Lecture 4: September 18

Project Planning: Planning in Practice

Agenda

- Mentor Intros
- Planning in Practice
- Tech Interviews
- Upcoming Deadlines

Agenda

- **Mentor Intros**
- Planning in Practice
- Tech Interviews
- Upcoming Deadlines

Austin Blackman, GW ‘15

Austin is a software engineer within Oracle's Cloud Infrastructure division. His focus is on UI development and he specializes in building applications with frameworks like React and Angular.

Eli Mattson, GW ‘14

Eli lives in San Francisco and has been working in various startups in silicon valley since graduation.

Currently, he is the Director of Engineering at Indigov – a startup selling software to Congress and elected officials across the country with the mission to improve communication in our democracy.

Rian Shambaugh, GW ‘15

Rian graduated from GW with a BS in CS in 2015. From there she went to UMass for her masters.

She’s been working at FINRA (Financial Industry Regulatory Authority) in the Transparency Services group for 5 years. Rian started as an SDET and now is a Software Engineer.

Her team generates statistical reports based on trade data and sends it to various stakeholders both inside and outside FINRA.

Lucas Chaufournier, GW ‘15

Lucas is currently a backend software engineer at Square, working on their restaurants product.

Prior to Square, he worked at Capital One, a startup, and spent 4 years living the graduate student lifestyle.

He’s worked all over the stack from the low level OS, to security, to web and mobile apps.

In his free time, he plays a lot of pickleball, host events for folks in Seattle, and is building a mobile app to help make finding friends easier!

Pat Cody, GW ‘20

Pat graduated in 2020, and has been working as a production engineer at Meta ever since.

He spent 3 years working on the infrastructure that powers Messenger, before pivoting last year to ads working on infrastructure reliability.

He enjoys building distributed systems, firefighting when code breaks, and sysadmin / devops work.

Sarah Morin, GW ‘21

Sarah works on distributed systems and consensus protocols.

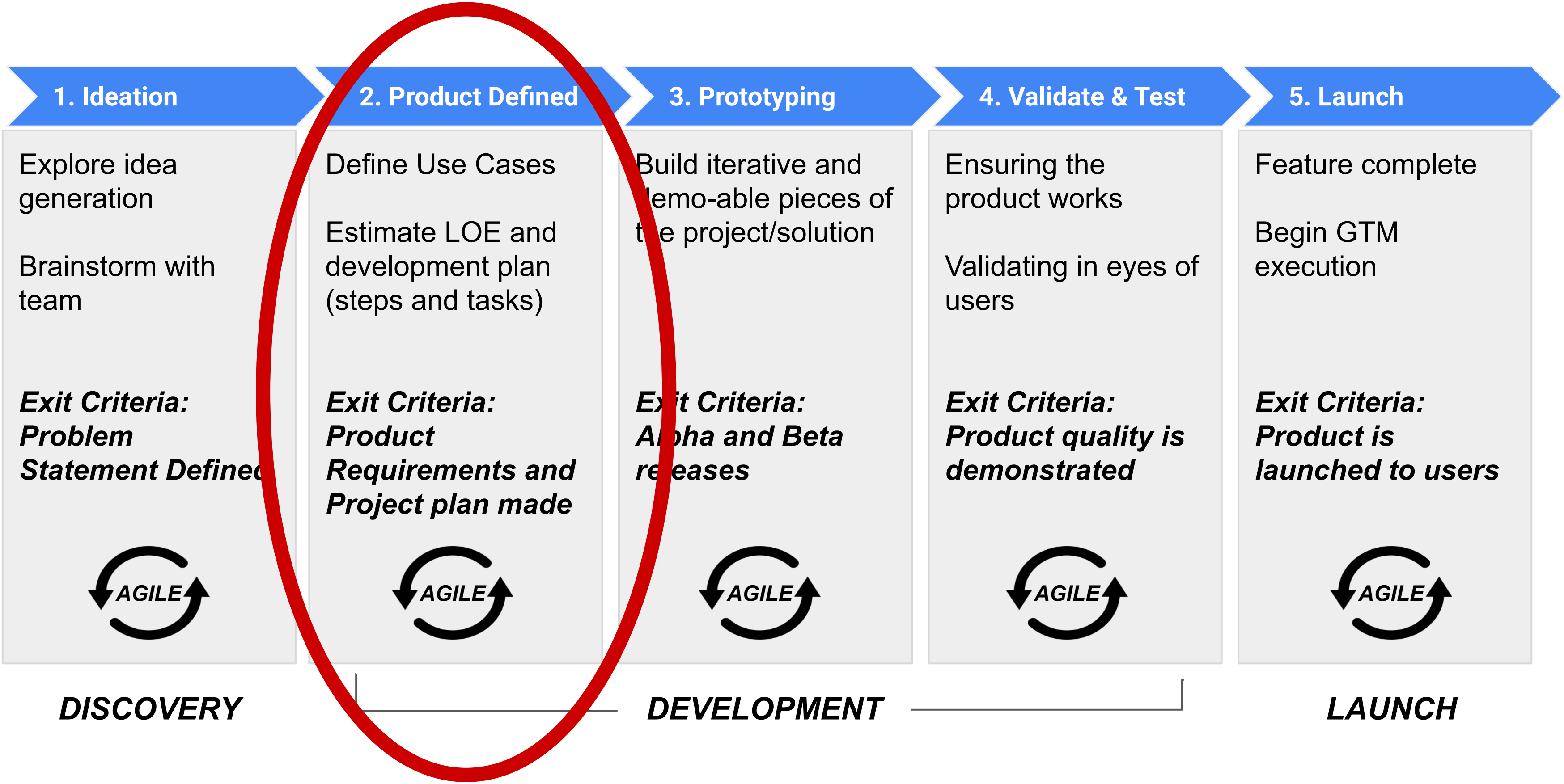
She spent the last 3 years as a software engineer at Oracle Cloud Infrastructure, specifically on the data-plane team for File Storage Service.

