

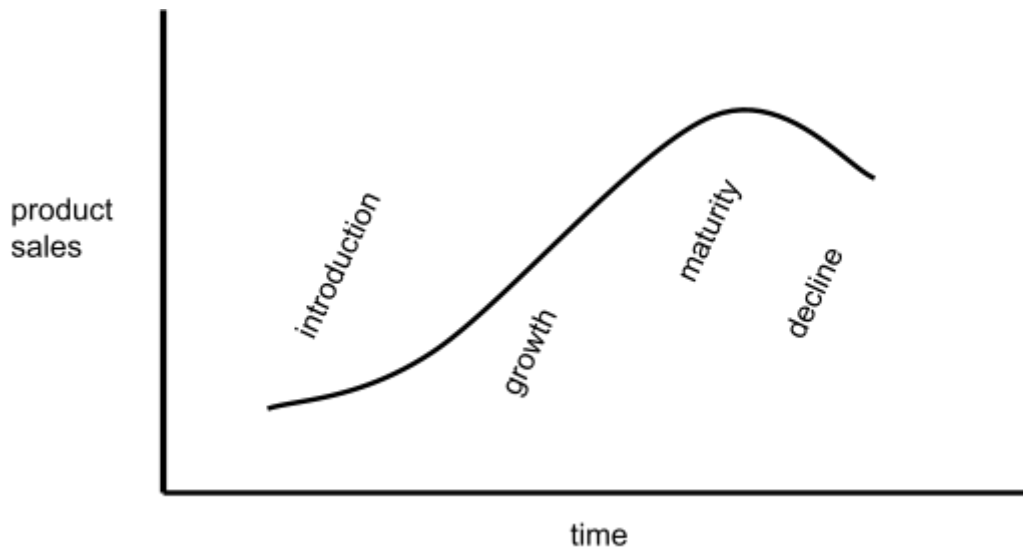
→ Section 2: Introduction to Product Management

- ◆ does not manage people
 - cross-functional collaboration/enabling
- ◆ responsible for the ultimate success of product
- ◆ typically in charge of a section of a product, not whole product, i.e.
 - newsfeed feature
 - user profiles
 - messaging feature
 - commenting feature
- ◆ can be split by platform, i.e. mobile, web, etc.
- ◆ types of PMs primarily differ by stakeholders
 - Internal PM
 - internal tools PM'ing for the other people or teams within org
 - Business to Business PM
 - solutions for other companies
 - i.e. Oracle
 - a lot of interfacing with sales teams
 - Consumer PM
 - product for average customer
 - most common
 - i.e. Facebook, Instagram, Twitter
- ◆ Product vs Project Management
 - product managers responsible for reaching certain metrics
 - success defined by KPIs or metrics
 - project manager typically responsible for accomplishing a project
 - project usually has a specific timeline and budget as a constraint
- ◆ day in the life
 - emails, research
 - tools: techmeme, hacker news, launch ticker, product hunt, twitter buzz

→ Section 3: Introduction to Product Development

- ◆ four major phases of the product lifecycle (market stages)
 1. Introduction
 - product first introduced to market
 - little to no competition
 - typically loses \$\$\$
 - usually only early adopters buying
 2. Growth
 - accepted by marketplace
 - sales rise

- introducing improvements to stay competitive
- few competitors
- 3. Maturity
- 4. Decline
- ◆ phases/lifecycle is just a tool/framework



- ◆ product development process
 1. conceive
 - collect user problems
 - brainstorm solutions
 - identify focus areas
 2. plan
 - market research
 - business cases development
 - perform customer interviews
 - roadmap, timelines, etc.
 3. develop
 - make specific timelines
 - write out features
 - make user stories and specs
 - gather estimations from dev
 4. iterate
 - finish MVP
 - get feedback
 - testing
 - alpha/beta releases
 5. launch
 - work with marketing, PR, sales, legal, etc.
 6. steady

- collect metrics on how people are using product
 - how often are people buying
 - analyze product and optimize
 - marketing team continuing to market
 - sales team continuing to sell
- 7. maintain or kill
 - not always related to revenue or usage
- ◆ lean product development
 - cutting out unnecessaries
 - not using resources unless you have to
 - don't waste resources !!
- ◆ agile
 - scrum: tickets, sprints, sprint backlog, standup, etc.
 - kanban: tickets, kanban board, no timebox, certain # of tickets, etc.

