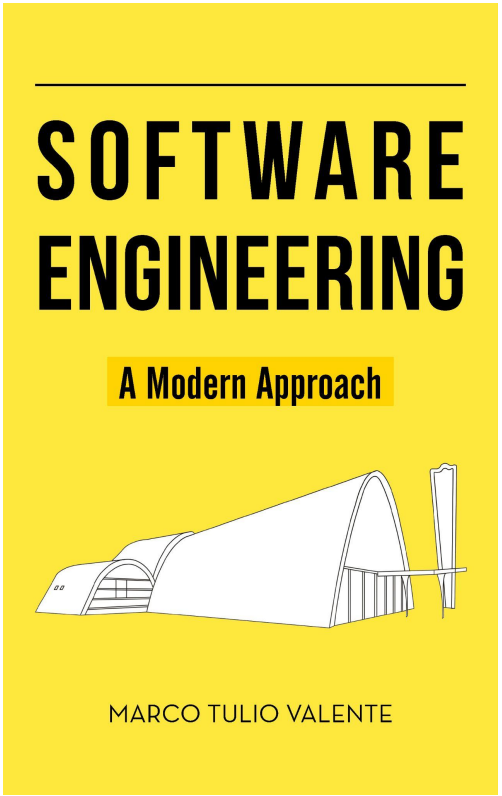


Chapter 3 - Requirements

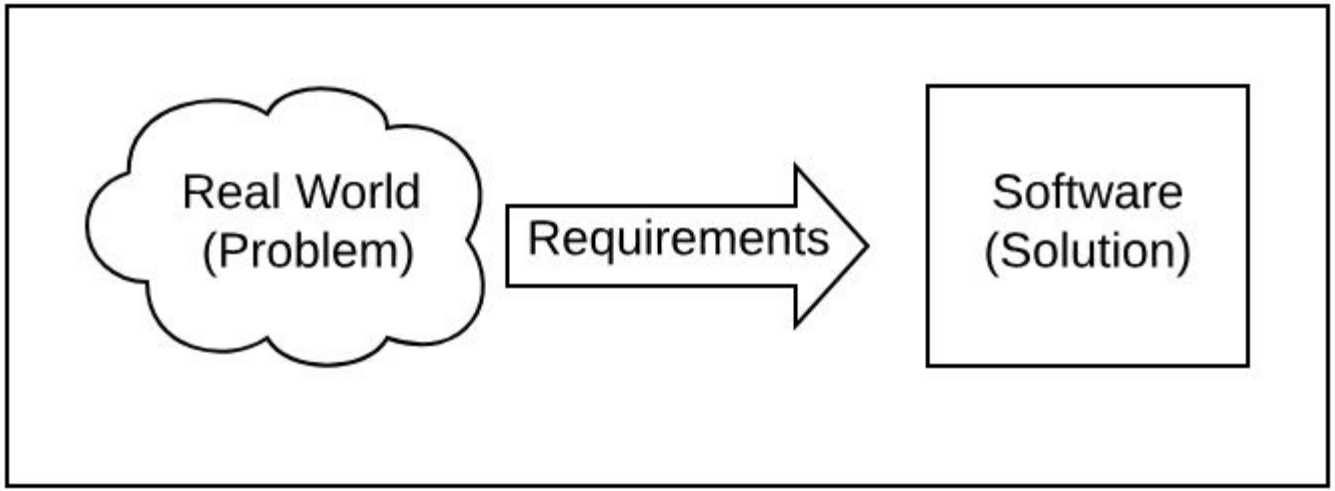
Prof. Marco Tulio Valente

https://softengbook.org

"The hardest part of building software is defining what to build" -- Fred Brooks

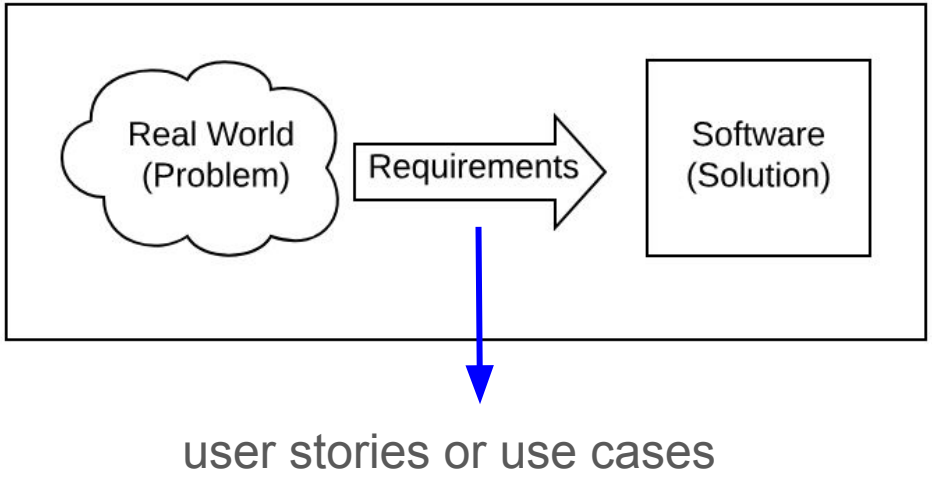
Requirements

- Functional: what a system should do (features)
- Non-functional: under what constraints
  - Performance, security, privacy, availability, etc


Real World
(Problem)
Requirements
Software
(Solution)

Topics of Study

- User Stories
- Use Cases
- MVP
- A/B Testing



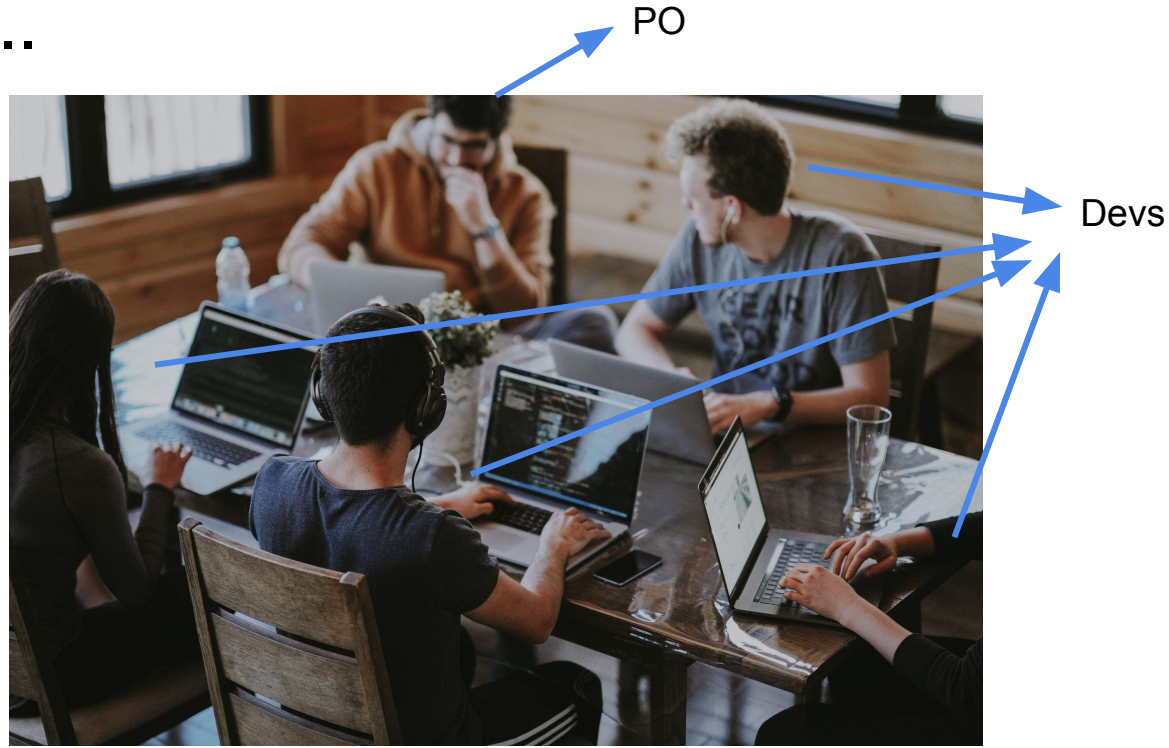
User Stories

Before ...