She recently left Oracle to return to research and pursue a PhD in CS. She lives in San Francisco with her boyfriend (also a GW CS alum) and their cats Zuko and Kiwi.

Agenda

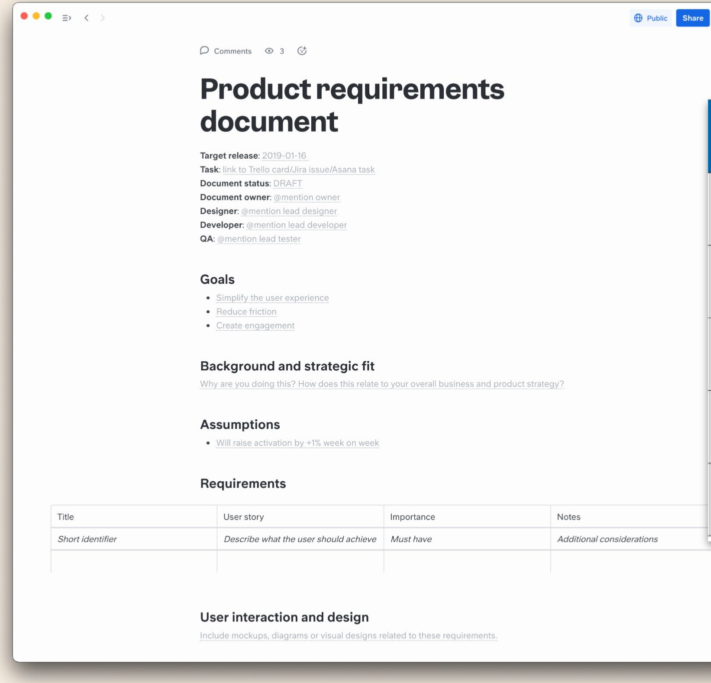
- Mentor Intros
- **Planning in Practice**
- Tech Interviews
- Upcoming Deadlines

Product Definition Phase



1. Ideation	2. Product Defined	3. Prototyping	4. Validate & Test	5. Launch
Explore idea generation Brainstorm with team	Define Use Cases Estimate LOE and development plan (steps and tasks)	Build iterative and demo-able pieces of the project/solution	Ensuring the product works Validating in eyes of users	Feature complete Begin GTM execution
Exit Criteria: Problem Statement Defined	***Exit Criteria: Product Requirements and Project plan made***	***Exit Criteria: Alpha and Beta releases***	***Exit Criteria: Product quality is demonstrated***	***Exit Criteria: Product is launched to users***
AGILE	AGILE	AGILE	AGILE	AGILE
DISCOVERY		***DEVELOPMENT***		***LAUNCH***

Exit Criteria: PRD and Gantt Chart



Product requirements document

Target release: 2019-01-16
Task: link to Trello card/Jira issue/Asana task
Document status: DRAFT
Document owner: @mention owner
Designer: @mention lead designer
Developer: @mention lead developer
QA: @mention lead tester

Goals

- Simplify the user experience
- Reduce friction
- Create engagement

Background and strategic fit

Why are you doing this? How does this relate to your overall business and product strategy?

Assumptions

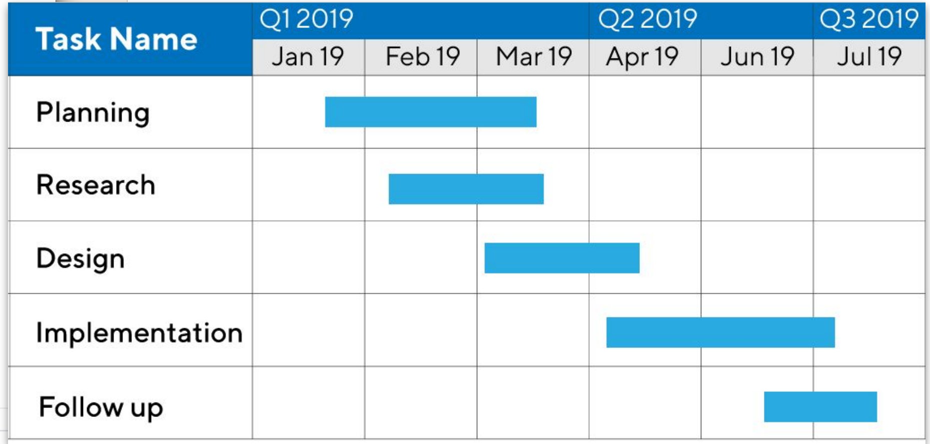
- Will raise activation by +1% week on week

Requirements

Title	User story	Importance	Notes
Short identifier	*Describe what the user should achieve*	*Must have*	*Additional considerations*

User interaction and design

Include mockups, diagrams or visual designs related to these requirements.



