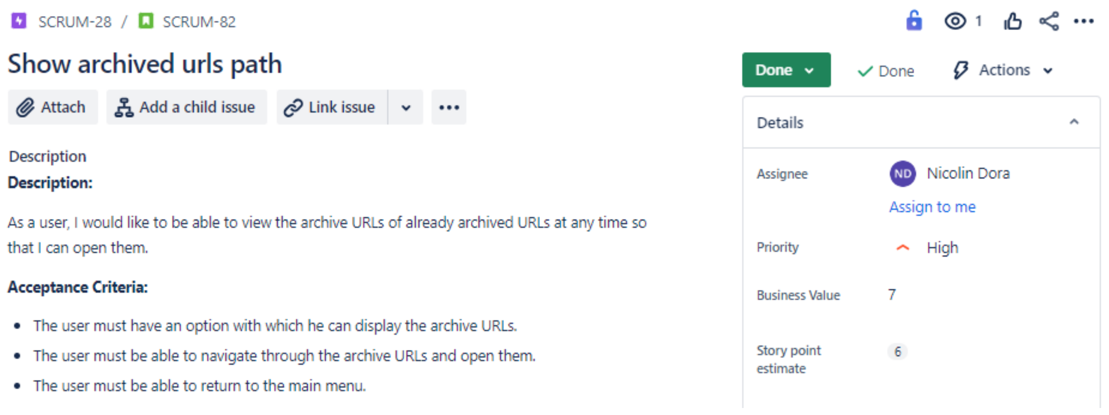
Assignee: Abidin Vejseli

Priority: Medium

Business Value: 4

Story point estimate: 13

Figure 1.35: User Story Detail for "Create / Fix Tests"


SCRUM-28 / SCRUM-82

Show archived urls path

Description

**Description:**

As a user, I would like to be able to view the archive URLs of already archived URLs at any time so that I can open them.

**Acceptance Criteria:**

- The user must have an option with which he can display the archive URLs.
- The user must be able to navigate through the archive URLs and open them.
- The user must be able to return to the main menu.

Assignee: Nicolin Dora

Priority: High

Business Value: 7

Story point estimate: 6

Figure 1.36: User Story Detail for "Show archived urls path"

In the fifth sprint, the burn down chart looks like this:

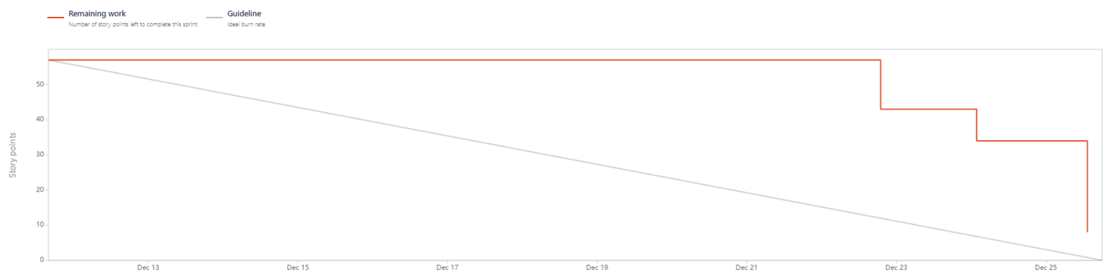


Figure 1.37: Sprint 5 Burn Down Chart

In sprint 5, our time was limited due to commitments in other modules and the upcoming Christmas period, which resulted in team member absences. We encountered an unexpected challenge with the extensive time required for test refactoring, as we prioritise quality, which often demands more time. As a result, we were unable to complete all planned user stories and had to carry them over to the next sprint. These challenges are reflected in the burn down chart.

### 1.1.6 Scrum Adaptionen

As part of Project 1, we have adjusted Scrum in order to use it in the best possible way. The adjustments are explained in this chapter.

#### Definition of Ready (DOR)

Our DOR includes conditions that ensure that all team members understand the user stories and know when a user story can be included in a sprint. The DOR was set in line with the INVEST[2] criteria. A user story in the product backlog must meet the DOR before it can be included in a sprint.

