

Becoming a Product Manager



- ⚠ Product Manager is not a manager of anybody
- ⚠ Product Manager is organizer and enabler
- ⚠ Product managers can be split up by features or platforms
- ⚠ types of product managers
 - internal PMs
 - consumer PMs
 - business to business PMs (SaaS PMs)

→ Primary difference : stakeholders

⚠ B2B PM talks to sales team a lot

- ⚠ internal PM
 - great intro role
 - integrate with other systems
 - project management
 - less risk
 - small # of users

△ B2B PM

- good intro role
- small # of users
- flexible and creative
- priorities weighted by \$
- tight deadlines
- one or very few platforms

△ B2C PM

- uncertainty
- pressure
- millions of users
- multiple platforms
- can lose company \$\$
- extensive user testing

△ Product managers are responsible for success of their product

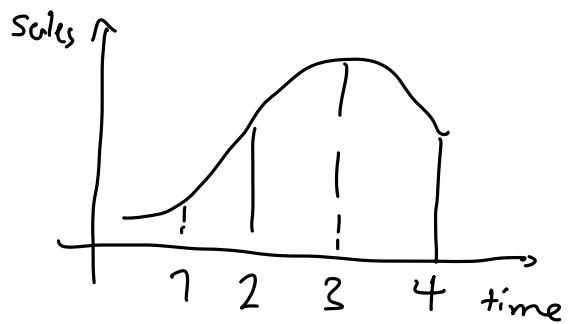
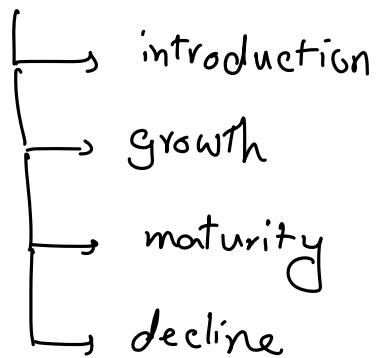
↳ success is defined by KPI or metrics

△ Project manager is responsible for accomplishing a project
not a goal

↳ A project usually has a timeline and a budget
as a constraint

⚠ keeping up with industry is very important as a PM

⚠ product life cycle



⚠ introduction phase

- ↳ Product first introduced to the market
- ↳ little to no competition
- ↳ typically loses \$\$\$

⚠ growth phase

- ↳ product is accepted by marketplace
- ↳ sales rise
- ↳ start improving product
- ↳ few competitors

⚠ Maturity phase

- ↳ sales peak
- ↳ competitors enter market

Δ Decline phase

- Sales diminish
- Products phased out
- deemed old / irrelevant

Δ Product development process

- conceive
- plan
- develop
- iterate
- launch
- steady state
- maintain or kill

Δ Lean is just about building something in an intelligent way
where i'm not using the resources until i know i actually
have to

Δ Agile is just a way of applying the lean mindset to
software development



△ SCRUM main parts

- sprint planning meeting
- using tickets for tasks
- standup meeting
- retrospective meeting

△ Kanban is not stricted like SCRUM in terms of time and meetings

△ Kanban says That only a certain number of items can be in progress at one time

△ when Waterfall is better than agile

- building operating system
- building mission critical products like a car
- building a skyscraper

△ where do ideas & needs come from

- E : Employees
- M : Metrics
- U : Users
- C : clients → pays for the product

⚠ things to do when getting a user need

- Is this solving an actual problem?
- Can this have unintended side effects?
- What is behind this request?
- What is the main reason?
- 3 times ask why? why?! why?!!

⚠ Market sizing

- top down
- bottom up (more accurate)

⚠ Feature Triage

- get you more users
- make your users happy
- enhance your brand

⚠ competitors

- direct : solving the same problem
- indirect : solving the same problem in a different way
and for a different target customers
- Potential : some target group but solving other problem
- substitute : does not target same people

⚠ actions to do with competitors

- direct : try to be competitive to these
- indirect : not losing too many customers
- potential : make sure they can't do it easily
- substitute : be at least better than these

⚠ five criteria for understanding competitors

- product core
 - ↳ How good is competitor's product team?
- know the size of their user base
- design quality
- brand
- speed

⚠ Monitoring competitors

- Funding (crunchbase.com)
- Acquisitions
- New features / products (mention.com)

