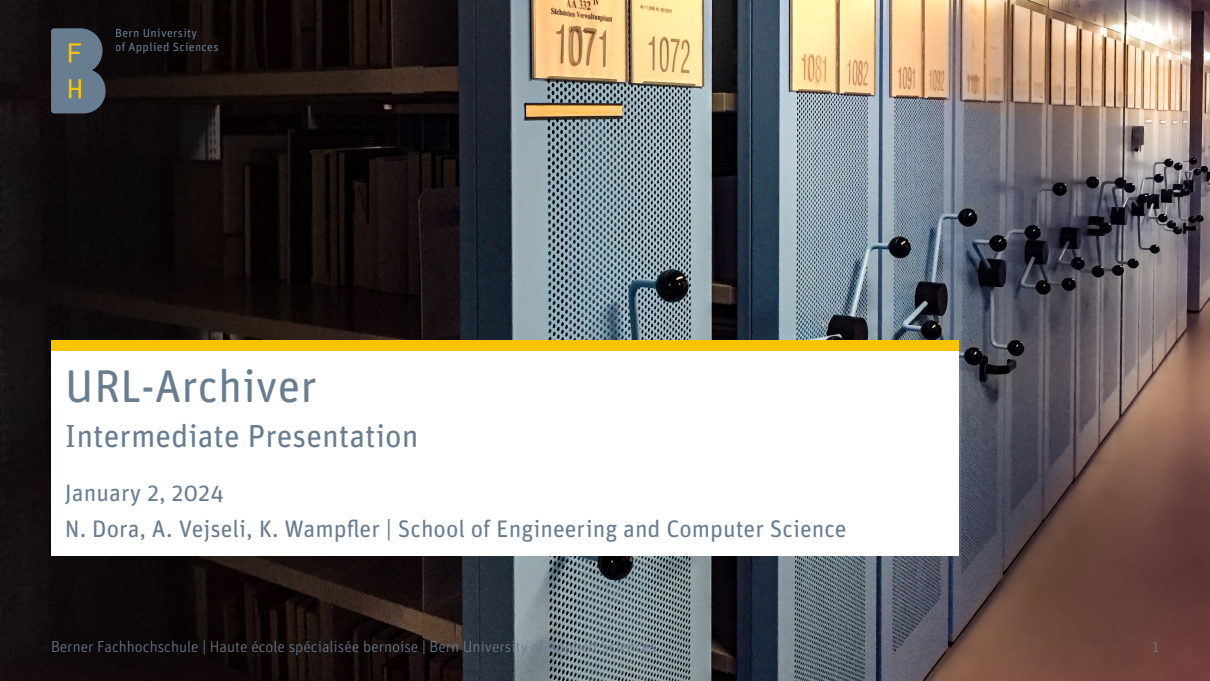


Bern University
of Applied Sciences

URL-Archiver

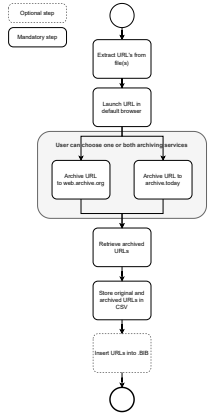
Intermediate Presentation

January 2, 2024

N. Dora, A. Vejseli, K. Wampfler | School of Engineering and Computer Science

Table of Content

Topics Covered



Problem Statement

Problem Statement ”URL-Archiver”

The ever changing internet

- **ever-changing internet** (websites are taken down or changed)
- not a problem for everyday internet users but…
- …what about **documents** (e. g. theses, documentations etc.)
  - invalid links to content-relevant information
  - invalid quote links
  - old documents may contain numerous broken hyperlinks
- another case: **Forensic**
  - e.g. **legal report** about a statement on the Internet

Problem Statement

Simple Solution

- Archive the **actual state** of the linked websites
  - Wayback machine
  - archive.today
  - Memento Time Travel
  - and more…
- But who has the **time** and **motivation**?
  - …to search each hyperlink
  - …to manually archive each website

