UNIVERSITATEA DIN BUCUREȘTI

VIRTUTE ET SAPIENTIA

tremend

A company of publicis sapient

Course 5
Requirements Analysis and Design Definition (part 2)

500 Technology Fast 500
2019 EMEA WINNER
Deloitte.

IMPACT STAR Award
by Deloitte. Technology
FAST 50 CENTRAL EUROPE 2020

Proud Member Of
EBRD Exclusive Blue
Ribbon Global Network Of
Best Performing SMEs

Agenda

Delivering just enough

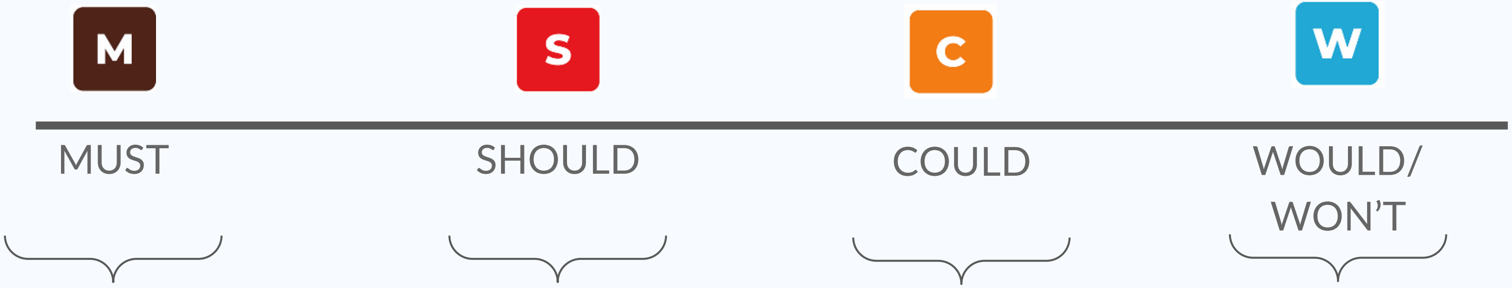
- Prioritization techniques
- Roadmap, MVP & MMP

Visual representation/modelling techniques

MoSCoW prioritization

Prioritization ensures that the most critical parts of a system are built first and everyone understands and agrees on what those parts are.

A well-organized requirements list will also ensure that the development team will build the most essential components in accordance with business objectives.



MUST

First priority functionality. Non-negotiable.

SHOULD

Important functionality. Not vital, but adds value.

COULD

Nice to have functionality. Can be left for later without much impact.

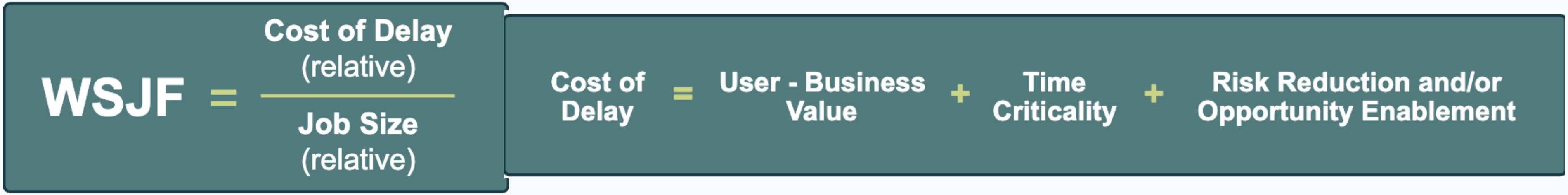
WOULD/ WON'T

Functionality that does not represent a current priority. Can be postponed for much later, or

WSJF - Weighted Shortest Job First

- (WSJF) is a prioritization model used to sequence work for maximum economic benefit
- Backlogs are **continuously** prioritized (each iteration)
- Uses **relative** measures of value, time criticality, risk reduction and/or opportunity enablement, and job size.
- Specific monetary elements of CoD components are unnecessary, significantly reducing complexity and the time spent on prioritization.
- This model encourages splitting large jobs into smaller ones. Otherwise, critical big jobs might never get done.

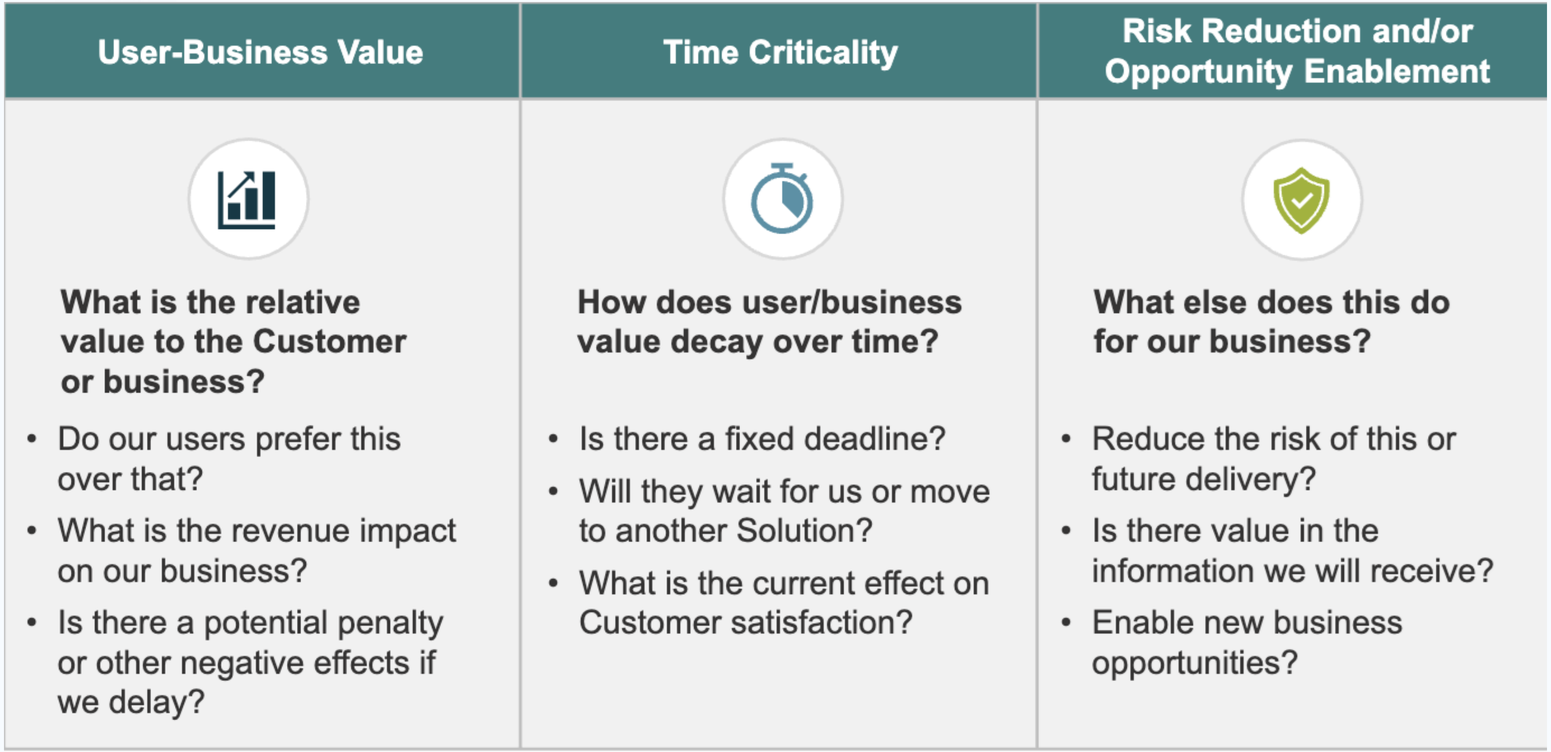
© Scaled Agile, Inc



$$\text{WSJF} = \frac{\text{Cost of Delay (relative)}}{\text{Job Size (relative)}}$$

