# Project Plan

## Table of Contents

## Contents

- Table of Contents
- Introduction
  - Project Overview
  - Project Description and Scope
    - Automated Note-Taking Process
    - Database Utilization for Transcript Management
    - Summary Generation with Generative AI
    - Interactive Chatbots for Enhanced Learning
  - Roles and Responsibilities
  - Team Communication
- Process Definition
  - Lifecycle Model
- Schedule
  - Work Breakdown Structure
  - Work Packages
  - Activities Dependencies
  - Work Package Details
- Project Estimates
  - Code Size Estimation using Function Points
    - Unadjusted Function Points
    - Adjusted Function Points
    - Lines of Code
  - Efforts, Duration and Team Size Estimation
    - Distribution of Effort
  - Cost Estimates
  - Product Checklist
  - Best Practice Checklist
  - Risk Management
    - More changes to requirements than anticipated
    - Specification Delays
    - System size underestimated
    - Staff leaving before project complete
    - Problems coordinating within the group

- Customer cancels project
- Quality Assurance
  - Test Procedures
  - Testing Methodologies
  - Testing Aspects
  - Use of Test Cases
- Monitoring & Control

# Introduction

## Project Overview

Nanyang Technological University (NTU) is addressing the substantial challenges presented by its traditional approach to lecture note-taking, which is both expensive and inefficient. The reliance on student note-takers has led to inconsistencies in note quality due to variations in accuracy, completeness, and the note-takers' understanding of the material. This system incurs significant financial costs and does not scale well to the volume of lectures at NTU.

To resolve these issues, NTU is proposing an innovative solution that leverages state-of-the-art speech-to-text and AI technologies to automate and improve the note-taking process. This comprehensive strategy includes:

**Automated Note-Taking**

The introduction of advanced speech recognition technology to automatically convert lecture audio into precise transcripts, supporting a range of audio and video formats for both live and uploaded lectures. This aims to enhance the accuracy and reliability of lecture documentation while reducing manual labor and associated costs.

**Transcript Management Database**

The development of a sophisticated database system for the effective storage, retrieval, and management of lecture transcripts. This system will feature an advanced search function, a user feedback mechanism for quality control, and a threshold policy for upload limits to ensure the maintenance of high-quality content.

**AI-Generated Summaries**

The use of generative AI to create succinct summaries of lecture transcripts, customizable by difficulty level and length, and further enriched by the optional inclusion of lecture slides for context. This feature is designed to provide students with quick, tailored access to the core content of lectures, facilitating efficient revision and study.

**Interactive Chatbots**

The implementation of AI-powered chatbots to simulate a personalized tutoring experience, enabling students to ask questions and receive accurate, contextually relevant responses based on the lecture materials. This initiative aims to enhance students' understanding and engagement with the material.

Despite the innovative aspects of this proposal, NTU recognizes the potential challenges and limitations inherent in such a technological undertaking, particularly the reliance on external APIs for transcription and summarization services, which may impact the quality of the outputs. Additionally, the manual input of metadata for the database could introduce human error and inefficiencies.

In conclusion, NTU's proposed enhancements to its academic support systems represent a significant advancement in educational technology, offering scalable, efficient, and user-friendly alternatives to conventional note-taking methods. By adopting these technological solutions, NTU aims to improve the educational experience for its students while also achieving operational efficiencies.

## Project Description and Scope

### Automated Note-Taking Process

Implement advanced speech-to-text technologies to automate the conversion of lecture audio into accurate transcripts. This automation will be facilitated through two primary methods: direct recording of live lectures and the option for users to upload existing lecture recordings. By leveraging cutting-edge speech recognition and processing capabilities, this objective aims to eliminate the need for manual note-taking and reduce associated costs, while ensuring comprehensive and accurate documentation of lecture content.

### Database Utilization for Transcript Management

Develop a robust database system to facilitate efficient storage, retrieval, and management of lecture transcripts. This system will allow users to search for specific transcripts based on multiple parameters such as course code, semester and year, lecture number, and week. Additionally, the database will incorporate a user feedback mechanism through an upvote/downvote system to assess the quality and accuracy of the generated transcripts. Transcripts with a net score (upvotes minus downvotes) below -100 will be flagged for review or removal, ensuring the maintenance of high-quality content within the database.

