Lecture 3: September 11

Project Planning: Discovery & Research

Update your slack topics

- Include a 1-2 sentence description of your proposed project and list of relevant technologies you plan to use.
- We will be sharing these with our alumni mentors so they can pick projects most related to them.

TravelGPT: We will develop an AI-based itinerary generator that combines user preferences and location reviews to produce optimized travel plans (Technologies: python, open-AI, generative AI)

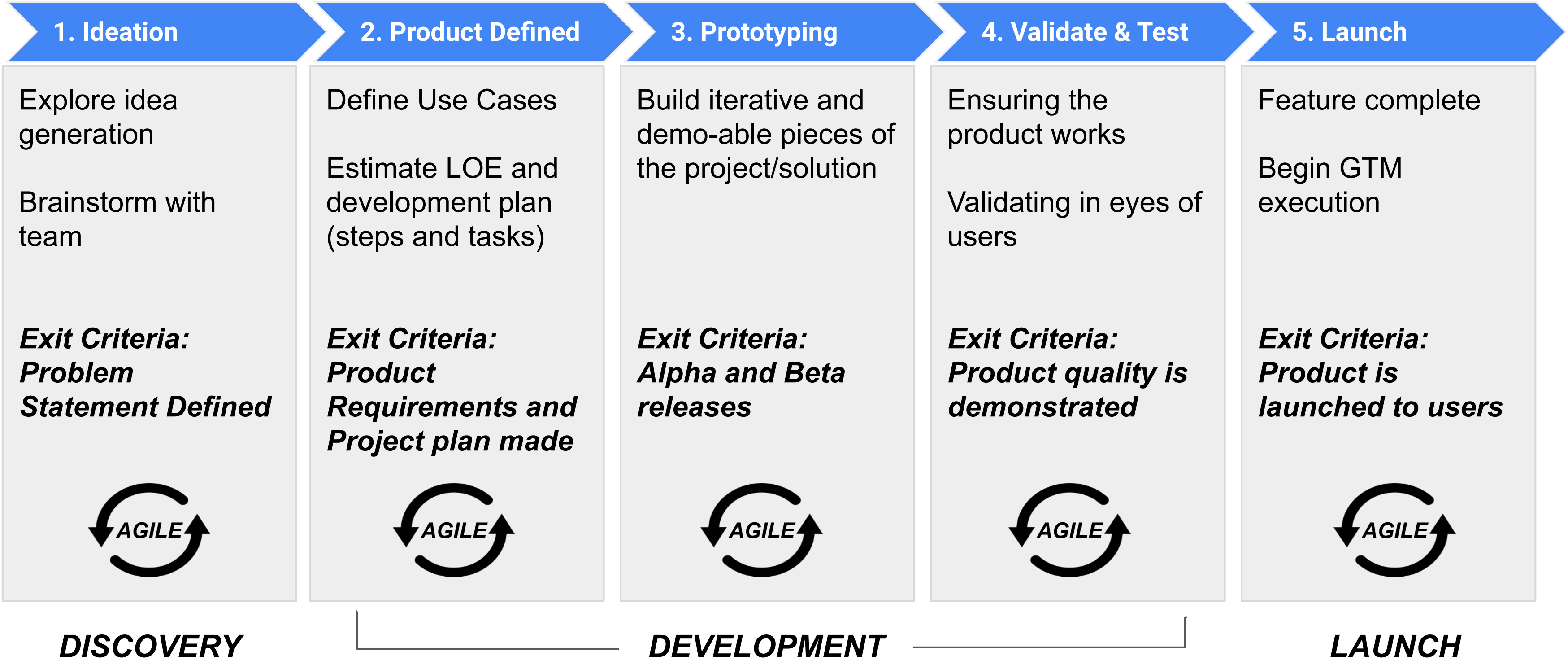
Agenda

- Lecture: Discovery & Research
- Assignment: Draft Project Proposal
- Tech Labs

Agenda

- **Lecture: Discovery & Research**
- Assignment: Draft Project Proposal
- Tech Labs