Product Requirements Document

Section 1: Proposal

Section 2: Specs

Section 3: Considerations, Constraints, Dependencies

Section 4: Open Questions

Section 1: Proposal

Your elevator pitch to your team, stakeholders, investors, etc.

- **Goals**

  What are you hoping to accomplish? What does Success look like?

- **Users**

  Who are you building for?

- **Use Cases**

  What will the user be able to do once you reach Success?

- **Background & Strategic Fit**

  What context is needed to understand your idea? How is your product different?

Section 2: Specs

The meaty “what” section to describe the work to be done.

- User Stories & Requirements
  - Features and functions from a user’s perspective
- UX Flow and Designs
  - Mockups
  - Flow Diagrams
- System and Environment Requirements
  - Web or Mobile?
  - OS needed?
  - Libraries or other tooling needed?

Section 3: Considerations, Constraints, Dependencies

Giving awareness into all aspects of the project.

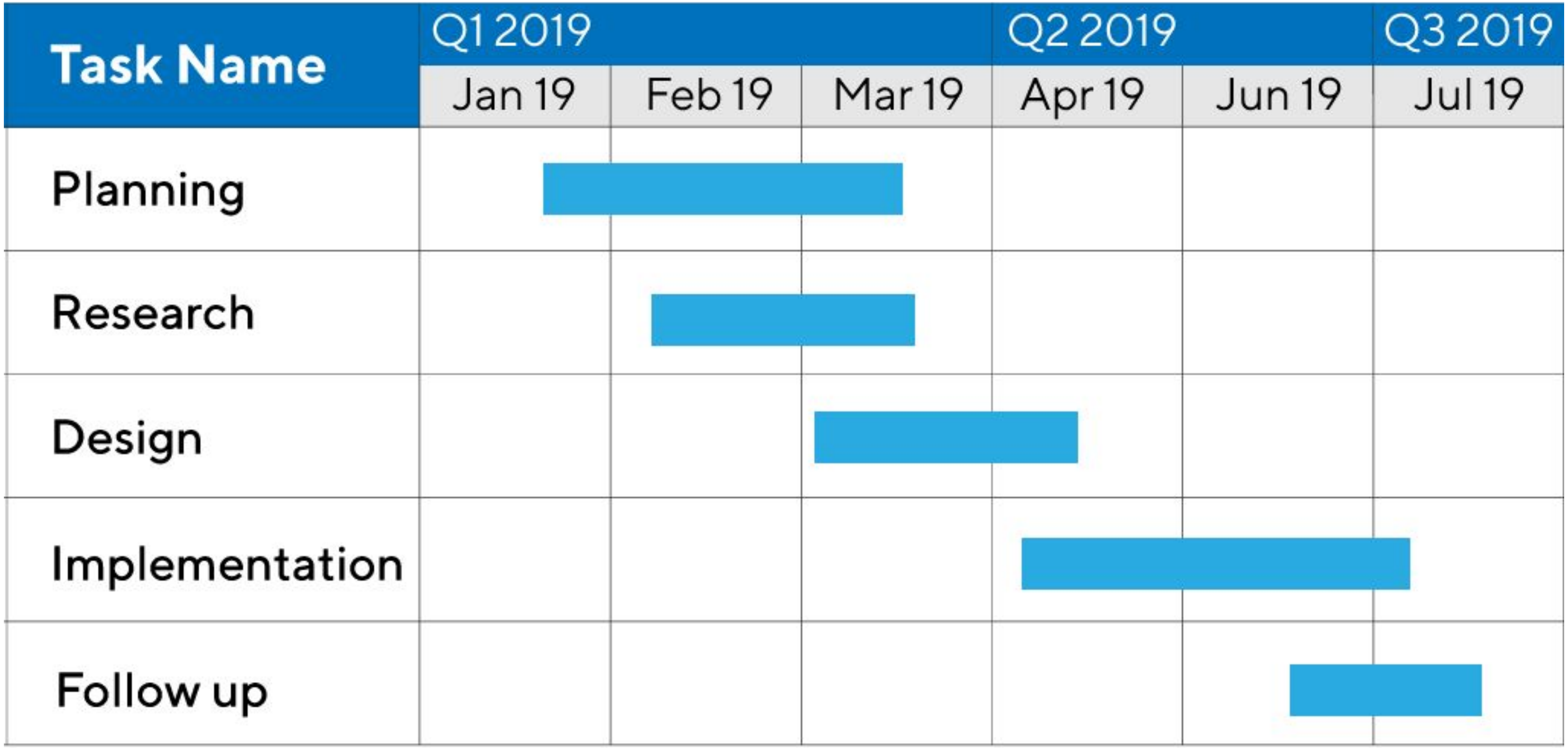
- Define your “non-goals”
- Limitations to be aware of
- Outside factors that would affect the project outcome
- Other team’s or individual’s work that must be complete beforehand
- Risks

Section 4: Open Questions

Central location for all team members and stakeholders to unblock each other and move the project forward. Helps with transparency into blockers/risks.