→ Section 4: Ideas and User Needs

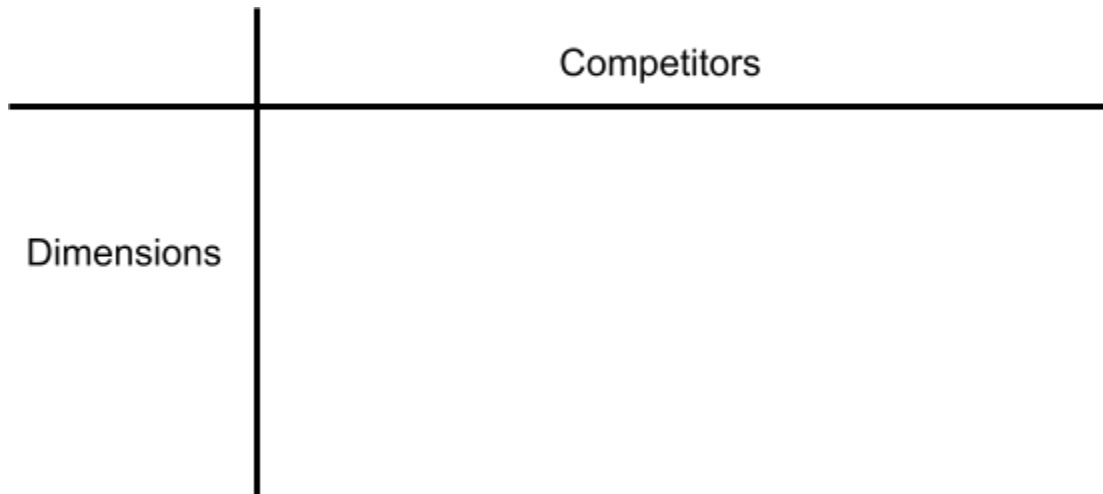
- ◆ where do ideas come from?
 - everywhere
 - E - Employees: coworker, mgmt, self, etc.
 - M - Metrics: problems, inefficiencies, how users use product, etc.
 - U - User feedback: reviews, forums, etc.
 - C - Clients/Customers (usually more common in B2B)
 - internal PM
 - stakeholders
 - B2C PM
 - users, metrics, coworkers
 - B2B PM
 - employees, clients
- ◆ getting to real user needs
 - is this solving an actual problem?
 - can this have unintended side effects?
 - what is behind this request?
 - ask why? why? why?
 - find the REAL pain
- ◆ users vs. customers
 - feedback
 - Macro-level (marketing, directors)
 - Everyday users (more granular)

→ Section 5: Competitive and Market Analysis

- ◆ part of PM'ing is making sure that market you're attempting to address is large enough to make it worth going into
- ◆ approaches to market sizing
 - Top down

- finding total market then estimating what your share of the market is
 - problem is you need to make assumptions
- Bottom up
 - find current sales of similar products then estimate how much sales you can capture
 - more accurate but requires more time and effort
- ◆ tools:
 - alexa.com, semrush.com, google adwords
- ◆ finding competitors
 - feature triage
 - need to know which potential opportunity, feature, or project is most likely to get more users, make users happier, enhance brand, etc.
 - build list of potential competitors then carve out
 - different types of competitors
 1. direct competitors
 - ◆ typically offering same/similar product or solving the same problems
 - ◆ user will have to make conscious decision to use product versus other
 2. indirect competitors
 - ◆ solve same problem in different way and for different target customer group
 3. potential competitors
 - ◆ offer something to same target customer group (or similar) but don't address same problem that you do
 - ◆ in periphery
 4. substitute competitors
 - ◆ solves same core problem but not angled or delivered in same way
 - ◆ generally doesn't target same people
- ◆ five criteria for understanding competitors
 - product core
 - the product team
 - ask: how good is the competitor's product team?
 - good competitor product team can outperform, engineer, execute, etc. you
 - know the size of their user base
 - large user base = market dominance
 - easier to negotiate/strike deals
 - design
 - people are more inclined to use well-thought out/good design
 - brand