Analysts
(PRD: Product Requirement Document)

Programmers
(Software Factory)

Today ...



Product Owner sits with the developers and explains the requirements to them

User Stories = 3C's

- **C**ard + **C**onversations + **C**onfirmation
- Card: reminder for conversations during sprint
- Conversations: as in previous slide
- Confirmation: tests the PO will make to confirm the story implementation; aka acceptance tests (or criteria)

Example: Online Bookstore

- Card: As a user, I want to buy a book
- Conversations: PO explains payment methods; delivery methods, discounts options, etc
- Confirmation:
  - Test purchase with a discount code
  - Test payments with two credit cards: Visa and Mastercard
  - Test with two types of delivery: standard and express

Writing Workshop (or Inception)

- When: project beginning
- Participants: key users
- Goal:
  - Define what the software will do ⇒ initial list of stories
  - Define what the software will not do

Setup Sprint

- Before starting, some teams also conduct a "setup sprint"
- Goal: set up the environment and development tools

Format of User Stories

- As a [certain type of user],

  I want to [do something with the system]

Example: Library Management System

Types of Users

- Students
- Instructors
- Staff members

Students' Stories

- As a student, I want to borrow books.
- As a student, I want to return books I have borrowed.
- As a student, I want to renew my book loans.
- As a student, I want to search for books.
- As a student, I want to reserve books that are currently on loan.
- As a student, I want to receive notifications about new acquisitions.

Instructors' Stories

- As an instructor, I want to borrow books for an extended period of time.
- As an instructor, I want to recommend books for acquisition.
- As an instructor, I want to donate books to the library.
- As an instructor, I want to return books to other campus libraries.

Staff Stories

- As a staff member, I want to register new users.
- As a staff member, I want to add new books to the system.
- As a staff member, I want to remove damaged books from the system.
- As a staff member, I want to access statistics about the collection.
- As a staff member, I want the system to send reminders to users with overdue books.
- As a staff member, I want the system to apply fines for late book returns.

Characteristics of Good User Stories (**INVEST**)

- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimable
- **S**hort
- **T**estable

User Stories ⇔ Functional Requirements

But how to define Non-Functional Requirements?

Non-functional requirements (NFR)

- Team should:
  - Define NFRs with the Product Owner
  - Include NFRs in the Done Criteria

Example

- Suppose performance is an important NFR
- The team can define that stories to be "done" should:
  - Pass a code review focused on performance
  - Pass performance tests with real load

Exercises about User Stories

1. Write three user stories for a banking app.
2. Suppose a learning management system, similar to Moodle.
    (a) Propose an epic story for this system
    (b) Break this story into smaller stories.

3. Suppose an online forms app, similar to Google Forms.

Suppose the following user story of this system:

> “As a form creator, I would like to close my forms to receive new answers”

Write an **acceptance test** for this story.

Possible answer:

- Story: As a form creator, I would like to close my forms to receive new answers
- Acceptance test:
  - Create a form (or select an existing form)
  - Submit two responses (as a regular user)
  - Close the form to receive responses
  - Try to submit a new response and check that it will not be possible

4. Suppose an e-commerce system. Suppose the following user story of this system:

> “As a customer, I would like to list the status of my purchases”

Write an **acceptance test** for this story.

Possible answer:

- Story: As a customer, I would like to list the status of my purchases
- Acceptance test:
  - List purchases as in “track_status.xls” (this is an example of user's purchases, with all the possible data and statuses)
  - When clicking on a purchase, its data should be displayed, as shown in the “purchase data” tab in the previous spreadsheet
  - It should be possible to sort the list by date, value, and status
  - We do not need support for searching and filtering purchases

5. Suppose you are the technical leader of a team. Thus, you are responsible for defining the **done criteria** of this team. List at least three checks that you would recommend.

Possible answer:

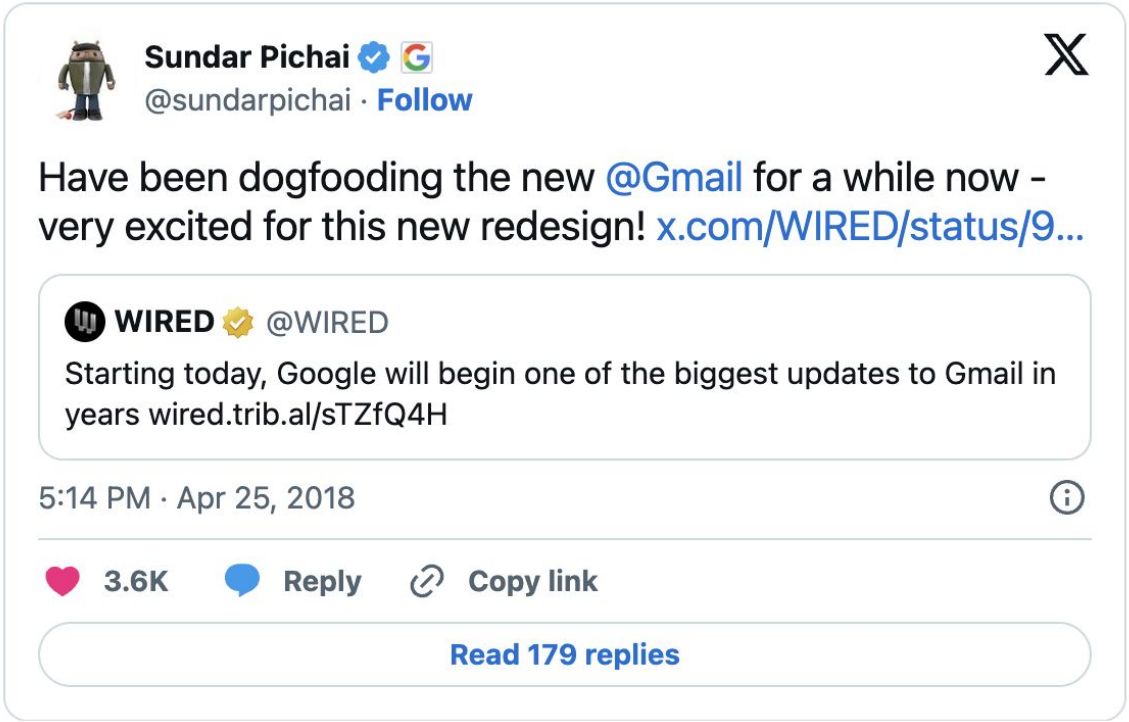
- Done criteria:
  a. Test coverage ≥ 80%
  b. Linter tool executed with no warnings
  c. UX designer has approved changes in the frontend (if any)
  d. Code has been reviewed by at least one other team member
  e. Code has been merged into the “development” branch

Acceptance Tests	Done Criteria
● Create a form ● Submit two responses (as a regular user) ● Close the form to receive responses ● Submit a new response and check that it will not be possible	● Test coverage ≥ 80% ● Linter tool executed with no warnings ● UX designer has approved changes in the frontend (if any) ● Code has been reviewed by at least one other team member ● Code has been merged into the “development” branch

6. In Software Engineering, anti-patterns are solutions that are not recommended for a certain problem.

Describe five anti-patterns for user stories, i.e., stories that are not recommended or that do not have desirable properties.