**Definition of Ready**

- Ensure a clear definition
- Define the functionality or requirement to be implemented
- Clearly defined and testable acceptance criteria
- Ensure there are no or minimal dependencies
- Understood by the whole team
- The user story has been estimated
- The scope of the user story is small enough that it can be implemented in a single sprint.

#### Definition of Done (DOD)

Our DOD contains all the characteristics and standards that a user story must meet to be considered complete. Once it satisfies the necessary quality requirements (acceptance criteria), the story can be considered complete and can be closed. The goal of our DOD is to create transparency so that everyone has a common understanding of when a story can be closed. A story that does not comply with the DOD may not be finalised.

**Definition of Done**

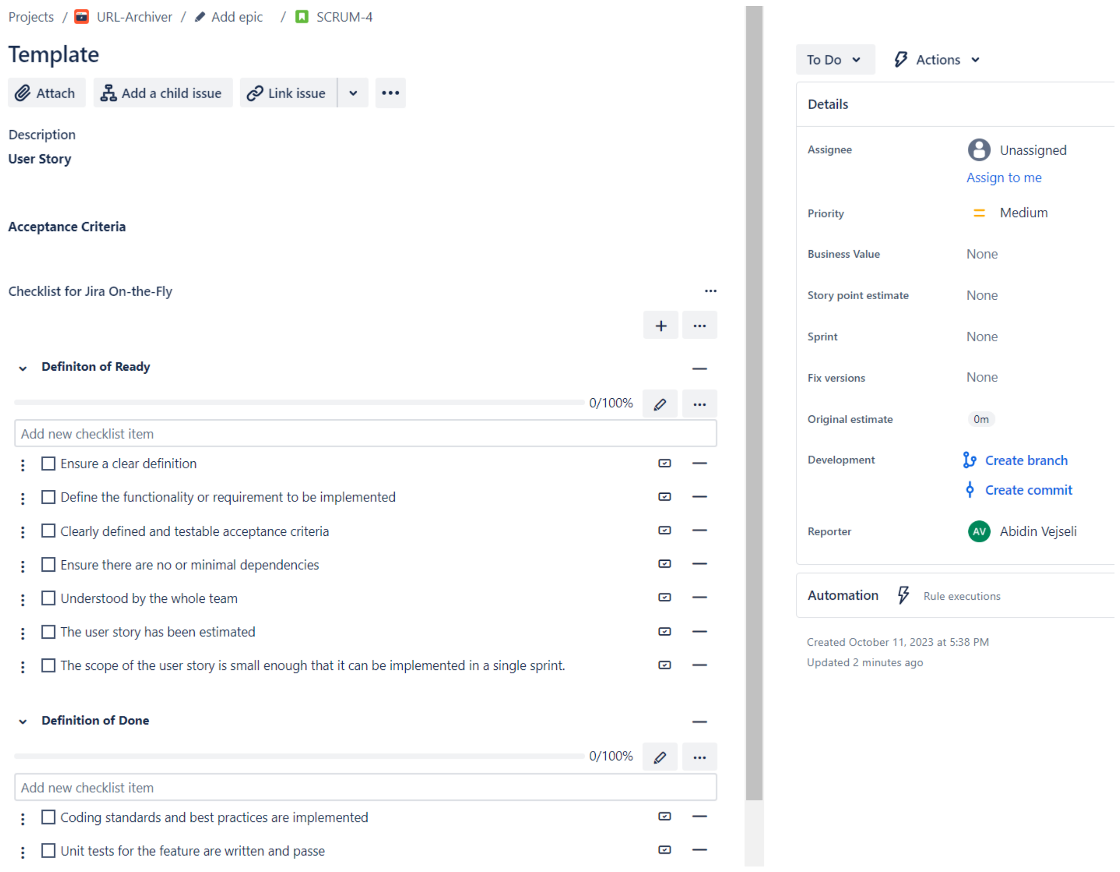
- Coding standards and best practices are implemented
- Unit tests for the feature are written and passe
- Any changes to the code or functionality are documented
- The code and functionality are reviewed by peers
- The feature works across multiple platforms

[2] XP123 article: Invest in good stories and smart tasks

- Code is integrated with master branch
- Documentation has been updated
- Acceptance criteria are met

## User Story Template

For the creation of a user story, we have defined a template so that the user stories contain all the necessary information. Below is a screenshot of our template. It includes all the relevant fields for us: Assignee, Priority, Business Value, Story Points estimate and assigned Sprint. Furthermore, we describe the user story in the "Description" field in the format "AS A <user role> I WANT TO <the goal> [SO THAT <reason>]" as well as the Acceptance Criteria. To ensure that we always have the DOR and DOD to hand, we also work with the Jira On-the-Fly add-on, which enables us to record both for each user story and tick off the individual points accordingly when they have been completed. This allows us to immediately recognise whether a user story can be included in a sprint and whether a story has been fully completed.