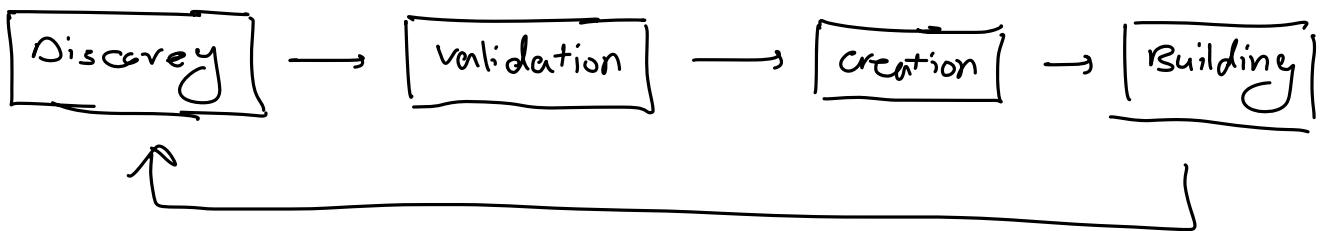
⚠ feature table



| | | competitors |
|--|--|-------------|
| | | dimensions |
| | | |

⚠ Customer development

↳ the practice of establishing a continuous and iterative communication line with your customers

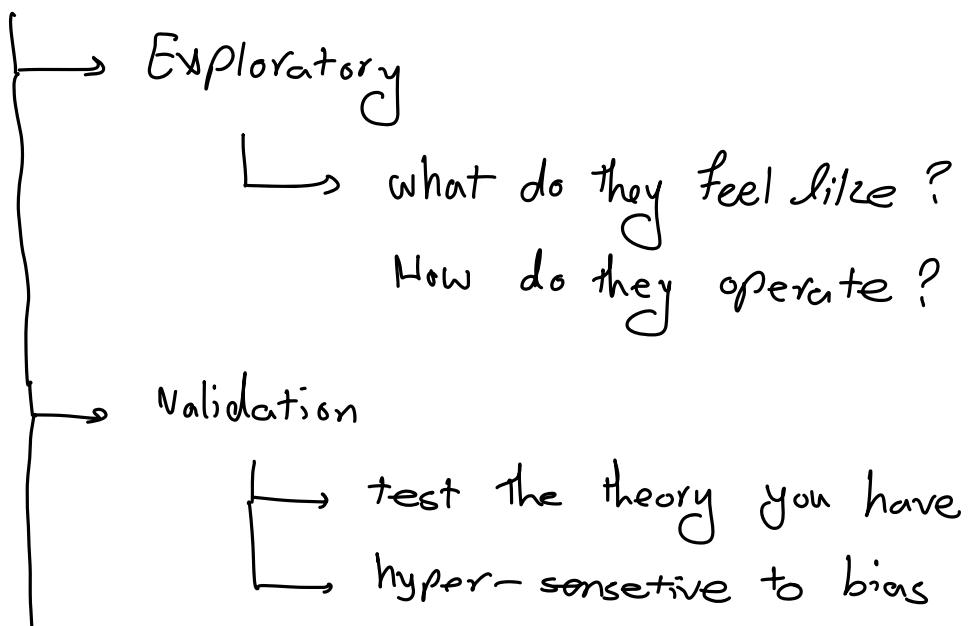


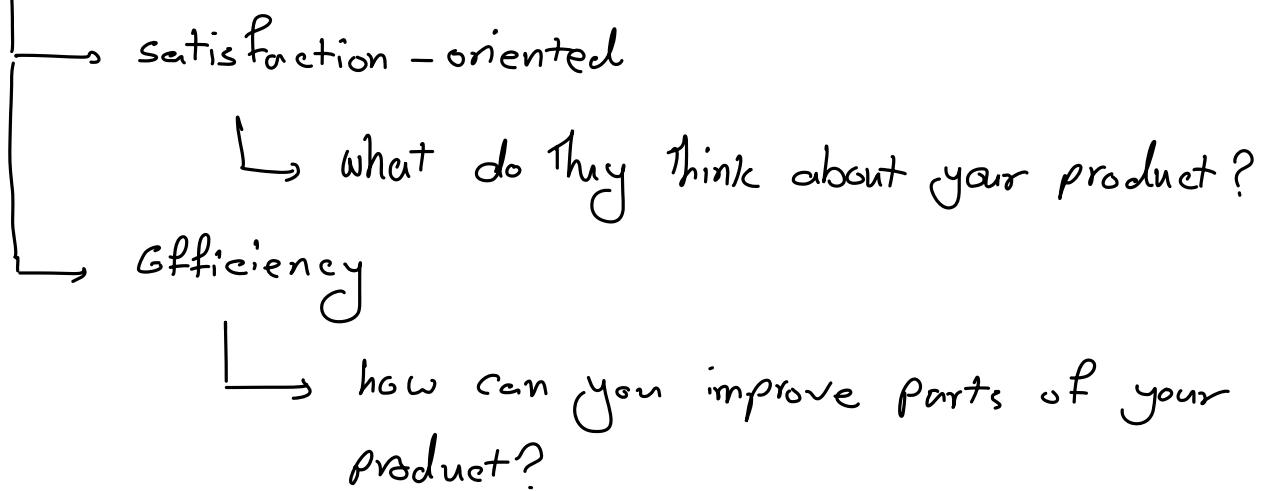
⚠ why customer development

- ↳ Risk mitigation
- ↳ Opportunity recognition

⚠ Validation phase ↗ potential customers (MVP)
development ↗ customers (v1)

⚠ types of interview





△ Pre-product vs. post-product

| pre-product | post-product |
|--|---|
| Potential customers They don't know you | existing customers they know of you |
| focus on pain points + validation finding interviewees can be difficult | focus on satisfaction usability + pain points finding interviewees is like shooting fish into barrel! |

△ pre-product

- validation
- what should go in it?
- Do people want it?
- Is there a market?

⚠ 3 question stud

| # | Group 1 | Group 2 | Group 3 | Potential Users |
|---------------------|---------------|----------|----------|-----------------|
| market size | | | | |
| size | | | | |
| Pain: Payment | | | | |
| how easy to contact | Accessibility | | | |
| | | Σ | Σ | Σ |

⚠ Post - product

- established customer base
- People paying and engaging with the product
- leads

⚠ Places to find interviewees

- LinkedIn
- Forums
- Twitter

⚠ ways to find people who will talk to you

- use live chat
- your blog
- Power users → segment by frequency, payment, ...
- twitter

- ⚠ How to get them to talk
 - be short and specific
 - be personal
 - mention how you find them
 - be valuable

⚠ tips when connecting to people

- mention that you're not from sales
- make them feel special

- ⚠ Your interview topic is ~~your product~~ customer needs

⚠ interview tips

- don't talk about your solution
- don't talk about your opinions
- create a comfortable environment
- don't force the conversation, guide it

⚠ question tips

- always ask open-ended questions
- don't ask binary questions (yes/no)
- don't ask hypothetical questions
- don't ask leading questions

↳ don't ask questions that might make them lie

⚠ when you stuck in interview

↳ I'm interesting. Tell me more!

⚠ User Personas

↳ groups of certain users that behave in similar ways

⚠ How to make user persona

- ↳ interview large # of users
- ↳ find a user behavior
- ↳ give it a real name
- ↳ give a description

⚠ Product manager's common data sources

- ↳ internal feedback
- ↳ user test data
- ↳ online feedback
- ↳ analytics stats
- ↳ news
- ↳ market trends
- ↳ watching competitors