### Summary Generation with Generative AI

Utilize generative AI technologies to produce concise and coherent summaries of the generated transcripts. These summaries will be tailored based on varying levels of difficulty (beginner, intermediate, advanced) and desired length (25%, 50%, 75%), catering to different student needs and preferences. The system will also allow users to upload relevant lecture slides, which the AI can use as additional context to enhance the accuracy and relevance of the summaries. This feature aims to provide students with quick and customizable access to the essence of lecture content, facilitating efficient study and revision processes.

### Interactive Chatbots for Enhanced Learning

Implement interactive AI-powered chatbots that can engage with users in a conversational manner, allowing them to ask questions related to the lecture summaries. These chatbots will utilize the information from the transcripts, summaries, and any uploaded slides to provide accurate and helpful responses. This interactive component is designed to mimic a personalized tutoring experience, enabling students to clarify doubts, explore topics in depth, and reinforce their understanding of the lecture material.

## Roles and Responsibilities

| Role | Member Name | Responsibilities |
|---|---|---|
| Project Manager | Nathanael Axel Wibisono | Overall delivery of the product |
| Lead Developer | | Overall technical lead, responsible for technical aspects of product release |
| QA Manager | Sherwin Samson | Overall product and process quality, implementation of QA processes |
| QA Engineer | | Devise test plans, conduct tests |
| Front-end Developer | Jiang Jiaxi, Liu Changsong | React front-end programming. Participate in the entire SDLC, generating work products including documentation, source code, unit and integration tests |
| Back-end Developer | Yang Yida, Sharan S/O | Server, application, and database programming. Participate in the entire SDLC, generating work products including documentation, source code, unit and integration tests |
| Release Engineer/Manager | Zhang Mengao | Create baselines and build and integrate changes for delivery. Manage releases of the product prototype. |

## Team Communication

TLA communication channels include the following: • Bi-Weekly meetings are held on Wednesday. • Group announcements and updates are sent through telegram • Telephone discussions are held as necessary. • Split up into subgroups as necessary, in order to work more co-operatively on specific problems.