Dogfooding ("eat your own dog food")



Another Example: Uber

Why Uber's CPO delivers food on weekends | Sachin Kansal

Behind Uber's extreme dogfooding and...

JUN 1

link

Either once or twice a month I will set aside half a day and then I'll go out and I'll drive and deliver.

We design these amazing features. They look so good, and then you get in the car and you have a phone which is sitting three feet away from you.

You're driving at 45 miles per hour. The world just changes. This thing that was looking so great in an office setting, now maybe makes no sense.

Dogfooding the restaurant app

So we have a cafe here at Uber on campus. So we enable the barista to be an Uber Eats restaurant, and they have an Uber Eats tablet and I can actually order from that barista.

So now you're dogfooding being an eater, you're dogfooding being a courier and you can dogfood being a merchant.

So some of my product team, they actually will go to the barista, they'll stand behind the counter and they'll experience what it feels like for these orders to come in.

Use Cases

Use Cases

- Detailed requirement specifications
- Usage is not common with agile methods
- Describe an actor performing an operation with the system
- Two flows: normal flow and extensions
- Extensions:
  - Exceptions (or errors)
  - Additional details for specific steps

Transfer Values between Accounts

Actor: Bank Customer

Main Flow:

1 - Authenticate Customer

2 - Customer enters destination account and branch

3 - Customer enters the amount for transfer

4 - Customer specifies the transfer date

5 - System executes the transfer

6 - System asks if the customer wants to make another transfer

Extensions:

2a - If the account or branch is incorrect, system requests new account and branch

3a - If transfer amount exceeds current balance, system requests new amount

4a - Transfer date must be the current date or within one year in the future

5a - If the transfer date is the current date, system processes transfer immediately

5b - If the transfer date is in the future, system schedules the transfer

Important

- Use cases are not algorithms
- We are still defining the requirements:
  - Understanding and delimiting the problem
  - Instead of proposing solutions (algorithms)

Example: Process Sale at POS (Point-of-Sale)

Source: Craig Larman. Applying UML and Patterns. Pearson, 2004

Main Flow

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.
   Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

Extensions: [we will show only two steps]

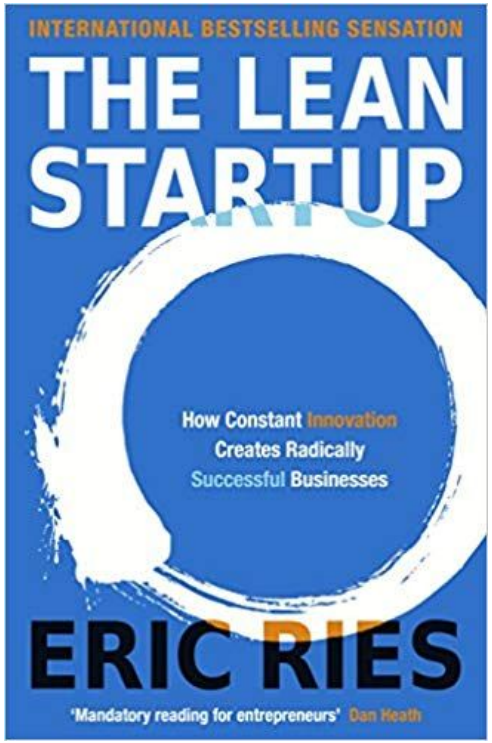
7a. Paying by cash:

1. Cashier enters the cash amount tendered.
2. System presents the balance due, and releases the cash drawer.
3. Cashier deposits cash tendered and returns balance in cash to Customer.
4. System records the cash payment.

7b. Paying by credit:

1. Customer enters their credit account information.
2. System sends payment authorization request to an external Payment Authorization Service System, and requests payment approval.
    - 2a. System detects failure to collaborate with external system:
        1. System signals error to Cashier.
        2. Cashier asks Customer for alternate payment.
3. System receives payment approval and signals approval to Cashier.
4. System records the credit payment, which includes the payment approval.
5. System presents credit payment signature input mechanism.
6. Cashier asks Customer for a credit payment signature. Customer enters signature.

Minimum Viable Product


INTERNATIONAL BESTSELLING SENSATION
THE LEAN STARTUP
How Constant Innovation Creates Radically Successful Businesses
ERIC RIES
'Mandatory reading for entrepreneurs' Dan Heath

2011

Two types of systems

1. Low risk and well-known users
2. High risk and uncertain outcome

Low risk and known users

- Example: library management system
  - Well-known system, essential in every library
  - Viability is clear
  - User stories work well

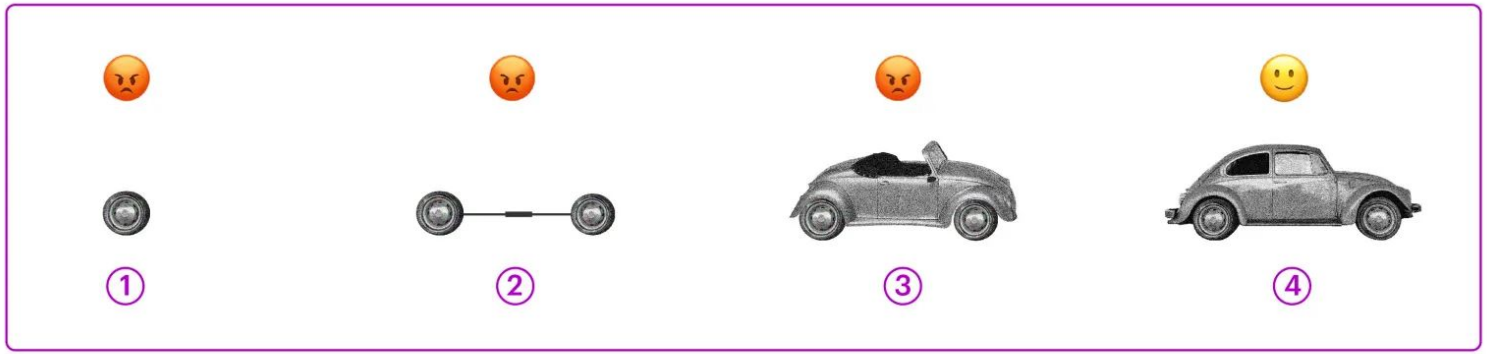
High risk and uncertain market

- Example: e-book store with payments via Bitcoin
  - Typical systems of startups, but not exclusive to them
  - As risk is high, the idea must be validated with real users

