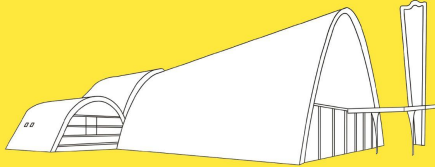


# SOFTWARE ENGINEERING

A Modern Approach



MARCO TULLIO VALENTE

## Chapter 3 - Requirements

Prof. Marco Tulio Valente

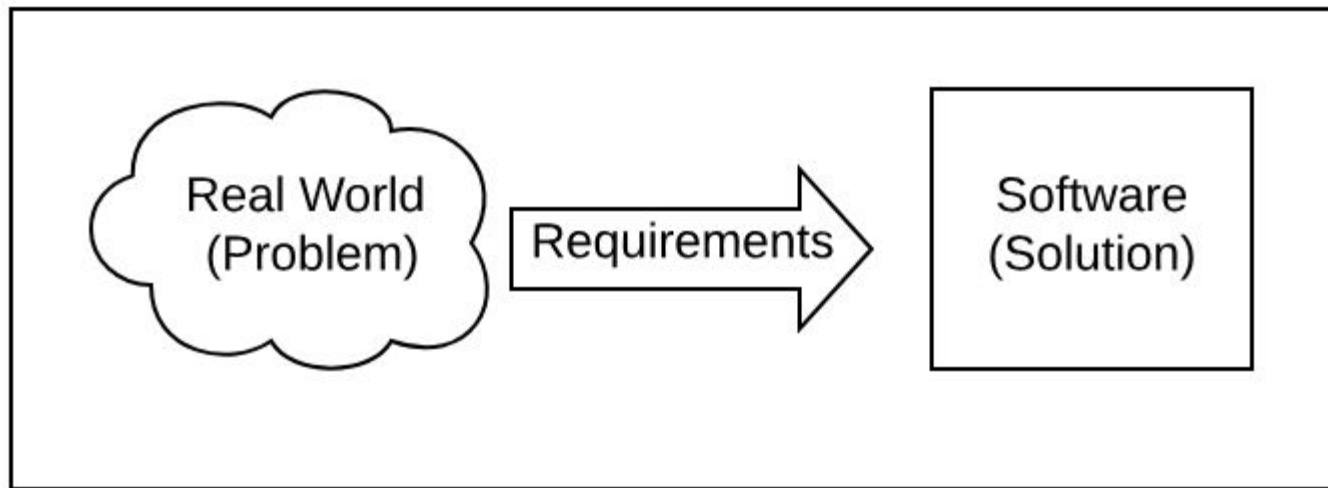
<https://softengbook.org>

CC-BY: This license enables anyone to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the author.

