

Starting a Project

- Often adding a feature to existing app/site
- Sometimes a new site/app

Key Questions:

- What do you do?
- What do you NOT do?

How it begins

You are given the idea of how it should "work"

- Could be a wireframe/mockup
- Could be a prototype
- Could be a spec document

The Real Questions

Confusingly, they may not actually ASK these aloud

- Can you do this?
- What else do you need?
- How long will it take?

But they NEED the answers to all 3

Goal is Shared Understanding

- Can you do this?
- What else do you need?
- How long will it take?

They expect you do what they are thinking

- Unclear means unreasonable expectations
- Clarity means no rude surprises for anyone

Ways they convey their hopes

- Wireframe/Mockup
- Prototype
- Spec document

Mockup/Wireframe

- **Mockup**: A picture of the page
 - Content usually faked
- **Wireframe**: A rough picture of the page
 - Images just boxes, content faked

Prototype

- A web-based app that shows certain interactions
- Like a facade or stage prop
 - Demonstrates the look/feel/behaviors

Distinct from a **Proof-of-Concept** (POC)

- A POC shows that the tech actually works
- Usually a sloppy proof it works to get support/funding
- POCs are sometimes called "prototypes" by devs
 - From manufacturing terminology vs design

Spec Document

- **Spec Document** is *specification*
- You may be given one
- You may be asked to create!
- Lists all the relevant behavior and design
- Similar to my assignments, but with more detail
- Given the same spec:
 - Devs should create the same basic app

What should you do?

- NEVER immediately accept
 - BAD: "Looks good"
 - BAD: "Sure"
 - BAD: "a couple of days"
- Your role is to give GOOD answers
 - Requires close consideration

You need to identify **needs** and **concerns**

- Making the app/feature is not your job YET
- Your job is to say *if you can*
- Your job is to say *at what cost*(resources)
- Your job is to *accurately estimate*
 - And we know how hard that is

Agreeing before consideration is AGAINST your job

Some Questions about the Data

- Do you have the needed data?
 - Do services exist?
- Are there security/access concerns?
 - Can the user SEE/change this data?
- How many service calls will this need?
 - Performance? Scalability?

Questions about the Presentation

- Mobile? (responsive/adaptive)?
- a11y?
 - Often overlooked until hard to fix
- i18n/l10n?
 - Swapping text/numbers/dates?
 - Room for different text sizes?
- Mockup gaps?
 - Does the UI fit data of different sizes?
- Validation
 - Where do validations/errors go?
 - When does validation happen?

Questions about the Architecture

- Does Security team need to review?
- Authentication? Authorization?
 - Expiration?
- What is server-generated and what is front-end?
 - Any changes/extensions to services?
- Database impacts?
- Does this impact existing code?
 - Adding features often complicated!
- What docs(internal/external) are needed?
 - Existing docs changed?

Example: A Social Media clone

Users

- Can login using same logins as Project 2
- Have Profile of personal data
 - Name, Location, Pic, Followers, Followed
- Can search for other users to Follow
- Can Follow other users
- See their Timeline of posts by Followed
- Can make posts (250 characters)
- Can "amplify" others' posts (show to Followers)

How long will it take you?

Do not answer that! Do not start work!

Simple questions:

- What is default page?
- What do they see when not following anyone?
- What are details about user info?
- What info does a "post" have beyond the words?
 - What info about posters on post?
- What "pages" do we have?
 - Deeplink URLs for seeing a person/post?
- What requires user be logged in?
- Validations/Error messages?

These are just the start

Incomplete Advanced questions:

- How many posts are shown?
- Infinite scrolling/Pages/Load More?
- Is Polling expected? Details?
- Can a user prevent someone following them?
 - Other Block/Mute/Safety behavior?
- Can a post be edited? (complex!)
 - Can Profile be edited?
 - Where can we edit Follow status?
- Languages?

Delayed Questions means Wasted Work

- Chosen data structures may be completely wrong
- Created services may be completely wrong
 - Chosen *resources* may be completely wrong

Worse, if you assume wrongly, *product* may be wrong

- Everyone frustrated

Asking later and waiting for answers:

- Counts under "development" time
- Not "planning" time
- In other words: counts against your success