- What else do you need to know? Known unknowns & unknowns unknowns?
- Do you need help or answers from other teams or stakeholders?

Project Plan: Gantt Chart



GIST Framework

GIST: Goals, Ideas, Steps, Tasks

Goals: What are your goals?

Ideas: What are your ideas to achieve your goal?

Steps: What are the ***steps*** to complete your idea?

Tasks: What are the ***tasks*** to completing that step?

Project Planning using GIST

What are your ***goals***?

- Provide users with walking directions that factors in environmental safety

What are your ***ideas*** to achieve those goals?

- Front end app
- Routing algorithm + safety score

GIST: Steps and Tasks

What are the ***steps*** to complete your idea?

- Set up Android app
- Implement Google Maps API
- Build safe routing algorithm
- Connect app to server/DB

What are the ***tasks*** to completing that step?

- Android App:
  - Build profile/login/logout capabilities for mobile user
  - Build activity for Social page
- Google Maps API:
  - Get Google Maps API keys and integrate
  - Test Google Maps API
- Routing algorithm
  - Research safety scoring
  - Research routing algorithms & apis
  - Integrate safety scores into routing function
  - Build and test
- Server/DB
  - Set up server
  - Connect and test with routing algorithm and Android app

Task

Lowest level of work to be done towards completion of the project

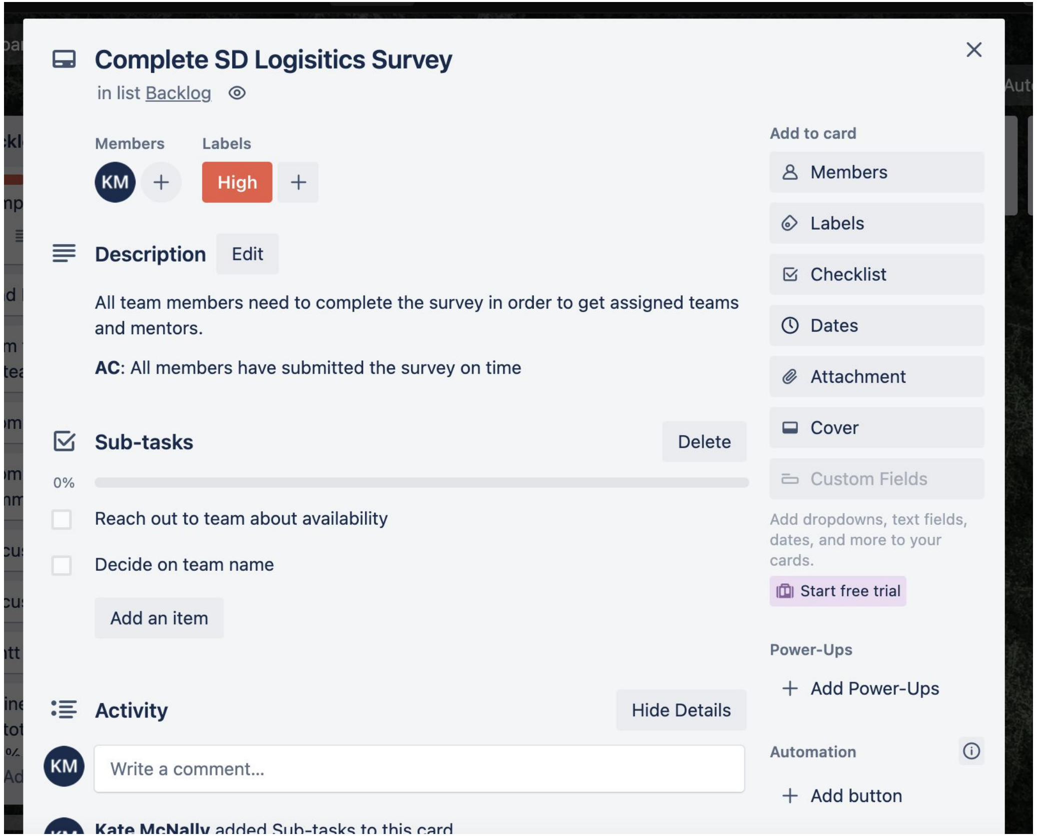
Description/Background

Assignment

Sizing*

Acceptance Criteria

Code Review**



Complete SD Logisitics Survey

in list Backlog

Members: KM

Labels: High

Description

All team members need to complete the survey in order to get assigned teams and mentors.

AC: All members have submitted the survey on time

Sub-tasks

0%

□ Reach out to team about availability

□ Decide on team name

Activity

Write a comment...

