

Project Management 101

CS 346 Application Development

<https://student.cs.uwaterloo.ca/~cs346>

Contents

Introduction	2
What is a Project?	3
Project Management	5
Software Projects	6
Project Components	7
Project Members	8
Requirements	9
Deliverables	11
Milestones	12
Project Tracking	14
What we want to track	15
GitLab	16
Bibliography	17

Introduction

What is a Project?

A **project** is a planned activity which results in some desired outcome e.g., building a fence, baking a birthday cake.

- As software developers, our projects usually include designing and building a product for the use of our customers.
- A project may be a software release, or a set of targeted changes to a product for an existing customer.

A **stakeholder** is any person or entity that has an interest in your project's outcome. e.g. a user, purchaser, even the business or individual responsible for your project.

- The success (or failure) of your project will have an impact on a large number of people! Each of them has a different interest in your project. [1]

What is a Project?

Stakeholders in a project will each have their own goals and interests.

They're often aligned on high-level goals, but they may disagree about the details and project priorities. [2]

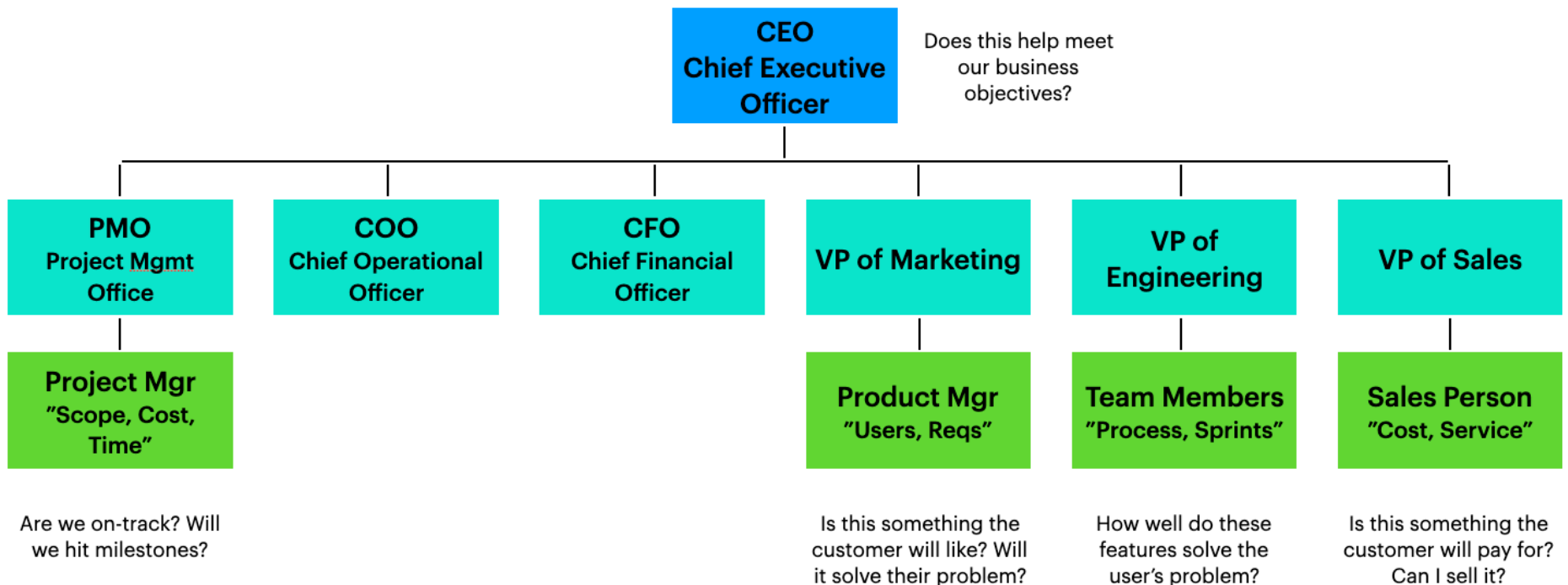


Figure 1: If you're a software developer working for the VP of Engineering, you need to consider not just how well your software operates, but all of the other stakeholder goals, including cost.