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

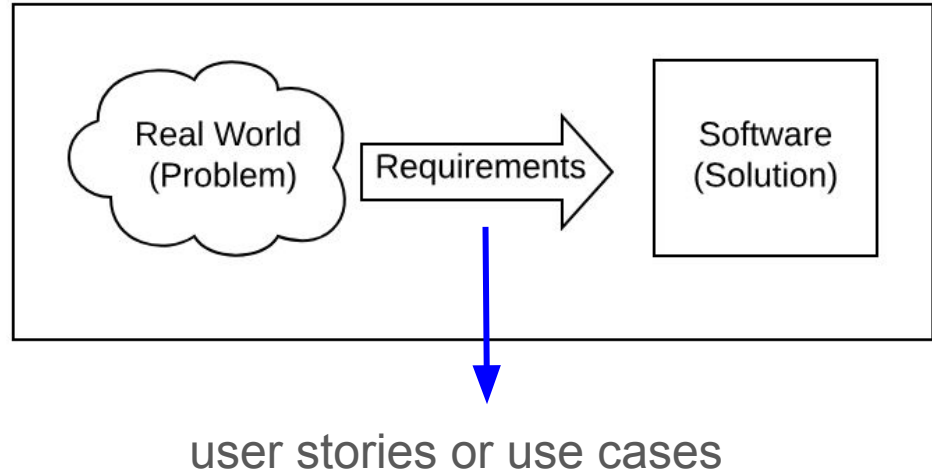
# Requirements

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



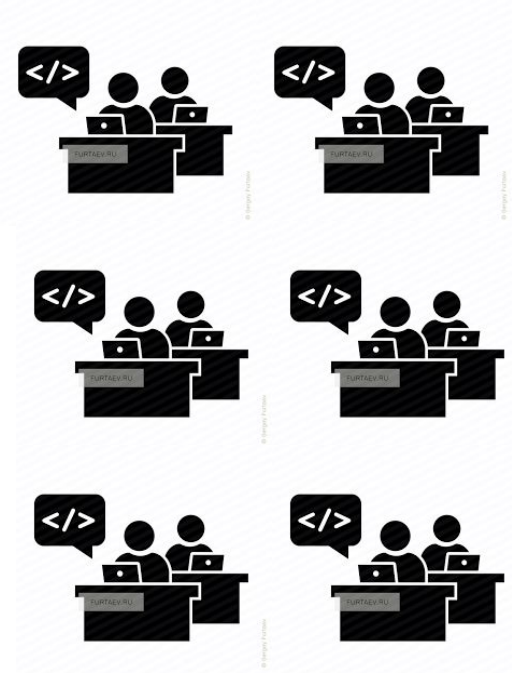
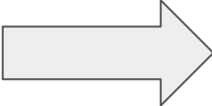
# Topics of Study

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



# User Stories

# Before ...



**Analysts**  
(PRD: Program Requirement Document)

**Programmers**  
(Software Factory)

Today ...



PO

Devs

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 would like to [do something with the system]

# Example: Library Management System

# Types of Users

- Students
- Teachers
- Staff members

# Students Stories

- As a student, I want to borrow books
- As a student, I want to return a book I 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 borrowed books
- As a student, I want to receive emails about new acquisitions



# Teachers Stories

- As a teacher, I want to borrow books for an extended period of time
- As a teacher, I want to recommend books for acquisition
- As a teacher, I want to donate books to the library
- As a teacher, I want to return books in other 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 collection
- As a staff member, I want to discard damaged books
- As a staff member, I want to access statistics about the library
- As a staff member, I want the system to send emails to users with overdue books
- As a staff member, I want the system to apply fines in the case of late book returns

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

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

User Stories  $\Leftrightarrow$  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 on a code review focused on performance
  - Pass on 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 the all the possible data and status)
  - When clicking on a purchase, its data should be displayed, see example 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  $\geq 80\%$
  - b. Linter tool executed with no warnings
  - c. UX designer has approved changes in the frontend (if any)
  - d. Code was reviewed by at least another team member
  - e. Code was merged into the “development” branch

Acceptance Tests	Done Criteria
<ul style="list-style-type: none"><li>● Create a form</li><li>● Submit two responses (as a regular user)</li><li>● Close the form to receive responses</li><li>● Submit a new response and check that it will not be possible</li></ul>	<ul style="list-style-type: none"><li>● Test coverage <math>\geq 80\%</math></li><li>● Linter tool executed with no warnings</li><li>● UX designer has approved changes in the frontend (if any)</li><li>● Code was reviewed by at least one other team member</li><li>● Code was merged into the “development” branch</li></ul>

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.



# 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)
  - Details about the steps

## **Transfer Values between Accounts**

**Actor:** Bank Customer

### **Main Flow:**

- 1 - Authenticate Customer
- 2 - Customer sets destination account and branch
- 3 - Customer sets the amount for transfer
- 4 - Customer sets the transfer date
- 5 - System executes the transfer
- 6 - System asks if the customer wants to make another transfer

### **Extensions:**

- 2a - If incorrect account and branch, request new account and branch
- 3a - If transfer amount exceeds current balance, request new amount
- 4a - Date must be the current one or no more than one year in the future
- 5a - If the date is the current one, process transfer immediately
- 5b - If the request lies in the future, schedule 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:

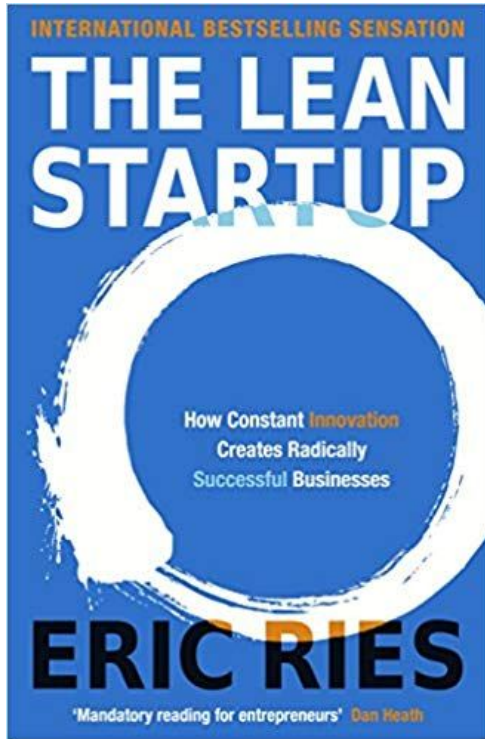
1. Cashier enters the cash amount tendered.
2. System presents the balance due, and releases the cash drawer.
3. Cashier deposits cash tendered and returns balance in cash to Customer.
4. System records the cash payment.

### 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





2011

# Two types of systems

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

# Low risk and known users

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

# High risk and uncertain market

- Example: ebook store with payments via bitcoins
  - Typical systems of startups, but not exclusive
  - As risk is high, idea must be validated with real users