Kate McNally added Sub-tasks to this card

Building the stack-ranked backlog

GOAL	IDEA	STEP	TASK
Optimize Walking Directions	Android App	Setup App	Build Home
Optimize Walking Directions	Android App	Setup App	Build Profiles
Optimize Walking Directions	Android App	Google Maps	Get API Keys
Optimize Walking Directions	Android App	Google Maps	Show Map on app
Optimize Walking Directions	Android App	Social Feed	Build table for feed
Optimize Walking Directions	Android App	Social Feed	Set table data source to get all user’s friends’ activity
Optimize Walking Directions	Routing Algorithm	Algorithm	Research safety algo.
Optimize Walking Directions	Routing Algorithm	Algorithm	Research routing algo.
Optimize Walking Directions	Routing Algorithm	Algorithm	Implement routing with safety score
Optimize Walking Directions	Routing Algorithm	Algorithm	Test with dummy data
Optimize Walking Directions	Routing Algorithm	Server/DB	Implement server
Optimize Walking Directions	Routing Algorithm	Server/DB	Connect algorithm to app

Reviewing and Sizing (story points / t shirt sizes)

Gauge relative effort of tasks to facilitate better planning, prioritization & tracking of work

- **Promotes relative sizing:** compare tasks relative to each other instead of by time estimates
- **Reduces pressure for exact time predictions:** time estimates are hard
- **Enhances collaboration**: encourages members to discuss task complexity up front
- **Facilitates iteration planning:** helps teams schedule balanced workloads

Reviewing and Sizing

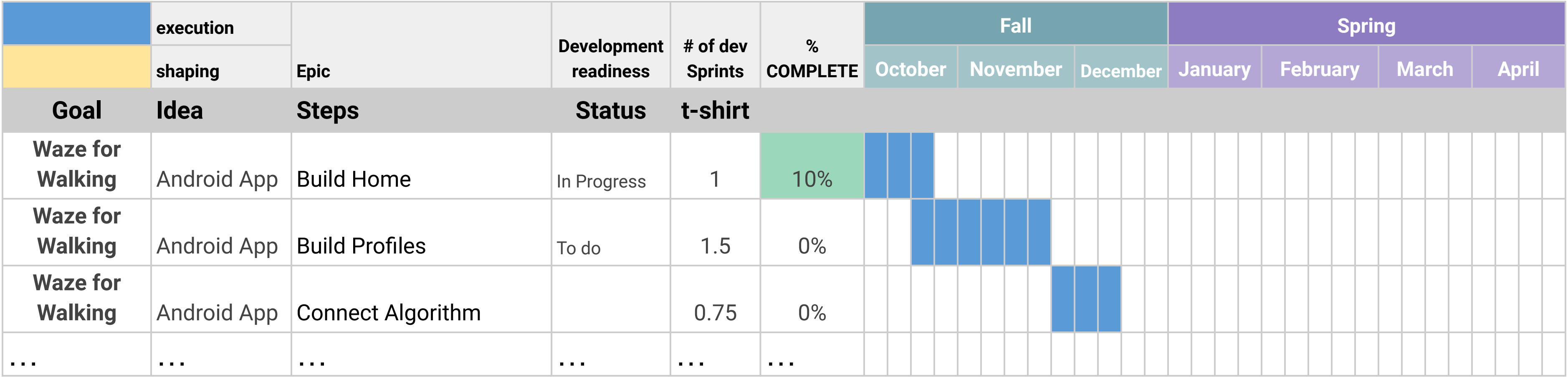
T-Shirt Sizing

Fibonacci Pointing System

Size	Pointing Point	Weeks
XS	1	0.25
S	2	0.5
M	5	1
L	8	2
XL	13	3

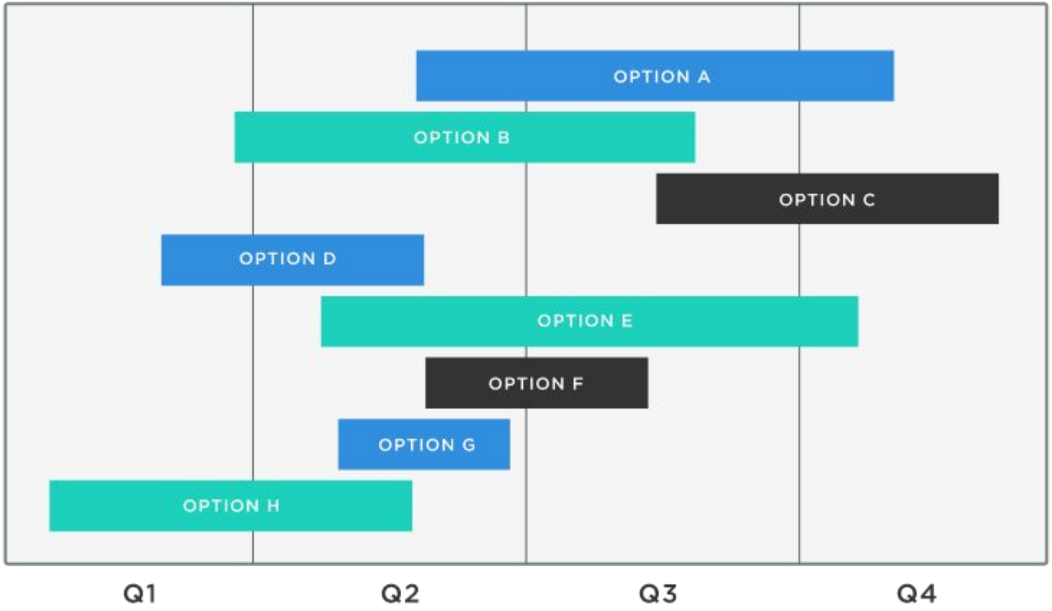
Put it all together

Goal	*Idea*	*Step*	*Task*	*Points*	*Start Date*
Optimize Walking Directions	Android App	Setup App	Build Home	2	October 1
Optimize Walking Directions	Android App	Setup App	Build Profiles	3	October 15



