

# Startup Studio

**AI-Accelerated Building and Validation**

01.29.26 C3W2

---

# Agenda

- Class size and waitlist
- Class structure recap
- Homework 1 Discussion
- Exercise
  - Fermi Estimation

# Instructor Intro

# Who am I?

## Ken St. Clair

- Early-stage start-up Product Leader with 20+ years experience
  - Software engineering, sales engineering, product design, product leadership
  - Two robotics products from 0→1
  - Experience with 0→1, growth, acquisition, and deprecation
  - B2B, B2B2C, B2C
- BS Computer Science, Georgia Tech
- Masters of Human-Computer Interaction, Carnegie Mellon



acescape



Carnegie  
Mellon  
University



# Whom have I Worked with?

## Select Partners and Clients

Gyms + Spas

EQUINOX

LIFETIME<sup>®</sup>  
HEALTHY WAY OF LIFE



LOTTE NEW YORK PALACE

REMEDY<sup>PL</sup>



Print Media

The New York Times

Bloomberg

THE  
WALL STREET  
JOURNAL.

*The Atlantic*

THE  
HUFFINGTON  
POST

VICE

Broadcast

TURNER

Disney

ESPN



NBC

VIACOM

©CBS

Streaming/Sports



twitch

ROKU



Food Service

OUTBACK<sup>®</sup>  
STEAKHOUSE

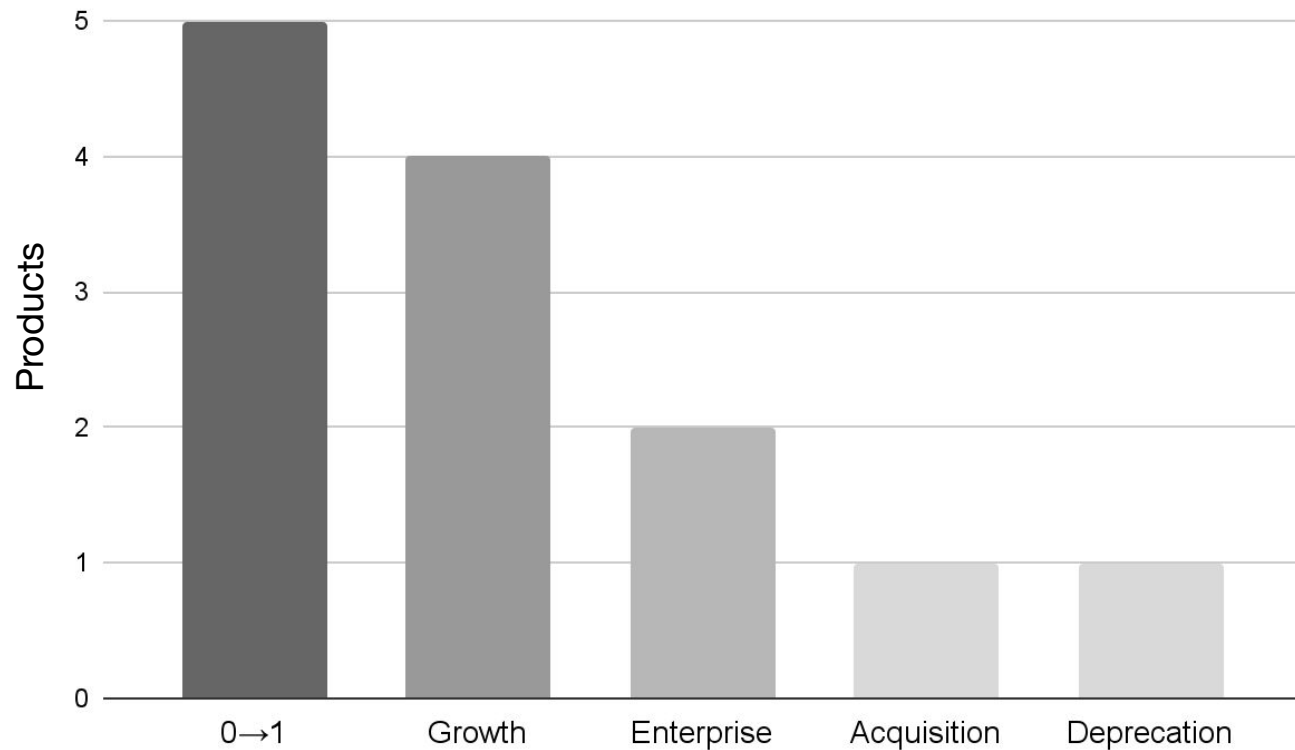


More...

tinder



# Product Stages I've Worked On



# **Class Size and Waitlist**

---

# Waitlist

- Very impressive amount of demand for this course (>150 registrations)
- Course is capped at 50. This is the final decision from the administration.
- We are limited by the studio nature of the course and the final, which will include industry leaders in a Demo Day
- Final enrollment list published after class Thursday



# Class Intro

---

# What's a “Studio”

A course focused on making, iterating, and critique.