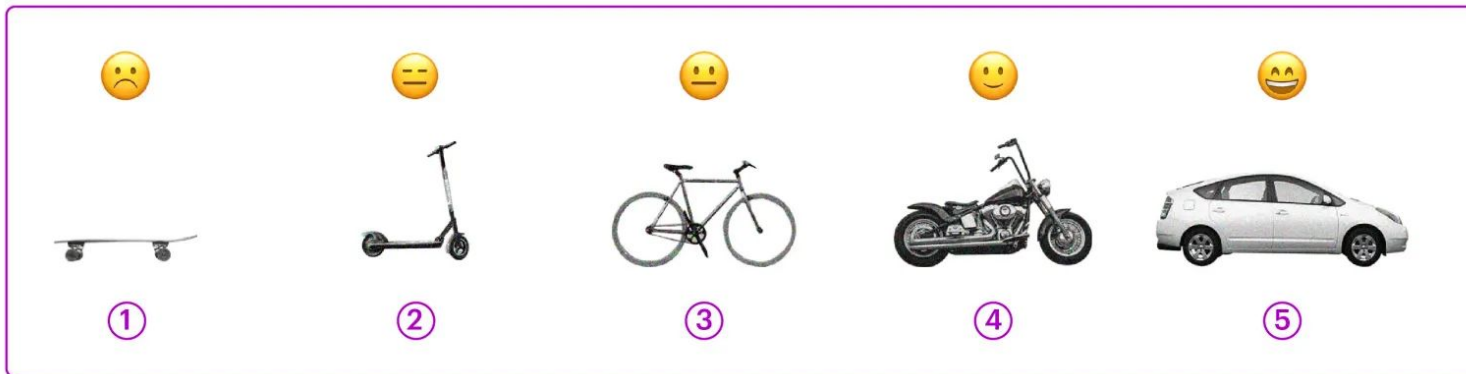
# MVP

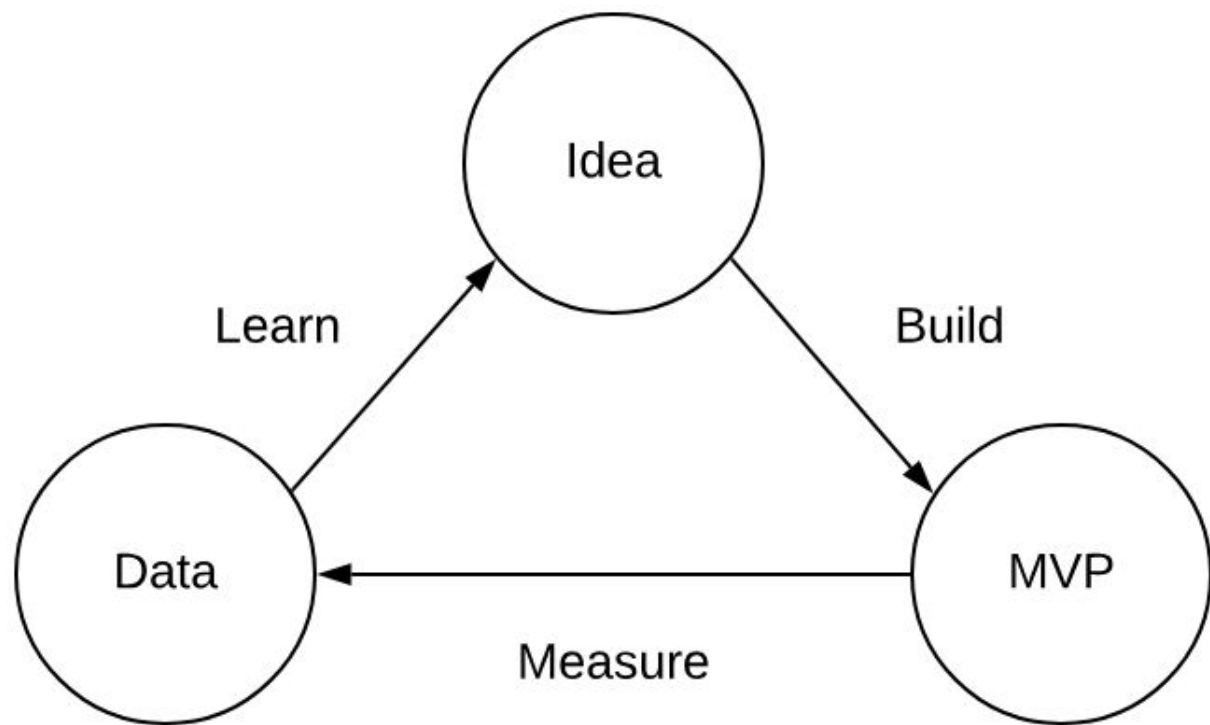
- Product = can be used
- Minimum = small number of features (lowest cost)
- Viable = will it have market?

Using MVPs



Without MVPs





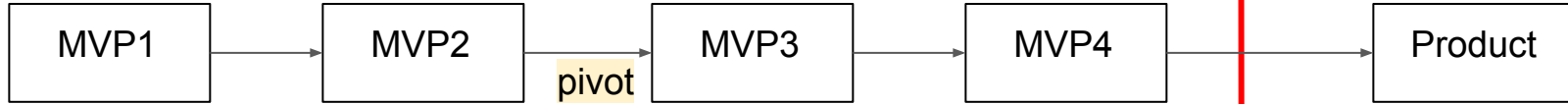
# At the end of a cycle, we can:

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

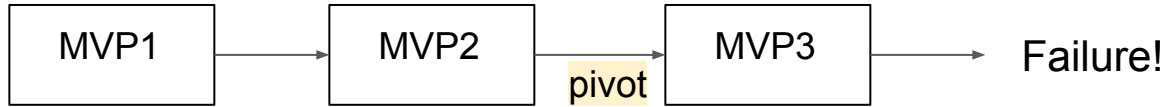


## Validation

## Market Fit



## Validation



---

## Market Fit

# MVP vs Interviews

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



# MVP Examples

# Zappos

- Online shoe store, later acquired by Amazon
- Hypothesis: Will 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](mailto:freeshoes@zappos.com) with your name and email address. A winner will be notified by email every Wednesday!

featured brands.



Zappos Special:

Free Shipping &  
No Sales Tax

home

women's dress

women's casual

women's athletic

men's dress

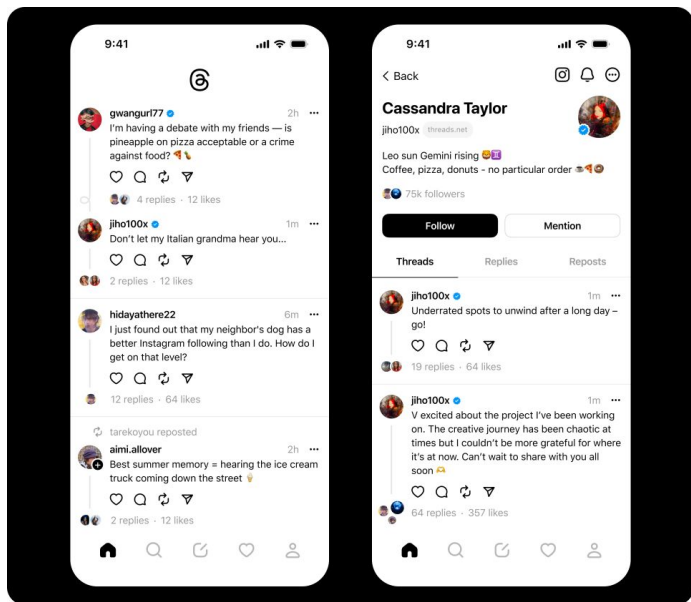
men's casual

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

# 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, NFR such as performance, usability, availability, etc

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

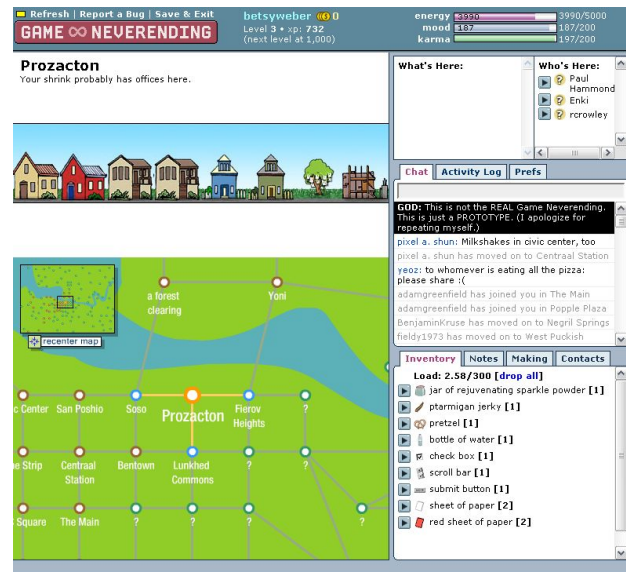
- It depends and varies a lot, but it must be rapid
- For example, two weeks...
- “If you're not embarrassed by the first version of your product, you've launched too late” (Reid Hoffman)

# 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 massive 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
- Also originated from an online RPG