- perception people have of you and products determine a lot
 - demand higher level of customer loyalty
 - being pigeonholed can be dangerous
- speed
 - how quick is turnaround
- ◆ feature table
 - dimensions → features and factors
 - e.g. price, reliability, feature X, ability to do Y



- example: GoPro

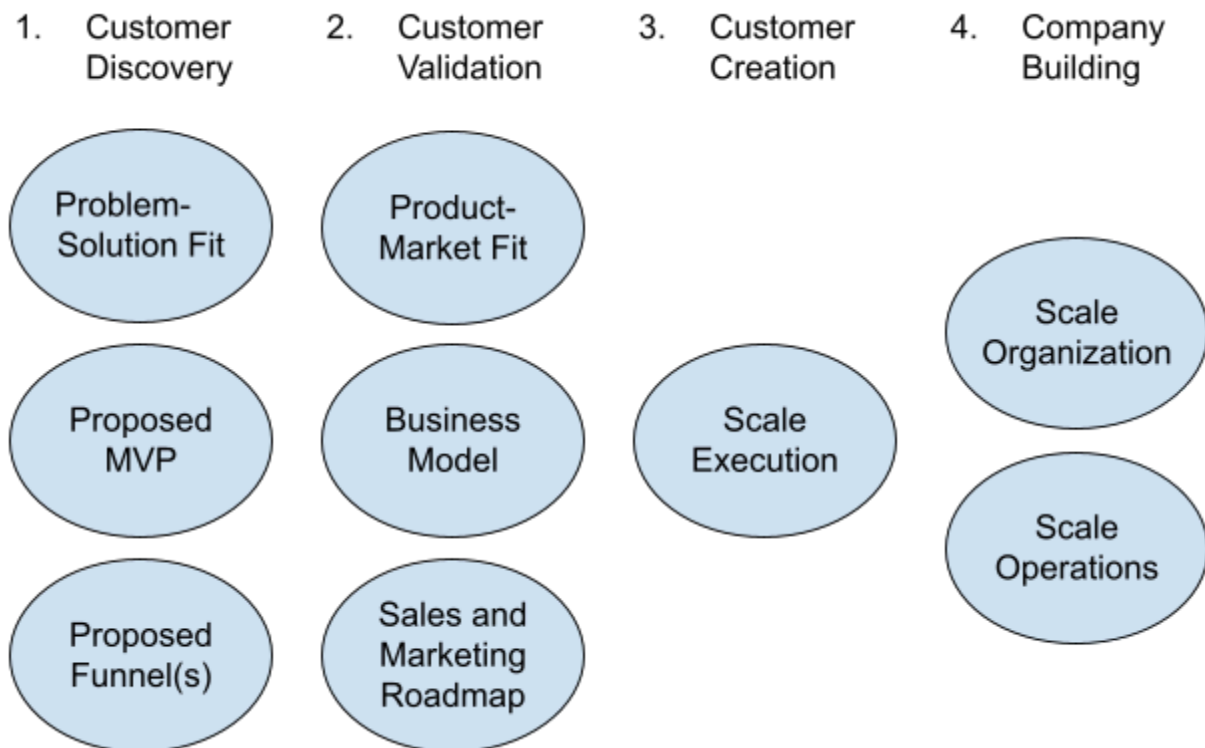
	Tom Tom Bandit	Veho	Sony FDR	GoPro
Price	\$349	\$200	\$399	\$499
Aesthetics	4/10	6/10	6/10	7/10
Weight	190g	84g	114g	89g
4k	Yes, 12 FPS	No	Yes, 30 FPS	Yes, 30 FPS
Max FPS	60 FPS	60 FPS	120 FPS	120 FPS
Accessories	10+	15+	25+	100+