🟦	**execution**	**Epic**	**Development readiness**	**# of dev Sprints**	**% COMPLETE**	**Fall**													**Spring**																
🟨	**shaping**					**October**				**November**					**December**				**January**				**February**					**March**				**April**			
Goal	**Idea**	**Steps**	**Status**	**t-shirt**																															
Waze for Walking	Android App	Build Home	In Progress	1	10%	■	■	■																											
Waze for Walking	Android App	Build Profiles	To do	1.5	0%			■	■	■	■	■	■																						
Waze for Walking	Android App	Connect Algorithm		0.75	0%									■	■	■																			
...	...	...	...	...	...																														

Gantt Chart - Negotiating Timelines



Creating a stack-ranked backlog

What is everything that you could possibly do for the project?

What are the dependencies?

- Will you need to use another library?
- Is there an order to which the work has to be done?

What are the priorities?

- Is everything important to build?
- Can you determine which are requirements vs nice-to-have?

User Story

User experience/behavior that you are building

Background

User Story

“As <*user persona*>, I would like to ___ so that I can ___”

Success Criteria

Testing

Documentation/Published Update

User Story Example

User Story

“As an android user, I would like to enter a start & end destination so that I can get walking directions.”

Success Criteria

- User is able to enter start & end locations
- Routes are generated based on input location
- Routes are walkable (not too long, not logistically dangerous)

Alpha → Beta → Final (GA)

Alpha

“Initial MVP of value”

Beta

“You can take on more customers, but not everyone”

Final (GA)

“Everyone can use it”

When to know if it’s ready for Alpha, Beta, or GA; “Drawing the line” on stack rank

Agenda

- Mentor Intros
- Planning in Practice
- **Tech Interviews**
    - What they are / why
    - Different stages of interview pipeline
    - Resume Deep Dive
    - Typical new grad hiring timelines
    - Extra resources
- Upcoming Deadlines

What interviewers are testing for

- **Analytical Skills**
  - Could you solve the problem? Did you solve it optimally? For system design, did you structure the problem well & think through tradeoffs?
- **Coding skills**
  - Could you translate your algorithm to code? Was it well-organized? Did you use good style?
- **Technical knowledge**
  - Do you have a strong foundation in the relevant technologies?
- **Experience**
  - Have you made good technical decisions in the past? Have you built interesting projects?
- **Culture fit / Communication skills**
  - Do your values fit with the company and team? Can you communicate your thoughts clearly?

Engineering Interview Process (one possibility)

1. Recruiter Screen
2. Hiring Manager Screen (optional)
3. Coding screen/take home project
4. System Design
5. Team Fit / Behavioral
6. Subject-specific interview / coding problems

Why companies use this process

- **Problem Solving skills & clear communication is valuable**
  - You’ll spend a good amount of time talking through problems with other engineers. Coding challenges are a (sometimes poor) proxy for this

- **Data structures & algorithms are useful**
  - Knowledge of fundamentals is a good proxy for how versed in CS you are
  - They do come up in work, and when they do it's important to know the basics

Why companies use this process

- **False negatives are ok, false positives are not**
  - Better for companies to reject good candidates than accept poor fits.

Why companies use this process

**Being good at interviews is a practiced skill,
and is only tangentially related to being a good engineer**

Recruiter Screen

- **Purpose**: identify if you meet the minimum requirements on paper
- ~15 minutes long
- Interviewer likely knows very little about the technical requirements for the position
- Your resume plays the largest role in this stage
    - Should be easily skimmable
    - Skills & technologies should be clearly listed

Hiring Manager Screen

- **Purpose**: pitch you on the company, confirm you meet minimum technical requirements
- ~30 minutes long
- Interviewer will be technical, acts as a filter before you begin the rest of the interview process
- This interview is more common in smaller companies & startups

Coding Screen / Take Home Project

- **Purpose**:
    - problem solving ability
    - coding ability
    - algorithms fundamentals
- 45 mins - 1 hr long
- Interviewer will be technical

Coding Screen: Tips

- Use python – it saves you a ton of boilerplate
- Communication:
  - Talk out your thought process, the worst thing is a silent candidate
  - Ask clarifying questions to understand the problem
  - If you don’t know the syntax, ask your interviewer
  - If you have a brute force solution, explain it before moving to a more optimal approach
- Coding:
  - Walk through examples & define edge cases before writing code
  - Start with pseudocode (or just use python)
  - Start with a brute force solution
  - Walk through your code with examples
  - Practice algorithms & data structures
- Resources
  - Cracking the Coding Interview
  - https://www.tryexponent.com/practice
  - https://www.structy.net/
  - neetcode.io