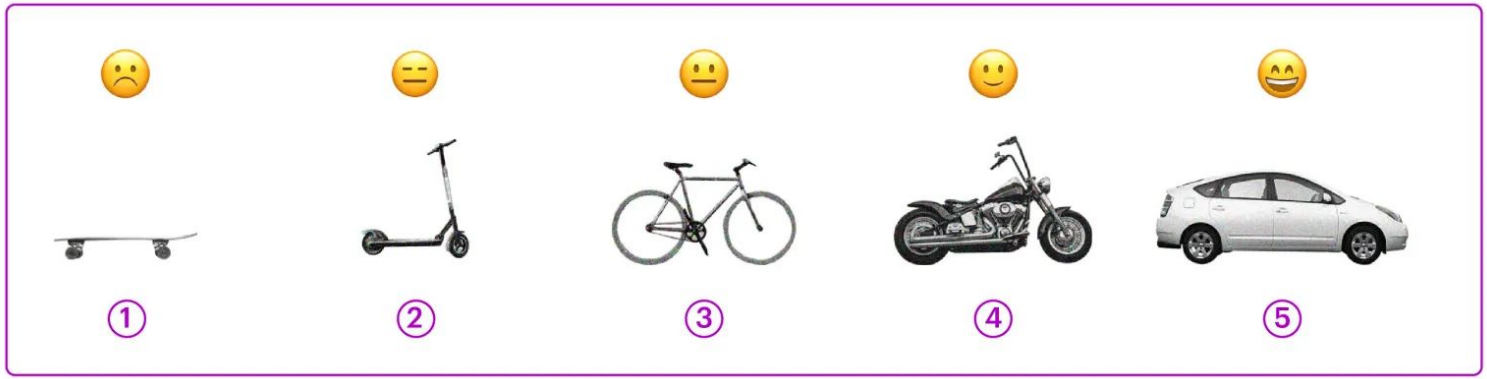
MVP

- Product = can be used
- Minimum = small number of features (lowest cost)
- Viable = will it have a market?

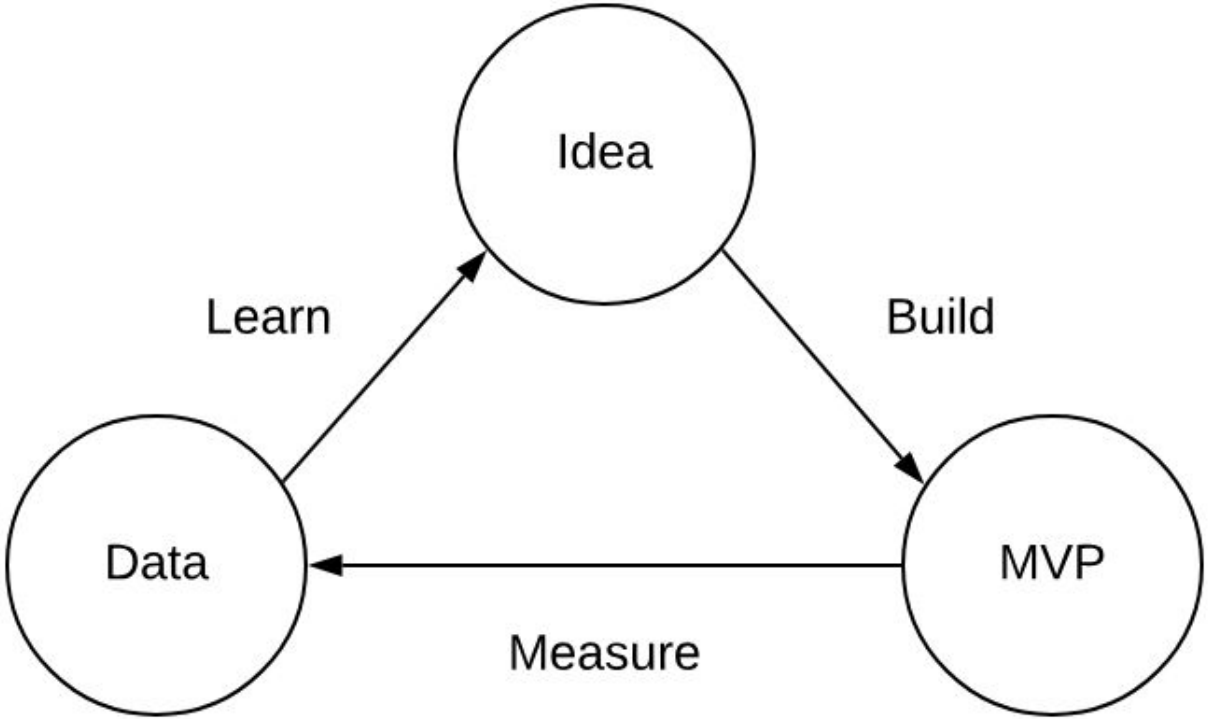
Using MVPs



Without MVPs

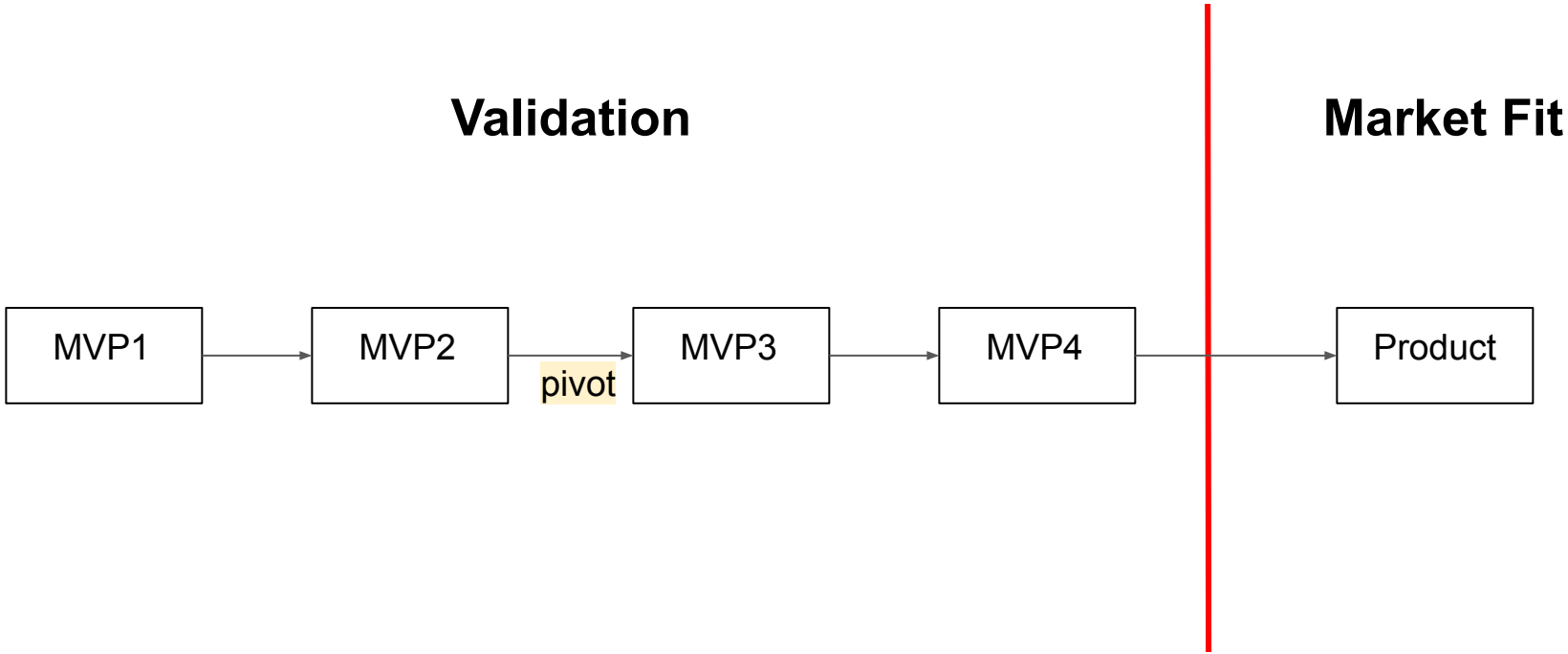


Source: https://blog.nubank.com.br/mvp-o-que-e-produto-minimo-viavel-e-como-fazemos-no-nubank/