# Process Definition

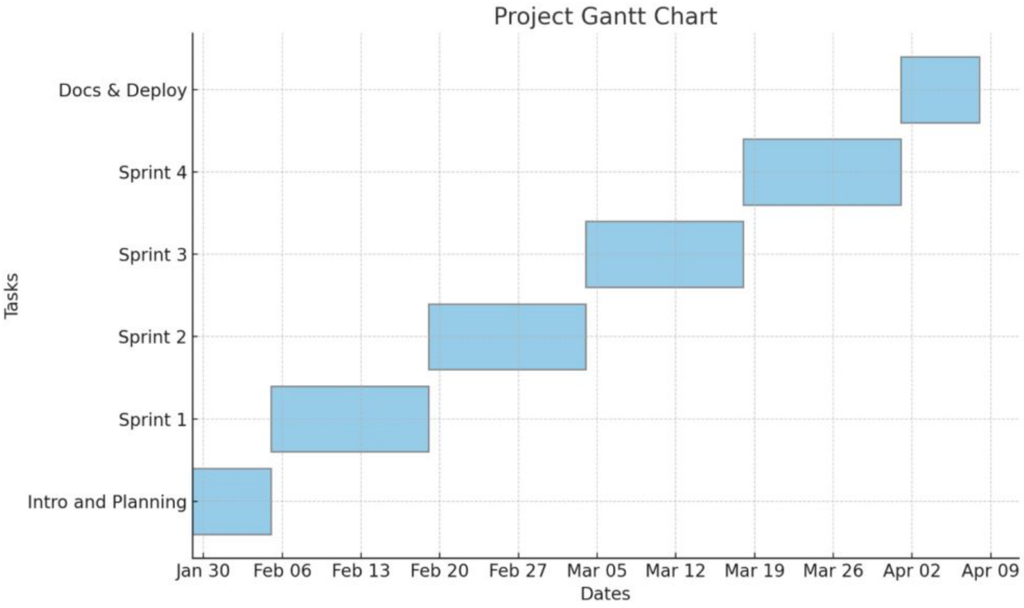
## Lifecycle Model

In addressing the complex requirements and evolving nature of our project at Nanyang Technological University (NTU), we intend to adopt the Agile lifecycle model as our guiding framework. This decision stems from our recognition of the need for a flexible, iterative approach that allows for continuous feedback and rapid adaptation to changes. The Agile model, with its emphasis on collaboration, customer feedback, and iterative development, aligns perfectly with our project's goals of developing a state-of-the-art automated note-taking system. By leveraging Agile methodologies, we will be able to more effectively manage the multifaceted aspects of our project, from the implementation of advanced speech-to-text technologies and AI-generated summaries to the creation of an interactive chatbot and a robust transcript management database.

This approach will enable us to deliver incremental value to our stakeholders, ensuring that each component of the project is developed to the highest standards of quality and efficiency, while also remaining responsive to the dynamic needs of our academic community.

# Schedule

=== Activities Dependencies and Schedule The project will adopt an Agile project management approach, allowing for iterative development, frequent reassessment, and adaptation of plans. This approach will enable the team to respond to changes in project requirements and priorities effectively. The project will be divided into two-week sprints, with each sprint focusing on a specific set of features and tasks.



Gantt Chart illustrating the project timeline and milestones.

### Week 3: Introduction and Planning

- Introduction to project scope, objectives, and team members.
- Initial planning phase, including setting up development environments and defining sprint goals.

### Weeks 4-5: Sprint 1 - Initial Development

- **Front-End Team**: Start developing the basic UI components using Next.js, Tailwind CSS, and TypeScript.
- **Back-End Team**: Set up the initial server infrastructure and database schema in PostgreSQL.
- **AI Integration Team**: Research and prototype integration methods for the Whisper API and OpenAI GPT API.

### Weeks 6-7: Sprint 2 - Feature Development and Integration

- **Front-End Team**: Continue UI development and start integrating with back-end services.
- **Back-End Team**: Develop API endpoints for handling lecture recordings and user data.

- **AI Integration Team**: Implement Whisper API for real-time transcription and integrate initial summarization features using the OpenAI GPT API.

### Weeks 8-9: Sprint 3 - Advanced Features and Testing

- **Front-End Team**: Implement advanced UI features and ensure responsive design. Start preliminary user testing.
- **Back-End Team**: Optimize database interactions and ensure security measures are in place.
- **AI Integration Team**: Fine-tune AI summarization features and test for accuracy and reliability.

