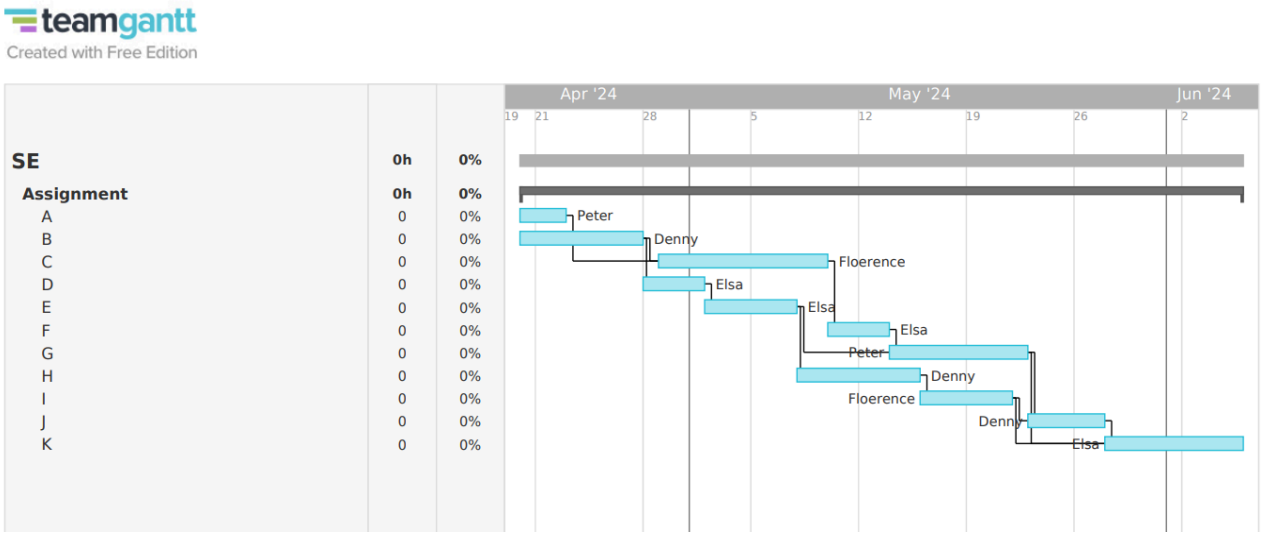
HW3 - Group 15

1. Project Planning and Control

I. Gantt Chart

We use [teamgantt] to create the Gantt Chart.

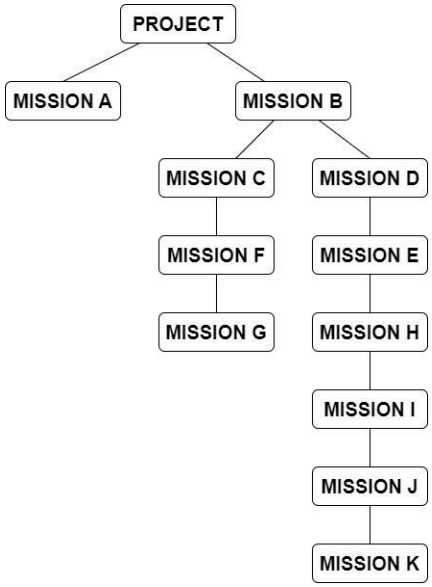


II. Work Breakdown Structure (WBS)

- Work Breakdown Structure as Indented List:

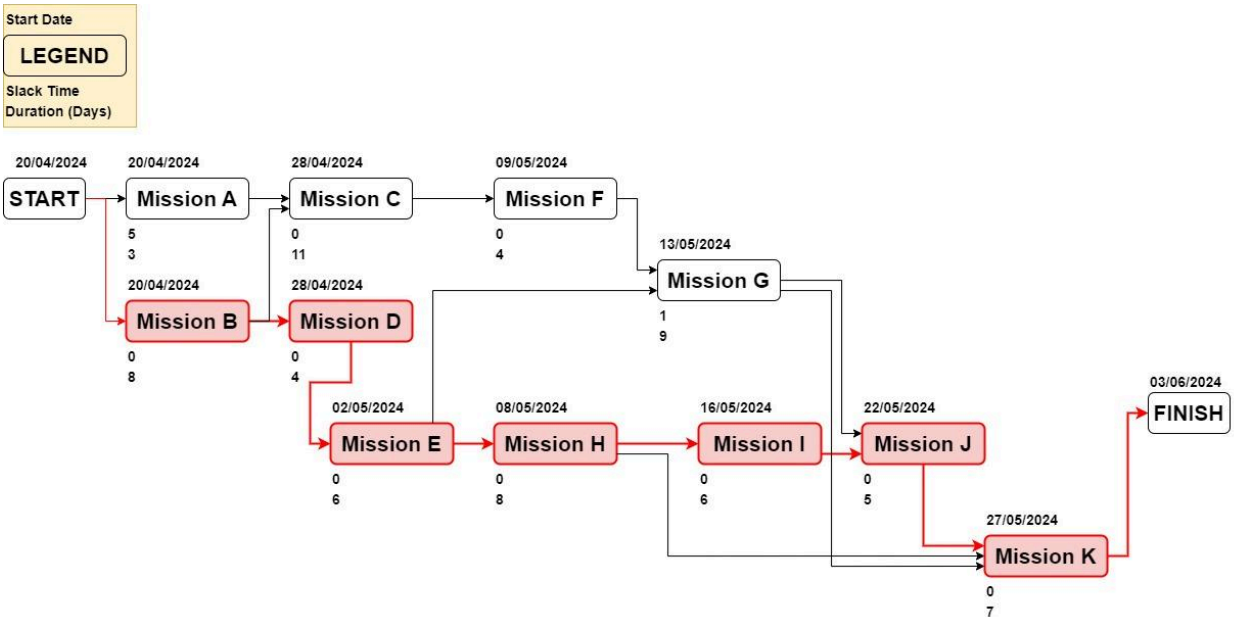
1. Project
    1.1. A
    1.2. B
        1.2.1. D
            1.2.1.1. E
                1.2.1.1.1. H
                    1.2.1.1.1.1. I
                        1.2.1.1.1.1.1. J
                            1.2.1.1.1.1.1.1. K
        1.2.2. C
            1.2.2.1. F
                1.2.2.1.1. G

- Work Breakdown Structure as a Tree (We use draw.io to create the WBS Tree) :



III. PERT chart (Program Evaluation and Review Technique)

We use draw.io to create the PERT chart. The parts that are highlighted in red color is the critical path because the missions on the critical path cannot be delayed without delaying the entire project. Therefore, the critical path helps project managers identify which tasks are most critical to the project's timeline.



2. Imagining ourselves as project managers of a startup company

I. Project Selection and Objectives

For our project, our group has selected “Disney+”. Disney+ is streaming service which was developed by the Disney Company and it is mainly focusing on providing family genre movies, series and other original content. For example, from big franchises such as Disney, Pixar, Marvel, Star Wars and so on.

Expected Functionalities and Objectives:

- **Streaming Content:** Capability to transmit videos of different qualities, from streaming services 4K resolution.
- **Search and Discovery:** Search function that copes with complexities in order to get desired results; personalized recommendations based on how the user behaves.
- **Multi-Platform Support:** Working both with multifunctional devices like smartphones, tablets, personal computers or smart TVs.
- **User Profiles:** Besides having multitude of user profiles with parental controls, you can modify the environment for each user.
- **Offline Viewing:** Download feature for content to be viewed offline.

The paragraph says (page 65- 68) from Chapter 3, details like a scope, requirements and software features with user in mind. Thus, it clearly outlines comprehensive project planning for this to be successful.

II. Resources and Team Members

Although we believe that Disney+project requires numerous amount of talented workforce, our group believe that the roles mentioned below are a must and the core of the project. For a project like Disney+, the following resources and team members are required:

- **Software Engineers:** Backend/Frontend development.
- **Quality Assurance Testers:** Ensure the application runs smoothly on all supported devices.
- **UI/UX Designers:** Design an user-friendly and easy to use interface.
- **Data Scientists:** See user analytics and help make data driven decisions in the future.
- **Project Managers:** Meet certain deadlines and be organized.
- **Security Experts:** Handle user data and system protection.