Project Goal

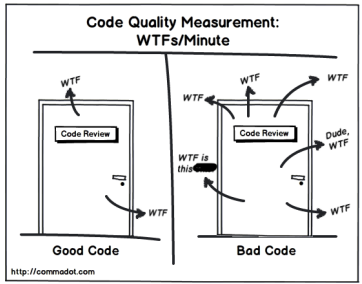
- Develop a software that…
  - …**searches hyperlinks** within documents
  - …is capable of **archiving websites**
  - …can **store** current and archived links in a file
  - …is **easy** and **intuitive** to use

Functional Requirements

- **The software must be…**
  - …a **Java application** that is compatible with multiple platforms
  - …Free and Open Source Software licensed **(FLOSS)**
  - …capable of archiving websites on **archive.ph** or/and **WayBackMachine**
  - …capable of generating a **CSV-file** with key-value (URL, archived URL)

Non-Functional requirement

- Coding Norm
- publicly accessible
- Project-Method: **SCRUM**
- public documents in English
- project report with **LaTeX**

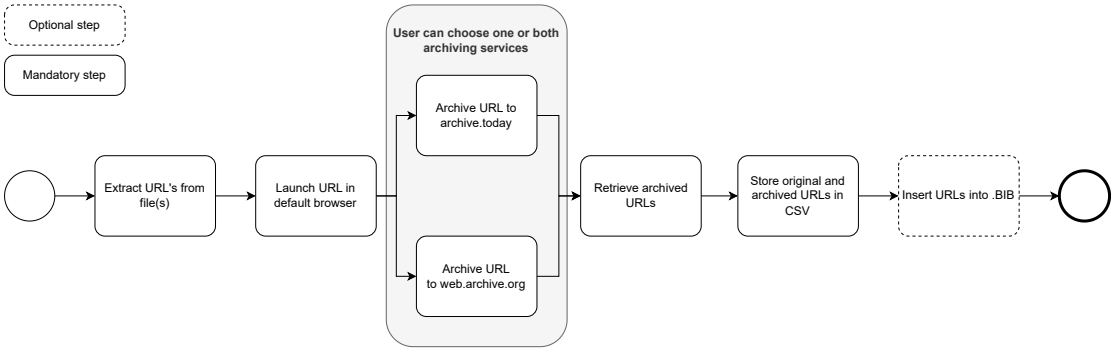


Solving The Problem

Process Model

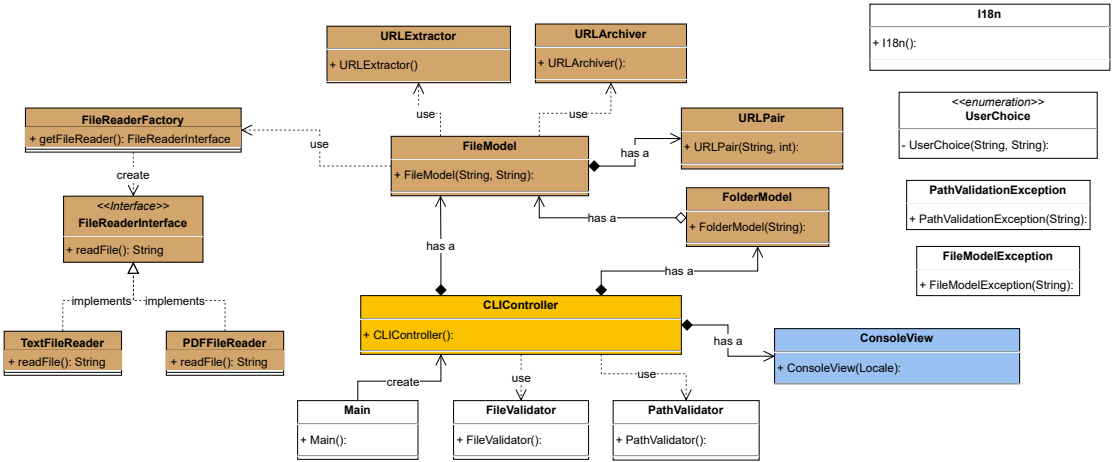
Workflow for URL Extraction and Archiving

