

# 総理大臣ページ生成課題計画書

作成者 佐々木順平 (PM)

## プロジェクト予定期間

2021 年 10 月 29 日 ～ 2021 年 12 月 29 日

## プロジェクトメンバー

佐々木順平 953661(プロジェクトマネージャ)  
岡本悠里 954480(サブプロジェクトマネージャ、プログラマ)  
近藤英雅 953580(設計)  
山内龍我 854003(設計)  
岡山紘大 953300(プログラマ)  
梶原隆太郎 953373(チェッカ)  
杉橋真輝 953760(チェッカ)

## 0. 要旨

本プロジェクトは京都産業大学情報理工学部 2021 年秋学期「プログラミング言語」の長期課題の一部である。総理大臣の情報を記した CSV ファイルを読み込み、その情報を一覧形式で表示するプログラムを Java と Python の両方で作成する。また、歴代の徳川幕府の将軍に関する情報を記した CSV ファイルも同様にして変換する。今回の課題では複数人でプロジェクトを行う事は必須ではないが独自にプロジェクトチームを編成する。クライアントを本講義の担当講師青木淳先生と仮定し、プロジェクト演習を行う。クライアントから要求仕様書を受取り、要求仕様書に従ったプログラム群を完成させることを目標とする。本チームではこの目標を達成するために、クライアントとの協議を始め、試行錯誤を繰り返し、その過程を記録、反省することにより良い成果物を提出する方針で開発を進める。開発はプロジェクト管理の知識体系 PMBOK[1]に沿って行う。本計画書ではプロジェクトの計画について記す。

## 1. プロジェクトスコープマネジメント

### 1.1. 立ち上げ

総理大臣ページ作成課題を行うにあたって、プロジェクトチームを立ち上げる。プロジェクトマネージャ(以下、PM)を佐々木順平とし、プロジェクトを発足する。  
(発足日:2021/10/29)

### 1.2. スコープ計画

クライアント(青木先生)より、要求仕様書を受け取る。要求仕様書より作業範囲を確認する。確認内容は以下の通りである。

#### ・基本方針

起動時: Web ページが一枚開く

設計: UML を使って設計。青木先生から配布された設計図を活用する。

実装: コーディング規則に準ずる。

全ての名称は省略せず、意味を明確に示す。

テスト方針: 内容はテスト仕様書にまとめる。

Junit や unittest を用いてテストケースを行う。

リリース方針: オープンソース

リポジトリからチェックアウトして全ソースプログラムを獲得。ソースプログラムから実行モジュールであるアプリケーションを構築するための手段(Makefile など)を提供する。アプリケーションは簡単なコ

マンドにより起動する。マニュアルをリリースに含める。

#### ・基本要件

アプリケーションの用途：総理大臣の情報を Web ページで閲覧できるようにする。

対象者：コンピュータの利用者

ハードウェア：Apple Macintosh シリーズ

Processor：Intel Core 2 Duo 2GHz クラス以上

Memory：1GiB 以上

Hard Disk：250MiB 以上の空き容量

ソフトウェア構成：OS X Yosemite (10.10) / OS X El Capitan (10.11) / macOS Sierra (10.12) 以上

プログラミング言語：Java：JDK 16 以上 Python：Python 3.8 以上

目標性能：一つのウィンドウの Web ページに表示させるようにする

総理大臣の情報が表として表示されるようにする

CSV という形式の言語を、html という形式の言語に変換する

#### ・開発体制

PM は開発スケジュールの段階ごとに開発書と仕様の確認をする。開発の不明点はクライアントに確認する。

#### ・開発スケジュール

開発スケジュールは開発計画の Web ページ内にガントチャートとして記載する。レベルごとに開発期間を決め、全体の開発期間は課題提出期限である 2021 年 12 月 31 日とする。本演習で行うプロジェクトは、レベル 1 の完成が最低条件とされ、レベル 7 までの完成は必須ではなく、自由挑戦課題である。そのため、レベル 1 を仕上げることを第一目標とし、完成するごとにレベルを上げるような開発スケジュールを立てる。