III. Software Process Model

Since the platform is meant for streaming, most Agile methodology will be suitable for the following reasons:

- **Iterative Development:** Empowering the project manager to pursue a steady monitoring of the project's trajectory and a corruption free adjustment of the operational requirements.
- **User Feedback Integration:** Offer frequent updates working on users feedback, the product can be improved on and in turn the users will be more happy.
- **Collaboration and Communication:** Ensure engagement between device manufacturers, designers and stakeholders for optimal outcomes in the performance.

Chapter 2 considers the numerous software process models, and it accordingly comes out that Agile is most suitable for projects that require dynamic requirements, as well self-adaptability.

IV. Model Stages and Tasks

Model Stages:

1. **Requirements Planning**
    - Request project requirements during initial meetings with stakeholders.
    - Define minimal viable product (MVP) features for early release.
2. **Design and Prototyping**
    - Develop initial design and user interfaces.
    - Create prototypes for user testing.
3. **Development Sprints**
    - Implement features in two-week sprints.
    - Conduct daily stand-ups to discuss progress and obstacles.
4. **Testing**
    - Continuous integration and testing of new features.
    - User acceptance testing to gather feedback.
5. **Deployment and Evaluation**
    - Regular deployment of updates to the production environment.
    - Monitor system performance and user satisfaction.

While the Tasks and stages are presented in Chapter 3, the agile philosophy is highlighted out via the iterative and the continuous features at the core of methodologies. Thus, the group will also become provided with a comprehensive foundation that will help us to master the execution of the developmental process from strategic planning to the evaluation of performance in the methodologies mentioned above.

3. Applying Lean Manufacturing Principles to Agile Software Development: A Case Study in Crowdfunding Platform Development

I. We decided to choose Kickstarter as a crowdfunding platform. The existence of Kickstarter will allow both individuals and small businesses to obtain financial assistance publicly through their creative projects. This platform also leads to a project that involves several categories, such as arts, fashion, technologies, etc. Moreover, Kickstarter provides us the compatibility between one and another devices, giving both PC users and mobile users seamless access to this platform, which clearly increases the level of effectiveness

and efficiency. Overall, Kickstarter has some nice features and services such as user comments, project showcase pages, and funding progress tracking.

II. The 7 types of waste during the project development process:

- **Transportation**
  This type of waste sometimes may force us to pay some additional fees like machinery, space, and time. This could happen because when we want to move material or resources from one place to another, the movement doesn't add value to the product. But in most cases, the excessive movement could damage the quality of our product.

- **Inventory**
  Although excessive inventory will increase storage and depreciation costs, in reality, these excessive inventories often don't meet customers' needs and don't increase our product's value. In these situations, businesses overstock to cover unforeseen demand, and guard against poor quality, manufacturing delays, and other issues.

- **Motion**
  Motion that includes employee and machinery movement sometimes can cause injuries, extended production time, and more. Overall, make all the necessary arrangements to ensure that employees are required to perform as little as feasible in order to complete their tasks.

- **Waiting**
  Waiting waste is one of the types of waste that we often encounter in our daily lives. This could happen because whenever goods or tasks are not moving, this "waiting waste" occurs. Some instances of waiting waste are waiting for third-party integrations, documents waiting for the executive's approval, equipment waiting that needs to be fixed, etc.

- **Overproduction**
  Producing more entails exceeding consumer demand, which raises expenses. In fact, the other six wastes could arise as a result of overproduction. It can cause other waste because if we do overproduction, it means we require additional transportation, greater waiting time, excessive motion, etc.

- **Over-processing**
  Over-processing occurs when we do too much processing in our product but it doesn't bring any additional value. These things can include giving a product

more features that no one will use, yet they raise the expenses of running your firm.

- **Defects**
  Another type of waste that we often meet is defects. Defects could happen in cases like design flaws or bugs in code, resulting in additional testing and correction, extra reworking areas where those come with further exploitation of labor and equipment.

III. 1**. Define Value**

- Customer Segments: Project creators plus backers and niche communities.
- Customer Needs and Problems: Interviews, surveys, and market research help to figure out what people truly want from a crowdfunding platform.
- Value Proposition: Define the unique benefits that your platform will offer based on these user needs and what sets you apart from others.