System Design

- **Purpose**: Can you break down a large ambiguous problem into manageable pieces
- 45 mins - 1 hr long
- More typical for non-junior candidates
- Evaluates your ability to be a tech lead, not a programmer

Tips

- Ask lots of questions, the problem will be ambiguous
- Take good notes so the interviewer has something to review
- Talk through tradeoffs, there is rarely a “correct” answer

Team fit / behavioral

- **Purpose**: Do you work well with others, do your values align with the company & team
- 45 mins - 1 hr long
- Interviewer is usually someone you will work closely with

Tips

- Be prepared to discuss prior experiences from your resume
- Have a few anecdotes on hand about challenges you faced and how you addressed them

Subject-specific

- **Purpose**: Do you have the specialized skills needed for the role
- 45 mins - 1 hr long
- This will depend a lot on the position (ml, security, data science)
- Interviewer is usually a senior engineer
- May ask questions about projects on your resume

Overall Interview Fundamentals

- If you join a zoom call and the interviewer isn’t there, give it 5 minutes before dropping/reaching out to the interviewer
- If you need to reschedule, do so as early as possible
- Dress appropriately
- Turn your camera on
- Ask thoughtful questions
- Send thank you notes
- Once employed, try to interview as soon as you can!

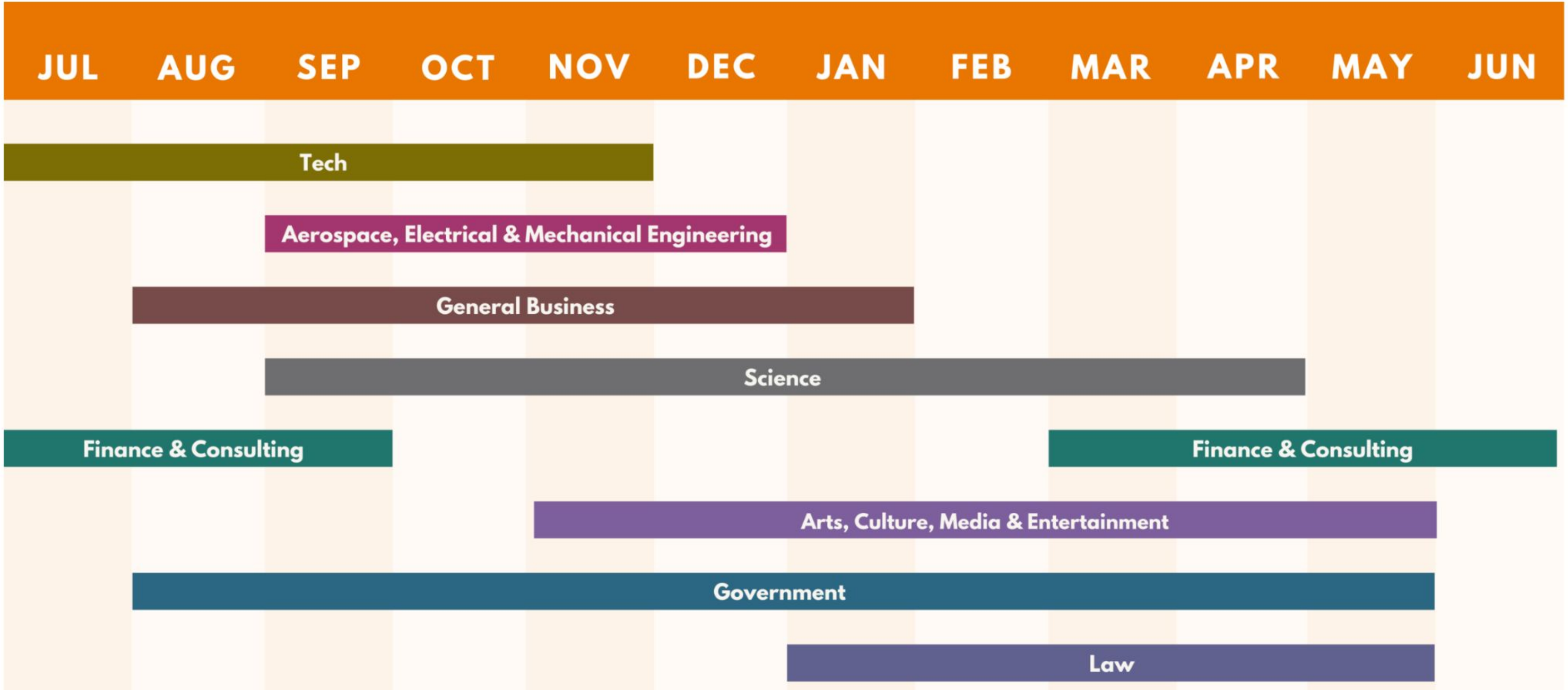
Agenda

- Mentor Intros
- Planning in Practice
- **Tech Interviews**
    - What they are / why
    - Different stages of interview pipeline
    - **Resume Deep Dive**
    - Typical new grad hiring timelines
    - Extra resources
- Upcoming Deadlines