Projects / URL-Archiver / Add epic / SCRUM-4

Template

Attach | Add a child issue | Link issue

Description

User Story

Acceptance Criteria

Checklist for Jira On-the-Fly

Definiton of Ready 0/100%

Add new checklist item

- ☐ Ensure a clear definition
- ☐ Define the functionality or requirement to be implemented
- ☐ Clearly defined and testable acceptance criteria
- ☐ Ensure there are no or minimal dependencies
- ☐ Understood by the whole team
- ☐ The user story has been estimated
- ☐ The scope of the user story is small enough that it can be implemented in a single sprint.

Definition of Done 0/100%

Add new checklist item

- ☐ Coding standards and best practices are implemented
- ☐ Unit tests for the feature are written and passe

To Do | Actions

Details

| | |
|---|---|
| Assignee | Unassigned<br>Assign to me |
| Priority | Medium |
| Business Value | None |
| Story point estimate | None |
| Sprint | None |
| Fix versions | None |
| Original estimate | 0m |
| Development | Create branch<br>Create commit |
| Reporter | Abidin Vejseli |

Automation Rule executions

Created October 11, 2023 at 5:38 PM

Updated 2 minutes ago

Figure 1.38: Screenshot from the user story template.

## Estimation method

We have chosen the ”T-shirt sizes” method because it is a simple way to estimate effort (story points). This method is based on the fact that everyone knows T-shirt sizes and that large sizes mean more work than small sizes. As a result, this method enables us to make efficient estimations, despite the lack of shared experience in the team.

Below is the scale we use:

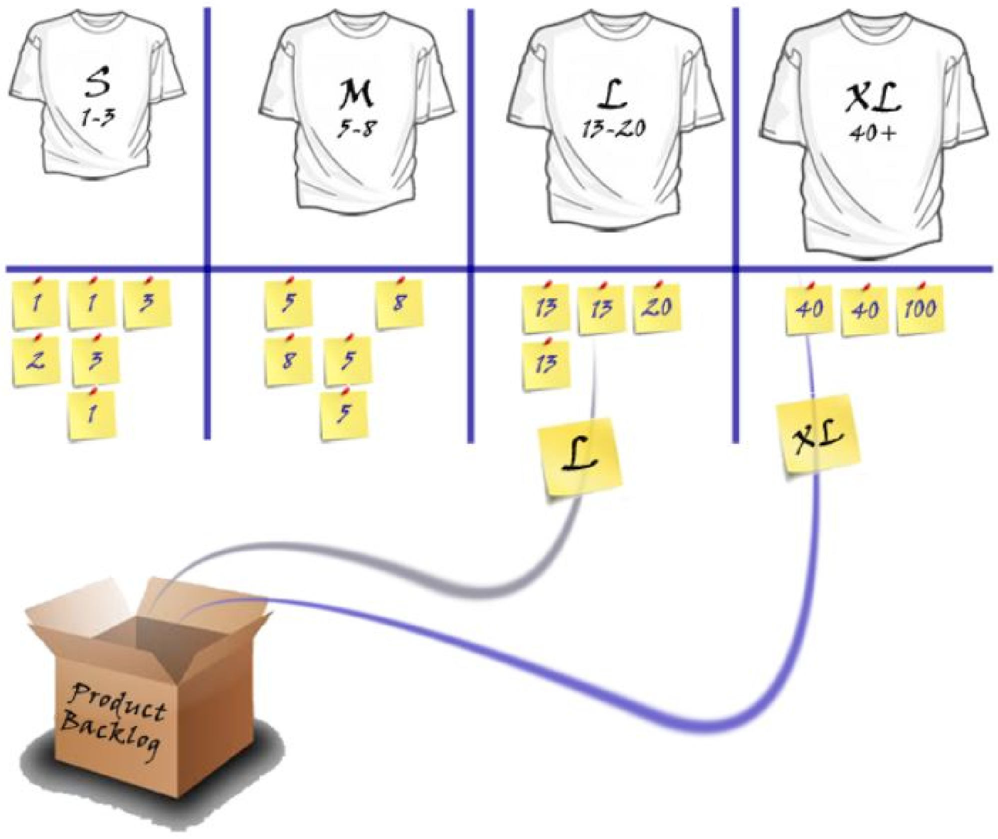


Figure 1.39: Ilustration of T-Shirt Sizes

### Velocity

To select the user stories and tasks to be worked on, a suitable criterion, velocity, is applied to estimate what can be completed in the upcoming sprint.

We have decided that we have a velocity of 30 story points per sprint. Therefore, our workload per sprint should not exceed this threshold. We consciously take this into account during sprint planning.

### Sprint

As a team, we have decided that our sprints will take place at two-week intervals. For each sprint we define a SMART sprint goal, which specifies the relevant user stories.

We decided in favour of the two-week rhythm because regular feedback is important to us and thus creates a greater learning effect. Additionally, we ascertained that one-week sprints would result in excessive overheads due to the administrative work involved in Scrum. Likewise, we consider sprints longer than two weeks to be impractical, as the interaction would suffer.

### Sprint Planning

As part of sprint planning, we make decisions about which user stories can be implemented based on the sprint goal, the story points and the velocity. Before a user story can be included in the sprint, it must be estimated by the Scrum team. This task is always carried out at the start of our sprint planning. A sprint goal is then defined based on the estimated and prioritised user stories. The user stories we select for the sprint are based on the business value, priority, story points and velocity of the team. The sprint planning takes place on the first Monday of each sprint. We have decided not to have a second sprint planning as we have already defined in our DOR that the user stories should be as small as possible. In addition, each developer has the opportunity to divide their user stories into tasks within the sprint. This allows a better overview of the progress in the sprint.

### Daily Scrum

As a team, we have decided not to have daily Scrum meetings, as this is not possible because all team members do work part times. Instead, we have two weekly meetings (weeklys), on Wednesday and Friday at 17:00, which last a maximum of 15 minutes. In addition, we have chosen to hold the meetings through Microsoft Teams as it is easier to organise. The goal of these meetings is to share the current progress, address issues, update the team and briefly discuss the next steps.

#### Sprint Review

In the sprint review, we check the intermediate result of the processed user stories. We check whether all the stories that should be completed meet the DOD. Furthermore, we discuss in the team what went well, what problems we encountered and how we solved them. Based on these results, the product increment is created. In our case, the product owner, who represents the customer, tests the product increment against the requirements. The review is done from the customer's perspective by testing the product increment. The outcomes are utilized to update the product backlog. The sprint review meeting takes place on the last day of the sprint.

#### Sprint Retrospective

In the sprint retrospective, we gather information as a team about what went well and what didn't go as planned in the previous sprint. We then derive specific improvements and plan their implementation. Our goal is to improve the efficiency, quality, communication and speed within our team. To achieve this, we give ourselves constructive criticism and are open to feedback. The sprint retrospective takes place on the last day of the sprint.

### 1.1.7 Review Project Setup

#### Initial situation

#### Subject analysis

#### Stakeholder(-Management)

#### Organisation

#### Installations

### 1.1.8 Review

This chapter provides a review of our project, discussing the management of our sprints and backlogs, and our progress towards achieving our product goal.

#### Product goal

Achieving the product goal for the URL Archiver application was a journey of iterative development and unwavering commitment to our goals. We successfully developed a platform-independent, CLI-based Java application that met all the specified functionalities, including efficient URL extraction, archiving options and CSV file generation. The application adheres to FLOSS licensing, ensuring open-source availability. Accompanying this, we created detailed user manuals, installation instructions and thorough software documentation. Our code is characterised by minimalism and modularity, with self-explanatory sections that underline our commitment to quality and maintainability.