- User can **skip URLs**, **launch them**, **access help** or **quit** the program at **any time**



High Level Class Diagram

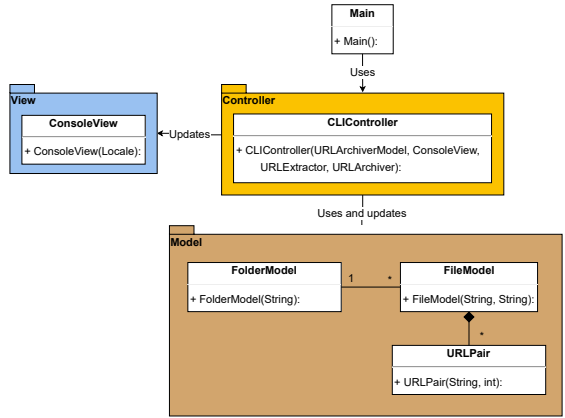
Overview



Architecture

Employing the **MVC** Pattern

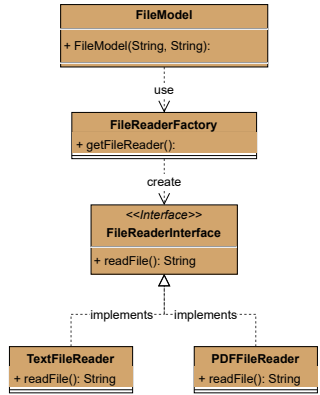
- Modular design for **easy extension**
- Separate data, view, and control flow
- Facilitates the potential **addition of a GUI**



Architecture

Factory Pattern

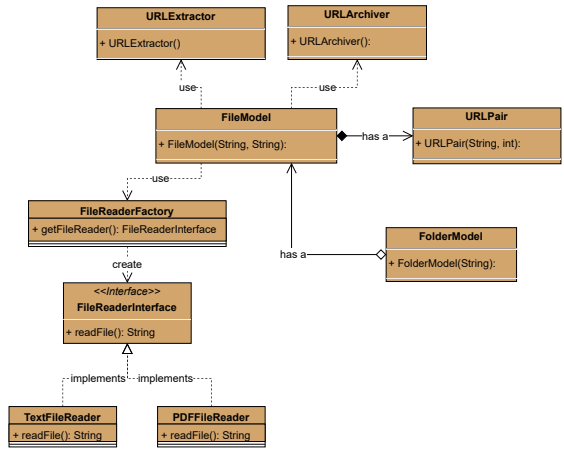
- **Scalable** and **robust** design
- Ensures **maintainability** and clear code structure
- Commitment to **best programming practices**



Key Components of the Data Model

Overview

- **FileModel**: Represents and handles individual files
- **FolderModel**: Manages a collection of FileModel instances
- **URLPair**: Links extracted URLs with their archived counterparts
- **FileReaderFactory**: Organizes and maintains URLPairs
- **URLExtractor**: Used for URL extraction
- **URLArchiver**: Used for URL archiving



Technologies I/II

- **Java**: The primary programming language for our application
- **LaTeX**: Used for documentation and presentation
- **JUnit 5**: Utilized for unit testing
- **PDFTextStripper (PDFBox library)**: Used for extracting text from PDF documents
- **Selenium**: Web automation tool used for tasks like inputting URLs, handling **captchas**, and retrieving archived URLs due to the absence of an official API from archive.today
- **Maven**: Essential for compilation, dependency management, and building the project

Technologies II/II

- **Archive.today**: One of the services used by the URL-Archiver to archive URLs
- **Wayback Machine**: One of the services used by the URL-Archiver to archive URLs
- **JetBrains IntelliJ IDEA**: Used to develop the program and create the report and presentation in Latex
- **GitHub**: A platform for code hosting and collaboration, allowing us to manage our program's development and version control
- **Atlassian Jira**: A tool for tracking progress and managing our project goals and tasks