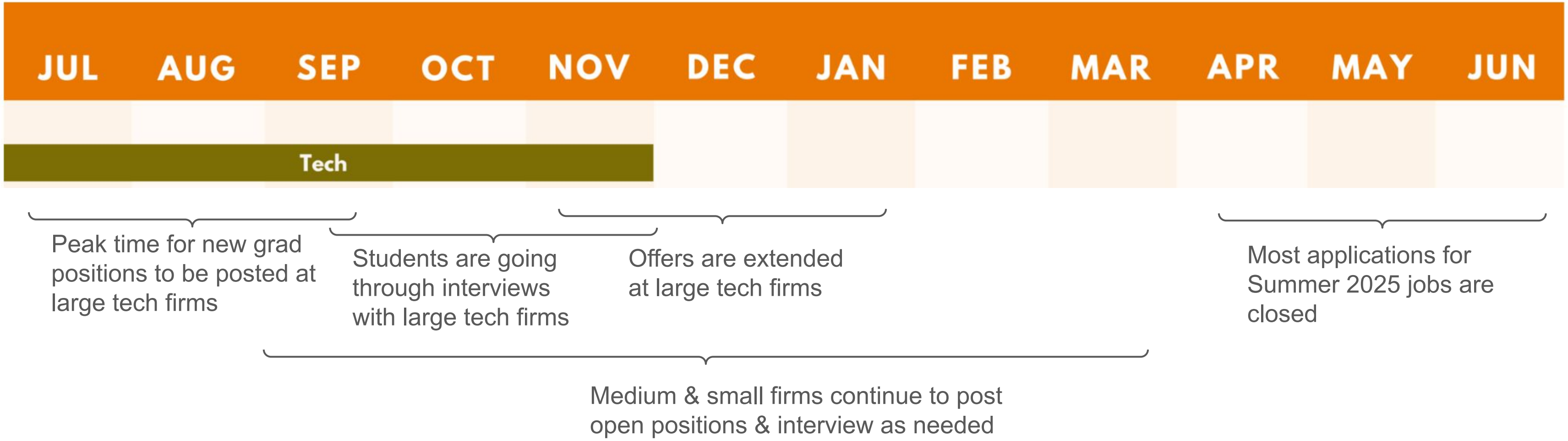
Resume FAQs

1. Who is it for?
    a. **A recruiter / screening tool**
    b. **Your interviewers**
2. How long should it be?
    a. **1 page – used LinkedIn for more detail**
3. What about the format?
    a. **Submit as a PDF**
    b. **Keep it easy to read – be consistent w/ fonts & styling**
    c. **Check for spelling errors!**
4. Should I include coursework & GPA?
    a. **Include GPA if >= 3.0, not as relevant after you have work experience**
    b. **List courses that are unique/stand out (not general courses), remove after you have work experience**
5. How do I list technical skills?
    a. **Includes skills that you can use quickly with minimal supervision. Split by experience level.**

New Grad Hiring Timelines



New Grad Hiring Timelines



Peak time for new grad positions to be posted at large tech firms

Students are going through interviews with large tech firms

Offers are extended at large tech firms

Most applications for Summer 2025 jobs are closed

Medium & small firms continue to post open positions & interview as needed

Advice

1. Update your resume
2. Apply to jobs ASAP
3. Practice as many coding challenges as possible
   a. Cracking the Coding Interview
   b. exponent.io

Extra Resources

1. Schedule a practice interview with a mentor (first come first served)
2. Schedule resume review / career advice w/ SEAS advising

Extra Resources

Timelines

- https://careerdevelopment.princeton.edu/internships-jobs/recruiting-timelines
- https://www.brandeis.edu/hiatt/connections/recruiting/recruiting.html
- https://www.careereducation.columbia.edu/news/june-2023/heres-whats-happening-tech-recruiting-timelines-2023-2024
- https://www.careereducation.columbia.edu/news/april-2024/essential-guide-industry-recruiting-timelines-2024-2025
- https://careerhub.students.duke.edu/blog/2024/04/06/application-timeline-for-tech-data-2/

Resume/Cover letter

- https://capd.mit.edu/channels/make-a-resume-cover-letter-cv/
- https://career.caltech.edu/documents/25752/2020WinterCareerGuide.pdf
- https://careered.stanford.edu/career-resources/discover#researching

Internships

- https://career.caltech.edu/learn/summer-internships

GW Resources

- https://careerservices.gwu.edu/resume-cover-letters

Agenda

- Mentor Intros
- Planning in Practice
- Tech Interviews
- **Upcoming Deadlines**

For Next Week

Deadlines

- [Team]: Meet with your mentor (**Sept. 18**)
- [Team]: Updated project proposals, blackboard (**Sept. 22**)
- [Team]: Hardware / Software Requests (**Sept. 22**)
- [Individual]: Writing 1 - Executive Summary (**Oct. 6**)

Suggestions

- Sign up for mock interviews
- Reach out to career services
- AIAA call for student paper submissions

Next Lab

- Coding Interview Practice (bring headphones)