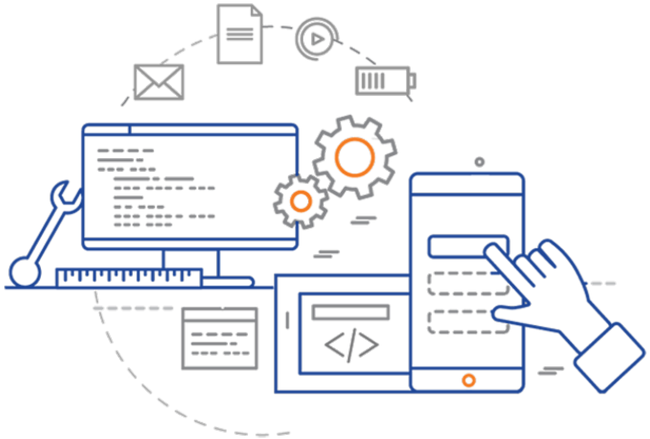


CS 398: Application Development

Week 02 Video: Planning

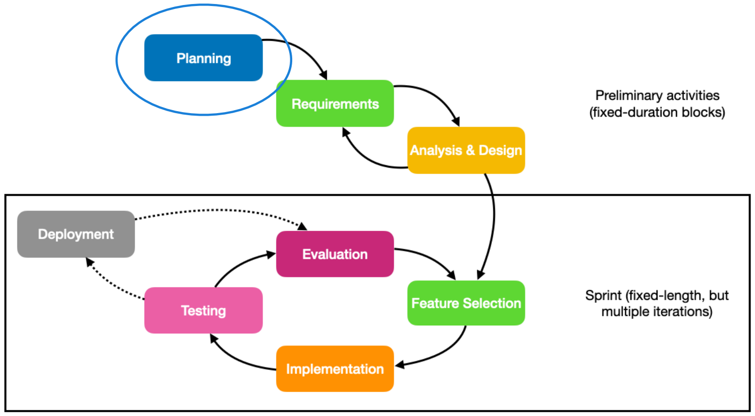
Phase 1: Project Planning

Planning is the set of activities related to the structure and management of our software project.

- Identifying **project goals**, which can include direct goals (e.g. build product x, have something to demo to a customer at the trade show), and indirect goals (e.g. provide Jerry with the opportunity to grow design skills).
- Identifying **resources** (e.g. hardware, software, staff that are available) and **constraints** (e.g. must run in Windows).
- **Identifying risks to your project**, and devising a plan to adjust and accommodate these risks.
- Determining an overall schedule, including **milestones** and **time constraints**.

Traditionally, you are trying to determine:

- **Time**: how long the project will take to complete, what milestones exist.
- **Scope**: what will be completed in this project.
- **Cost**: what is the total cost (tracked at different levels).



Software Development Lifecycle (SDLC)

Triple Constraint Model

The triple constraint model ("project management triangle") is a model of the constraints of project management.

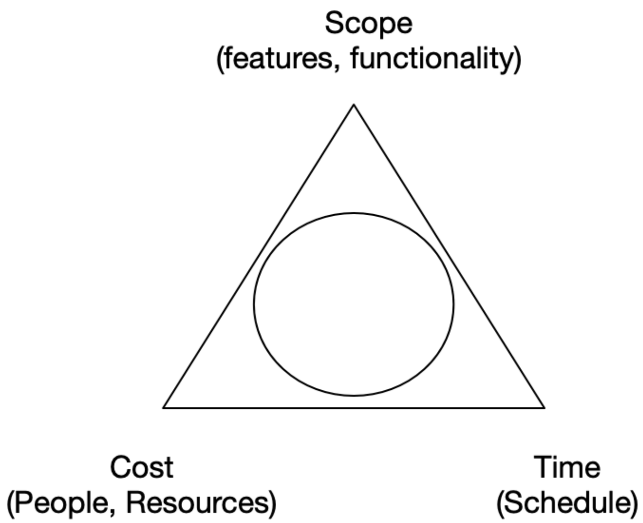
The quality of work is constrained by the project's budget (cost), deadlines (time) and scope. In other words, quality depends on the relationship between project costs, what is being done and time required to do it properly.