Students learn by building real artifacts and receiving continuous feedback, rather than by absorbing theory first.



---

# What's “Startup Studio”

A course focused on making, iterating, critique, **experimentation and launching**.

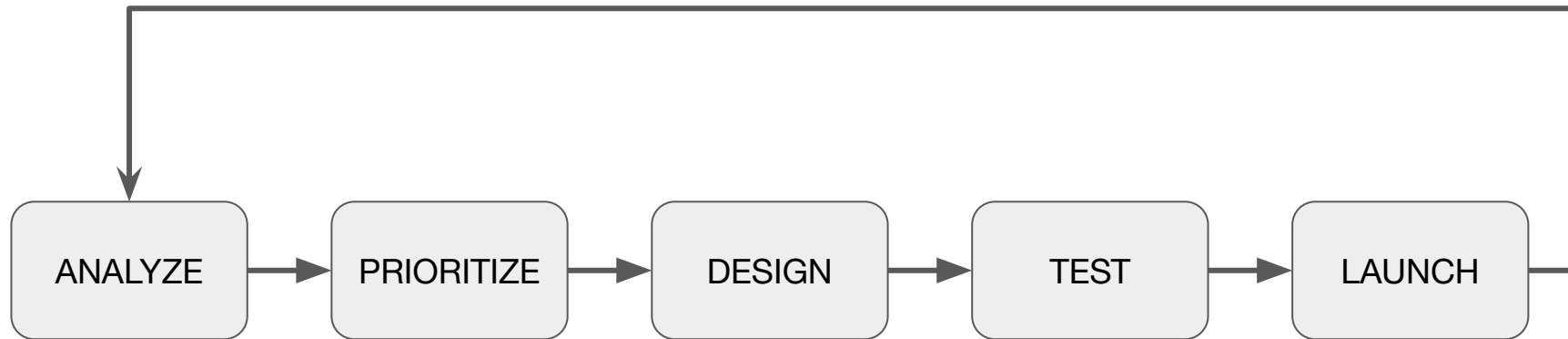
Students learn by building real artifacts and receiving continuous feedback, rather than by absorbing theory first.



---

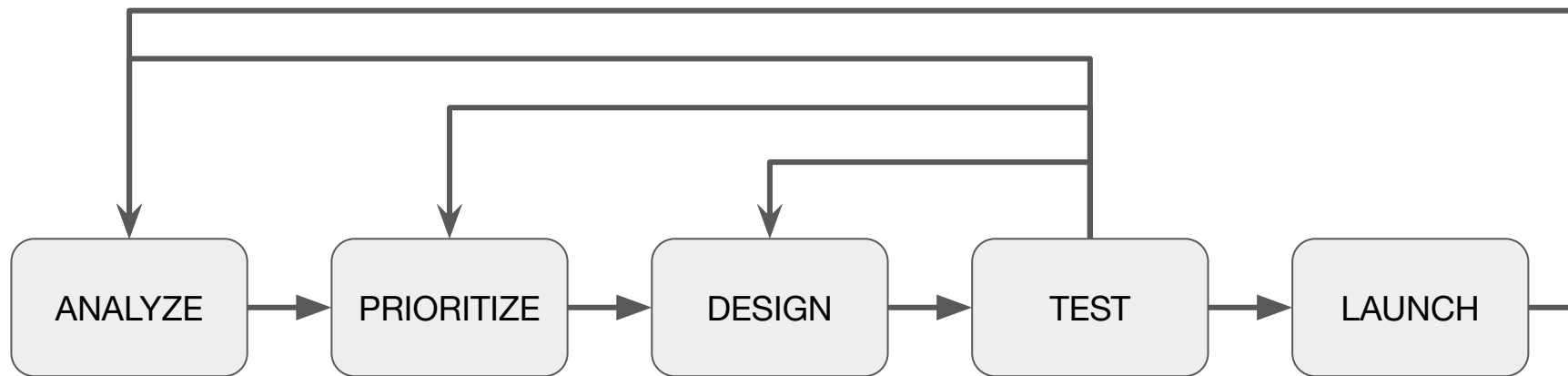
# Startup OS

The operating system for this class



# Startup OS

The operating system for this class



---

## What We're NOT Focusing On

- Hardware
- B2B
- Research/Publishing
- Theory/Deep Foundational Learning



This is not an A+ course

---

## **Class Position on AI**

Use AI for everything.

The AI you use will eventually come with a monetary cost.



---

# Waitlist

- The course is highly oversubscribed. If you are on the fence in any way, kindly make way for those that will find the most value from the course
- Be sure to let your administrators know you'd like to see the course offered again.

# Homework 1 Discussion

---

# What did you Build? krs2212

- What did you create?
- Which AI tools did you use?
- What was your process?
- How long did it take?

Gmail chat me your link and I'll show it (krs2212)

---

# What did AI do well?

What surprised you about what AI could do? Where did it exceed your expectations?