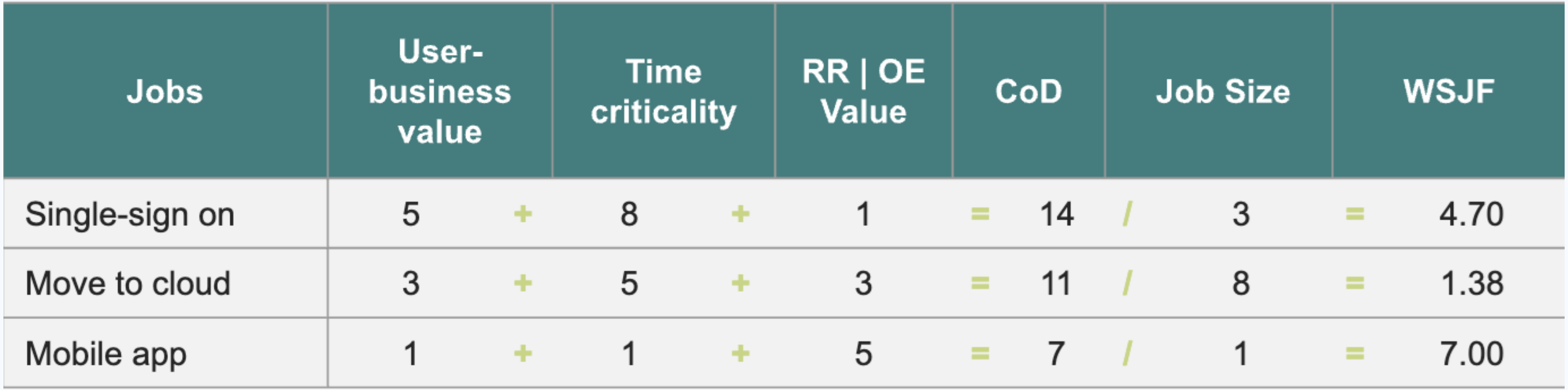
$$\text{Cost of Delay} = \text{User - Business Value} + \text{Time Criticality} + \text{Risk Reduction and/or Opportunity Enablement}$$

WSJF - Weighted Shortest Job First



User-Business Value	Time Criticality	Risk Reduction and/or Opportunity Enablement
What is the relative value to the Customer or business? • Do our users prefer this over that? • What is the revenue impact on our business? • Is there a potential penalty or other negative effects if we delay?	**How does user/business value decay over time?** • Is there a fixed deadline? • Will they wait for us or move to another Solution? • What is the current effect on Customer satisfaction?	**What else does this do for our business?** • Reduce the risk of this or future delivery? • Is there value in the information we will receive? • Enable new business opportunities?

WSJF - The highest WSJF is the next most important job to do



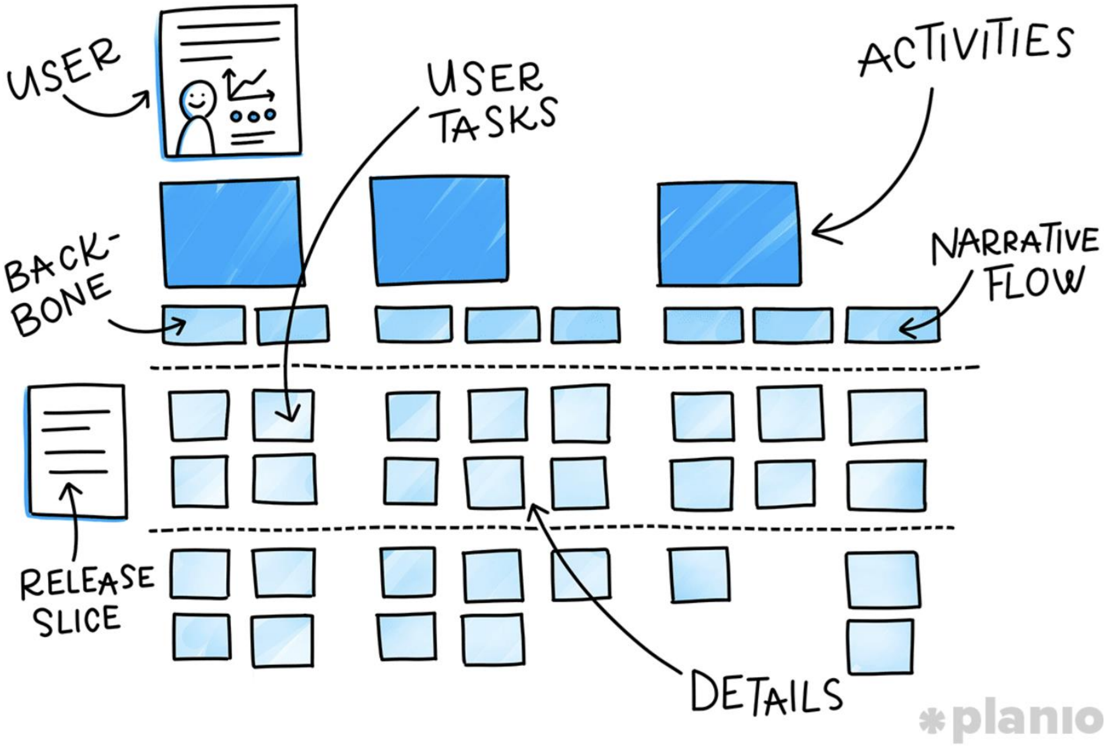
Jobs	User-business value	Time criticality	RR | OE Value	CoD	Job Size	WSJF
Single-sign on	5 +	8 +	1	= 14	/ 3	= 4.70
Move to cloud	3 +	5 +	3	= 11	/ 8	= 1.38
Mobile app	1 +	1 +	5	= 7	/ 1	= 7.00

Notes:

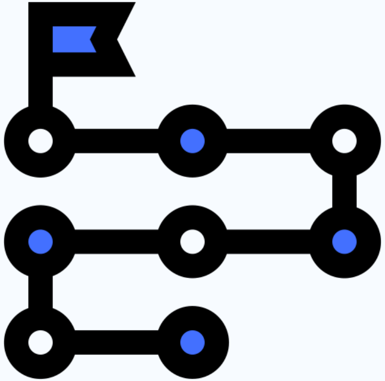
- Scale for each parameter: 1, 2, 3, 5, 8, 13, 20
- Do one column at a time, start by picking the smallest item and give it a “1”
- There must be at least one “1” in each column

User Story Mapping

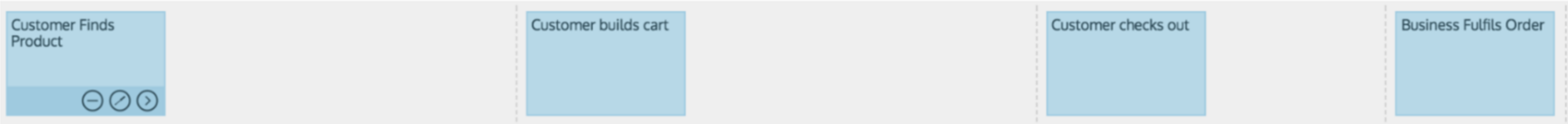
Story mapping is a simple technique that helps you, your team and your stakeholders talk about the right things and gain shared understanding on what you need to build to make your users happier than they were before your product came out.