Project Management with SCRUM

Our Scrum Team

Nicolin Dora
Product Owner & Developer

Abidin Vejseli
Scrummaster & Developer

Kilian Wampfler
Developer

Other Roles

Dr. Simon Kramer
Stakeholder & Customer

Frank Helbling
PM-Advisor

Our Scrum Adaptions

Sprint

Sprint	Sprint planning	Daily Scrum	Sprint Review	Sprint Retro
Two Weeks Sprint Goal (SMART)				

Our Scrum Adaptions

Sprint Planning

Sprint	Sprint planning	Daily Scrum	Sprint Review	Sprint Retro
	First Monday Estimate User Stories Define Goal Choose User Stories			

Our Scrum Adaptions

Daily Scrum

Sprint	Sprint planning	Daily Scrum	Sprint Review	Sprint Retro
		~~Daily~~ Twice a week Current state MS Teams		

Our Scrum Adaptions

Sprint Review

Sprint	Sprint planning	Daily Scrum	Sprint Review	Sprint Retro
			Last Day Completed? Problems? Testing Poductincrement	

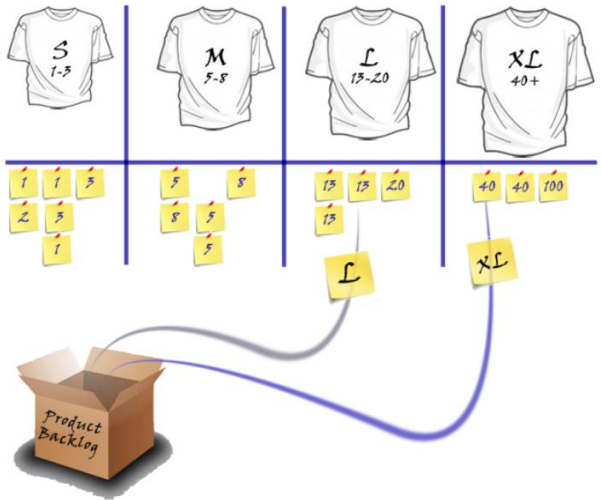
Our Scrum Adaptions

Sprint Retro

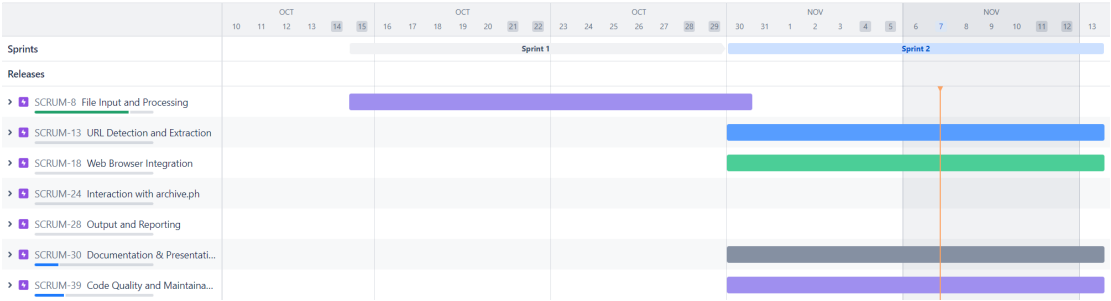
Sprint	Sprint planning	Daily Scrum	Sprint Review	Sprint Retro
				Last Day Feedback Do's & Dont's