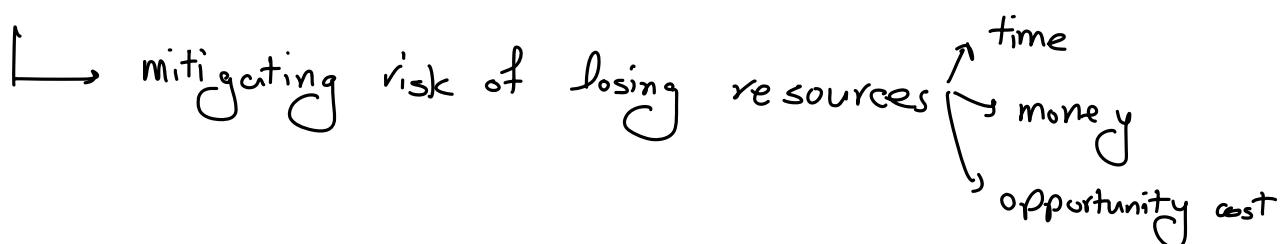
⚠ Customer interview vulnerabilities

- don't scale
- the data is qualitative

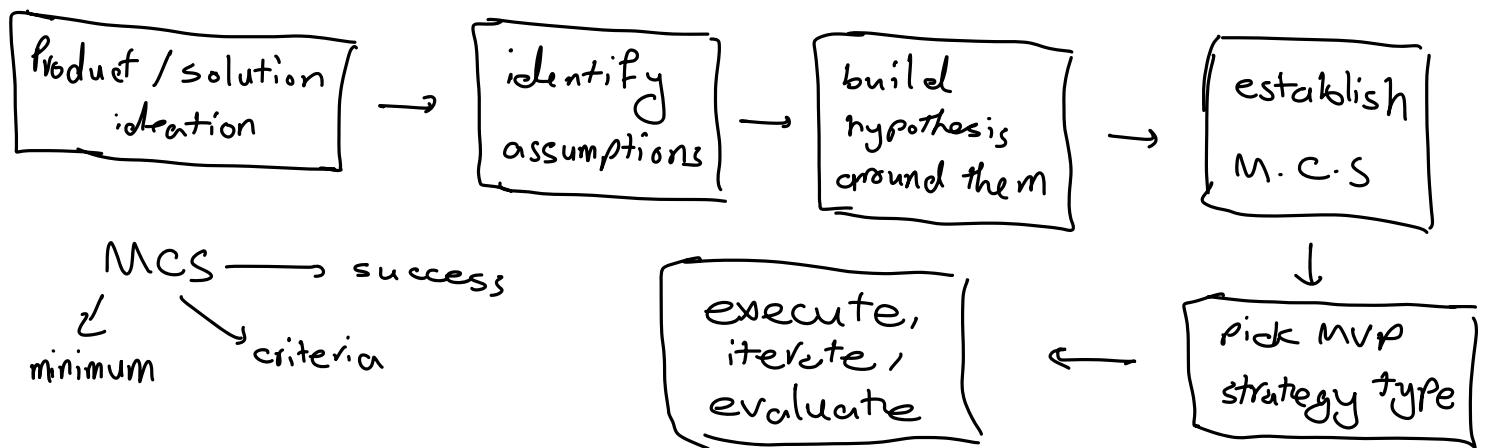
⚠ what is MVP

- a Minimum Viable Product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort
- is all about idea validation
- is not a prototype