Idea
Learn
Build
Data
MVP
Measure

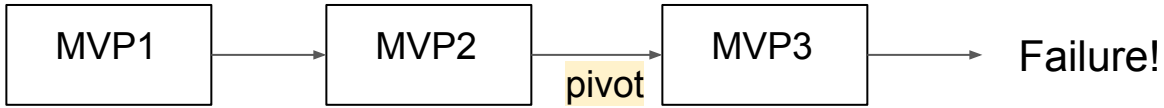
At the end of a cycle, we can:

1. Make minor changes and run the cycle again
2. Pivot: make major changes and run the cycle again
3. Give up (out of money!)
4. Success: we achieved Product-Market Fit (PMF) and will now build a robust product


Validation
Market Fit
MVP1
MVP2
pivot
MVP3
MVP4
Product

Validation

Market Fit



MVP vs Interviews

- “If I had asked my customers what they wanted, they would have said a faster horse” (Henry Ford)

MVP Examples

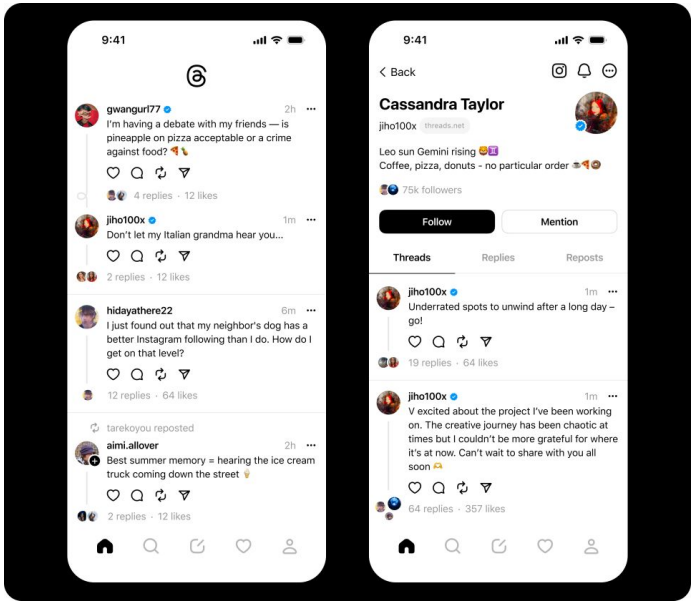
Zappos

- Online shoe store, later acquired by Amazon
- Hypothesis: Would people buy shoes online (in 1999)?
- MVP: Simple Web page, with photos of shoes from local stores
- Manual backend
- Goal: Just validate business hypothesis


Zappos.com
the world's largest shoe store!
free shipping & no sales tax
customer service
shopping bag
account & register
women's
men's
dress
casual
athletic
dress
casual
athletic
kids
Pick a category to shop from:
Category
Register now & Save Money
registered customers
username
password
login
Measure Your Foot
The world's largest shoe store!
WHAT WE'RE HEARING!
Welcome to Zappos.com - the shoe store! We have a selection of over 100 brands to shop from! We offer FREE SHIPPING (U.S. orders only) and NO SALES TAX.
Live Customer Service!
Mon. - Fri. 10am-6pm PST
Sat. 10am-5pm PST
Click here!
Free Shoes!
Congratulations to the November 24 winner,
Adriana Rodriguez of Oakland, CA
We will be giving away a FREE PAIR OF SHOES (up to $150 in store credit) every Wednesday until the year 2000!
To enter become a registered user or send an email to freeshoes@zappos.com with your name and email address. A winner will be notified by email every Wednesday!
featured brands.
BOSTONIAN
Sudini
rieker
L.B.Evans
Men's Comfort
Zappos Special:
Free Shipping &
No Sales Tax
home
women's dress
women's casual
women's athletic
men's dress
men's casual
men's athletic
kids

MVP “Simple App”

- Example: Meta's Threads (Twitter competitor)



- No Web version
- No hashtags
- No trending topics
- No DMs
- No chronological timeline
- No post editing

July 2023

MVP ≠ 1st version of a product

MVP is an experiment (so, it may fail)

- Assuming:
  - There is a market
  - The client will pay you
  - You have the skills to implement
  - The clients know what they want
- Thus, there is no risk, and we don't need an MVP
- If we know the software is viable, it is not an experiment

MVP & Software Engineering

- MVP does not need to use the best SE practices
  - Unit tests, refactoring, complex architectures, etc
- If the idea is validated, the system can be re-implemented
- However, certain requirements are important
  - Specifically, NFRs such as performance, usability, availability, etc

How long does it take to build the 1st MVP?

- It depends and varies a lot, but it must be rapid
- For example, two weeks…
- “If you're not embarrassed by the first version of your product, you've launched too late” (Reid Hoffman)

Common Types of Pivots

- Zoom-in (a specific feature becomes a product)
- Customer Segment
- Application to Platform
- Technology

Zoom-in: Flickr

- Started as an online massively multiplayer role-playing game
- Feature for sharing photos became very successful and turned into Flickr

Zoom-in: Slack

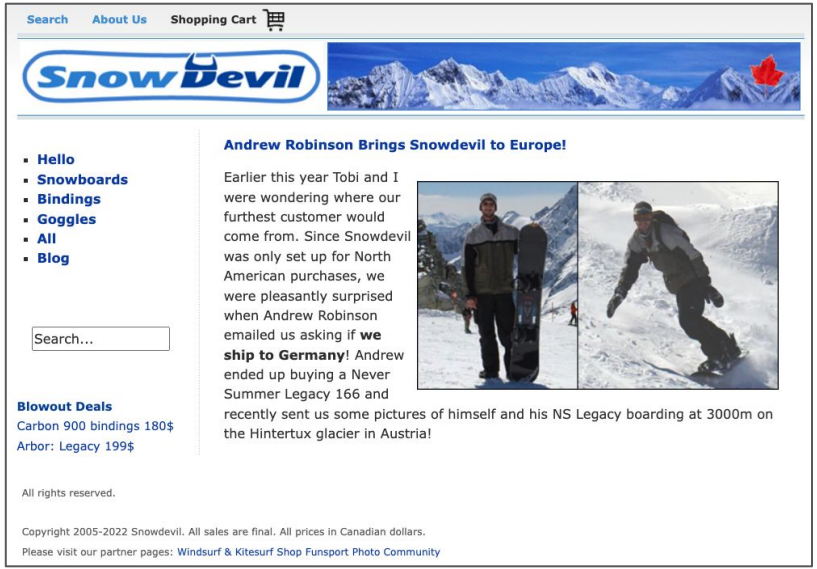
- Similar case to Flickr
- Slack is a messaging app used by companies
- It also originated from an online RPG

Customer Segment: Twitch

- Live streaming app
- Initial audience: any user (justin.tv)
- Later: gamers (twitch.tv)

Application to Platform: Shopify

- From an online store for renting skiing equipment
- To a platform for hosting online stores