Often discussed in terms of tradeoffs:

- "If you increase scope, then I have to add resources".
- "If you decrease time, then I have to increase costs".

Brook's Law:

- "Adding manpower to a late software project makes it later"

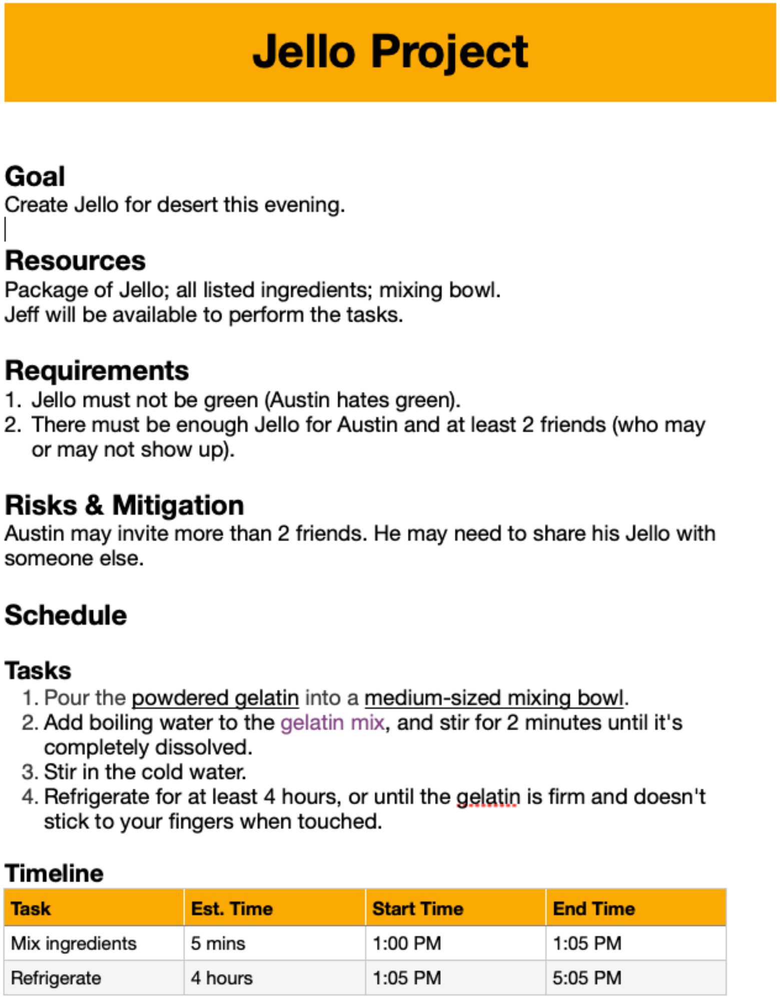


Project Planning

Project Plan

You should aim to capture this information in a project plan:

- **Goal**: You typically want a succinct project statement that captures the overall purpose of this project. What are you trying to accomplish?
- **Resources**: Identify resources required. Typically this includes an estimate of staff required, plus any additional hardware or software systems that are needed.
- Include resource constraints. Is someone going on vacation? Can they only work PT?
- **Risks**. What risks can you identify at this stage? These can be technical or non-technical. For every risk, you should identify a mitigation strategy e.g. We have identified Windows 11 as a target but it is not releasing until midway through the project. If needed we will drop Windows 11 support and revert to Windows 10 for testing.
- **Schedule**: Finally, you need a list of critical dates and deadlines. e.g. must be able to demo feature X at a conference in October.



Jello Project

Goal

Create Jello for desert this evening.

Resources

Package of Jello; all listed ingredients; mixing bowl.
Jeff will be available to perform the tasks.

Requirements

1. Jello must not be green (Austin hates green).
2. There must be enough Jello for Austin and at least 2 friends (who may or may not show up).

Risks & Mitigation

Austin may invite more than 2 friends. He may need to share his Jello with someone else.

Schedule

Tasks

1. Pour the powdered gelatin into a medium-sized mixing bowl.
2. Add boiling water to the gelatin mix, and stir for 2 minutes until it's completely dissolved.
3. Stir in the cold water.
4. Refrigerate for at least 4 hours, or until the gelatin is firm and doesn't stick to your fingers when touched.