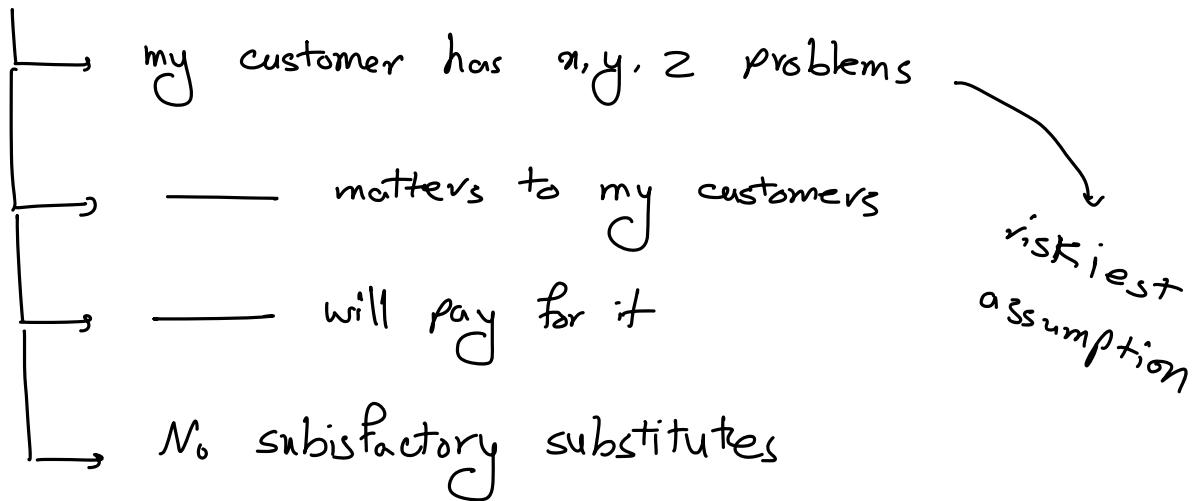
⚠ why MVP is good



⚠ steps to running an MVP experiment



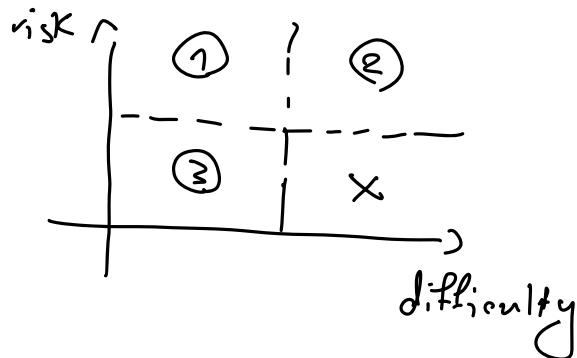
⚠ assumption formats



⚠ ideas are usually based on observations & intuitions

⚠ what to consider when choosing assumptions to test

risk
difficulty



⚠ assumption simple, readable, testable, hypothesis

⚠ hypothesis is actionable → specific target group
→ specific expected outcome

⚠ MCS gives clarity and meaning
minimum → criteria → success

⚠ hypothesis format

- simple → we believe subject / target will predicted action
because reason.
- better → if we action, we believe subject
will predicted action / outcome because reason
- The "PN" way → we believe subject has a problem
because reason. If we action,
this metric metric will impact

⚠ Outcomes of MVP test

- your hypothesis is false and not worth doing
- your hypothesis is true without question
- you're somewhere in the middle

⚠ validation metric

- demonstrates real interest from potential customers
- % of people sign up
- % of people that share your post
- average purchase price
- # of people that open your email

⚠ 2 sides : cost and reward

| Cost | Reward |
|-----------------------|------------------------|
| Developers time | Increased revenue |
| your time | time spent on page |
| other coworker's time | # of shares |
| labor wages | increased satisfaction |
| advertising costs | number of likes |
| brand effect | conversion rate |
| legacy issues | open rate |
| opportunity costs | customer LTV |
| Σ | Σ |

⚠ MVP techniques

- Email
- shadow button
- 404 / coming soon page
- fake landing / pitch experiment
- explainer
- concierge
- piecemeal
- wizard of oz

⚠ wizard of oz is most resource intensive technique
but is the most protective to your brand

⚠ MRP experiments primarily return quantitative data
remember: customer development → qualitative data

⚠ quantitative → what
qualitative → why

⚠ what is a wireframe

↳ a visual guide for a website, web app or mobile app that basically lays out a rough structure for where the content is going to go