- ◆ monitoring competitors
 - things worth noting

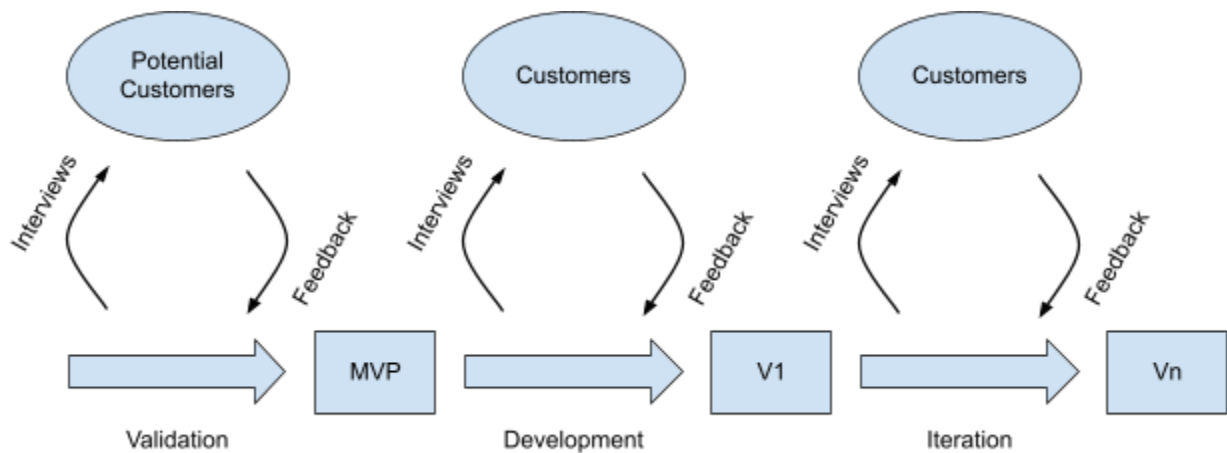
- funding
 - ◆ more money can make a difference
 - ◆ tool: crunchbase.com
- acquisitions
 - ◆ something companies buy other company because they want to acquire product team, user base, design team
- new features/products
 - ◆ tool: medium.com, google alerts

→ Section 6: Customer Development

- ◆ the practice of establishing a continuous and iterative communication line with your customers
- ◆ communication is crucial
- ◆ customer development framework



- customer interview
 - increase product IQ by understanding the real reason why your customers use or do not use your product
- customer development cycle



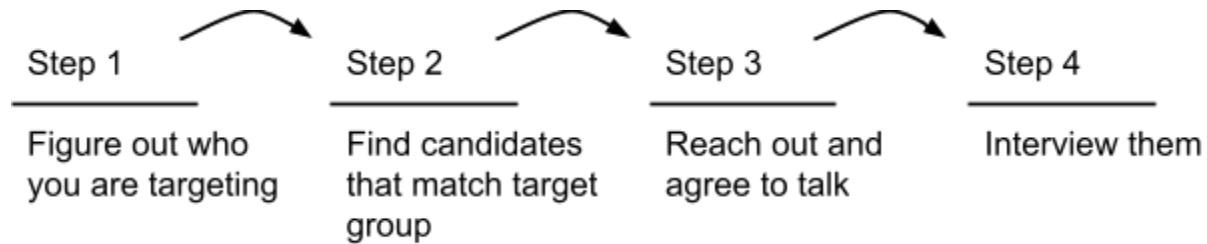
◆ four types of interviews

- exploratory interview
 - exploring what customer are excited about, have a problem with, etc.
 - examples
 - ◆ “what’s the worst part of your commute?”
- validation interview
 - test a theory
 - hypersensitive to bias
 - present idea then collect honest feedback
- satisfaction oriented interview
 - figure out which parts of product is working, which are not
 - root of satisfaction/dissatisfaction
- efficiency interview
 - who uses product, when do they use it, under what circumstances, etc.