Map the big picture next



- Identify the critical users of your product and map their interaction with your product;
- Map the activities the users will undertake & think what tasks are part of those activities for reaching their goal
- Add any other users as you tell the story and discover that their tasks are also part of the story you are telling



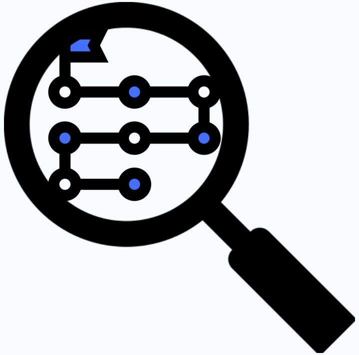
Customer Finds Product

Customer builds cart

Customer checks out

Business Fulfils Order

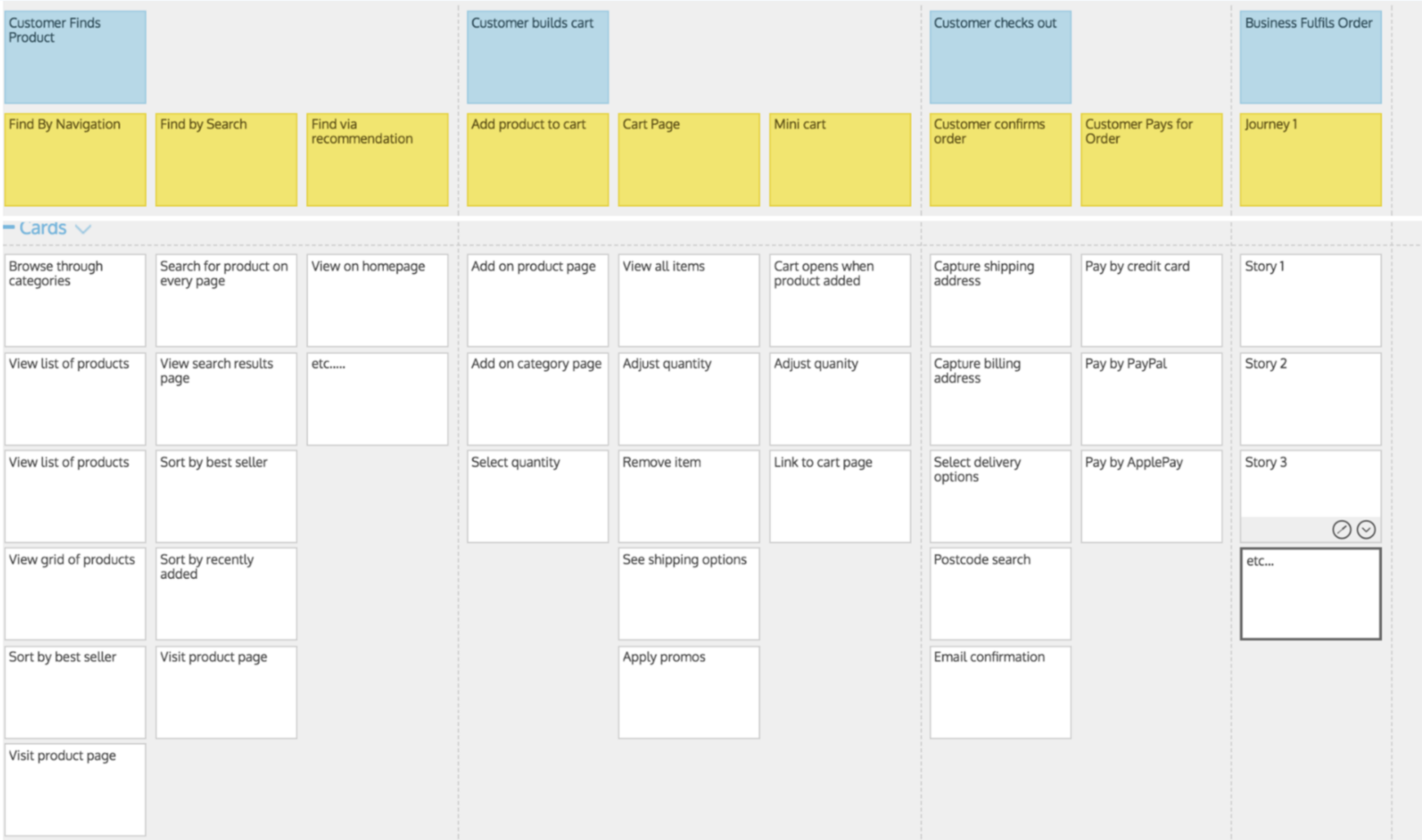
Explore the story in depth



Once you’ve got the backbone of your map, start drilling down:

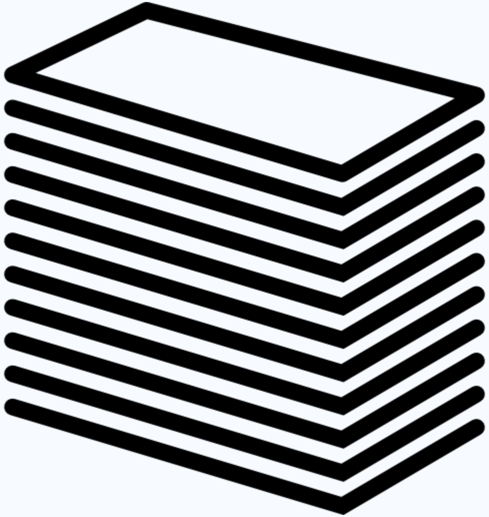
- **Split** the activities that are part of your story into tasks; the tasks will make great user stories for when you build your backlog later;
- Think of the **ideal journey** for your users as well as **what could go wrong**; identify **variations** in your users behaviours and possible **exceptions;**
- Add in details with **wireframes**, **business rules**, **data** etc.
- Ask others to listen to your story and **fill the gaps** that they point out (developers will point out **constraints**, **risks** and find tech **solutions** for your problems).

User Story Mapping - Example



Customer Finds Product			Customer builds cart			Customer checks out		Business Fulfils Order
Find By Navigation	Find by Search	Find via recommendation	Add product to cart	Cart Page	Mini cart	Customer confirms order	Customer Pays for Order	Journey 1
Cards								
Browse through categories	Search for product on every page	View on homepage	Add on product page	View all items	Cart opens when product added	Capture shipping address	Pay by credit card	Story 1
View list of products	View search results page	etc.....	Add on category page	Adjust quantity	Adjust quanity	Capture billing address	Pay by PayPal	Story 2
View list of products	Sort by best seller		Select quantity	Remove item	Link to cart page	Select delivery options	Pay by ApplePay	Story 3
View grid of products	Sort by recently added			See shipping options		Postcode search		etc...
Sort by best seller	Visit product page			Apply promos		Email confirmation		
Visit product page								