## 2. プロジェクトタイムマネジメント

### 2.1. アクティビティ

アローダイアグラムを作成する。今回のプロジェクトでは、ベースとして青木先生から提供されたものを用いる。

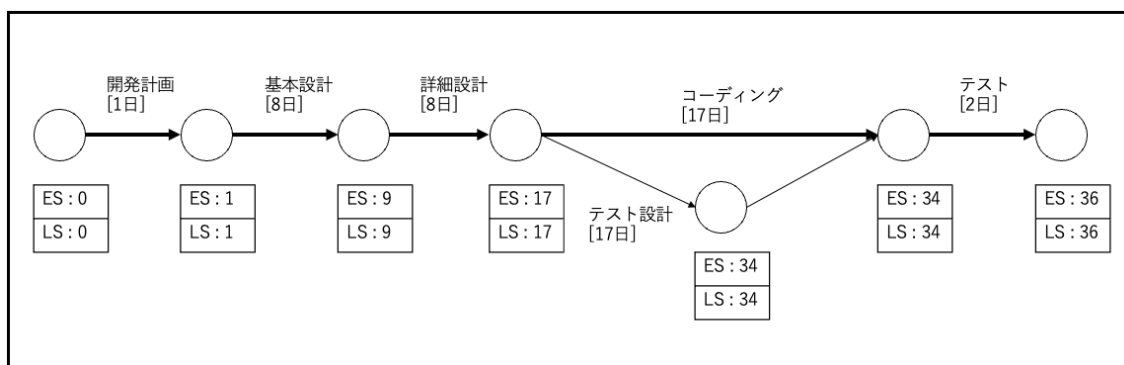


図 1 PERT 図

コーディングの PERT 図は以下のようにする。

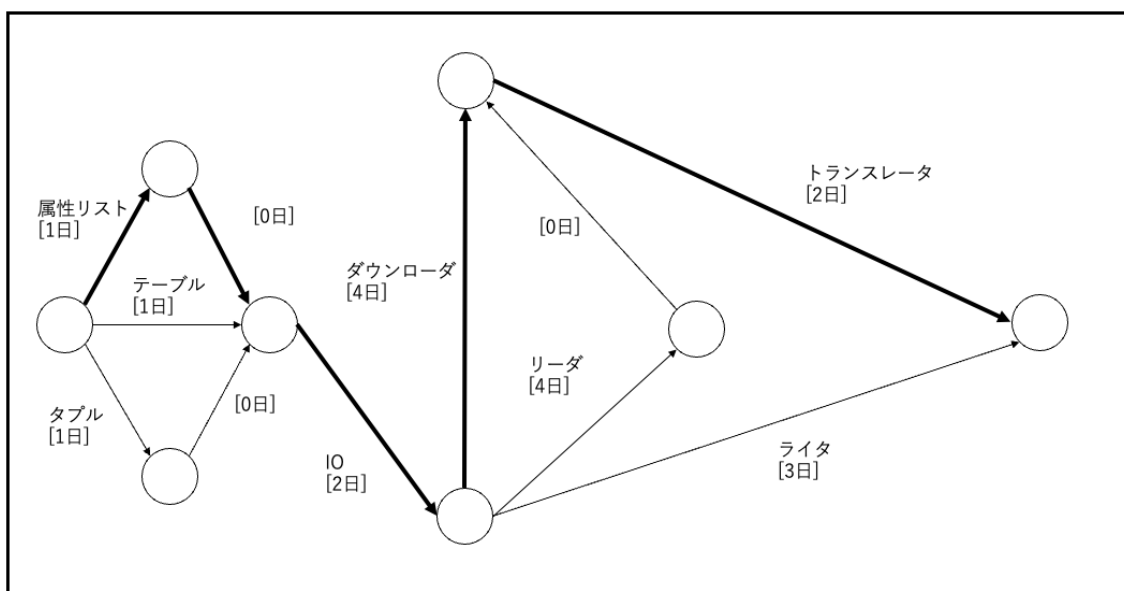


図 2 コーディングの PERT 図

## 2.2. スケジュール作成

2.1 をもとにガントチャートを作成する。作成したガントチャートを以下に示す。使用ツールとして GantProject[2]を用いる。

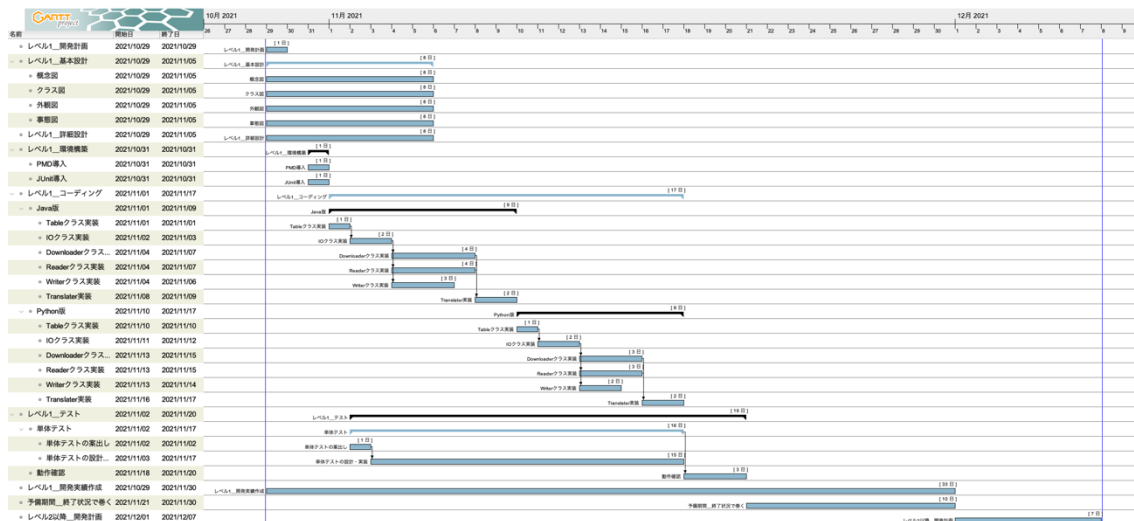


図3 ガントチャート(レベル1)

レベル1の完成を最低限の目標するため、レベルごとの大まかな開発期間を定める。レベル1のプログラムが完成、または、完成目前になった場合、進度に応じてレベル2以降の開発期間を決定する。

(2021/11/28)

レベル2のガントチャートを以下に示す。

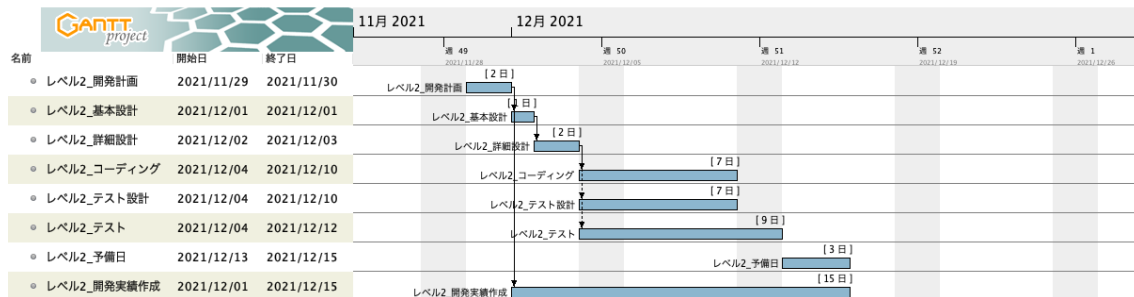


図4 ガントチャート(レベル2)

レベル2のプログラムが完成、または、完成目前になった場合、進度に応じてレベル3以降の開発期間を決定する。

(2021/12/10)

レベル3のガントチャートを以下に示す。



図5 ガントチャート(レベル3)