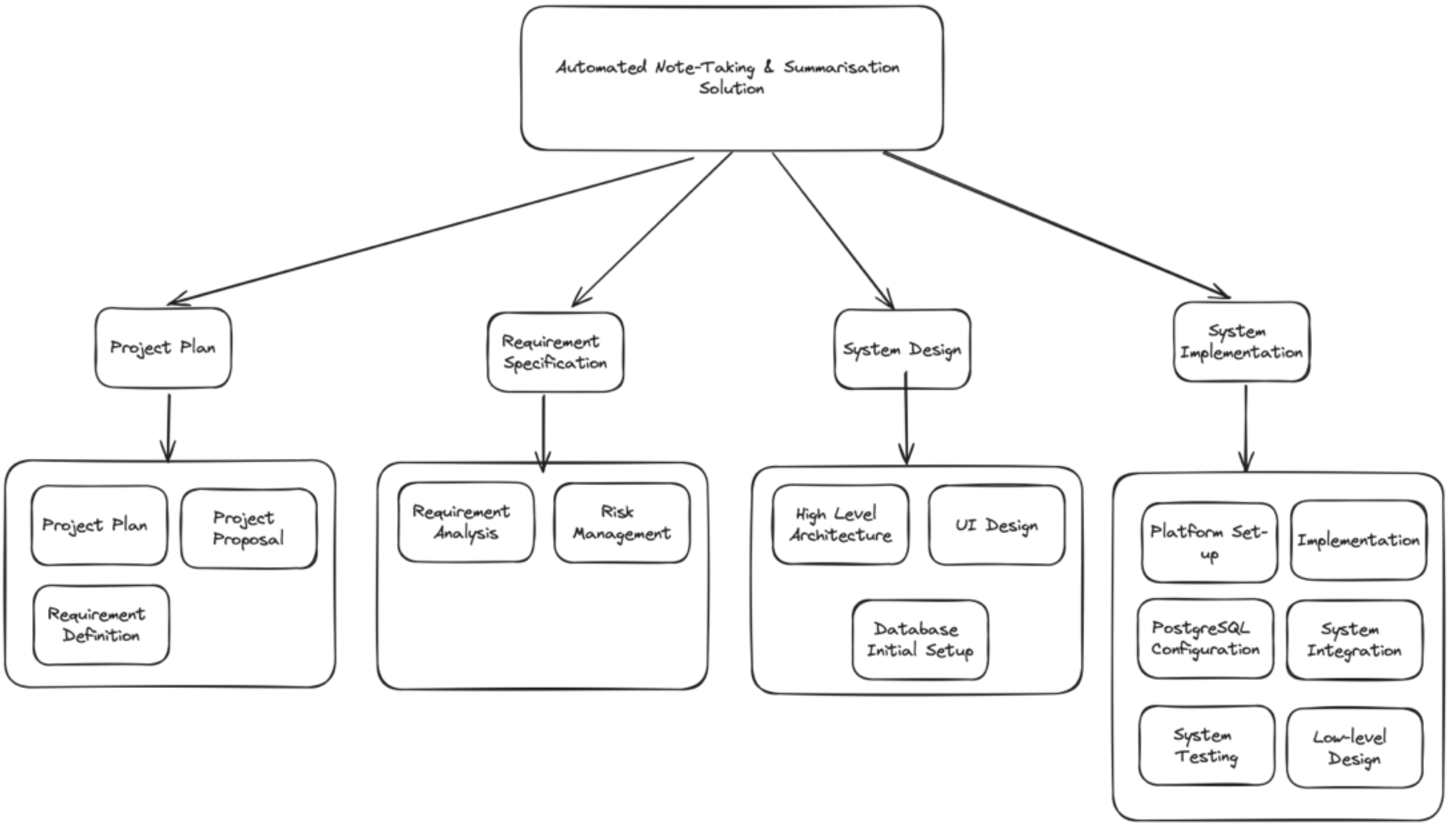
### Weeks 10-11: Sprint 4 - Finalization and Testing

- **All Teams**: Focus on bug fixing, performance optimization, and finalizing features.
- **QA Team**: Conduct extensive testing, including unit tests, integration tests, and user acceptance testing.

### Week 12: Documentation and Deployment

- **All Teams**: Prepare user manuals, technical documentation, and deployment guides.
- **Deployment Team**: Deploy the application to the production environment and perform final checks.

## Work Breakdown Structure



## Work Packages

The entire project work is broken down by the important phases of the software development life cycle. They include the following:

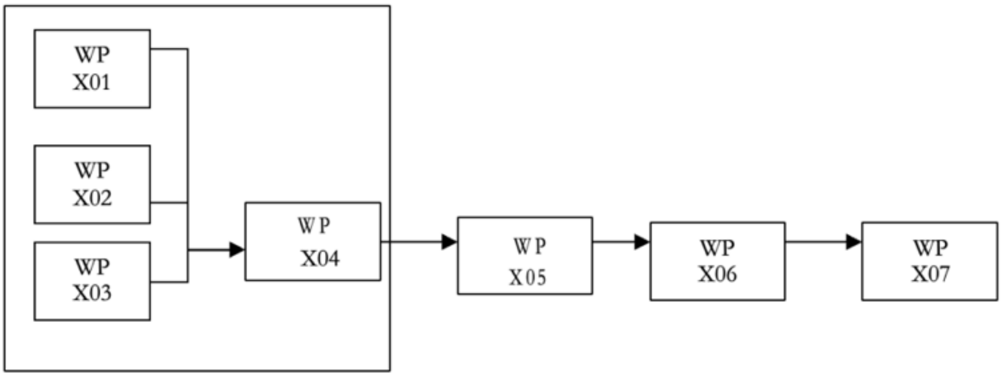
1. **Project Plan**
2. **Requirement Specification**
3. **User Interface**
4. **Technical Architecture**
5. **Data Modeling**
6. **Coding & Unit Testing**
7. **Integration & Quality Assurance**