Technology: Android

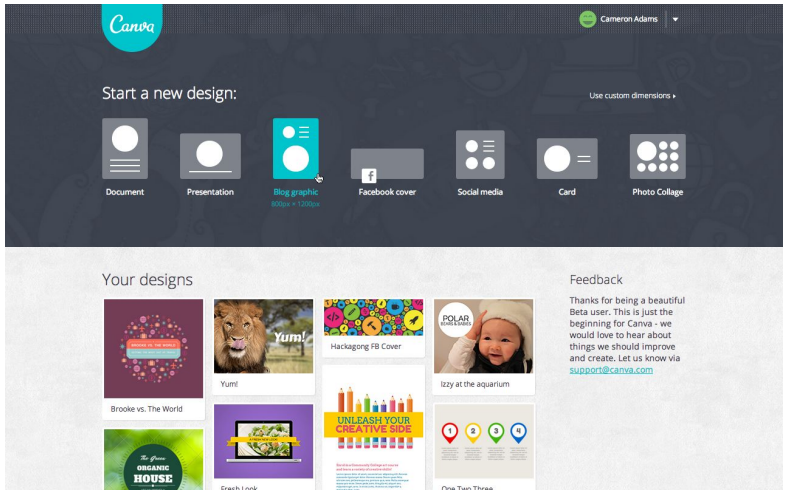
- From: Operating system for cameras
- To: Operating system for smartphones

Exercises about MVPs

1. What is the difference between an MVP and a market survey?
2. Suppose you had the idea of creating a system to organize car rides for the students in your university.
    a. How would you implement an MVP for this system without writing code?
    b. Suppose this MVP failed, describe a pivot you can try.
3. Describe a domain where it is more challenging to create an MVP.

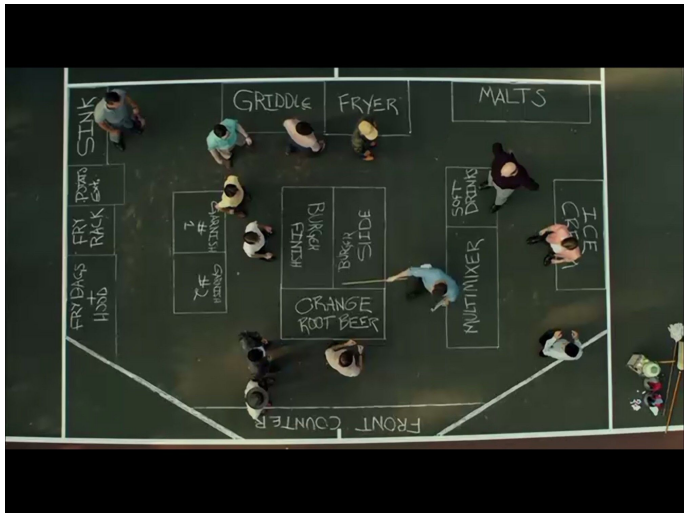
Example: Canva

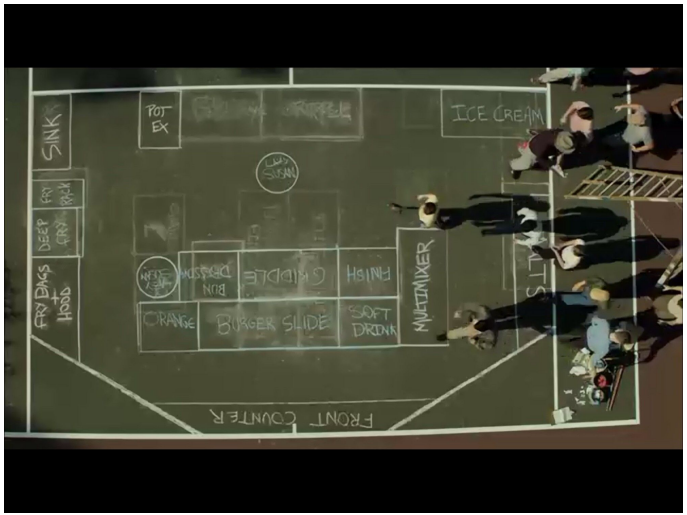
- First version took a year to launch; thus, it should not be called an MVP



https://techcrunch.com/2013/08/26/backed-by-3-million-in-funding-canva-launches-a-graphic-design-platform-anyone-can-use/

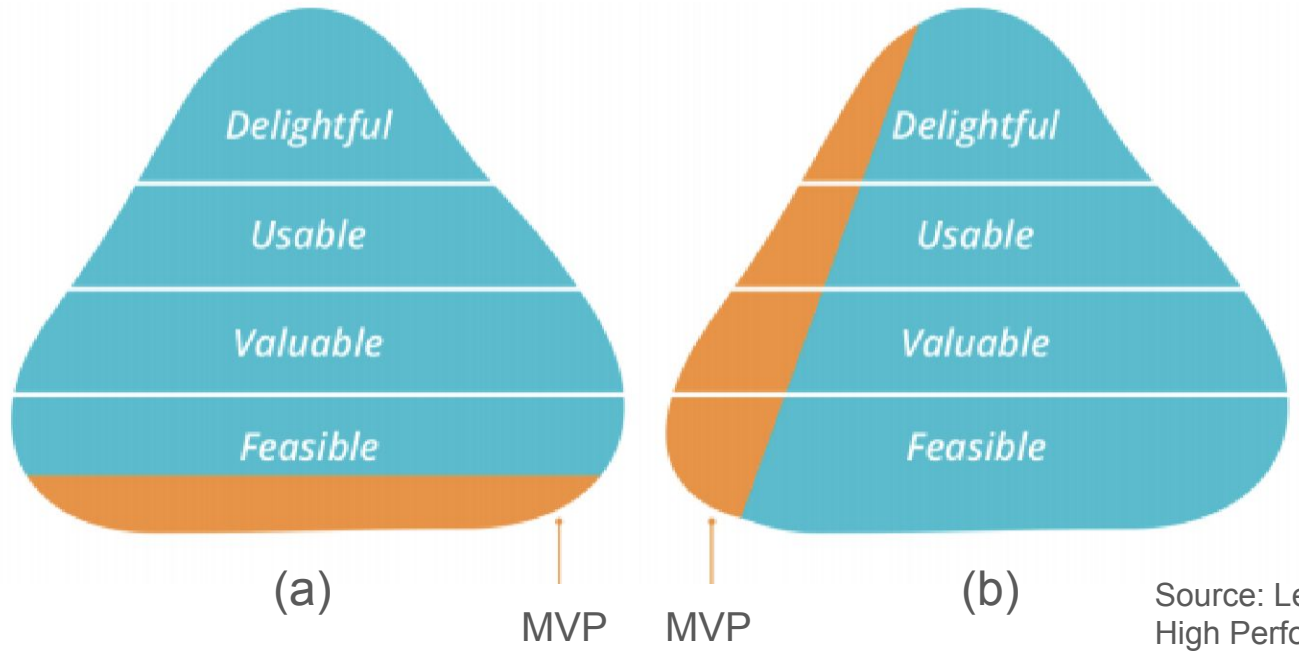
4. In the 1940s, prior to founding McDonald's, the company's founders conducted several experiments using kitchen plans drawn with chalk on a tennis court. The objective was to test various kitchen layouts before selecting one. Do these drawings qualify as MVPs? Justify.





Source: The Founder movie

5. Which of the two figures below better represents a best practice for implementing MVPs? In the figures, the orange part describes the concerns that were considered in the MVP design.