2. **Map the Value Stream**

- To create projects on your platform, it is important that you understand the key processes in which users are involved.
- Equally important is to scrutinize these steps for wastage; every step should be subjected to scrutiny with a view of identifying any non-value adding processes that can be curtailed or eliminated. An example would be the elimination of redundant approval processes like overly complex registration forms which do not contribute meaningfully towards achievement of project objectives.
- You will need to come up with a plan on how such operations can be streamlined: some approvals could easily be automated while others might require simplifying user interfaces in order for them to become less time-consuming yet effective.

3. **Create Flow**

- The impediments to be overcome: to secure the unobstructed current of work throughout its channels, such as those arising from bottlenecks in project approval or payment processing.
- Tools and technologies that are advanced: adopt software tools which ease workflow control and ensure that information flows freely within the platform.
- An uninterrupted progression along the line: see that all operations initiated are carried through promptly without any stoppage for lack of material, equipment or

manpower— unless their continuance is prevented by some insurmountable barrier.

4. **Establish Pull**
   - Development of features should be determined based on what users actually need and not based on assumptions. Develop new functionalities as they are needed and validated by users to avoid overproduction of unused features — do not develop a feature until you need it. Develop systems to gather continuous feedback from users, which should be used to determine the priority of development tasks. Feedback from real people is more important than any other abstract criteria that you can think about.

5**. Strive for Flawlessness**
   - Adopt a continuous improvement approach by periodically evaluating and enhancing all facets of the system using user input and operational data.
   - Embrace iterative development by employing agile methodologies that allow consistent evolution of the platform through periodic updates and feature rollouts.
   - Continuously monitor the changes' effects via metrics and key performance indicators, enabling us to spot any lacunas that deserve more attention.

IV. Below are five methods we can adopt to reduce costs during the development of Kickstarter:

A. **MVP Approach (Minimum Viable Product)** = First, we can release a beta version of the product with minimal features that early adopters will use. Through this, we can gather sufficient customer value and collect feedback to build a better product. Using the MVP approach, we can quickly and inexpensively release improvements to our product. Although this may take longer, we can still deliver customer value.

B. **Automated Testing** = Implementing automated testing can help catch early errors and reduce the cost for fixes required if the errors are caught in the later parts of the development process, or even after release. As a rule of thumb, it's best to automate 20% of the tested functions, according to the 80/20 Pareto rule.

C. **Continuous Integration/ Continuous Development (CI/CD)** = We can use a VCS (Version Control System) such as GitHub by writing code and committing it to the VCS, then triggering the CI/CD tool to fetch the code and run the tests. If it passes the tests, we can move on to the next step of the development. Hence, utilizing CI/CD can reduce manual processes in building and deploying software, further reducing labor costs.

D. **Cloud-Based Developmen**t = By merging the benefits of cloud platforms and utilizing lean manufacturing, we can get a modern system that efficiently controls

and manages the workflow. Cloud services such as ERP (Enterprise Resource Planning) or SCM (Supply Chain Management) help adjust the scale of our development environment, thus cutting down costs/ investments in infrastructure or workforce.

E. **Outsource Non-Core Activities** = Outsourcing non-core activities such as graphic design, financial services, content creation, etc. can help us free up valuable time and resources, enabling us to focus fully on our company's core initiatives.

V. Some challenges we may encounter are as follows:

A. **Misalignment or miscommunication** = Different priorities across different departments, or simple miscommunication can lead to conflicts which may hinder our progress needed for Lean success.

B. **Over-reliance on tools** = Focusing too much on lean methodologies might stray us away from our company's values which can lead to ineffective implementation.

C. **Resistance to change** = Some team members may be more accustomed to traditional SE methods, and unable to adapt to Lean principles.

Some key points we should implement are as follows:

a. **Training** = All team members must receive adequate and comprehensive training on Lean principles so they can understand the specific methodologies used.

b. **Leadership** = Establishing strong leadership to help guide and support team members is crucial in helping transition to Lean practices.

c. **Iterative improvement** = We must also believe in the idea of continuously improving, hence frequently collecting and reviewing feedback from customers to help further refine our product.