Estimation Method

T-Shirt Sizes



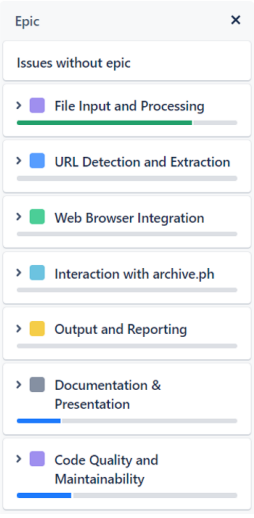
Velocity

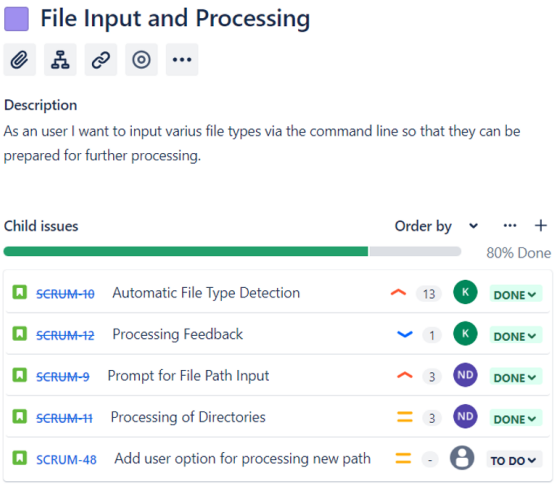

OCT
10 11 12 13 14 15
OCT
16 17 18 19 20 21 22
OCT
23 24 25 26 27 28 29
NOV
30 31 1 2 3 4 5
NOV
6 7 8 9 10 11 12 13
Sprints
Sprint 1
Sprint 2
Releases
SCRUM-8 File Input and Processing
SCRUM-13 URL Detection and Extraction
SCRUM-18 Web Browser Integration
SCRUM-24 Interaction with archive.ph
SCRUM-28 Output and Reporting
SCRUM-30 Documentation & Presentati...
SCRUM-39 Code Quality and Maintaina...

Backlog

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	-	
SCRUM-37 Implementation documentation	DOCUMENTATION & P...	TO DO	-	

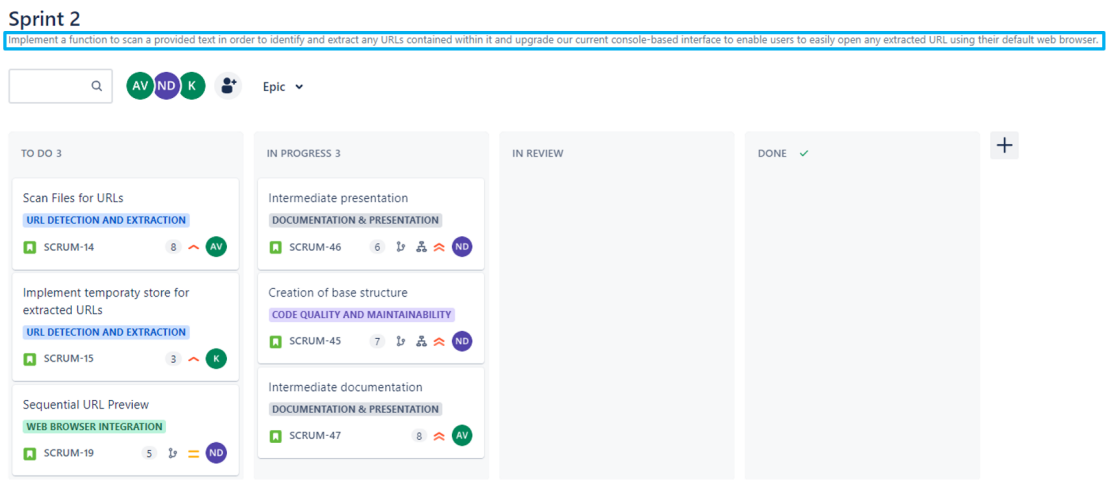
Epics


Epic
Issues without epic
File Input and Processing
URL Detection and Extraction
Web Browser Integration
Interaction with archive.ph
Output and Reporting
Documentation & Presentation
Code Quality and Maintainability