## 2.3. スケジュールコントロール

計画通りに行われているかどうかの進捗状況をPM、またはサブPMが把握しておく。予定の日程からの遅れは、各週に設けた全体会議において、全員で共有し、状況の再確認と開発体制の見直しを行う。予定の日程より早い場合、完成するごとに次のレベルのものを作成するため、そのまま進行する。

## 3. プロジェクトコストマネジメント

### 3.1. 資源計画

必要となる人員は7名である。プロジェクト開始から全員が必要となる。メンバを以下に示す。

佐々木順平 953661

岡本悠里 954480

近藤英雅 953580

山内龍我 854003

岡山紘大 953300

梶原隆太郎 953373

杉橋真輝 953760

### 3.2. コスト見積もり

今回のプロジェクトは大学の授業で行われる演習である。新たな人員以外の資源を必要としないため、今回は考慮しない。

### 3.3. コストの予算化

省略

### 3.4. コストコントロール

省略

## 4. プロジェクト品質マネジメント

### 4.1. 品質計画

品質は単体テスト、及び、全体の動作確認をメンバ全員で検査することによって行う。unittest によってテストを行う。項目は以下の通りである。

- ・ html への変換が正しくできているかの確認テスト
- ・ .csv ファイルが正しく読み込めているかの確認テスト
- ・ テーブルがタプルの中身を正しく返してくれるかの確認テスト
- ・ 日数計算が正しくできているかの確認テスト