◆ key differences in customer development

Pre-Product	Post-Product
• Potential customers • They do not know you • Focus on pain points + validation • Finding interviewees can be difficult	• Existing customers • They know of you • Focus on satisfying usability and pain points • Finding interviewees is like shooting fish in a barrell

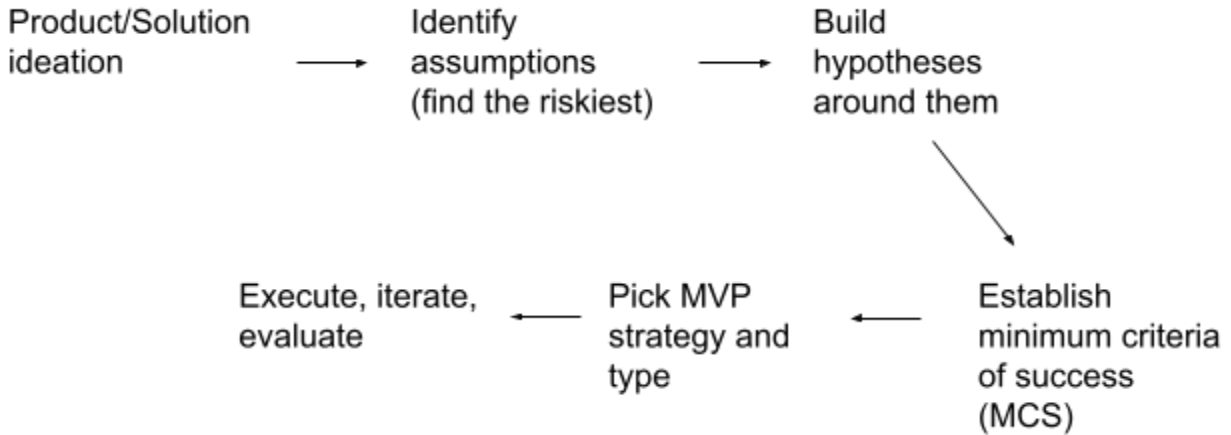
◆ talking to customers



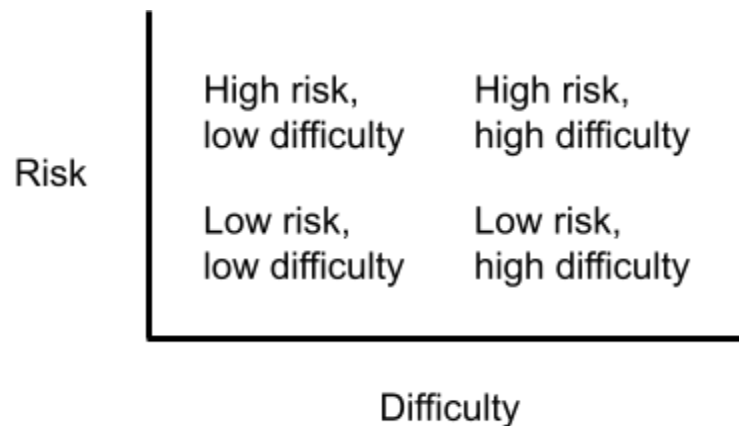
- tenets of cold email
 - be short
 - ◆ focus on brevity
 - be personal
 - be valuable
 - TIPS:
 - ◆ mention not from sales
 - ◆ make them feel special
- running customer interviews
 - don't talk about your solution
 - don't talk about your opinions
 - create a comfortable environment
 - don't force the conversation. guide it.
- remember:
 - what?
 - why?
 - who?
 - when?
 - where?
- build user personas off interviews
 - user persona: a representation of behavior by many users in the form of one fictional person that you can understand and refer to easily

→ Section 7: Designing and Running Experiments

- ◆ minimum viable product (MVP)
 - version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort
 - designed to test hypotheses and assumptions
 - way to mitigate risk
- ◆ steps to running an mvp experiment



- identifying assumptions
 - “in order for my idea to be successful the following must be true...”
 - ex. google glasses
 - ◆ assume people like/are comfortable wearing glasses
 - ◆ assume people willing to pay \$\$\$
 - framework: I assume that....
 - ◆ my customers have X, Y, Z problems
 - ◆ _____ matters to my customers
 - ◆ _____ will pay for it
 - ◆ no satisfying substitutes
- ◆ prioritizing: which assumption to test first?