Project Management

Project management is the process of planning, executing, monitoring, and completing projects within a specific timeframe.

Project managers are concerned with ensuring that the project's goals are being met. They are also responsible for ensuring that all stakeholder concerns are being represented in the project e.g., the customer will be satisfied, costs don't exceed budgets and so on.

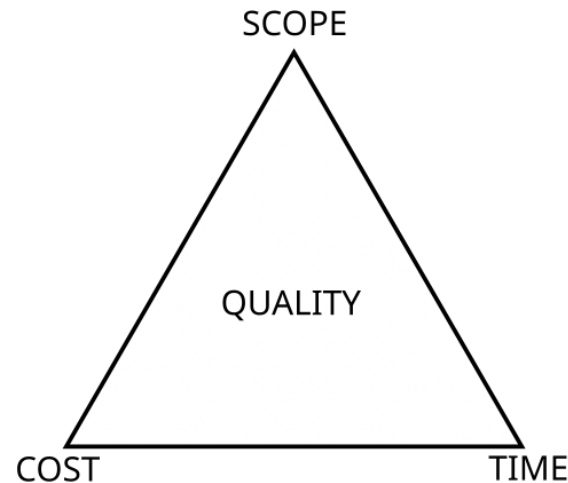


Figure 2: This is the [project management triangle](#), which describes the relationship between scope (what we're building), time (how long it will take) and cost (budget)

Software Projects

How is a software project different from a regular project?

We can think of a software project like any other project: it includes a set of steps that we will follow and track, to make sure our goals are adequately met.

However, software has its own peculiarities:

- The cost of manufacturing software is largely the cost of hiring and empowering people to do the work.
- There are no “raw materials” like you have in manufacturing e.g., compare building a calculator application to building a car.
- Once produced, each additional “unit” of software that you sell is zero-cost. All of the cost is up-front, incurred in design and development.

We'll talk more about the *process* of building software shortly.

Project Components

Project Members

Project members are the people who are staffing a project and responsible for it's success.

For a software project, this can include:

- **Project manager**: involved in high-level project tracking
- **Product manager**: represents customer and market needs
- **Architect**: addresses design and system integration
- **Software developers**: responsible for implementation
- **Quality assurance**: design and manage testing strategies
- **Support engineers**: support the product once it's deployed.

Depending on the project, you may also have other specialized roles e.g., a hardware engineers who help with manufacturing dongles, or specialized hardware to support your hardware.

Requirements

[Software requirements](#) are a detailed representation of what you want to produce. These are typically produced by multiple people, working in coordination to capture the needs of all stakeholders.

Types of requirements documents can include:

- **Business Requirements Document (BRD)**: Captures what the business needs, including the rationale and authority to launch the project.
- **Product Requirements Document (PRD)**: What the customer would like, or what functionality is required to address their needs.
- **Functional Requirements Document (FRD)**: Defines system behaviour.
- **Technical Requirements Document (TRD)**: Defines how it will be built.

It's critical to address all stakeholder needs! We want something that's useful for a customer, but engineering needs to be able to maintain what you produce, and the project itself needs to stay within budget so that the business remains profitable.