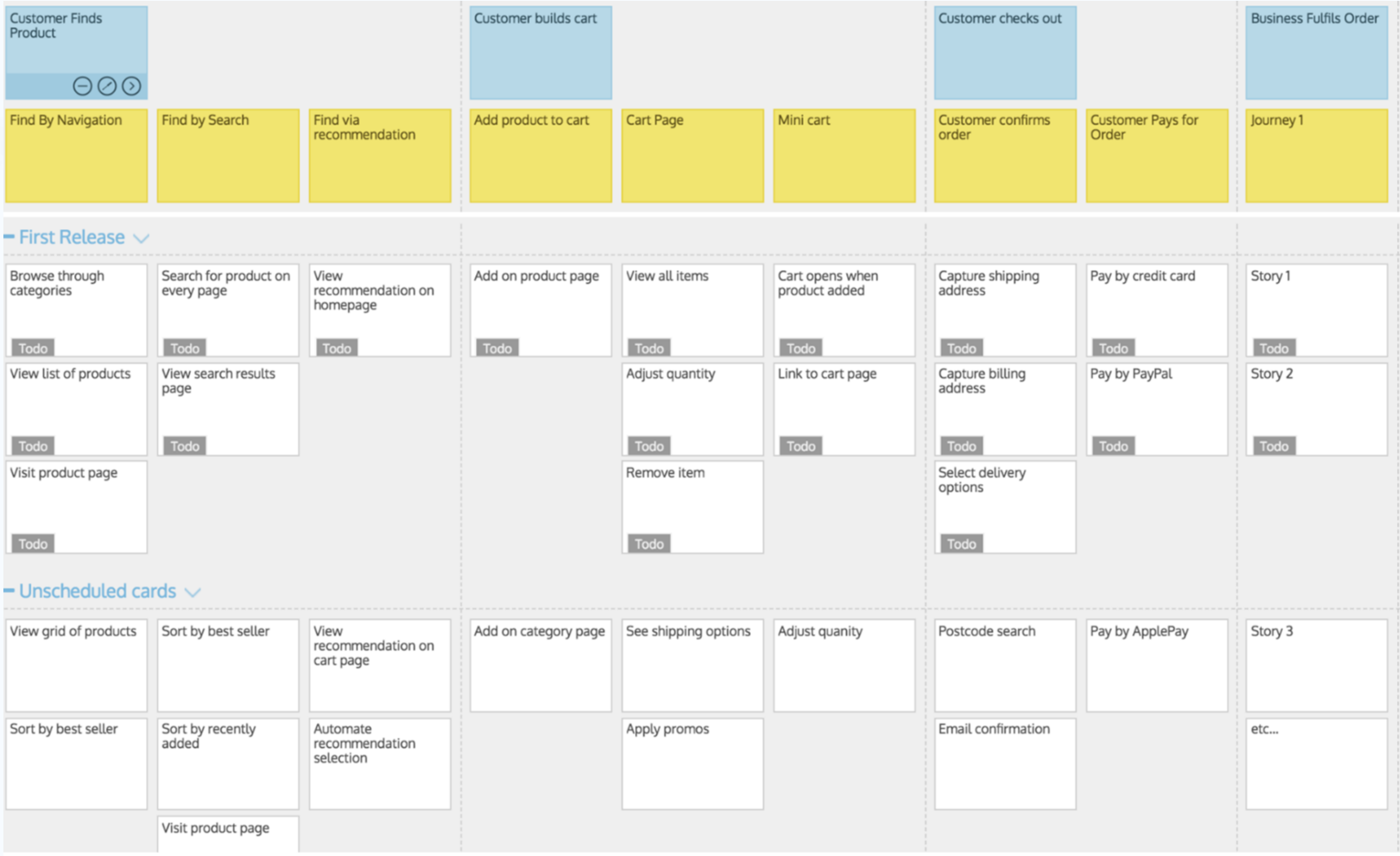
Slice out releases



Now that you’ve got a whole map, you can slice out viable releases:

- **Slice horizontally through your map** to identify viable incremental versions of your product that can be released;
- **Identify the goals and outcomes for each of your releases** to show how each release contributes to your products overall goals;
- **For each release, find the right product success metrics** to show you how well you are doing every step of the way.

User Story Mapping - Example



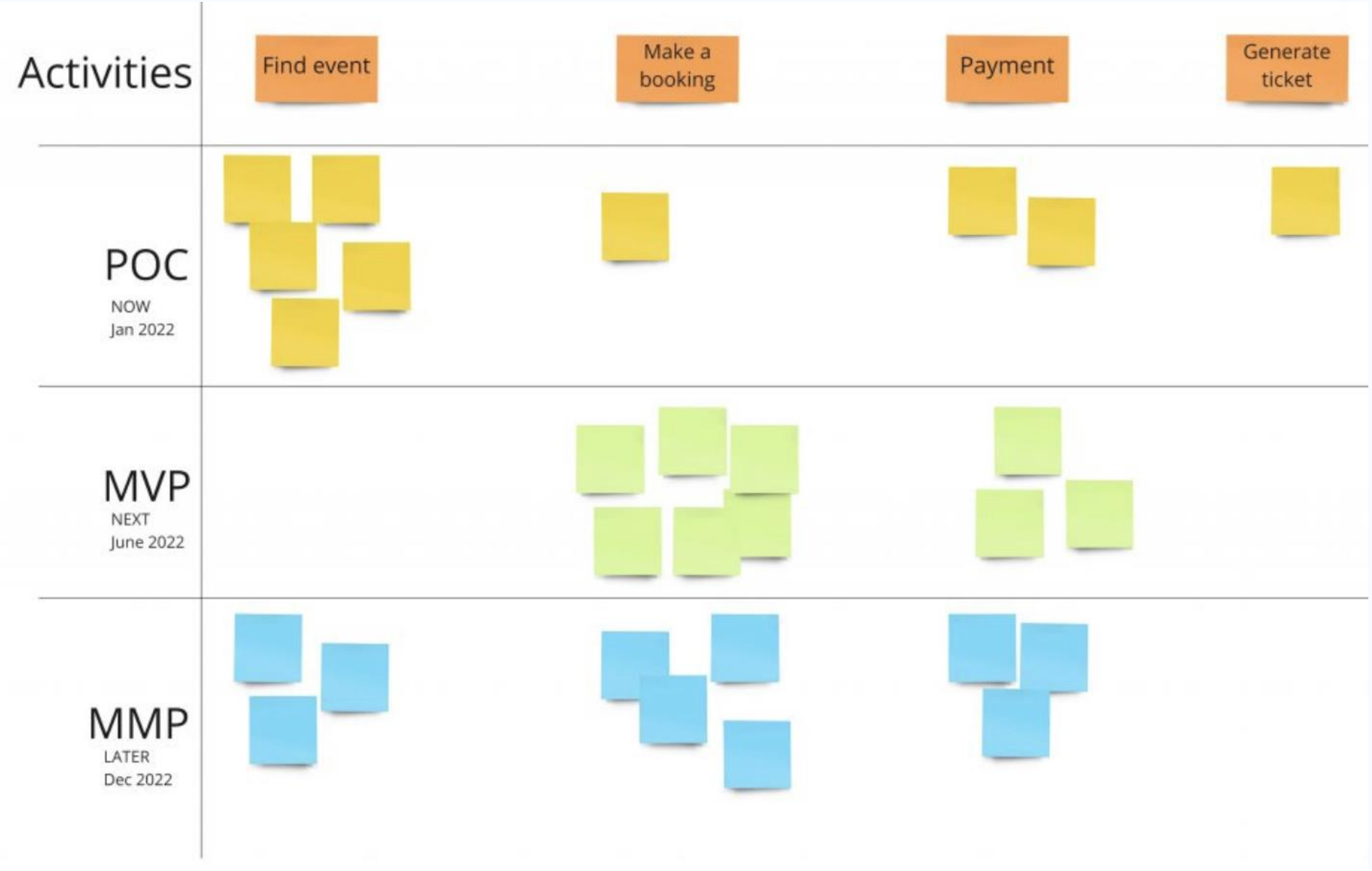
Customer Finds Product			Customer builds cart			Customer checks out		Business Fulfils Order
Find By Navigation	Find by Search	Find via recommendation	Add product to cart	Cart Page	Mini cart	Customer confirms order	Customer Pays for Order	Journey 1
First Release								
Browse through categories Todo	Search for product on every page Todo	View recommendation on homepage Todo	Add on product page Todo	View all items Todo	Cart opens when product added Todo	Capture shipping address Todo	Pay by credit card Todo	Story 1 Todo
View list of products Todo	View search results page Todo			Adjust quantity Todo	Link to cart page Todo	Capture billing address Todo	Pay by PayPal Todo	Story 2 Todo
Visit product page Todo				Remove item Todo		Select delivery options Todo		
Unscheduled cards								
View grid of products	Sort by best seller	View recommendation on cart page	Add on category page	See shipping options	Adjust quanity	Postcode search	Pay by ApplePay	Story 3
Sort by best seller	Sort by recently added	Automate recommendation selection		Apply promos		Email confirmation		etc...
	Visit product page							

Slice out a development strategy

While you may have sliced out potential releases, there's one more thing to consider - what will be your development strategy?