- maybe....

Priority



1. High risk, low difficulty
2. High risk, high difficulty
3. Low risk, low difficulty
4. Low risk, high difficulty

- high difficulty → resource-intensive
- low difficulty → not resource-intensive
- ◆ assumption vs. hypothesis?
 - hypothesis is actionable

Simple

We believe (subject/target) will (predicted action) because (reason).

Better

If we (action), we believe (subject) will (predicted action/outcome) because (reason).

The “PM” Way

We believe (subject) has a (problem) because (reason). If we (action), this (metric) metric will improve.

- more specificity → more focus
- ◆ outcomes of MVP test
 - your hypothesis is false and not worth doing
 - your hypothesis is true without question
 - you’re somewhere in the middle
- ◆ minimum criteria for success (MCS)?

Cost

- Developers' time
- PM time
- Other stakeholders' time
- Labor wages
- Advertising costs
- Brand effect
- Legacy issues
- Opportunity costs

Reward

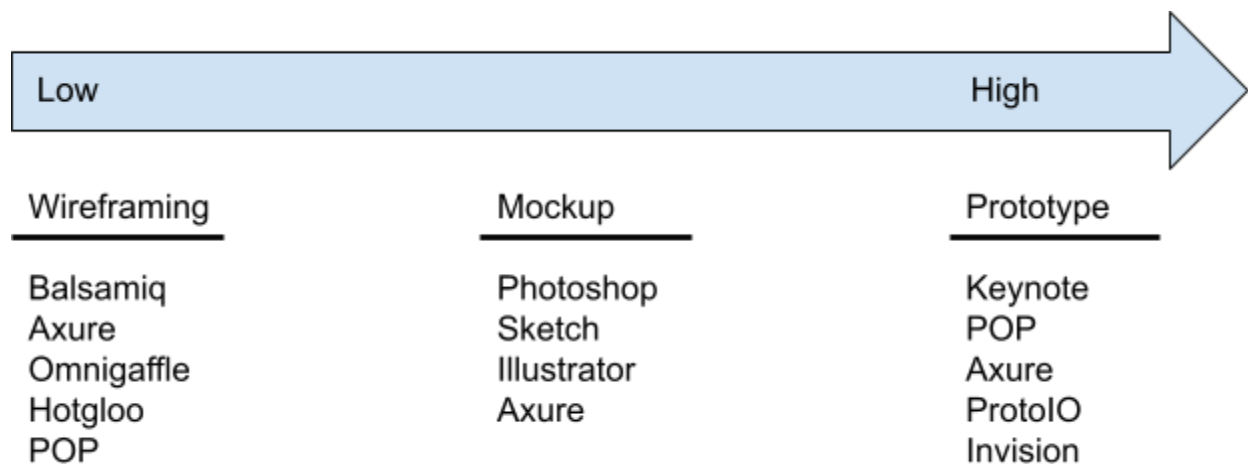
metrics like:

- Increased revenue
- Time spent on page
- # of shares
- Increased satisfaction
- # of likes
- Conversion rate
- Open rate
- Customer LTV

- ◆ at what point do benefits outweigh costs?
- ◆ general MVP types/techniques
 - email MVP
 - shadow button MVP
 - 404/coming soon MVP
 - explainer MVP
 - fake landing page/pitch experiment MVP
 - concierge MVP
 - piecemeal MVP
 - wizard of oz MVP

→ Section 8: Conceptualizing the Solution

- ◆ wireframe → visual guide of app, site, product, etc.
- ◆ mockup → more visual detail
- ◆ prototype → more usability



→ Section 9: Metrics for Product Managers - Defining Success and Measuring Results