Requirements

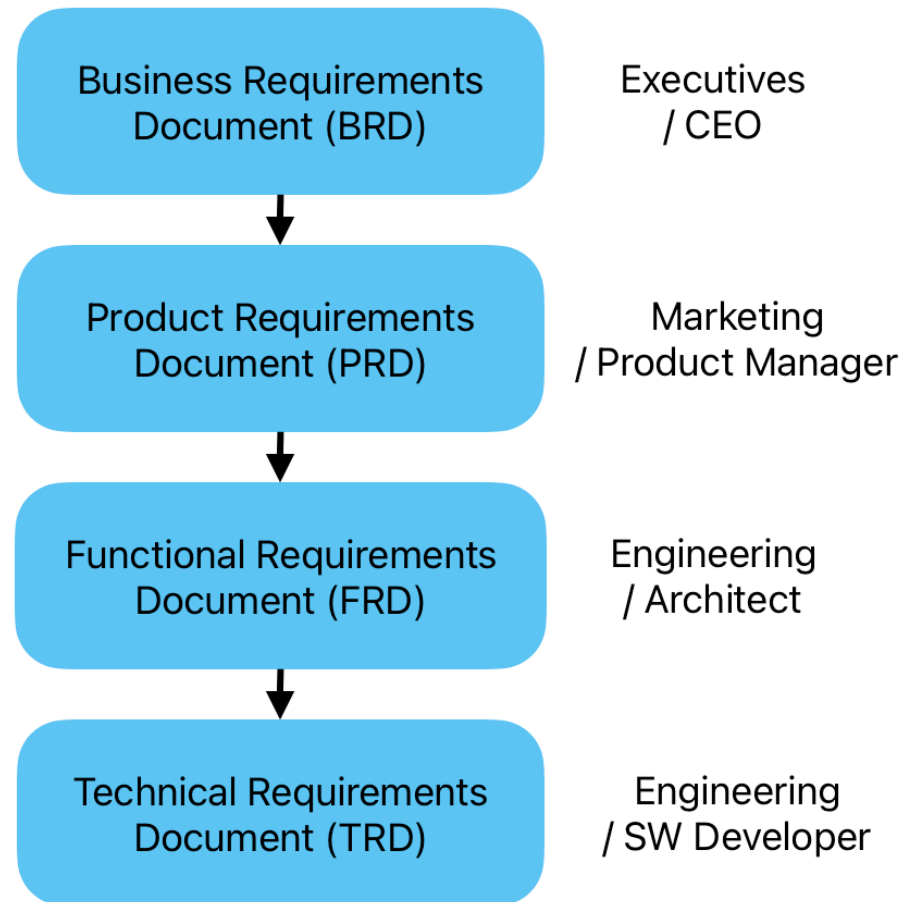


Figure 3: Each document in this series is “owned” by a different team, and represents their area of responsibility. This is idealized, and documents may sometimes be combined or even omitted.

Deliverables

If requirements define what we want to produce in our project, [deliverables](#) reflect those items that we actually deliver to the stakeholders.

Most of the time, this is some product e.g., a software release. However, we will also have items that are produced as outputs of the project process. They can include:

- A software product, in different installable formats, for different operating systems. We likely have multiple versions.
- Software documentation, either in printed or electronic form. It can be on media, or hosted online as determined by the requirements.
- Project documents, detailing how the project progressed, and the details of decisions that were made during the course of the project.
- Source code, saved in a version control system.

There are many business, engineering and legal reasons why you want to retain records of your project. Can you think of a few?

Milestones

Simply put, [milestones](#) are deadlines, or significant dates. Your project will typically have multiple milestones that you need to address.

e.g.,

- Demos to your customer to show your progress.
- A trade show where you want to present an early version of your software.
- Approval to proceed past a certain checkpoint in your project.

In our course, you will have multiple milestones that you need to reach - typically demos.