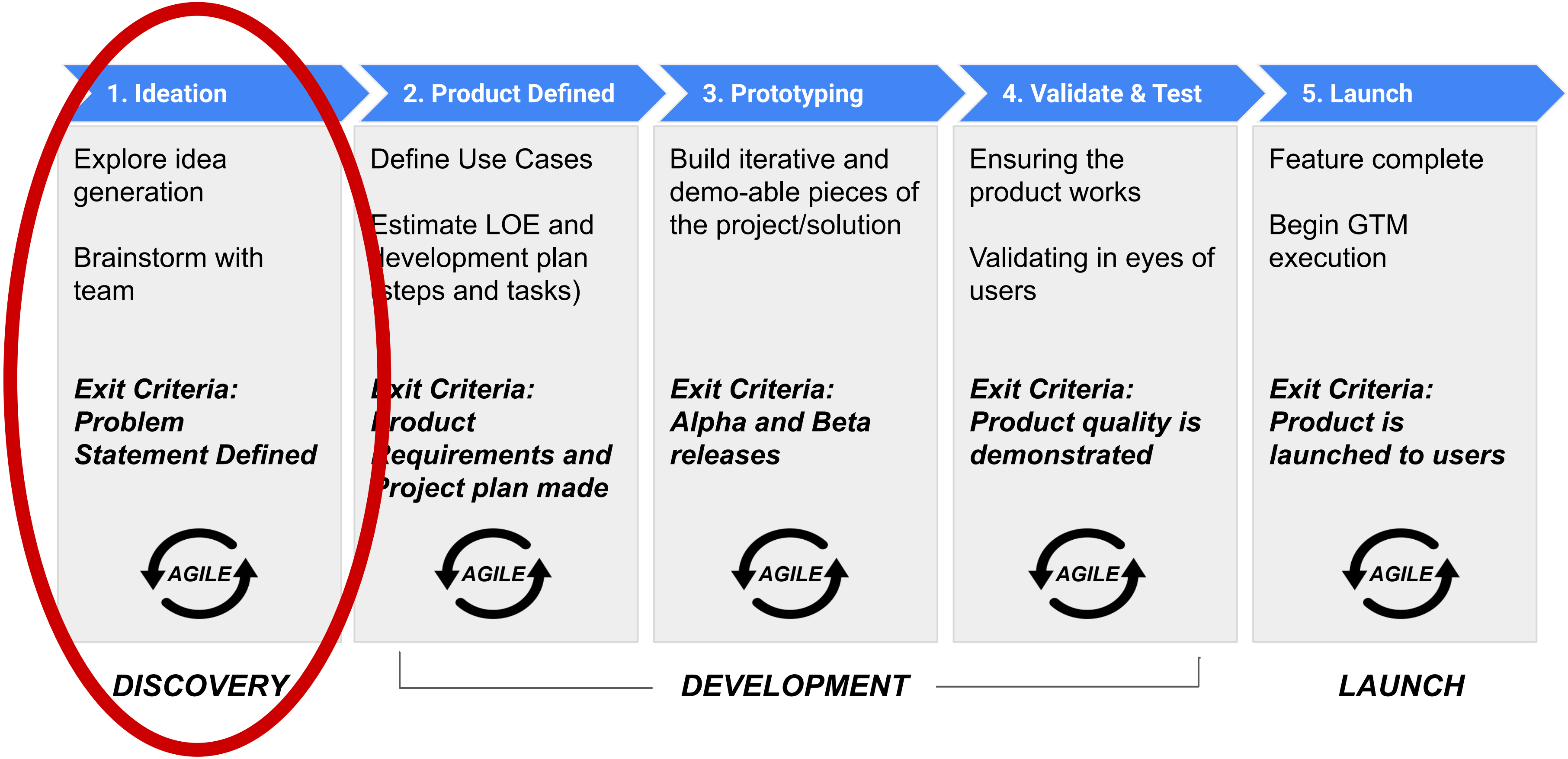
Product Development Lifecycle



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***

Discovery Phase



1. Ideation

Explore idea generation

Brainstorm with team

Exit Criteria: Problem Statement Defined

2. Product Defined

Define Use Cases

Estimate LOE and development plan (steps and tasks)

Exit Criteria: Product Requirements and Project plan made

3. Prototyping

Build iterative and demo-able pieces of the project/solution

Exit Criteria: Alpha and Beta releases

4. Validate & Test

Ensuring the product works

Validating in eyes of users

Exit Criteria: Product quality is demonstrated

5. Launch

Feature complete

Begin GTM execution

Exit Criteria: Product is launched to users

DISCOVERY

DEVELOPMENT

LAUNCH

Industry approaches to product discovery

Top-down

Leadership sets Objectives or “Top Projects” and team executes on solutions to those objectives / project ideas

Pros: Often aligns with broader business goals, more strategic

Cons: Can be disconnected from on-the-ground realities or actual user needs

Bottoms-up

Product team pitches roadmaps to leadership and priorities are set based on those ideas

Pros: Tends to be more user-centric, can uncover unique opportunities, often more agile

Cons: Risks being too narrow or not aligning with broader strategy

These approaches aren’t necessarily exclusive

Key Questions to Begin Discovery

Problem-centric Questions

User-centric Questions

Market-centric Questions

Problem-centric Questions

Questions
What problem are we solving?
Who faces this problem?