Timeline

Task	Est. Time	Start Time	End Time
Mix ingredients	5 mins	1:00 PM	1:05 PM
Refrigerate	4 hours	1:05 PM	5:05 PM

See the project plan template on the course website.

Project Tracking

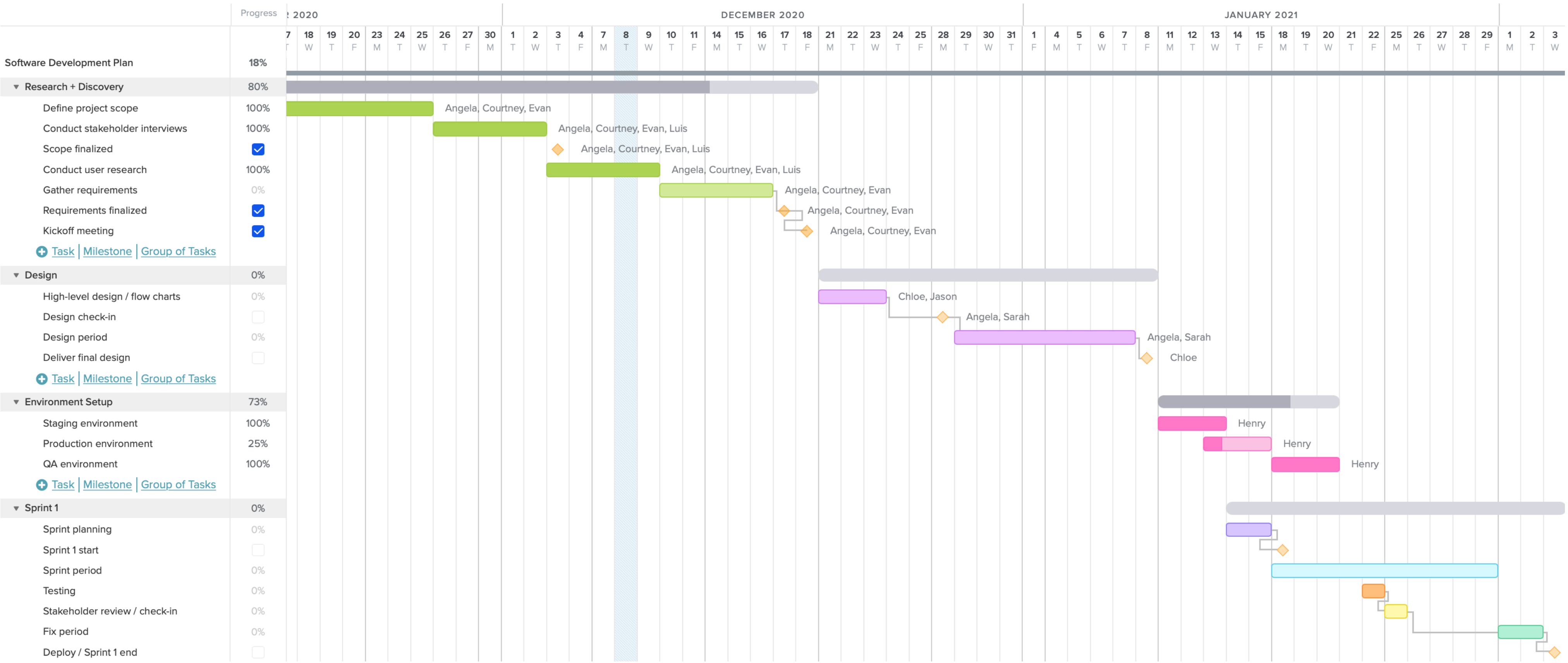
Project Tracking

You will want to track your progress against your project plan. This allows the team to assess their progress and take any necessary action to maintain the schedule.

The most common means of tracking project schedules is called a Gantt Chart, which is a high-level schedule showing all tasks and their dependencies.

Key concepts:

- **Task**: Some “piece of work” to perform. Usually an independent task though it may be have predecessors or successors.
- **Tasks and dependencies**: Work that needs to be complete (tasks) and order that needs to be enforced (dependencies).
- **Milestones**: Important deadlines that you want to highlight. You are planning to meet these.
- **Critical path**: The longest path in your project; the path that determines the overall duration of the project. Often there is enough “slack” in a project that tasks can go over scheduled time without harming your ability to hit milestones; the critical path is the one which, if delayed, will delay the project.