Source: Lean Enterprise. How High Performance Organizations Innovate at Scale

6. **Discussion question** (the answer is not necessarily objective)**:**

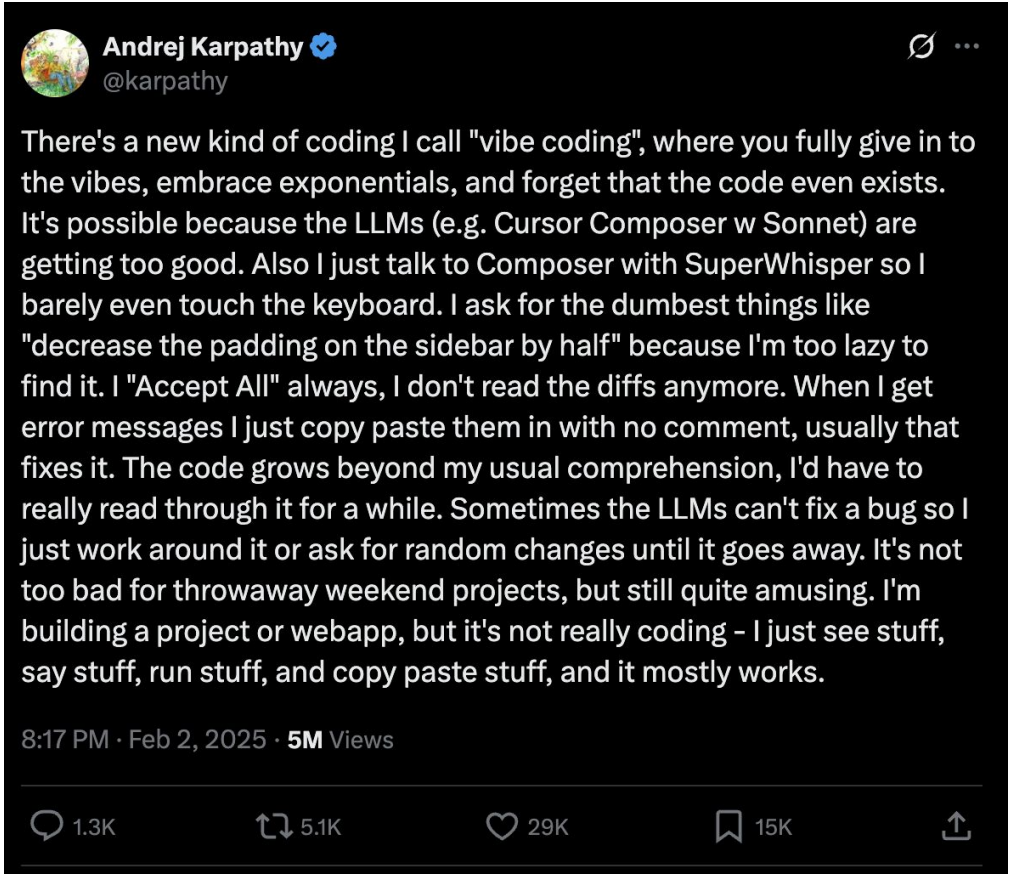
Nvidia emerged in the 1990s as a manufacturer of graphics processing units (GPUs), widely used, for example, in gaming. However, around 2015, the company began prioritizing the manufacturing of processing units for general parallel applications.

Can this shift (from graphics processing to parallel processing) be considered a pivot? Justify your answer.