- Plan out how to go about delivering your **MVP** to get early feedback and how to build on top based on your new found knowledge of your users' behaviour;
- Don't forget to factor in time and effort for refine and workshop stories with the development team, proper testing, actual release activities and closing feedback loops.

User Story Mapping - Example



Activities

Find event

Make a booking

Payment

Generate ticket

POC
NOW
Jan 2022

MVP
NEXT
June 2022

MMP
LATER
Dec 2022

Buy-a-Feature Prioritization Model Framework

- Interactive method (game) that helps you reveal what people value most in a product or service
- Organizations and customers usually want *all* of the features to be created, without accounting for the *resources available* to develop those features
- The Buy-a-Feature game gives a limited amount of “money” (e.g. for ⅔ of the features) to potential customers, places prices on features, and asks that they purchase features with the “money” they are given
- By putting the decision making power in the hands of the customer, you can see what features are ‘make it or break’ it features and which they are willing to forgo
- Consider making your own money with your products’ logo on it. This can be a fun souvenir for the players

Roadmap

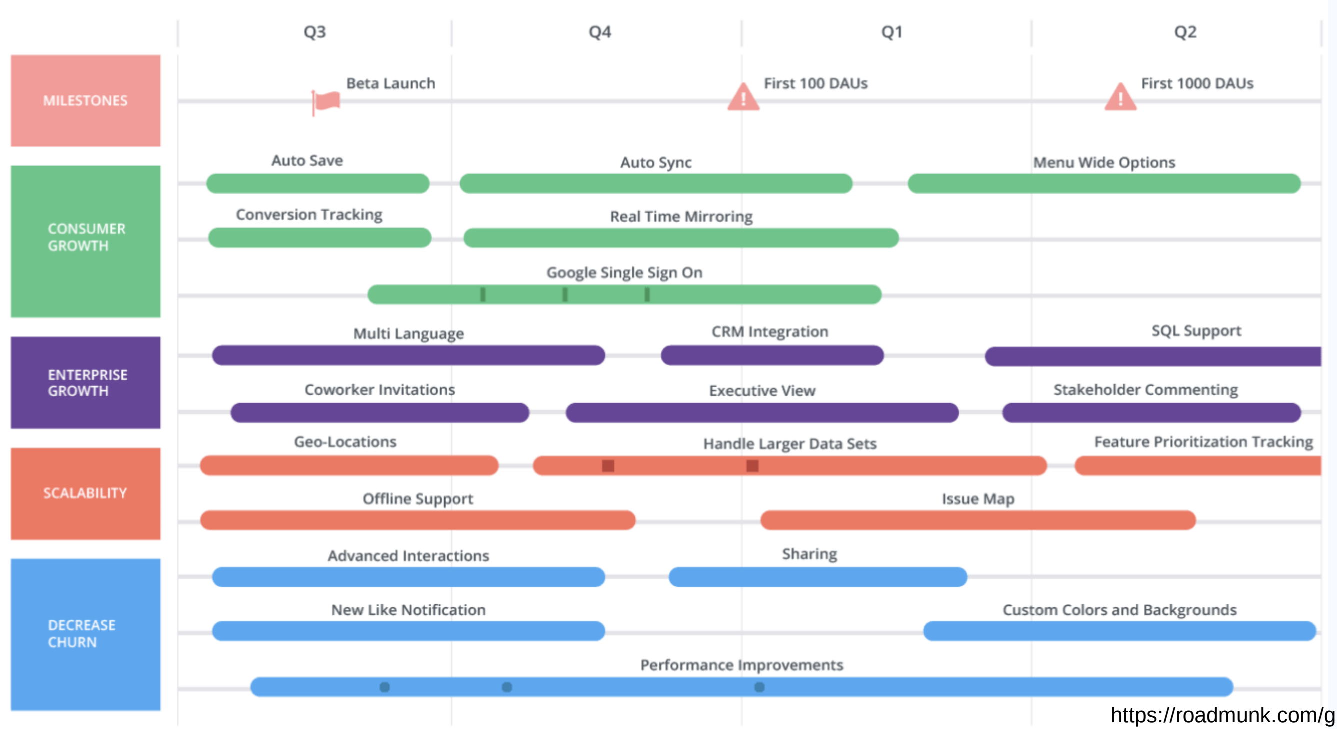
The **product roadmap** shows how the product strategy is put into **action by stating specific releases with dates, goals, and features.**

The product strategy describes how the long-term goal is attained; it includes the product’s value proposition, market, key features, and business goals.

Roman Pichler

Roadmap

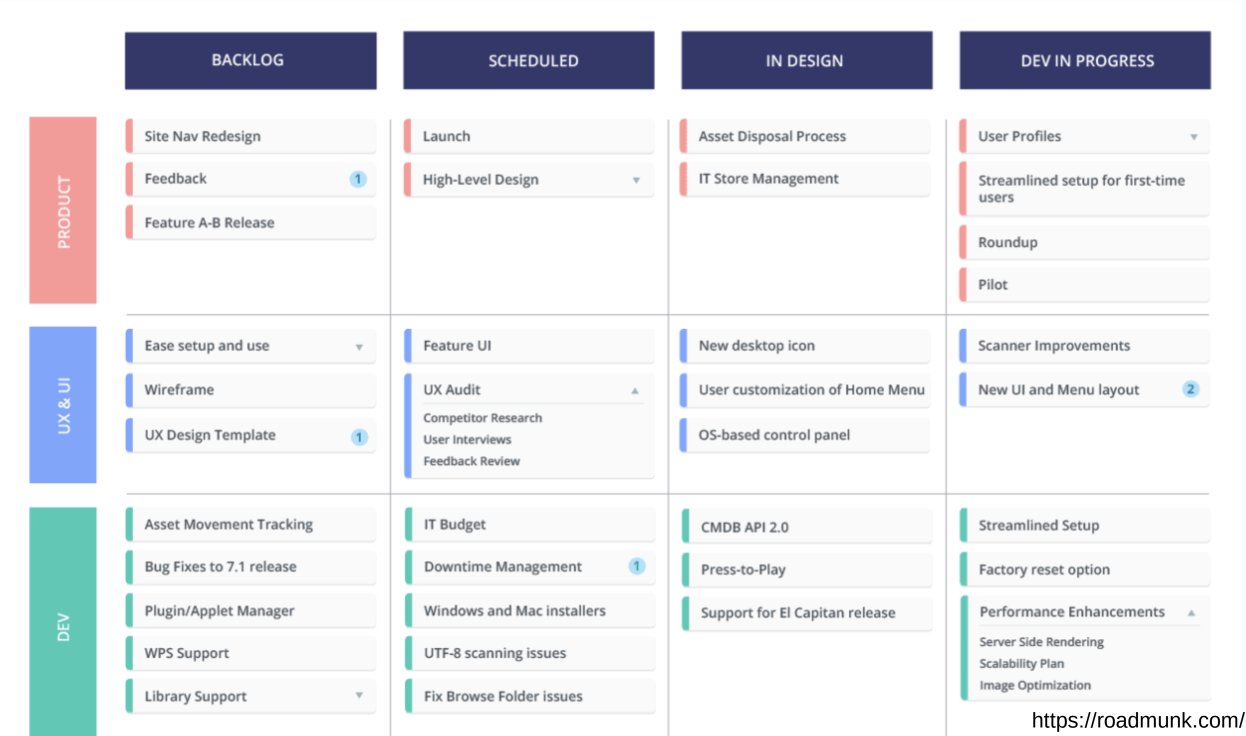
A product roadmap is a shared source of truth that outlines the vision, direction, priorities, and progress of a product over time.