### Sprint goals

Throughout the sprints, we've rigorously set our goals in alignment with the SMART criteria, ensuring clarity and measurable targets. From the foundational work of implementing a robust input handler in Sprint 1 to enhancing system stability in Sprint 4, each goal was carefully crafted and successfully achieved. Sprint 5's focus on asynchronous archiving refined our application's functionality, while Sprint 6's emphasis on design and documentation prepared us for the final review. This structured approach to goal-setting has been instrumental in our consistent delivery of sprint objectives.

### Delimitation

The delimitation in our project was implicitly conducted through disciplined backlog management and sprint planning. We confined our efforts to the most critical features, implicitly setting boundaries that guided our development focus. This implicit delimitation allowed us to concentrate resources on the highest priority tasks, ensuring the project's scope remained tightly aligned with our key objectives. While a formal delimitation review was not performed, our adherence to Agile principles effectively served the same purpose, keeping the project streamlined and on target.

### Delivery objects

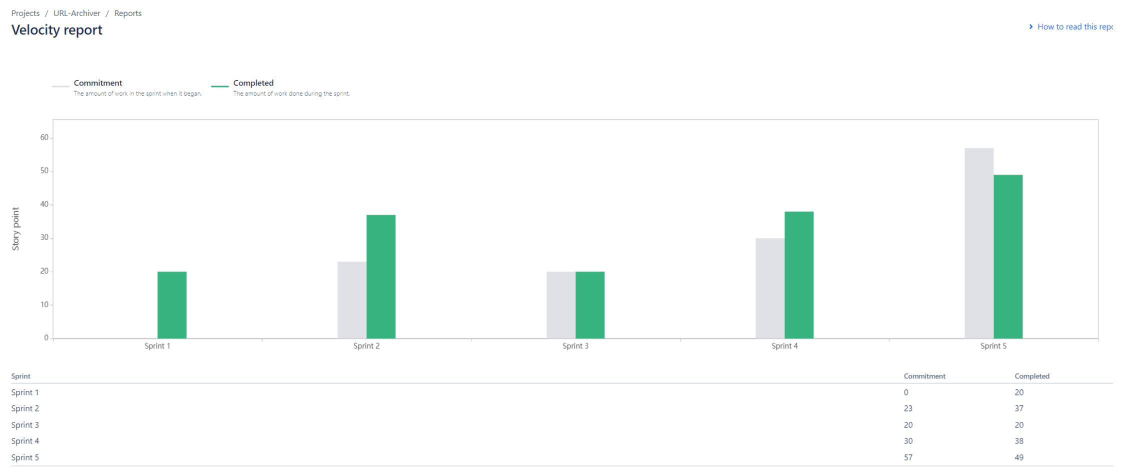
We successfully developed a platform-independent Java application that follows the principles of FLOSS. The application includes a command-line interface for ease of use, can process various file inputs, has URL extraction and archiving functionalities, and generates a CSV file documenting the archived URLs. In addition, we provided user manuals, installation scripts, and software documentation to improve user experience and understanding. Our attention to detail and commitment to our goals throughout the project's lifecycle resulted in successfully meeting our delivery objectives.

### Product Backlog

Our product backlog management was an exercise in adaptability and strategic foresight. In collaboration with stakeholders, we regularly assessed and updated our backlog, ensuring it remained relevant and reflective of current project needs. This iterative process, led by the Product Owner, involved introducing new user stories and de-prioritizing or removing those that had lost their relevance. Our focus was on high-priority tasks that directly contributed to achieving the product goal, while medium and low-priority items were cataloged for potential future development. This approach maintained a balance between addressing immediate project requirements and envisioning future enhancements. The careful maintenance of the backlog provided a clear roadmap for our efforts, crucial for the effectiveness of our sprint reviews and overall project progress.

### Sprint Backlogs

Throughout the sprints, our team demonstrated a commendable ability to effectively manage and complete the sprint backlogs. We consistently improved our workload management with each sprint, adapting our strategies based on previous experiences. There was only one instance of user stories being carried over to the next sprint, reflecting our overall planning efficiency and responsiveness to unforeseen challenges. The velocity report illustrates this progress, with an upward trend in completed story points, underlining our growing proficiency in sprint management.