## Activities Dependencies

The following table describes the dependencies of the deliverable work packages:

| Work Package # | Work Package Description | Duration | Dependencies |
|---|---|---|---|
| X01 | Project Plan | 7 days | -- |
| X02 | Requirement Specification | 7 days | -- |
| X03 | User Interface | 7 days | -- |
| X04 | Technical Architecture | 12.1 days | X01, X02, X03 |
| X05 | Data Modeling | 7 days | X04 |
| X06 | Coding & Unit Testing | 16.2 days | X05 |
| X07 | Integration & System Testing | 16.2 days | X06 |

The following Activity Network Diagram describes the above in more graphical detail:



## Work Package Details

Work package are listed below. a team member, indicated in bold, has been assigned as primarily responsible fo each work package and will coordinate that package.

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X01— Project Plan (1 of 7) |
| **Assigned To** | **Nathanael Axel Wibisono** |
| **Effort** | 7PD |
| **Start Date** | Lab 1 |
| **Purpose** | To determine an introductory overview of the project, to be refined in later work packages. |
| **Inputs** | None |
| **Activities** | This work package includes providing a brief overview of the project, its objectives, and a set of proposed project deliverables throughout the development of the software cycle. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report. |
| **Outputs** | A written document of the Project Plan Introduction. |

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X02— Requirement Specification (2 of 7) |
| **Assigned To** | **Nathanael Axel Wibisono** |
| **Effort** | 7PD |
| **Start Date** | Lab 2 |
| **Purpose** | To establish a common understanding between the customer and the software project team of the customers' requirements to be addressed by the project |
| **Inputs** | Customer's requirements |
| **Activities** | Identify "the customer", interview customer, write and inspect customer requirement and build requirements. |
| **Outputs** | A written document of the requirement specification. |

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X03— User Interface (3 of 7) |
| **Assigned To** | **Zhang Mengao**,Yang Yida, Liu Changsong |
| **Effort** | 7PD |
| **Start Date** | Lab 2 |
| **Purpose** | To build the user interface between the system and the customer, to make it easy to use, and friendly to the customer |
| **Inputs** | User information |
| **Activities** | To get the user information, user request, display the dialog between system and user, display the result of request |
| **Outputs** | User Interface |

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X04— Technical Architecture (4 of 7) |
| **Assigned To** | **Nathanael Axel Wibisono** |
| **Effort** | 12.1PD |
| **Start Date** | Lab 2 |
| **Purpose** | To do the high level architecture |
| **Inputs** | Project Plan Work Packages (X01 to X03 inclusive). |
| **Activities** | High level design entails defining the architecture of the software system and identifying the various components and how they are inter-related to and interactive with each other. Designers also need to decide on the software and hardware infrastructures, such as what operating system on which the software is built, the language used to implement the software, and so on. Design topics including maintainability, portability, and reusability will be addressed here as well. |
| **Outputs** | High Level Design and Architectural Specification. |

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X05— Data Modeling (5 of 7) |
| **Assigned To** | **Jiang Jiaxi**, Sharan S/O, Sherwin Samson |
| **Effort** | 7PD |
| **Start Date** | Lab 3 |
| **Purpose** | To build the project's database |
| **Inputs** | Project Plan Work Packages (X01 to X05 inclusive). |
| **Activities** | Analyze the data flow relationships, entity relationships |
| **Outputs** | A written document of the data modeling |