https://roadmunk.com/guides/what-is-a-product-roadmap/

Roadmap

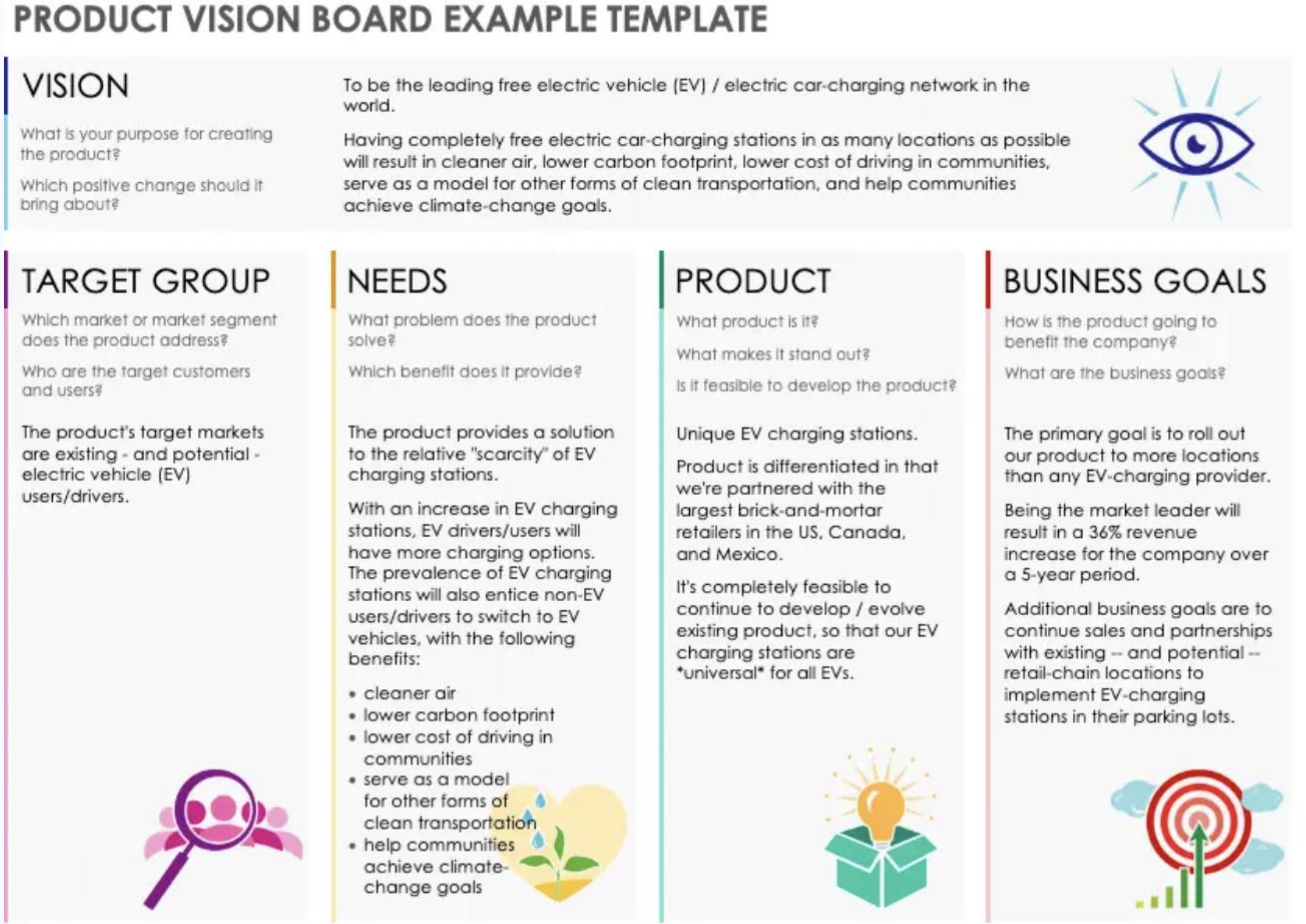
It is a plan of action that aligns the organization around short and long-term goals for the product or project, and how they will be achieved.



	BACKLOG	SCHEDULED	IN DESIGN	DEV IN PROGRESS
PRODUCT	Site Nav Redesign Feedback 1 Feature A-B Release	Launch High-Level Design ▾	Asset Disposal Process IT Store Management	User Profiles ▾ Streamlined setup for first-time users Roundup Pilot
UX & UI	Ease setup and use ▾ Wireframe UX Design Template 1	Feature UI UX Audit ▴ Competitor Research User Interviews Feedback Review	New desktop icon User customization of Home Menu OS-based control panel	Scanner Improvements New UI and Menu layout 2
DEV	Asset Movement Tracking Bug Fixes to 7.1 release Plugin/Applet Manager WPS Support Library Support ▾	IT Budget Downtime Management 1 Windows and Mac installers UTF-8 scanning issues Fix Browse Folder issues	CMDB API 2.0 Press-to-Play Support for El Capitan release	Streamlined Setup Factory reset option Performance Enhancements ▴ Server Side Rendering Scalability Plan Image Optimization

https://roadmunk.com/guides/what-is-a-product-roadmap/

Product vision - example



PRODUCT VISION BOARD EXAMPLE TEMPLATE

VISION

What is your purpose for creating the product?

Which positive change should it bring about?

To be the leading free electric vehicle (EV) / electric car-charging network in the world.

Having completely free electric car-charging stations in as many locations as possible will result in cleaner air, lower carbon footprint, lower cost of driving in communities, serve as a model for other forms of clean transportation, and help communities achieve climate-change goals.

TARGET GROUP

Which market or market segment does the product address?

Who are the target customers and users?

The product's target markets are existing - and potential - electric vehicle (EV) users/drivers.

NEEDS

What problem does the product solve?

Which benefit does it provide?

The product provides a solution to the relative "scarcity" of EV charging stations.

With an increase in EV charging stations, EV drivers/users will have more charging options. The prevalence of EV charging stations will also entice non-EV users/drivers to switch to EV vehicles, with the following benefits:

- cleaner air
- lower carbon footprint
- lower cost of driving in communities
- serve as a model for other forms of clean transportation
- help communities achieve climate-change goals

PRODUCT

What product is it?

What makes it stand out?

Is it feasible to develop the product?

Unique EV charging stations.

Product is differentiated in that we're partnered with the largest brick-and-mortar retailers in the US, Canada, and Mexico.

It's completely feasible to continue to develop / evolve existing product, so that our EV charging stations are *universal* for all EVs.

BUSINESS GOALS

How is the product going to benefit the company?