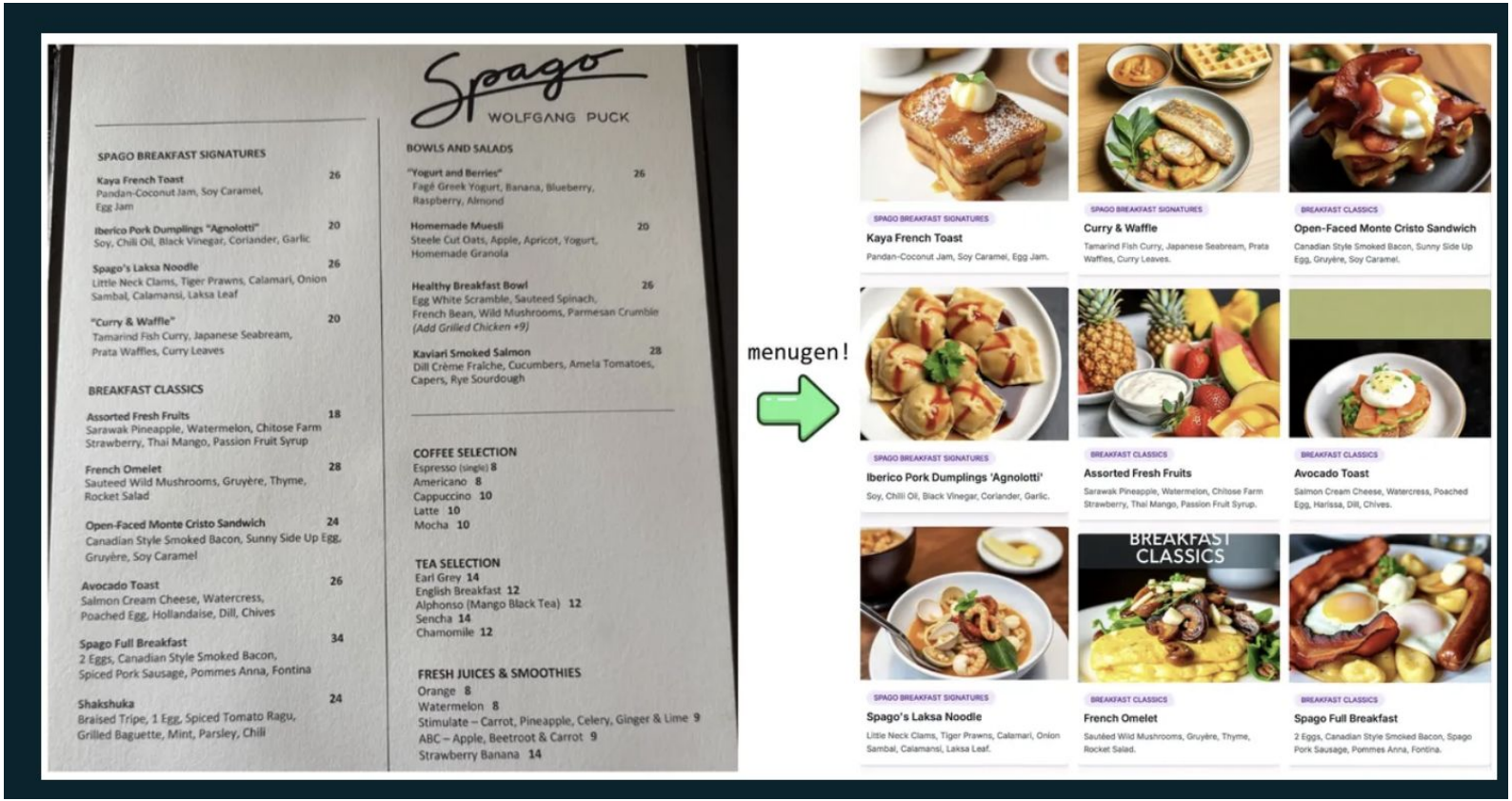
Vibe Coding

Vibe Coding

- Programming using LLM prompts, without writing a single line of code



https://x.com/karpathy/status/1886192184808149383



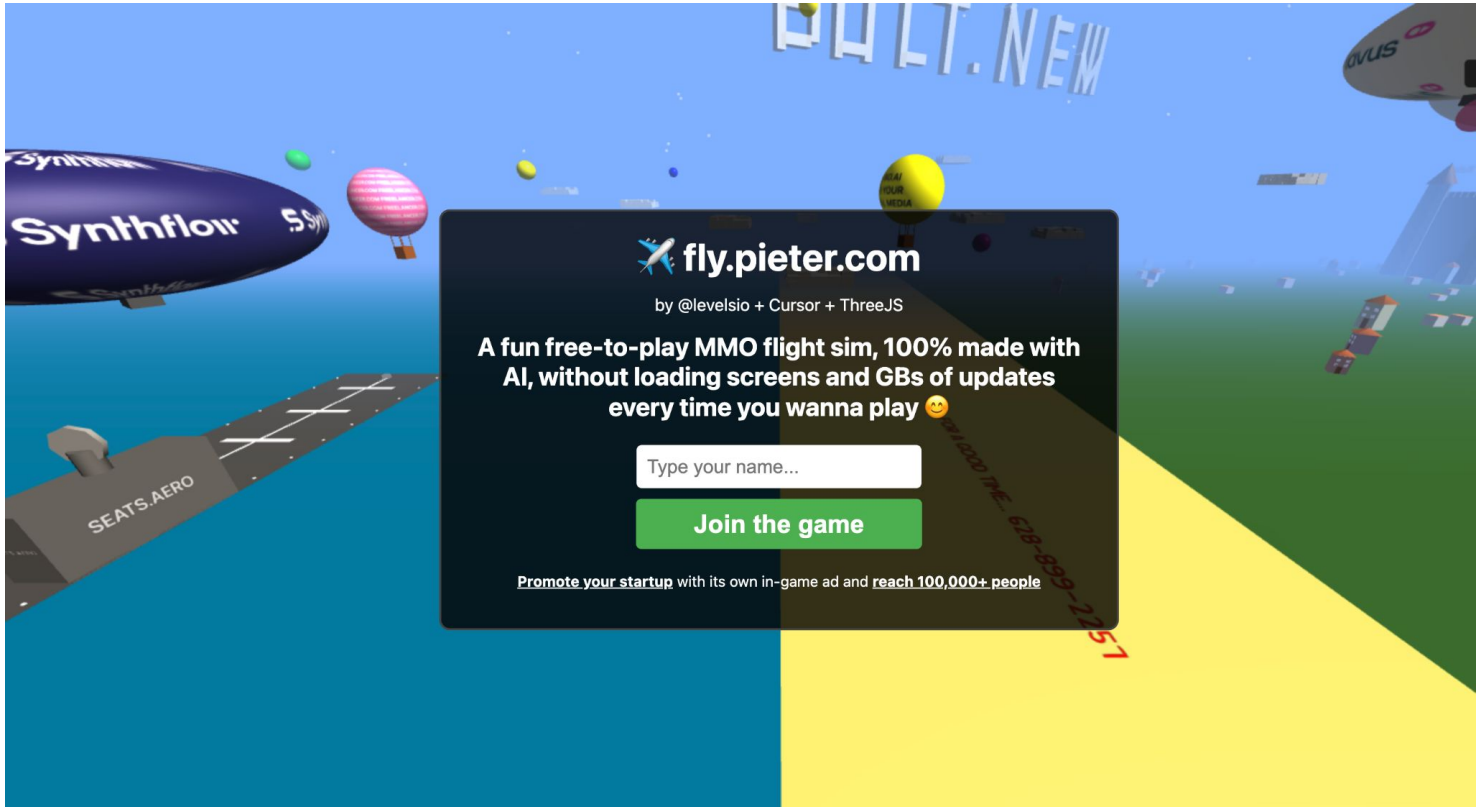
https://karpathy.bearblog.dev/vibe-coding-menugen/

"The fascinating thing about menu gem for me is that the code, well, the vibe coding part, the code was actually an easy part of vibe coding menu gem.

And, most of it actually was when I tried to make it real so that you can actually have authentication, payments, domain name, and personal deployment.

This was really hard, and all of this was not code."

https://www.youtube.com/watch?v=LCEmiRjPEtQ



https://fly.pieter.com

Vibe coding can be useful for building MVPs and/or prototypes

A/B Testing

A/B Testing

- Used when two implementations compete against each other
- Example: recommendation system in an online store
- Customers who buy P also buy X,Y,Z
- Two versions:
  - Version A: Original version
  - Version B: New version, proposed by some devs
- Is it worth switching to the new version?
- A/B Testing: lets the users decide

Control and Treatment Versions

- Version A: control
- Version B: treatment

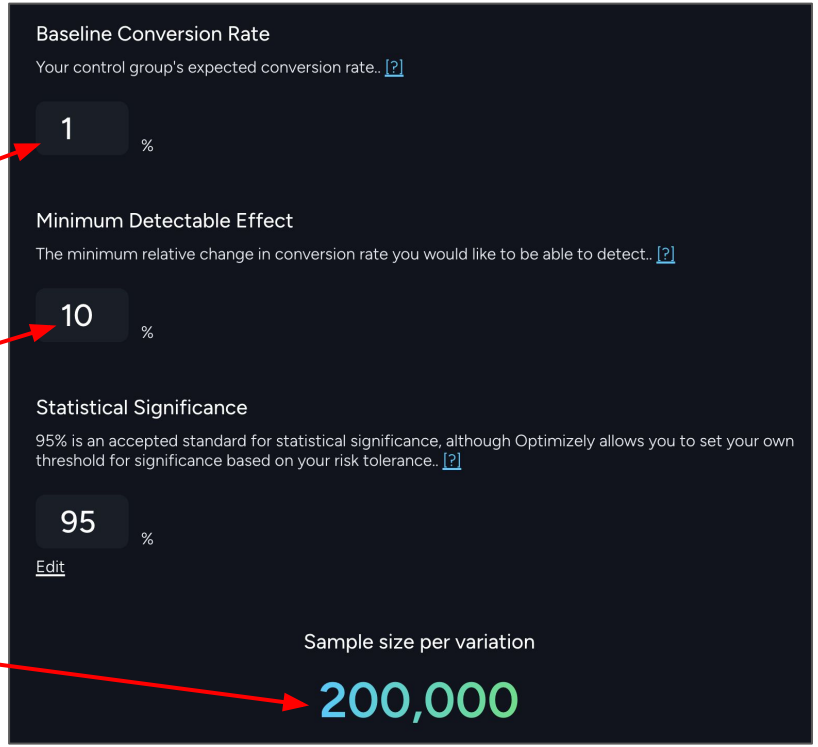
```
version = Math.Random(); // random number between 0 and 1
if (version < 0.5)
   "execute the control version"
else
   "execute the treatment version"
```

At the end of the experiment

- The data is analyzed, using some metric
- Example: conversion rate from visits to purchases
- Does version B have statistically significant gains?
  - Yes, let's switch to it
  - No, let's stick with the original implementation (version A)

Sample Size Calculators

- Input:
  - Current conversion rate (1%)
  - Intended gain (10%)
- Sample size:
  - 200K clients per version



https://www.optimizely.com/sample-size-calculator/#/?conversion=1&effect=10&significance=95

Final Comments

- A/B Testing requires large samples
- Used by major internet companies

Exercises

1. Consider a first A/B test as follows:

- Conversion rate of the control group: 1%
- Intended increase in this conversion rate: 10%

Now, consider a second A/B test as follows:

- Conversion rate of the control group: 1%
- Intended increase in this conversion rate: 30%

Which of these tests requires a larger sample size? Justify. If you wish, perform the test using this sample size calculator.

End