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X06— Coding & Unit Testing (6 of 7) |
| **Assigned To** | **Nathanael Axel Wibisono**,Zhang Mengao,Yang Yida, Liu Changsong, Jiang Jiaxi, Sharan S/O, Sherwin Samson |
| **Effort** | 16.2PD |
| **Start Date** | Lab 3 |
| **Purpose** | To implement the system as per the requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing. |
| **Inputs** | Project Plan Work Package X06. |
| **Activities** | Programmers will implement the modules according to the design specifications noted in the Specification document. |
| **Outputs** | Source code and header files |

| | |
|---|---|
| **Project** | Automated Note-Taking & Summarisation Solution |
| **Work Package** | X07— Integration & System Testing (7 of 7) |
| **Assigned To** | **Nathanael Axel Wibisono**,Zhang Mengao,Yang Yida, Liu Changsong, Jiang Jiaxi, Sharan S/O, Sherwin Samson |
| **Effort** | 16.2PD |
| **Start Date** | Lab 4 |
| **Purpose** | To identify and fix logical and syntactical errors produced during the implementation of the System, and setting up drivers and stubs to see how the module responds to various inputs. Black box testing as well as white box testing might be conducted to check for logical errors. All the testing procedures will be documented in the Test Plan report. If problems are found, they will be noted and fixed at the earliest possible time. |
| **Inputs** | Project Plan Work Package X07. |
| **Activities** | The Integration testing team may try to simulate how a user might interact with the system. Similar to Unit Testing, Integration Testing may require the development of stubs and drivers as well, but here this is more geared towards the higher (overall system) level. Testers may also examine issues such as system performance and integrity. Heuristics assessment plays an important role in this work package, as intelligence components will define eventual system success. |
| **Outputs** | A test report. |

# Project Estimates

## Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

## Unadjusted Function Points

The project support the following proposed functions:

| Component | Description | Complexity |
|---|---|---|
| **Inputs** | Automated speech-to-text conversion, User uploads for lecture recordings, User edits to transcripts, Slide uploads for contextual enhancement | Medium |
| **Outputs** | Text transcripts, Summarized texts, Edited transcripts, Integration with database for transcript management | Medium |
| **Inquiries** | Database searches based on multiple parameters, User feedback mechanism through upvote/downvote, Chatbot interactions for information and clarification | High |
| **Logical Files** | Database for storing and managing transcripts and summaries, User accounts and preferences storage, Metadata storage for lecture information | High |
| **Interfaces** | User interface for transcript and summary interaction, Administrative interface for database management, Chatbot interface for user interaction | Medium |

| Characteristic | Low (3 FP) | Medium (4 FP) | High (6 FP) | Total |
|---|---|---|---|---|
| **Inputs** | 1 × 3 | 0 × 4 | 0 × 6 | 3 |
| **Outputs** | 0 × 4 | 0 × 5 | 1 × 7 | 7 |
| **Inquiries** | 0 × 3 | 1 × 4 | 0 × 6 | 4 |
| **Logical Files** | 0 × 7 | 0 × 10 | 2 × 15 | 30 |
| **Interfaces** | 0 × 5 | 2 × 7 | 0 × 10 | 14 |
| **Total Unadjusted FP** | | | | **58** |

## Adjusted Function Points

| Influence Factors | Score | Detail |
|---|---|---|
| Data Communications | 3 | The application supports a standard communication protocol for data transfer. |
| Distributed Functions | 2 | The project involves some level of distributed processing, particularly with cloud storage and processing. |
| Performance | 4 | High performance is crucial due to real-time processing of audio and large text data. |
| Heavily Used Configuration | 2 | The system will be regularly used by students and educators, with moderate load expected. |
| Transaction Rate | 2 | The system will handle a moderate number of transactions, particularly during lecture hours. |
| Online Data Entry | 4 | A significant portion of the system's functionality involves interactive data entry, such as uploads and edits. |
| End-User Efficiency | 3 | The system is designed with several features to enhance end-user efficiency, like summary generation and search functionalities. |
| Online Update | 4 | Real-time update and editing of transcripts and summaries are critical features. |
| Complex Processing | 3 | The system includes complex processing, such as speech-to-text conversion and generative summary. |
| Reusability | 2 | Components of the system are designed to be reusable, although this is not the primary focus. |
| Installation Ease | 1 | The system is web-based, requiring minimal installation effort from the end-user perspective. |
| Operational Ease | 2 | The system includes features for easy operation, but some manual configuration is necessary for setup. |
| Multiple Sites | 1 | The system is designed to be accessed from multiple locations, but does not require significant customization for each site. |
| Facilitate Change | 3 | The system is designed to be flexible, with features allowing for easy updates and changes based on user feedback. |
| **Total score** | | **36** |
| **Influence Multiplier** | | **= Total score × 0.01 + 0.65 = 36 × 0.01 + 0.65 = 1.01** |
| **Adjusted FP** | | **= Unadjusted FP × Influence Multiplier = 58 × 1.01 = 58.58** |