What are the business goals?

The primary goal is to roll out our product to more locations than any EV-charging provider.

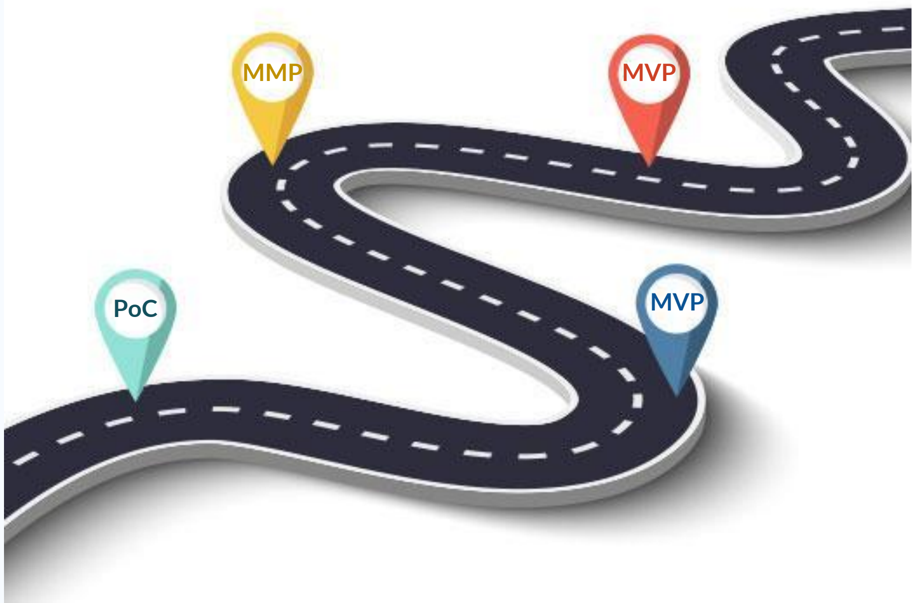
Being the market leader will result in a 36% revenue increase for the company over a 5-year period.

Additional business goals are to continue sales and partnerships with existing -- and potential -- retail-chain locations to implement EV-charging stations in their parking lots.

MVP vs. MMP

A **Minimum viable product (MVP)** is a version of a new product that is created with the least effort possible to be used for validated learning about customers. MVPs are created to help you to find the features that customers are actually interested in.

A **Minimum marketable product (MMP)** is the product that has the smallest possible feature set that addresses the current needs of your customers. This type of product has value to both the organization delivering it and the people using it.

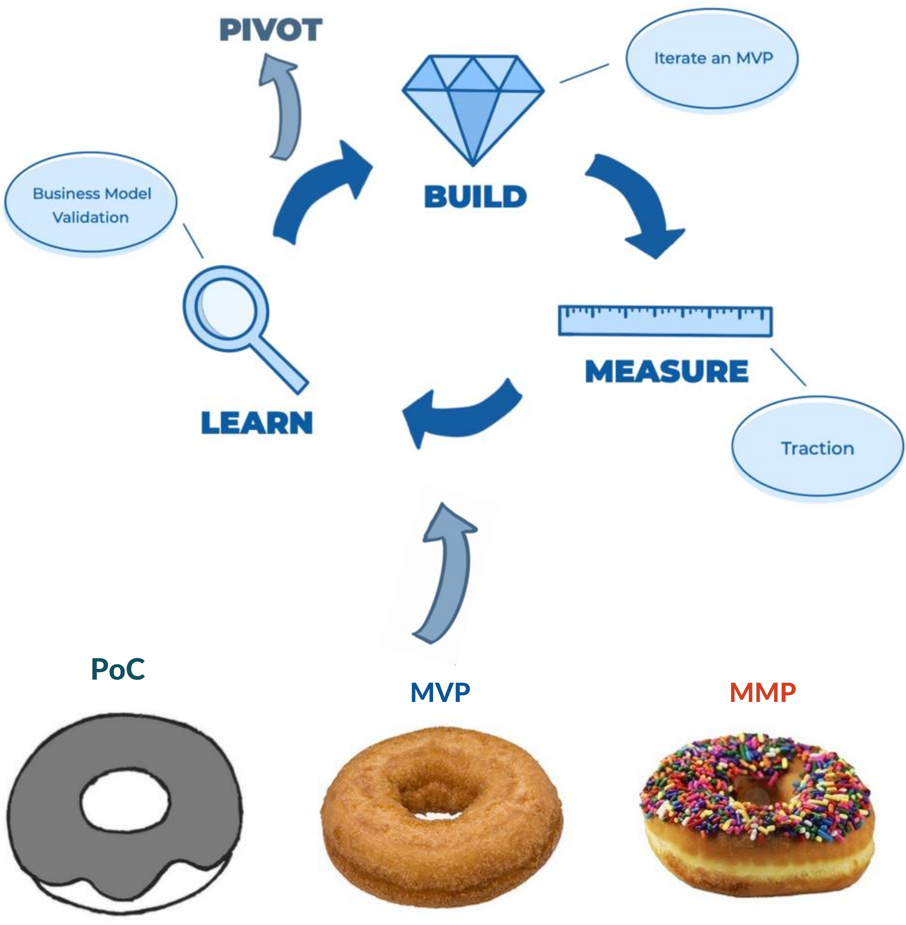


Why define an MVP?

Choosing to build an **MVP** will allow you to invest the absolute **minimum effort** in a version of the product that will allow the **users to test** it.

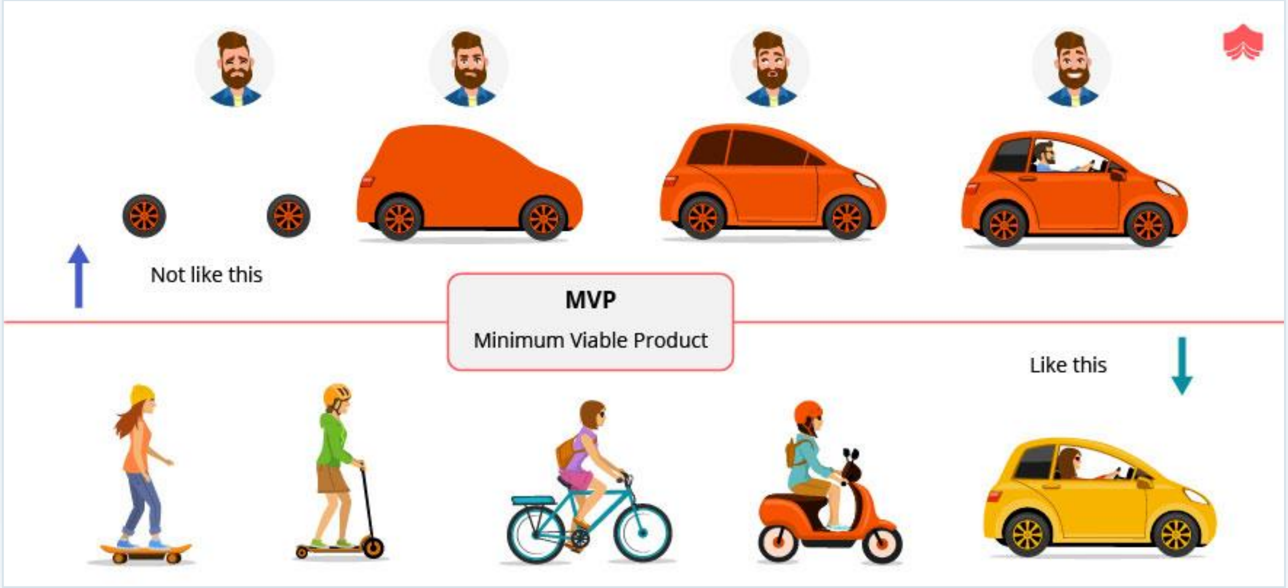
It’s quick & incomplete but will allow you to learn from the users and **adapt**. An MVP, **maximizes the work not done.**

The MVP will enable you to iterate with minimum costs.