Top-down:
Senior leadership or stakeholders identify a broad issue or market opportunity, and teams work to define the specifics. Often relies on market trends or competitive pressures.

Bottom-up:
Teams or individuals identify problems through their own experiences or insights. Can be more grassroots, originating from customer feedback or frontline observations.

User-centric Questions

Questions
Who are our users?
What are their needs, habits, and pain points?

Top-down:
Use broad market segments or personas defined by corporate strategy or marketing insights.

Bottom-up:
Gathered from direct user engagement (user interviews, feedback, ethnographic research). Often more granular and specific.

Market-centric Questions

Questions
What are the existing solutions?
What is the market opportunity?

Top-down:
Driven by market research reports, competitive analyses, and larger industry trends. May involve third-party consultants / research firms.

Bottom-up:
Grounded in direct observations of user behavior, competitor product analysis, and grassroots market feedback.

September Goal

Answer the question:
What are you building and why?

Discovery and Research Methods

If we have a product, what can our product already tell us about the problems?
If we don't have a product, how can we learn more about the problems?

Goal: ensure product decisions are user-centric, data-driven, and aligned with market needs

Research Methods:

- Qualitative approaches: focus on the why (user behavior)
- Quantitative approaches: focus on the what (patterns in data)

Qualitative vs Quantitative approaches

Qualitative

- Deals with descriptions & interpretations
- Offers insights into user behavior & motivations
- Understand the “why”

Quantitative

- Deals with measurable data to formulate facts & uncover patterns
- Provides broad, measurable insights
- Understand the “what”

Discovery and Research Methods

Qualitative Tests

- Usability Testing
- User Interviews
- Conferences, Social Events
- Competitive Research

Quantitative Research

- User surveys
- User billing, segmentation data
- Product data
- A/B Testing

Considerations when choosing research methods:

- Market size, Customer population size
- Type of product
- Cost

Research tradeoffs by customer

B2B

- Small customer pool
- Intimate convos
- Higher value per customer

B2C

- Large customer pool
- More quantitative methods used
- Lower value per customer

Research

- More explicit user groups

B2B2C, B2G, and more...

Role: UX Researcher

Who? Roles and Responsibilities?

- Uncovers user behaviors, needs and motivations to make products, services and websites more intuitive and enjoyable for users
- Uses qualitative and quantitative methods, they conduct comprehensive research
- Share the insights from research with the UX designers
- Works with Product Managers, Designers, and Engineers

Role: UX Designer

Who? Roles and Responsibilities?

- Builds out UX ideas, prototypes, and designs based on the product’s problem to be solved and success criteria
- Collaborates with Product team in order to enable engineers to build technical solution
- Works with Product Managers, Engineers, and other project/product team members

Role: Data Scientist / Analyst

Who? Roles and Responsibilities?

- Find patterns and trends in datasets to uncover insights
- Create algorithms and data models to forecast outcomes
- Deploy data tools
- Share insights with org and team
- ***Scientist***: Works to create data vision and strategy for organizations
- ***Analyst***: Usually works with a team to help uncover findings based on team’s needs

Agenda

- Lecture: Discovery & Research
- **Assignment: Draft Project Proposal**
- Tech Labs

Draft Project proposal (due 9/15)

- Team submission
- 3 slides minimum
    - Overview of the project: what are you building, and why?
    - Algorithmic challenges
    - Technical challenges/technologies used
- No need for diagrams/images, slides are just a way to structure your thoughts
- Submit through blackboard

Agenda

- Lecture: Discovery & Research
- Assignment: Draft Project Proposal
- **Tech Labs**

Tech Labs - Requirements

1. You can work on these labs together, but **each student must submit their own code**
2. Each team must complete at least 2 different tutorials (not everyone can work on the same thing)
3. You can choose one of the suggested topics, or choose your own

Tech Labs - Topics

1. Backends:
   a. Python backend web app (django, flask, fastapi)
   b. Node.js / Express.js
2. Frontends:
   a. React
   b. iOS
   c. Android
3. ML
   a. Google Colab
   b. Pytorch
   c. sklearn
4. IoT, Raspberry Pi, Arduino

Tech Labs - Python Web Apps

Common python frameworks for creating backends

1. Django
   - Full-featured all-in-one web framework. Includes ORM, authentication, admin UI, etc
   - Suitable for complex web applications, but comes with a steep learning curve
2. Flask
   - Lightweight library good for rapid development
   - Lacks a ton of built-in features, relies on additional extension libraries
3. FastAPI
   - Modern, asynchronous python framework good for rapid prototyping
   - Relies on type annotations for I/O interface, self-documenting
   - Relatively new, might lack mature solutions