### Lines of Code

According to Capers Jones statistics, each Function Point requires 29 lines of code if the application is implemented using Typescript. Therefore, we have: Lines of Code = 58 FP × 29 LOC/FP = 1682 LOC

## Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (1682 LOC) / (39 LOC/PD) = 43 PD
- Duration = 3 × (Effort)^(1/3) = 3 × (43)^(1/3) = 5.05 Days
- Initial schedule = 5.05 Days / 5 days a week = 1.01 Weeks
- Team size = 43 PD / 5.05 D = 8.51 P ≈ 8 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 43 PD × 8 hours = 344 PH

### Distribution of Effort

| 1990's Industry Data | Work Package | Distribution | Estimates (PD) |
|---|---|---|---|
| Preliminary Design 18% | Project Plan | 9% | 30.96 |
| | Requirement Specification | 9% | 30.96 |
| Detailed Design 25% | User Interface | 7% | 24.08 |
| | Technical Architecture | 11% | 37.84 |
| | Data Modeling | 7% | 24.08 |
| Code & Unit Testing 26% | Code & Unit testing | 21% | 72.24 |
| | Online Documentation | 5% | 17.2 |
| Integration & Test 31% | Integration & Quality Assurance | 31% | 106.64 |
| **Extrapolated total effort** | | | **344** |
| 2% for project management | | | 6.88 |
| 3% for contingency | | | 10.32 |
| **Total effort** | | | **361.2** |

These duration estimates are based on the assumption that each team member works an equal amount on any given work package.

## Cost Estimates

| Category | Description | Cost |
|---|---|---|
| **Hardware** | Developer workstations: 7 - (Specifications not provided, assumed covered by TLA or client) | $0.00 |
| **Software** | **Free License-based Software:**<br>- Vercel (Free Tier)<br>- Other Open Source Software as required<br>**Software License Provided by Third Party:**<br>- API costs for Whisper (Assumed cost)<br>- API costs for ChatGPT 4 (Assumed cost) | $0.00 (Vercel and Open Source)<br>API costs not specified |
| **Other Resources** | **Staff:**<br>- 7 Employees with 344 working hours at $30/hour | $72,240.00 |
| **Stationary** | Paper, photocopying, and other miscellaneous costs | $50.00 |
| **Total** | | **$72,290.00** |

## Product Checklist

| Project Deliverable | Estimated Deadline |
|---|---|
| Project Plan | Lab 1 |
| Requirements Specification | Lab 2 |
| Design Document | Lab 3 |
| Module/System Test Plan | Lab 4 |
| System Release (Demo) | Lab 5 |

## Best Practice Checklist

| Practice Area | Guidelines |
|---|---|
| Documentation | Document all processes; ensure documentation is in a standardized format. Include complete functional specifications in requirement documentation to avoid ambiguity, and ensure completeness, accuracy, and consistency. |
| Complexity Management | Keep it simple by minimizing interfaces between modules, procedures, data, and people to reduce communication costs. Avoid unnecessary features; design should meet customer requirements. |
| Visibility | Ensure visibility of the development process. This includes effective manager-employee communication, making code available for review, and reviewing designs for appropriateness. |
| Continuous Change | Plan for continuous change by using revision numbers and dates, revision history comments, and change marks on all manuals, designs, tests, and source codes. Ensure new revisions are approved and checked for quality and compliance. Utilize a configuration management system. |
| Estimation | Obtain accurate estimates for time, effort, overhead, meeting time, and especially for integration, testing, documentation, and maintenance. Avoid underestimation. |
| Code Reviews | Conduct code reviews as they are an efficient method to find software defects. Plan and manage code reviews between team members. |
| Software Testing | Implement both black box and white box testing methodologies, involving unit, functional, integration, and acceptance testing. |

## Risk Management

### More changes to requirements than anticipated

- **Impact Severity:** High
- **Probability:** 25%
- **Impacts:** Depending on the stage at which changes occur, could range from needing to update the requirements documentation to needing to do a complete redesign.
- **Risk Reduction:** Be rigorous in eliciting requirements. Make the customer aware of the potential repercussions of requirement changes.