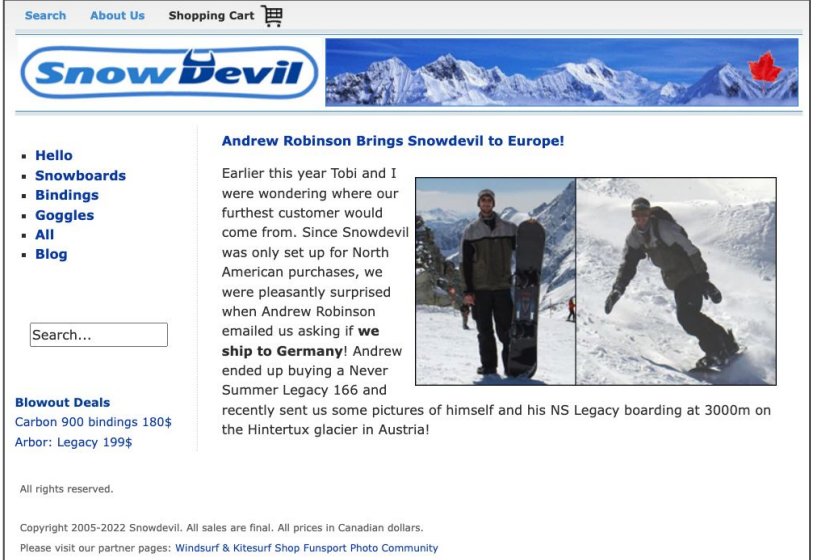
# Customer Segment: Twitch

- 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



The screenshot shows the Snowdevil website interface. At the top, there is a navigation bar with links for "Search", "About Us", and "Shopping Cart" (with a shopping cart icon). Below the navigation bar is a banner featuring the Snowdevil logo, a scenic image of snow-capped mountains, and a red maple leaf. The main content area is divided into two columns. The left column contains a vertical menu with links: "Hello", "Snowboards", "Bindings", "Goggles", "All", and "Blog". Below the menu is a search input field labeled "Search...". Underneath the search field, there is a "Blowout Deals" section listing "Carbon 900 bindings 180\$" and "Arbor: Legacy 199\$". The right column features a blog post titled "Andrew Robinson Brings Snowdevil to Europe!". The text of the post reads: "Earlier this year Tobi and I were wondering where our furthest customer would come from. Since Snowdevil was only set up for North American purchases, we were pleasantly surprised when Andrew Robinson emailed us asking if **we ship to Germany!** Andrew ended up buying a Never Summer Legacy 166 and recently sent us some pictures of himself and his NS Legacy boarding at 3000m on the Hintertux glacier in Austria!". Two photographs accompany the text: one shows a man standing with a snowboard, and the other shows a man snowboarding down a slope. At the bottom of the page, there is a footer with the text: "All rights reserved.", "Copyright 2005-2022 Snowdevil. All sales are final. All prices in Canadian dollars.", and "Please visit our partner pages: Windsurf & Kitesurf Shop Funsport Photo Community".



# Technology: Android

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

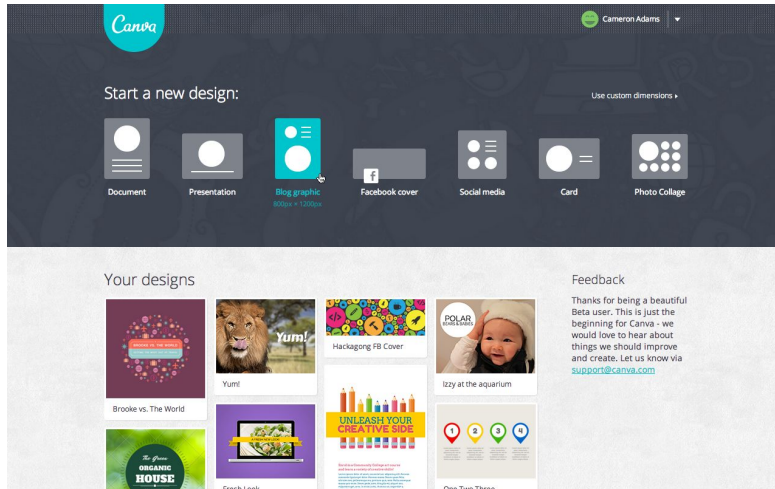


# Exercises about MVPs

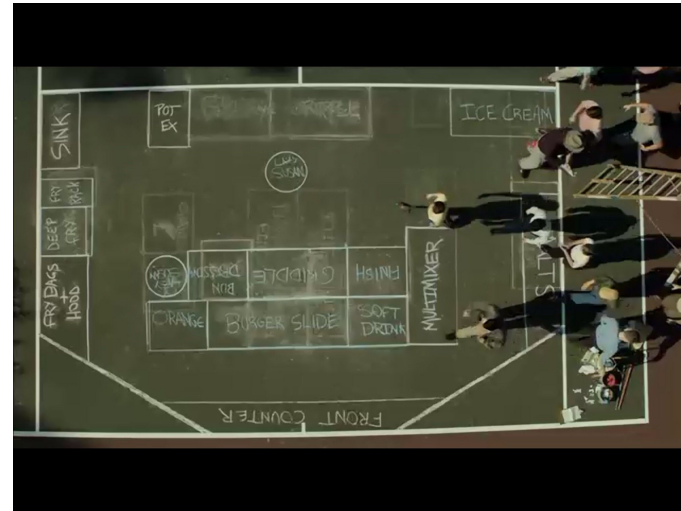
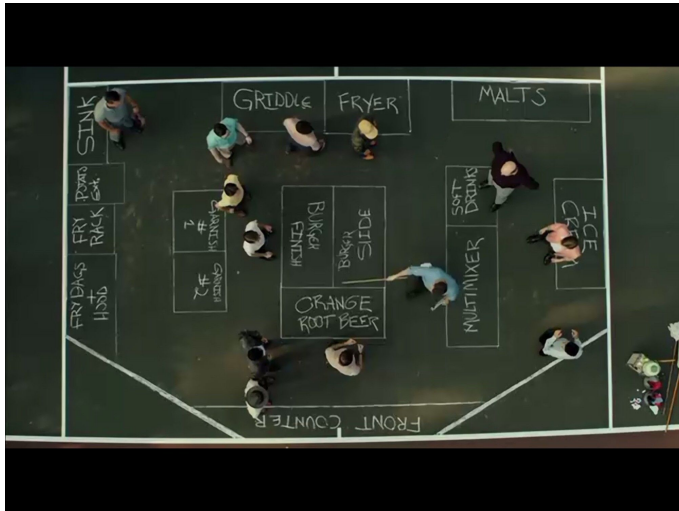
1. What is the difference between an MVP and a market survey?
2. Suppose you had the idea of creating a system to organize car rides for the students in your university.
  - a. How would you implement an MVP for this system without writing code?
  - b. Suppose this MVP failed, describe a pivot you can 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 shouldn't be called an MVP