Projects / URL-Archiver / Reports
› How to read this repo
Commitment
The amount of work in the sprint when it began.
Completed
The amount of work done during the sprint.
Story point
0
10
20
30
40
50
60
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5

Velocity report

| Sprint | Commitment | Completed |
|---|---|---|
| Sprint 1 | 0 | 20 |
| Sprint 2 | 23 | 37 |
| Sprint 3 | 20 | 20 |
| Sprint 4 | 30 | 38 |
| Sprint 5 | 57 | 49 |

Figure 1.40: Velocity Report

### 1.1.9 Retrospective I: Scrum Method

This chapter reflects on the distribution and implementation of Scrum roles as previously described. It also considers our insights from Scrum Events and Scrum Artifacts.

### Scrum Roles

Nicolin fulfilled the role of Product Owner exceptionally well. He consistently maintained and prioritized the Product Backlog. Moreover, he ensured we were in close communication with our customer, Mr. Kramer, guaranteeing that stakeholder requirements were discussed and incorporated. This was instrumental in the business success of the product being developed.

Abidin, as Scrum Master, executed his responsibilities with diligence, evident in the efficient Scrum operation of the team. Acting as a liaison between the Product Owner and the Developers, any questions regarding the agile methodology were promptly clarified. Additionally, obstacles beyond the team's capacity were resolved with the involvement of our coaches.

Kilian held the role of developer and played a key role in shaping the product. As there were only three of us in the project team, Abidin and Nicolin also contributed to the technical development. Our goal to satisfy our client, Mr. Simon Kramer, was achieved through the early and continuous delivery of our software. This ensured the software developed matched Mr. Kramer's vision and requirements. Moreover, we were able to implement new requirements during the project, as clearly demonstrated by integrating the Wayback Machine. Throughout the project, we maintained self-organization, a consistent pace, reflection, and expeditious work.

### Scrum Events

**Sprint** Our two-week sprint cycles have been effective in balancing workload and facilitating regular feedback. Setting SMART goals for each sprint provided us with a clear and focused direction and contributed significantly to our team's progress. Although we occasionally faced challenges in estimating workload, these instances offered valuable lessons in capacity planning. Exploring more precise estimation techniques could improve our ability to align tasks with our team's capacity, building on our already solid sprint planning process.

**Sprint Planning** Sprint planning was a strong point for our team, particularly with the upfront estimation of user stories and prioritization based on business value and team velocity. We ensured that the most impactful tasks were addressed each sprint. Encountering larger-than-expected user stories at times provided learning opportunities for better task evaluation. A more comprehensive review process for estimating user stories could bolster our sprint planning further.

**Daily Scrum** Adjusting to bi-weekly meetings complemented our part-time work schedules and kept team communication robust. Using Microsoft Teams for meetings brought added flexibility and convenience. While the format sometimes delayed the addressing of immediate issues, it generally promoted focused and efficient discussions. A mid-week check-in could enhance our responsiveness without burdening the team's schedule.

**Sprint Review** Our sprint reviews were invaluable, offering objective evaluations that kept product increments in line with customer needs and expectations. Responding to customer feedback, though sometimes challenging, was a dynamic learning opportunity that led to significant product enhancements. More frequent reviews during the sprint could make the feedback integration process even smoother.

**Sprint Retrospective** The sprint retrospectives were characterized by constructive, open discussions that effectively pinpointed our successes and growth areas. These sessions were instrumental in fostering a culture of continuous improvement, with actionable steps consistently identified to enhance team processes. While implementing these improvements took effort, the retrospectives were central to our team's development and unity. A structured follow-up mechanism for retrospective suggestions could maximize these sessions' impact.

Although we attempted to provide each other with honest and constructive feedback, we realised that it can be challenging to do so. This is often due to fear of the recipient's reaction. However, we believe that with time, giving feedback will become easier for a team that has worked together for an extended period.

### Scrum Artifacts

**Product Backlog** The Product Backlog was a dynamic, well-organized tool that effectively captured our project's requirements and priorities. Regular updates and refinements ensured it remained in sync with our evolving project goals and customer needs. Clear categorization and prioritization of items significantly aided our planning and decision-making. We are considering incorporating more frequent stakeholder feedback sessions to ensure that the backlog continues to reflect the most current and relevant project needs.