File Input and Processing
Description
As an user I want to input varius file types via the command line so that they can be prepared for further processing.
Child issues
Order by
80% Done
SCRUM-10 Automatic File Type Detection 13 K DONE
SCRUM-12 Processing Feedback 1 K DONE
SCRUM-9 Prompt for File Path Input 3 ND DONE
SCRUM-11 Processing of Directories 3 ND DONE
SCRUM-48 Add user option for processing new path - TO DO

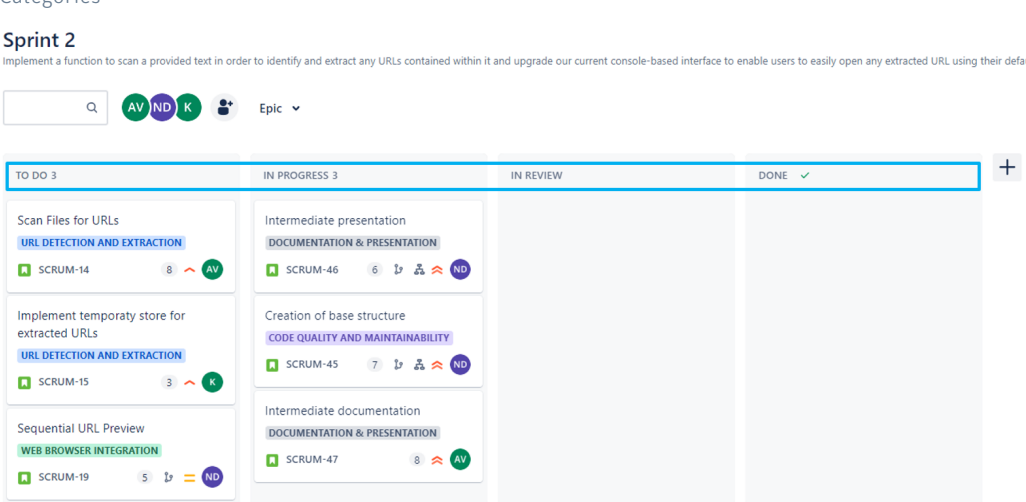
Spint 2

Sprint Goal



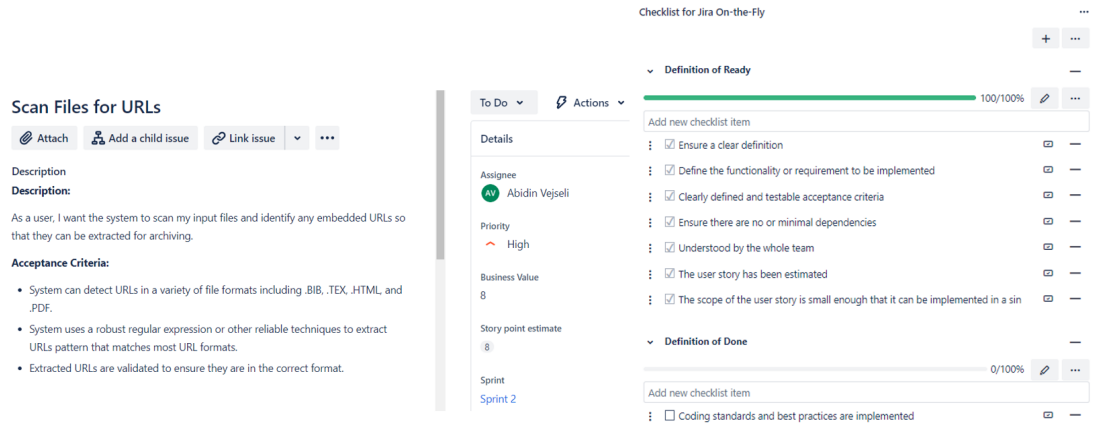
Spint 2

Categories



User Story

Description, Acceptance Criteria, DOR and DOD



Our learnings

Coding	Scrum
Branch per User Story	Create Tasks
Commit Messages	User Story good size
Code Reviews	Weeklys are valuable
Basic code structure	

Questions?