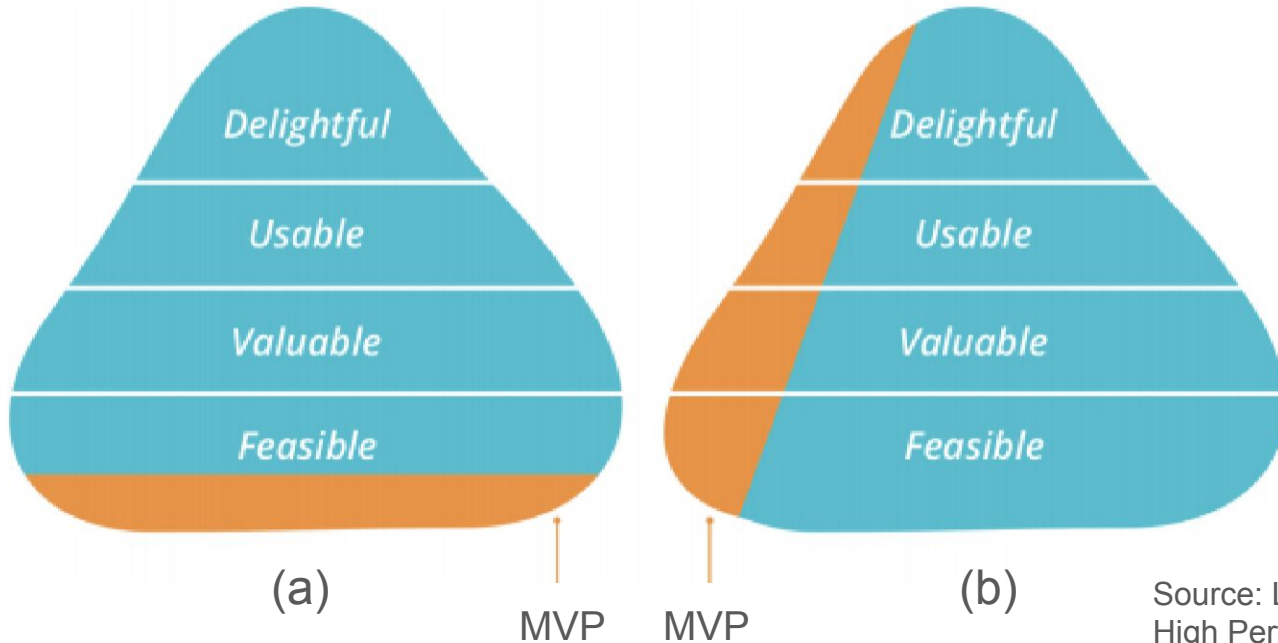


4. In the 1940s, prior to founding McDonald's, the company's founders conducted several experiments using kitchen plans drawn with chalk on a tennis court. The objective was to test various kitchen layouts before selecting one. Do these models qualify as MVPs? Justify.



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

# 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: let's 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 has statistically relevant 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

Baseline Conversion Rate  
Your control group's expected conversion rate.. [?]

1 %

Minimum Detectable Effect  
The minimum relative change in conversion rate you would like to be able to detect.. [?]

10 %

Statistical Significance  
95% is an accepted standard for statistical significance, although Optimizely allows you to set your own threshold for significance based on your risk tolerance.. [?]

95 %  
[Edit](#)

Sample size per variation

**200,000**

# Final Comments

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

# Exercises

1. 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 parallel applications in general.

Can this change (from graphics processing to general parallel processing) be considered a pivot? Justify.

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