⚠ why wireframe → the first rough translation of a product idea

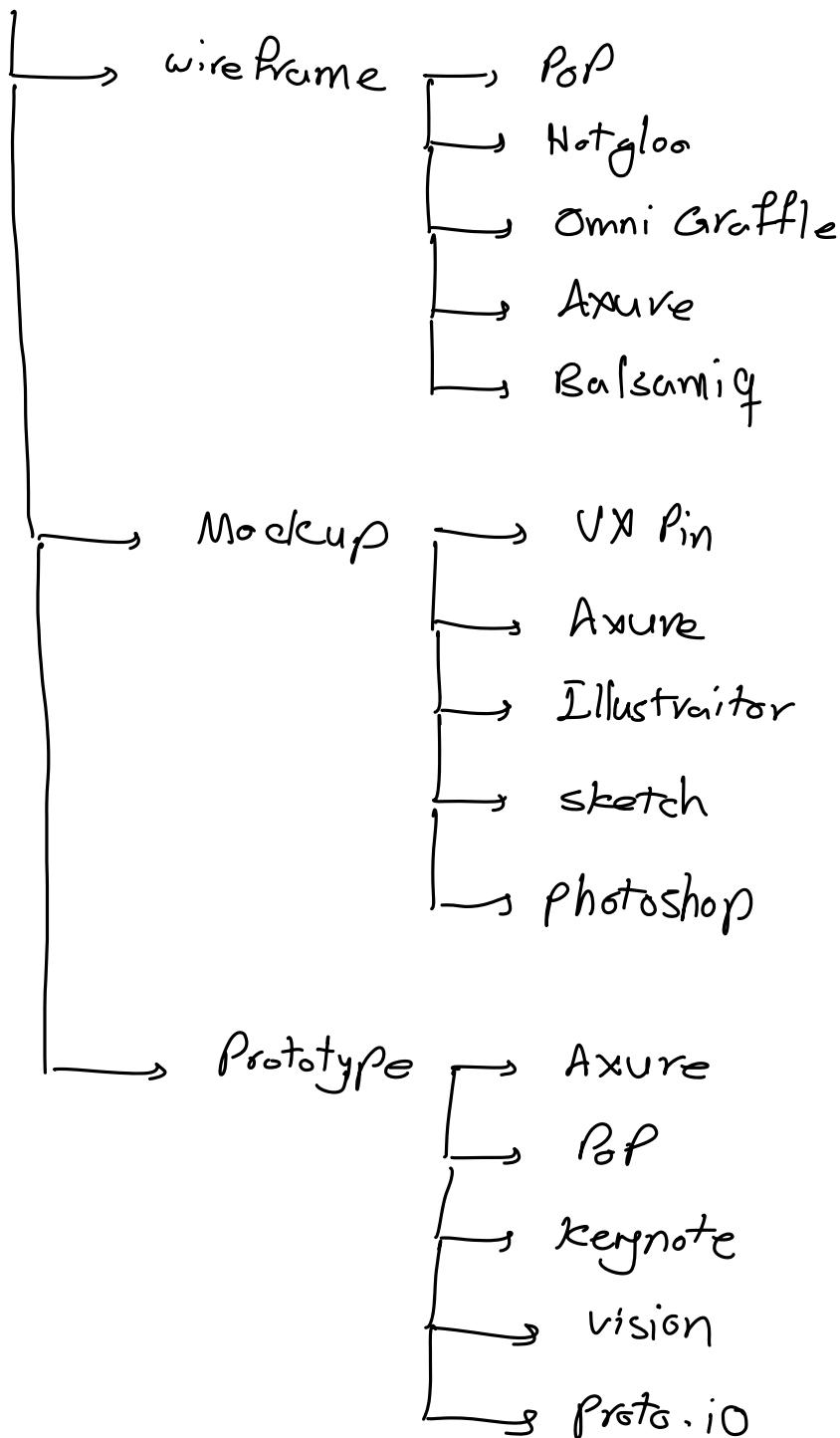
⚠ low-fidelity wireframe

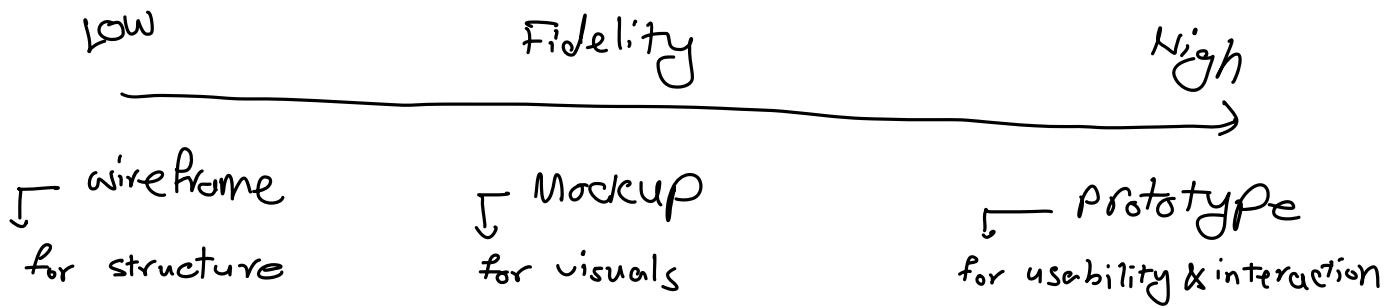
↳ not very exact / precise
↳ not a lot of detail

⚠ user's feedback adds more detail to wireframe

⚠ Product manager usually sketches out wireframes in small teams / startup

⚠ wireframing tools





 what's gets measured, gets managed

A metrics that shows what's going on the product

Metrics categories

- growth and activation (or acquisition)
 - engagement
 - retention
 - user happiness
 - revenue