### Specification Delays

- **Impact Severity:** High
- **Probability:** 15%
- **Impacts:** Delay in finalizing the specification will push the schedule for all following stages of the project.
- **Risk Reduction:** Monitor the progress of specification carefully.

### System size underestimated

- **Impact Severity:** Moderate
- **Probability:** 30%
- **Impacts:** More work will need to be spent on design and coding; could negatively impact the schedule.
- **Risk Reduction:** Update estimates often as the project progresses.

### Staff leaving before project complete

- **Impact Severity:** Extreme
- **Probability:** 5%
- **Impacts:** There would be more work for the remaining employees, and any specialized skills or knowledge would be lost.
- **Risk Reduction:** Offer benefits and incentives to staff.

### Problems coordinating within the group

- **Impact Severity:** Moderate
- **Probability:** 40%
- **Impacts:** Members may be unaware of what is expected of them; managers may not be able to measure progress; portions of projects not completed.
- **Risk Reduction:** Follow communication plans as documented.

### Customer cancels project

- **Impact Severity:** Super-Extreme!
- **Probability:** 1%
- **Impacts:** All work will have been wasted.
- **Risk Reduction:** Keep in close contact with the customer. Ensure that they have some market research indicating a demand for this product.

## Quality Assurance

The note-taking solution project will follow the organization's established quality assurance standards. The Quality Plan will detail specific procedures and guidelines to ensure product quality.

### Test Procedures

The Module/System Test Plan will document specific test procedures and details, guiding the testing phase of the project.

### Testing Methodologies

We will utilize two main testing methodologies for the project:

- **Unit Testing:** Individual components of the note-taking system, such as the speech-to-text conversion and summary generation modules, will be tested to verify their functionality.
- **In-Place Testing:** The entire system, including integration between different components like the user interface, database, and processing algorithms, will be tested as a cohesive unit.

### Testing Aspects

Key functionalities of the note-taking system will be subjected to thorough testing:

- **System Function:** Tests will focus on identifying and eliminating any software bugs to ensure a stable and reliable system.
- **Algorithmic Function:** The effectiveness and realism of algorithmic features, such as the accuracy of speech-to-text conversion and the relevance of generated summaries, will be tested to ensure they meet user needs and expectations.

### Use of Test Cases

Realistic test cases will be extensively used to evaluate the system. We will provide a detailed subset of test data relevant to typical note-taking scenarios, including challenging cases such as lectures with background noise or varied accents, to verify the system's robustness and adaptability.

## Monitoring & Control

Effective monitoring and control are crucial for the successful progression of the note-taking solution project. Key procedures include:

**Quantitative measurement of resource consumption**: Estimations of resource requirements, especially human resources, will serve as a quantitative gauge of project progression. By comparing these estimations with actual progress at various milestones, we can effectively track project advancement. This document provides percentage estimates of resource requirements for each milestone, facilitating straightforward progress monitoring.

**Identification of major project risks**: Prompt identification of significant risks allows for early implementation of preventative measures. The Risk Management section of this document details these risks and outlines strategies to mitigate them, ensuring the project remains on track.

**Regular reviews of project progress**: The project team will conduct regular meetings to review the progress across all tasks, including planning, design, development, testing, and documentation. These reviews are essential for maintaining alignment with the project objectives and timelines.

**Timeline Planning and task decomposition**: A detailed timeline for the project has been developed by breaking down tasks into measurable subcomponents. This approach not only aids in accurate timeline creation but also facilitates task assignment and workload distribution. During the implementation phase, these subcomponents enable precise measurement of progress. The project's subcomponents and their estimated timelines are outlined in the Estimates and Work Breakdown Structure sections of this document.

Retrieved from "http://155.69.100.30/s22324SCELMagnificent7/index.php?title=Project_Plan&oldid=164"