Milestones

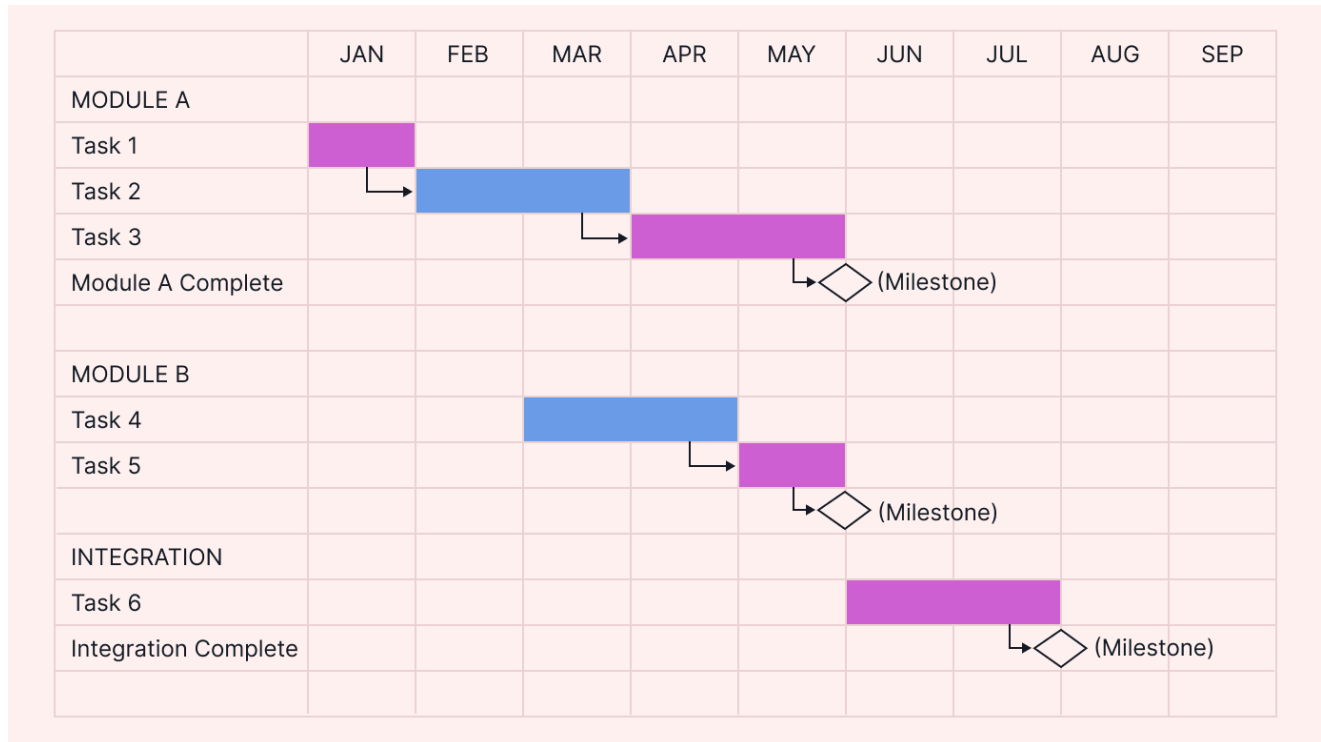


Figure 4: A Gantt or Milestone chart is often used to show a series of work items leading up to a milestone. This example is from usemotion.com.

Project Tracking

What we want to track

Project management has evolved away from manually tracking work to tracking all of the components of a project in an integrated software system.

Ideally, we want to track, in real-time:

- **Task management:** all of the work items for a project, representing the work that needs to be done for a requirement.
- **Timeline visualization:** when we expect to reach certain goals, based on current trajectory.
- **Milestone tracking:** keep track of significant dates or events, and track work against them.

The goal is not to necessarily “surveil” the project team members, but to make project information available to everyone to help keep the project moving forward.

GitLab

GitLab is a project management solution, comparable to GitHub. The University of Waterloo has a hosted GitLab instance where you will store your project.

The project components that we've discussed have corresponding representations in GitLab.

After you have [formed a team](#), use these instructions to setup GitLab.

1. [Setup GitLab](#) to create your own project space.
2. [Organize GitLab](#) to create project milestones.
3. [Log work items](#) to track work items against a milestone.

It's important that you track your work in both GitLab and your Git repository (with commit messages). This is how we will evaluate your work.

Bibliography

- [1] S. Berkun, *Making Things Happen - Mastering Project Management*. O'Reilly, 2010.
- [2] S. McConnell, *Code Complete*, 2nd ed. Microsoft Press, 2004.