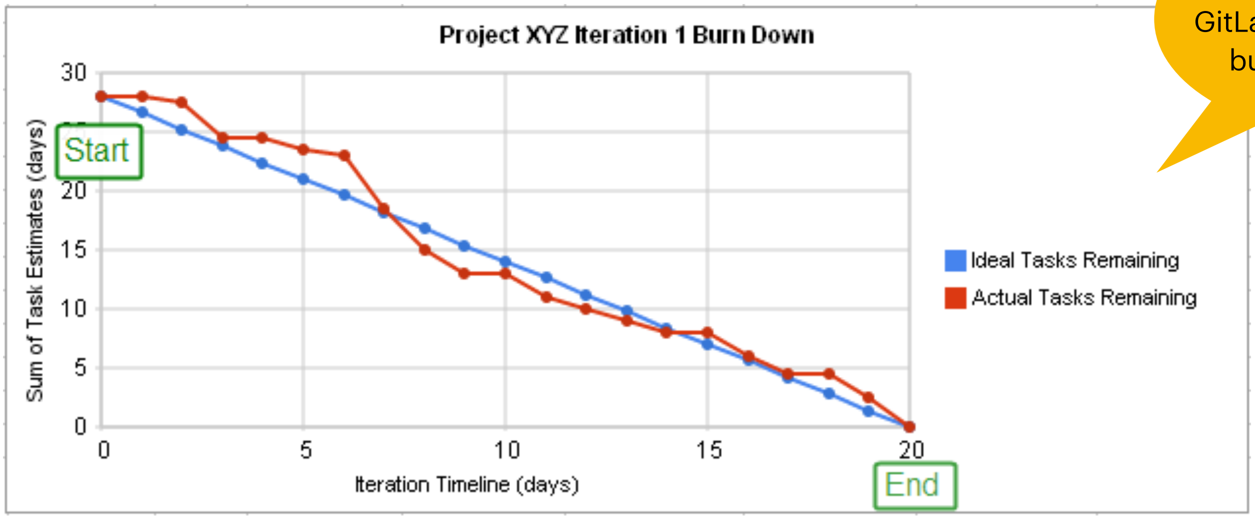


	Progress
Software Development Plan	18%
▾ Research + Discovery	80%
Define project scope	100%
Conduct stakeholder interviews	100%
Scope finalized	☑
Conduct user research	100%
Gather requirements	0%
Requirements finalized	☑
Kickoff meeting	☑
⊕ Task | Milestone | Group of Tasks	
▾ Design	0%
High-level design / flow charts	0%
Design check-in	☐
Design period	0%
Deliver final design	☐
⊕ Task | Milestone | Group of Tasks	
▾ Environment Setup	73%
Staging environment	100%
Production environment	25%
QA environment	100%
⊕ Task | Milestone | Group of Tasks	
▾ Sprint 1	0%
Sprint planning	0%
Sprint 1 start	☐
Sprint period	0%
Testing	0%
Stakeholder review / check-in	0%
Fix period	0%
Deploy / Sprint 1 end	☐

Burndown Charts

With the rise of Agile, Burndown Charts have become popular. They show the amount of work remaining in a Sprint at any given time, and can be often generated on-the-fly from an issue tracking system. They are significantly more useful for Agile projects, since they focus less on the details of what is being built, and more on the team's velocity.

Sadly, our version of GitLab doesn't support burndown charts.



Time Estimates

Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

-- Douglas Hofstadter. **Gödel, Escher, Bach** (1979)

It is extremely difficult to determine a realistic, *detailed* schedule at the beginning of a project. However, for planning purposes, we need *some* estimates.

The recommended process for determining how long a project will take to complete is:

1. Look at similar projects that you have delivered, and use those as a guideline for the resources that you will need, timelines and so on..
2. Identify major milestones: these are targets that you know you need to meet for the project to be considered successful. Work backwards as much as you can from the final date, and fill in intermediate goals.
3. Fill in project phases based on the work that needs to be done. e.g. research, design and setup phases before actual implementation. These phases may vary based on your style of project.