Tech Labs - Python Web Apps

Choose a framework and complete at least the first tutorial

1. Django
    - https://docs.djangoproject.com/en/5.0/intro/tutorial01/ (parts 1-4)
    - https://code.visualstudio.com/docs/python/tutorial-django
2. Flask
    - https://flask.palletsprojects.com/en/3.0.x/tutorial/
    - https://code.visualstudio.com/docs/python/tutorial-flask
3. FastAPI
    - https://fastapi.tiangolo.com/tutorial/ (basic & advanced tutorial)
    - https://www.tutorialspoint.com/fastapi/index.htm
    - https://code.visualstudio.com/docs/python/tutorial-fastapi

Tech Labs - Node.js / Express.js

If you're familiar with javascript, you can write your backend in javascript as well

Node.js: javascript runtime allowing developers to run javascript server-side

Express.js: a minimal, flexible web app framework for Node.js

Choose one of the following (do both if you have time)

- https://codexam.vercel.app/docs/project/xt/xt1
- https://codexam.vercel.app/docs/project/mernchat (fullstack + db + react)

Tech Labs - Front Ends

- **React**: common front end for web-apps, written in javascript

- **iOS**: mobile operating system in the Apple ecosystem. Defines a framework for developing mobile apps, written in Swift. Used for frontend, can also be used for backend.

- **Android**: mobile operating system from Google. Defines a framework for developing mobile apps. Used for frontend, can also be used for backend.

Tech Labs - Front Ends

- **React**: (choose one, do both if you have time)
  - https://react.dev/learn/tutorial-tic-tac-toe
  - https://www.freecodecamp.org/news/react-tutorial-build-a-project/
  - https://codexam.vercel.app/docs/project/mernchat (fullstack + db + react)
- **iOS**: (complete the first, get as far as you can in the second)
  - https://www.swift.org/getting-started/swiftui/ (focused on swift ui)
  - https://developer.apple.com/tutorials/app-dev-training (thorough but very long, won’t finish)
- **Android**:
  - https://developer.android.com/get-started/overview

Tech Labs - ML

Complete the intro to Google Colab tutorial. Then choose at least one of the pytorch tutorials OR the sklearn tutorials. Export the notebook and upload to github.

- **Google Colab**: web-based jupyter notebook that provides free access to gpu compute
  - https://colab.research.google.com/# (intro to colab)
- **Sklearn**: library providing non-deep learning ml algorithms + training utilities
  - https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.02-Introducing-Scikit-Learn.ipynb
- **PyTorch**: library for deep learning commonly used in industry
  - https://pytorch.org/tutorials/beginner/basics/intro.html
  - https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
  - https://colab.research.google.com/github/phlippe/uvadlc_notebooks/blob/master/docs/tutorial_notebooks/tutorial2/Introduction_to_PyTorch.ipynb
- **Datascience handbook:** useful resource on ml & datascience as a whole
  - https://github.com/jakevdp/PythonDataScienceHandbook/tree/master
  -

Tech Labs - IoT / Raspberry Pi / Arduino / etc

- Any tutorials with a hardware component. Bring your own hardware and we’re happy to help!
    - Arduino: https://docs.arduino.cc/built-in-examples/
    - Raspberry Pi: https://tutorials-raspberrypi.com/
- ROS: robotic operating system – used as part of the RTX projects
    - https://www.youtube.com/watch?v=979IZWOXC_0&list=PL8MgID9MCju0GMQDTWzYmfiU3wY_Zdjl5
    - https://www.youtube.com/playlist?list=PLy9nLDKxDN683GqAiJ4IVLquYBod_2oA6

For Next Week

- [**Individual**] push progress from tech lab to github **9/11**
- **[Individual]** fill out weekly status update – due before meeting w/ mentor or Wednesday lab, whichever comes first.

- [**Team**] Submit draft project proposal **9/15** (blackboard)
- [**Team**] Schedule first mentor meeting next week
  - Create a team zoom meeting & bookmark to your team slack channel

Tech Lab Submission

1. **Join Github Assignment** to create repo: https://classroom.github.com/a/PnF2mA90

2. **Push all code** from tech lab to main branch before end of lab.
   a. If you are doing an ML tutorial with google colab, export the notebook and copy it into your repo.

3. **Add a README.md** including:
   a. Which tutorial(s) you chose. If you found your own tutorials, link out to them & add a description of what they were.
   b. Who you worked with