- ◆ “what gets measured, gets managed”
- ◆ feedback loops: the more frequently you’re getting accurate feedback about something, the more effectively you are going to manage that
 - i.e. fitness devices that provide a lot of metrics to users/wearers
- ◆ real life examples
 - twitter
 - growth metrics:
 - ◆ total new users per _____
 - ◆ monthly/daily active users
 - ◆ activated users
 - engagement metrics
 - ◆ multiple logins per day
 - ◆ time spent
 - ◆ # of tweets
 - ◆ average # of likes, re-tweets, and follows
 - ◆ # messages sent
 - youtube
 - engagement metric
 - ◆ watching at least 30 seconds of video is indicator of quality
- ◆ types of metrics
 - growth and activation
 - track how product or company is growing
 - user acquisition
 - how are users finding you?
 - retention
 - actually retained users
 - resurrected users
 - engagement
 - specific behaviors
 - user happiness
 - level of satisfaction
 - revenue
 - example metrics
 - ◆ lifetime value (LTV)
 - ◆ cost of acquisition of customer (CAC)/cost of customer acquisition (CCA)
 - ◆ monthly recurring revenue (MRR)
 - ◆ annual recurring revenue (ARR)

Exploratory

- Things you're not always tracking
- Tracking an analysis
- i.e. # of times user clicks button somewhere

Reporting

- Things tracking over long periods of time to ensure product is doing well or heading in right direction

◆ HEART framework

- Happiness - How happy is your user?
- Engagement - How engaged is your user in the short term?
- Adoption - How many interested users have actually tried your product?
- Retention - How many users do you retain long term?
- Task - How successful are you at allowing users to perform the most valuable task?

◆ HEART chronological/funnel order → ATERH

- organizing

Goals	Signals	Metrics
What do you want to happen?	What is the thing we need to measure? i.e. ratings, survey scores, etc.	A signal expressed over time. This is the thing you watch.

	Goals	Signals	Metrics
H E A R T			

- ◆ AARRR framework
 - Acquisition - User acquired/arrived → users indicating interest
 - Activation - How activated
 - Retention - Always coming back
 - Referral - Sharing to others
 - Revenue - buying subscription, buying product, attracting ads
- ◆ Tracking metrics in practice and other tools
 - Google analytics
 - crazy egg
 - KISS metrics
 - mixpanel
 - optimizely
 - segment

→ Section 10: Building the Product - Product Mgmt for PMs

- ◆ product team typically builds out 3-5 epics during a quarter (in general)
- ◆ epic example: allow users to upgrade membership
- ◆ epic spec sheet
 - allow others to understand what you're building
 - four main sections
 - introduction
 - product requirements
 - design requirements

- engineering requirements
- ◆ user stories and acceptance criteria
 - acceptance criteria: set of conditions that software must satisfy to be considered complete
 - usually in ticket with user story
- ◆ sprint velocities → stories and story points calculated
- ◆ prioritization methods
 - assumption testing
 - remove risky assumptions
 - buck method
 - prioritize based on final score

$$\left(\begin{array}{c} \text{business benefits (1-10)} \\ + \\ \text{user benefits (1-10)} \end{array} \right) - \text{cost (1-10)} = \text{final score}$$

- may be challenging to score
- moscow method
 - MUST
 - COULD
 - SHOULD
 - WOULD

→ Section 11: Working with People and Stakeholders

- ◆ communication channels
 - meetings
 - conference calls
 - emails
 - informal meetings
- ◆ working with engineers
 - something goes wrong, it's your fault
 - when pitching, have a good idea of where feature will go in future
 - try to do work yourself before asking engineer
 - watch out for tech debt
 - do not treat engineers like an agency
- ◆ working with designers
 - give designers creative freedom
 - don't treat designers like an agency
 - you and designer are a team
 - don't tell them what to do
 - always talk about user problems first, solution second
- ◆ working with executives and others (legal, marketing, etc.)

- ◆ TIPS for executives
 - be brief
 - always speak in terms of business effect
 - communicate in their style
- ◆ TIPS for other stakeholders
 - talk to people on each team as much as possible
 - ensure other teams understand that you know user base, technology, and business very well
 - update them on latest developments as frequently as possible
 - be sure to tell stakeholders the reason solution is the way it is

→ Section 12: Technology for PMs

- ◆ blank