**Sprint Backlog** Our Sprint Backlog was used effectively to map out user stories, providing a clear outline of the sprint goals. Generally, we managed without creating detailed tasks for each story, favoring a broader view for simplicity and clarity. Introducing specific tasks for user stories in future sprints could improve granularity and task management. This approach could enhance tracking and aid in early obstacle identification.

**Product Increment** Our approach to product increments has highlighted our team's dedication to delivering high-quality results at the end of each sprint. The increments met the set criteria and client expectations, showcasing our ability to effectively transform backlog items into tangible, valuable product features. Regular reviews and refinements ensured alignment with project goals and customer needs.

### 1.1.10 Retrospektive II: Tools/Instruments

#### Insights tool use

During the project, we used various tools that were essential to our success. In this section, we will discuss these tools and our experiences with them.

**Java** Java, our chosen primary programming language, facilitated a smooth development process due to the team's pre-existing knowledge. This familiarity allowed us to focus on delivering a high-quality application efficiently.

**Maven** Maven's role in our project was critical, handling compilation, dependency management, and the building process. Our prior experience with Maven enabled its seamless integration into our workflow, contributing to a smooth development cycle.

**JUnit 5** JUnit 5 played a pivotal role in our testing framework, guaranteeing the stability and reliability of our application, especially following code refactoring activities, thus ensuring the integrity of our application's logic.

**JetBrains IntelliJ IDEA** The selection of IntelliJ IDEA as our integrated development environment provided an array of tools that streamlined our development processes, enhancing productivity, and facilitating a high level of code quality.

**GitHub** GitHub was the backbone of our version control system, offering a robust platform for collaborative code management. It enabled efficient teamwork and was essential in maintaining a consistent codebase.

**Atlassian Jira** Atlassian Jira was a cornerstone of our project management, enabling us to track progress, manage priorities, and streamline our workflow effectively, proving itself as an invaluable asset to our project organization.

**LaTeX** LaTeX was used for creating our project documentation and presentations. Despite initial challenges due to varying levels of experience within the team, it ultimately enabled us to produce professional and consistent documentation.

#### Communication

**Microsoft Teams** With Microsoft Teams, we were able to streamline our communication, enabling an effective platform for meetings and Scrum events that fostered rapid problem resolution and team agility. The platform's flexibility and immediacy were crucial to our effective problem-solving process.

**Email and BigBlueButton** For external communications, particularly with coaches, we used email for asynchronous exchanges and BigBlueButton for real-time video conferencing, ensuring clear and direct dialogue with our stakeholders.

#### Controlling

**Atlassian Jira** Jira served as the central control tool for our project, providing a comprehensive overview of project status and task distribution. It facilitated clear categorisation and prioritisation of tasks, thereby enhancing our workflow and project management.

**Burndown Charts** We utilized Burndown Charts to monitor our sprint efficiency and to inform future sprint planning. This allowed for a balanced workload management and realistic goal-setting based on our team's velocity and past performance.

### 1.1.11 Lessons learned

#### Team

#### Nicolin Dora

Developing and managing my first software project from scratch was a great experience. However, working with SCRUM was challenging as we could only work on weekends or evenings due to part-time study. This made it difficult to use SCRUM effectively, and planning and report preparation took almost as much time as product development. In principle, SCRUM is a useful method as it allows for easy tracking of project progress. However, the students could have been given more freedom in this project to simplify the method.

I enjoyed the teamwork and appreciated the goal-oriented approach. It was interesting to balance the different demands on quality. Mr. Kramer and Mr. Helbling were always available to answer questions, which made collaboration easier. For future projects, I plan to use a slimmed-down version of SCRUM, depending on the complexity of the project. I am determined to improve my planning and communication within the team right from the start.

### Abidin Vejseli

Project 1 has been an enriching yet demanding journey, requiring meticulous time management and organization from my side. The Scrum methodology, which I believe our team adapted effectively, and the complexities of part-time study contributed to this challenge. Personally, I felt the initial module introduction was rather lengthy and could be condensed to the essential elements. It was unfortunate not to have both coaches at the presentation, an aspect I missed.

Working within the team was a positive experience, and I'm proud of how we distributed the workload, ensuring valuable learning for everyone. The interaction with our coaches was a highlight, their readiness to assist was something I greatly valued. For the next project, I aim to initiate and ideally complete my user stories right at the start of the sprint.

### Killian Wampfler