

2 Project Plan

2.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

Which of agile, waterfall or waterfall+agile project management style are you adopting. Justify it with respect to the project goals.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

We plan on using a combination of waterfall + agile management style. The linear approach of the Waterfall process will allow us to take account for the initial requirements and help us in developing the application life cycle. The flexibility of the Agile App Development process will allow our team to easily adapt to changes regarding features (documents support), timing constraints, new technologies, or the overall scope of the project.

Our group plans on using a combination of GitLab and a shared Google Drive for this project. GitLab will be used to keep track of tasks and a repository for code. Our primary means of communication between team members will be through Discord. Microsoft Teams will be used for primary communication with BuilderTrend. Email will be used for communication with our faculty advisor.

2.2 TASK DECOMPOSITION

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

Task	Description
1. Project Planning	Flush out requirements received from client and figure out team dynamic
2. Develop Web Application User Interface (UI)	Create a UI for users to search using keywords and filters and see results of related files. UI will also need to allow uploading of files.
3. Develop REST API	Design and implement application functions which will communicate between the frontend and Elasticsearch, and between the text extractor and Elasticsearch.

4. Setup Elasticsearch Node	Setup Elasticsearch node to store uploaded data.
5. Develop File Text Extractor	Design and implement a way to extract text and metadata from various types of files.
6. Application Testing	Test individual components of the web application, text extractor and REST API

2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.2. How do you measure progress on a given task? These metrics, preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to those sprint).

Milestones:

1. Project Planning
 - a. Complete Team Initiation Assignment
 - b. Complete Professionalism Assignment
 - c. Complete Requirements & Standards Assignment
 - d. Complete Design Assignment
 - e. Complete Testing assignment
2. Develop User Interface (UI) for Web Application
 - a. Begin design of core components (search bar, filter bar, result view, and file uploader)
 - b. Design mockups are verified by client and professor
 - c. Begin implementation of search bar and result view
 - d. Begin implementation of filter bar
 - e. Begin implementation of file uploader
 - f. Web application prototype is complete (able to query keywords and retrieve relevant results)
3. Develop REST API
 - a. Frontend can successfully communicate with Elasticsearch
 - b. Text extractor can successfully communicate with Elasticsearch
 - c. Frontend receives a response within 10 seconds.
4. Setup Elasticsearch Node
 - a. Elasticsearch installation on server is complete
5. Develop File Text Extractor
 - a. Application is able to detect file type of .docx, .txt, .xlsx, .pptx, .pdf, .jpg, and .png

files

- b. Application is able to detect file metadata
 - c. Application is able to extract text from ...
 - i. .docx files with at least 90% accuracy.
 - ii. .txt files with at least 90% accuracy.
 - iii. .xlsx files with at least 90% accuracy.
 - iv. .pptx files with at least 90% accuracy.
 - v. .pdf files with at least 90% accuracy.
 - vi. .jpg files with at least 85% accuracy.
 - vii. .png files with at least 85% accuracy.
 - d. Extracted text and metadata is able to be formatted and inserted into Elasticsearch
6. Application Testing
- a. Text extractor passes unit tests with 90% code coverage rating
 - b. Frontend unit tests pass with 90% code coverage rating
 - c. Application passes users acceptance tests

Metrics of Interest

- User Interface Usability
- Text extraction accuracy
- Query speed and accuracy

Evaluation Criteria:

- User Interface Usability
 - responsiveness of UI
 - good user experience (get feedback from different users)
- Text extraction accuracy
 - accuracy of the results with keyword matching
 - meet user expectations
- Query speed and accuracy
 - time needed for the search
 - our implementation vs brute force

2.4 PROJECT TIMELINE/SCHEDULE

- A realistic, well-planned schedule is an essential component of every well-planned project
- Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity
- A detailed schedule is needed as a part of the plan:
 - Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.
 - Annotate the Gantt chart with when each project deliverable will be delivered

Agile projects can associate risks and risk mitigation with each sprint.

- 1. Project Planning
 - Risk probability: 0.3
 - The major risk involved with our project planning missing requirements/constraints or having to make major changes to our project requirements and constraints. Our initial project document did not contain a lot of detail regarding the project so we have to invest a lot of time asking our client for the requirements before making any major design decisions. If we end up missing a major requirement, it could result in us having to invest a lot of extra time fixing the issue.
- 2. Develop Web Application User Interface (UI)
 - Risk probability: 0.3
 - One big risk for developing our frontend application is having problems integrating the application with the backend and retrieving data from it. This could be the result of several factors such as incompatibility between languages.
 - Risk probability: 0.1
 - Another risk involving the frontend interface is, with the change or addition of requirements, we may need to redesign the user interface or make amends to it. In the case that this occurs, it will not be a major issue since our design for the UI is relatively simple and can be redesigned without much effort.
- 3. Develop REST API
 - Risk probability: 0.4
 - Since the REST API will be handling a lot of the communication between our applications, there is a lot of room here to run into issues. Whether it be between the frontend and Elasticsearch or Elasticsearch and Apache Tika, it is crucial that our implementation is carefully designed to ensure that we will not run into any major issues that will result in a large setback.
- 4. Setup Elasticsearch Node
 - Risk probability: 0.2
 - We shouldn't run into many issues with this. The major risk would be to have some technical problems setting the node up on the server, especially if it's from their side. In that case, we might have to contact Elasticsearch to resolve it.
- 5. Develop File Text Extractor
 - Risk probability: 0.4
 - The file extractor is a major part of the project and we could run into multiple issues. If the extractor isn't able to detect the file type, get metadata (within our accuracy %), or format and insert files into elasticsearch, we'll have to re-design from the very basics and dive deep down to solve the issue. As such, we need to work closely on designing this part of the project.
- 6. Application Testing
 - Risk probability: 0.1
 - There isn't much risk associated with testing. The biggest risk would be to have issues in our test code/implementation.

2.6 PERSONNEL EFFORT REQUIREMENTS

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

Task Name	Est. Hours	Explanation
Project Planning	75	Including most of the assignments for the first semester, research for learning how to utilize the APIs, and asking our client questions. The estimated time should be between 75 and 100 hours.
Develop Web Application User Interface (UI)	50	Our frontend requirements are not that rigorous so it should take less time relative to our backend applications
Develop REST API	100	The REST API will interface with all of our applications and can result in a lot of issues transmitting messages between our applications so the most time will be spent here
Setup Elasticsearch Node	75	Elasticsearch is one of the core parts of the backend so a large amount of time will be allocated to developing and refining it for maximum performance
Develop File Text Extractor	75	Apache Tika is the other core part of the backend. We need to make sure we get the correct info from the files as efficiently as possible.
Application Testing	25	We will need to spend some time testing some edge cases and the performance of the entire system. Although, there shouldn't be too many test cases depending on how many subsystems we implement from the stretch goals

2.7 OTHER RESOURCE REQUIREMENTS

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

There shouldn't be any need for additional resources for our project apart from the server. Since it's a software project we won't need any parts or materials. However, we might need some software licenses in the future.