⚠ Growth and acquisition metrics example

- total users per month / week
 - activated users per month / week

A retention metrics example

- ↳ retained users
↳ resurrected users

⚠ engagement metrics example

- ↳ tailored by each company
 - ↳ e.g. # of users tweeted today

⚠ user happiness metrics example

- ↳ net promoter score (NPS)
- ↳ app store rate

⚠ revenue metrics example

- ↳ LTV (lifetime value)
- ↳ CAC (customer acquisition cost)
- ↳ MRR / ARR (monthly / annual recurring revenue)

⚠ metrics types

- ↳ exploratory
 - ↳ tracks user behaviour
- ↳ reporting
 - ↳ ensure that product is doing well

⚠ what makes a metric good ?

- simple and understandable
- rate or ratio
- correlation
- changeable

⚠ HEART metrics framework

- Happiness (5)
- Engagement (3)
- Adoption (1)
- Retention (4)
- Task success (2)

⚠ HEART metrics is flexible, you can use it for anything

⚠ HEART metrics framework is used for reporting metrics

⚠ AARRR metrics framework

- Acquisition
- Activation
- Retention
- Referral
- Revenue

⚠ Metrics tracking tools

- Google Analytics
- crazy Egg
- Kiss Metrics
- Mixpanel
- Optimizely

⚠ what is a epic

- Features or functionalities we build to solve a problem
- A piece of work that takes longer than one sprint to build

⚠ what is a epic spec sheet

- allow anyone in your company to read it and understand exactly what you're building
- ① Introduction
- ② Product Requirements
- ③ Design Requirements
- ④ Engineering Requirements

Q: what is a user story

- just a way to describe a thing we're going to build that delivers some type of functionality to end user
- format : As a X, I want to do Y, so that i can Z

A: What is a acceptance criteria

- set of conditions that software must satisfy to be considered complete
- Purpose : to be very specific on how a feature should function

A: Story point → how hard is it to do this?

hours → how much does it take to do this?

A: velocity

- number of story points we were able to accomplish in a sprint period
- use average sprint velocity for rough estimation

⚠ why we use roadmaps

- executives and investors like to see quarter-based maps
- you could be against an actual deadline

⚠ How to prioritize tasks

- Assumption Testing
 - remove risky assumptions
 - Rank = value (1-10) + Important Score (1-10)
- The Buck method
 - consider business benefit
 - Final Score = business benefits (1-10) + User benefits (1-10) - cost (1-10)
- The MOSCOW method
 - Must, should, could, would

⚠ working with engineers tips

- if something goes wrong, it is your fault!
- when you're pitching, have a good idea of where the feature will go in the future
- Try to do the work yourself upfront before asking an engineer
- Watch out for technical debt
- do not treat engineers like an agency

⚠ working with designers tips

- give designers their creative freedom
- do not treat designers like an agency
- You and your designer are a team
- do not ever tell the designer what to do
- always talk about user problems first & solutions second

⚠ working with executives

- be brief
- always speak in terms of business effect
- communicate in their style

⚠ How to get a job

- have a side project
- brand yourself
 - ↳ blog, website, social media accounts

⚠ Where to look for job

- your own company
- networking
- online

⚠ Inside advice on your PM job hunt

- make sure the company has a good understanding of the product management role
- do research on the current product team or product management leader

⚠ PM resumes tips

- think about your relevant experience
 - ↳ make sure your resume reflect that experience
- make everything quantitative

⚠ PM interview tips

- Use every interview to make the next one even better
- Ask good questions
- Always ask clarifying questions, don't jump into solutions and mention the path and reason behind that

⚠ insider tips for getting the job

- do a demonstration project
 - research a current project
 - show them how would you approach that problem
- do research on
 - product, industry, technology, competition

⚠ The first things to do when landed on a job

- Schedule 1:1 meeting with all team members
 - ask about their goals and challenges
- Arrange a meeting with engineering leader
 - ask about stack and challenges
- Start talking to users
 - what is important to them

- read as many internal documents as you can
- look at the data
- Meet with the boss
 - ask about expectations and their goals
 - what are their concerns

Q: How would you describe product management

- ↳ have a deep understanding of the customer, have a deep understanding of the business, and marrying those together to identify opportunities & problems, prioritize those opportunities, and then devise the right solution for the right time