- Letting AI decide color was fairly clean
- Original UI looks strong
- Quick for tightly scoped
- Does well at making stuff up for quick pitches/prototypes
- API integration much easier
- Suggesting thing you don't know
- Feature by feature gave strong results

# What did AI Struggle with?

Where did you have to step in? What did AI get wrong repeatedly? What took longer than expected?

- Feature 3 breaks feature 1
- A lot of bloat
  - Garbage code - really big functions that aren't needed
  - Workflow Errors create new files
- Wanted to use a specific library/mathematical process, but wasn't able to bring it into the tool that was being used
- Made up user ratings and other marketing data that was not true
- More polished UX, especially around semantics, was lacking
- Dead buttons/links not leading anywhere
- Unrealistic expectations of humans in the UX
- Outdated versions of SDKs
- Large large codebases without a lot of direction
- CONTEXT IS LOST EVERY CHAT
- Get stuck on ideas, goes down rabbit holes
- Complex deployment

---

# The Reality of AI-Assisted Building

**AI is a productivity multiplier, not a replacement**

You still need:

- Clear vision & requirements
- Technical judgment
- Debugging skills
- Architecture understanding

# How do Humans do this?

For those that have worked in building software, what's the human process?

- Product manager analyzes business goals
  - Negotiate with other business stakeholders
- PM works with Engineering Manager
  - Divides work into sprints/chunks
- Storyboard for workflow from UX designer
  - Breaking down into stories/features for consumability
- Test
  - Unit/Regression
  - QA
  - Beta
  - User Testing
- Users
- Marketers
- Designer

What works for humans typically also works for AI

---

# How to Make AI Development Work

Clear requirements documents

→ helps AI understand context

Breaking work into specialized roles

→ matches AI's strength in focused tasks

Living documentation

→ gives AI persistent memory

Human oversight at key checkpoints

→ catches AI's blind spots



# A Quick Glance at Structure

docs		1	Dashboard Status Pill – Progress
features		2	
20260121.01_more_homepages		3	## Status: Complete
20260121.02_testing_updates		4	
20260121.03_metacortex		5	---
20260123.01_dev_chat		6	
20260123.02_frameworks		7	## Completed
20260123.03_quiz_vnext		8	
20260124.01_track_allocation		9	- [x] Phase 1: Backend StatusPillService with priority logic and copy mapping
20260126.01_engine_testing		10	- [x] Phase 2: API integration via SettingsSerializer
20260128.01_confirmation_polling		11	- [x] Phase 3: Frontend StatusPill component with tooltip and CTA handling
20260128.02_ux_polish		12	- [x] Phase 4: Dashboard integration replacing current getStatusInfo logic
20260128.03_dashboard_status_pill		13	- [x] Phase 5: Resume pause endpoint with immediate UX feedback
dashboard_status_pill_architecture.md	U	14	- [x] Phase 6: Analytics events
dashboard_status_pill_brief.md	U	15	- [x] Phase 7: Mobile polish
dashboard_status_pill_plan.md	U	16	- [x] Phase 8: Edge case handling (multiple blockers, timezone, fetch errors)
dashboard_status_pill_progress.md	U	17	
ongoing		18	---
diagrams		19	
api_request_flow.md	U	20	## In Progress
delivery_pipeline.md	U	21	
domain_model.md	U	22	_None_
inbound_message_flow.md	U	23	
metacortex.md	U	24	---
onboarding_flow.md	U	25	
user_states.md	U	26	## Remaining
ANALYTICS_CATALOG.md		27	
BRAND_POSITION.md	M	28	- [ ] Manual QA: verify all pill states render correctly
DELIVERY_SCHEDULING.md	M	29	- [ ] Manual QA: verify CTA actions work end-to-end
MESSAGE_ARCHITECTURE.md	M	30	- [ ] Manual QA: verify "Resume now" feels immediate
PRD.md	M	31	
STYLE_GUIDE.md		32	---
releases		33	
reports		34	## Notes
user_testing		35	
steady_thread		36	- Current implementation in `AppHeader.jsx` uses client-side logic that doesn't include policy checks (pause, quiet dates, cadence)
OUTLINE		37	- Backend has all the building blocks: `Deliveries::Eligibility`
TIMELINE			

# Exercise

---

# Fermi Estimation

## What is Fermi Estimation?

Breaking down complex estimates into smaller, manageable assumptions to calculate approximate audience sizes.

## Why It Matters

Helps you prioritize which personas to target. Larger audiences = more potential impact and validation.

---

# Fermi Estimation

## Example Framework

1. Start with a large population (e.g., US adults = 260M)
2. Apply filters based on your persona (e.g.,  $\times 40\%$  live in cities = 104M)
3. Narrow by behavior/need (e.g.,  $\times 15\%$  order takeout weekly = 15.6M)
4. Apply additional constraints until you reach your target audience

---

# Fermi Estimation

AGM Trick (Approximate Geometric Mean)

**Time per day a 15-yr old watches TV?**

2min - 400min

$2 \times 10^0 - 4 \times 10^2$

Average coefficients and powers

$3 \times 10^1$

**End of Class 3**