### 4.2. 品質保証

行ったテストの詳細をテスト仕様書、テストの結果をテスト結果のドキュメントにまとめ、提出する。

### 4.3. 品質管理

品質全体の方針決定、監督及び管理は PM が統括する。サポートとして、PM 経験者のサブ PM が担当する。

## 5. プロジェクト人的資源マネジメント

### 5.1. 組織計画

このプロジェクトは京都産業大学の講義、プログラミング言語の演習である。メンバはその講義の受講生となり、クライアントは教師である。メンバの技術的役割は以下に示す。

- (プロジェクトマネージャ) 佐々木順平
- (サブプロジェクトマネージャ) 岡本悠里
- (設計) 近藤英雅
- (設計) 山内龍我
- (プログラマ) 岡山紘大
- (プログラマ) 岡本悠里
- (チェッカ) 梶原隆太郎
- (チェッカ) 杉橋真輝

(2021/11/06)

プログラム班に設計班が兼任という形で配属。Python プログラムを担当。

## 5.2. 要員調達

5.1. に基づき、要員調達を行った。全員を調達することができた。

## 5.3. チーム育成

プロジェクト完遂のため、チーム意識を持ち、行動する方法として、定期的に会議を行い、報告などをこまめに行えるようにする。会議は音声通話を原則必須とする。会議の進行は PM が中心となって行い、全員が話せるようなオープンな議論を心がける。

# 6. プロジェクト伝達マネジメント

## 6.1. コミュニケーション計画

情報共有の手段として、メンバ間の連絡は Discord[3] を使用する。メンバ全員が参加したサーバを新たに立ち上げる。サーバ内は役職別などにカテゴリを分割し、それぞれチャンネルを設立する。また、会議の内容は議事録として保管する。

PM が中心となり、週 1 回以上の会議を行い、進捗状況を確認する。各役職から最低 1 人は参加できる日程を毎週全員で決定する。メンバ全員が連携をとりやすい環境を心がけ、全体が状況把握できるようにする。役職間での連携をより良くするため、役職ごとで自由に臨時的な会議を行ってもらい、その実績は全体がわかるようにする。

## 6.2. 情報配布

メンバ間の情報、資料の共有は 6.1. で設立した Discord で行う。バージョン管理を行い、メンバ内の情報が共通認識として得られるようにする。Discord 内の情報管理は PM が統括する。

完成した成果物はリポジトリである GitHub[4] を用いて、全ドキュメント、全ソースコードを提供する。

## 6.3. 実績報告・完了手続き

報告書を作成し、実績を記す。

# 7. プロジェクトリスクマネジメント

## 7.1. リスクマネジメント計画



リスクの特定はプロジェクトの進行過程で、その都度行う。新たなリスクが生じた際、Discord 内の全体共有チャットにて報告する。急を要する場合は、臨時会議をもうけ、最優先事項でなければ、定期会議で協議する。リスク管理は PM が統括する。

## 7.2. リスク識別

以下に、現在考えられるリスクを記す。

### ・環境が原因のリスク

自然災害や感染症の拡大により、授業の継続、プロジェクトの進行が困難になる。

### ・メンバに発生するリスク

PC やネットワークの故障によって作業が進められなくなる。

事故や病気により参加が困難になる。

感染症のクラスタがメンバ内で発生する。

### ・クライアントに発生するリスク

事故や病気などにより参加が困難になる。

## 7.3. 定性的リスク分析

7.2. で想定したリスクの評価を以下にまとめた。

表 1 定性的リスク分析結果

リスク ランキング	リスク	発生確率	影響度
1	自然災害や感染症の拡大により、授業の継続、プロジェクトの進行が困難になる。	中	大
2	PC やネットワークの故障によって作業が進められなくなる。	中	中
3	感染症のクラスタがメンバ内で発生する。	小	大
4	事故や病気により参加が困難になる。(メンバ)	小	中
5	事故や病気などにより参加が困難になる。(クライアント)	小	大

## 7.4. リスク対応計画

7.3. をもとに対応を計画する。

メンバが短期の間参加できない場合、各班 2 人いるため、参加できないメンバと同じ班の人が引き継ぎ、遅れが生じる可能性があれば、他の班から作業状況を考慮してサポートに入ってもらおう。長期の場合、教師に相談のもと、残りのメンバの業務の振り分けによって、作業を続行できるのか判断する。仮に、続行不可と判断される場合、新たなメンバを追加する、またはレベル 1 の完成を最終目標として取り組む。

授業継続不可となるような、自然災害、感染症の流行などがあった場合、教師の指示に従う。演習が継続されるのであれば、Discord にて作業を進める。集団感染のリスクを下げるため、原則オンラインによる会議とするが、対面で会話する場合はマスクの着用を必須とする。

## 8. 出典

[1] PMBOK® -Project Management Institute

<https://www.pmi.org>

[2] GanttProject

<https://www.ganttproject.biz>

[3] Discord

<https://discord.com>

[4] GitHub

<https://github.com>