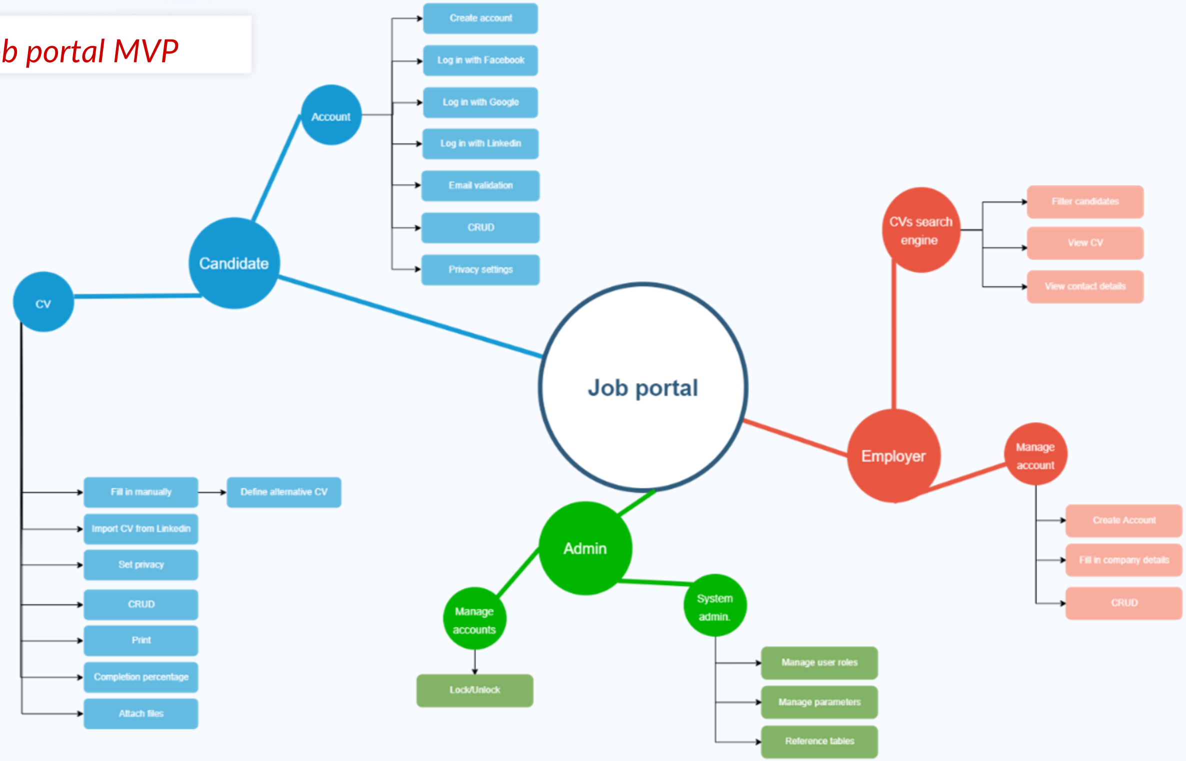
MValuableP

Mind that while the MVP will require the minimum effort, the effort must be invested in valuable functionalities. They may not be fully baked, but they must be valuable. Otherwise, the effort, whether minimum or not is wasted.

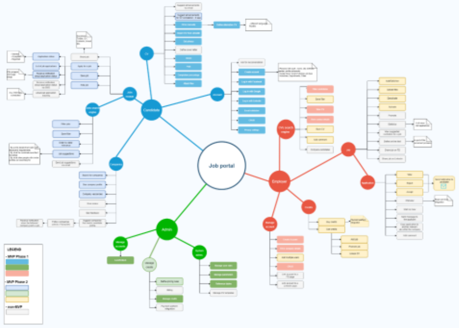


Job portal MVP - example

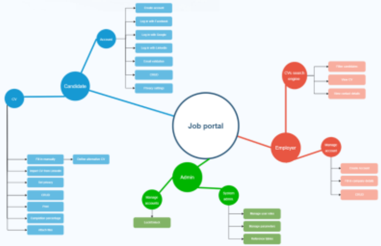
Job portal MVP

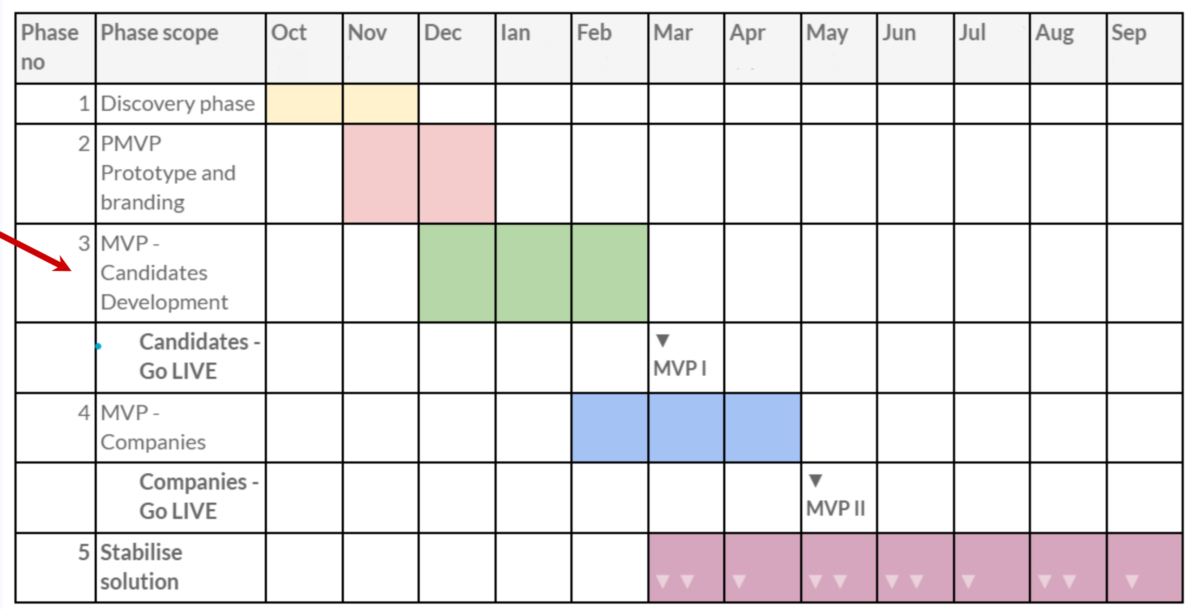


Job portal - full functionalities list



Job portal MVP - example





Phase no	Phase scope	Oct	Nov	Dec	Ian	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
1	Discovery phase	🟨	🟨										
2	PMVP Prototype and branding		🟥	🟥									
3	MVP - Candidates Development			🟩	🟩	🟩							
	Candidates - Go LIVE						▼ MVP I						
4	MVP - Companies					🟦	🟦	🟦					
	Companies - Go LIVE								▼ MVP II				
5	**Stabilise solution**						🟪 ▼ ▼	🟪 ▼	🟪 ▼ ▼	🟪 ▼ ▼	🟪 ▼	🟪 ▼ ▼	🟪 ▼

Visual representations

UML diagrams (activity/state diagram)

Context diagram

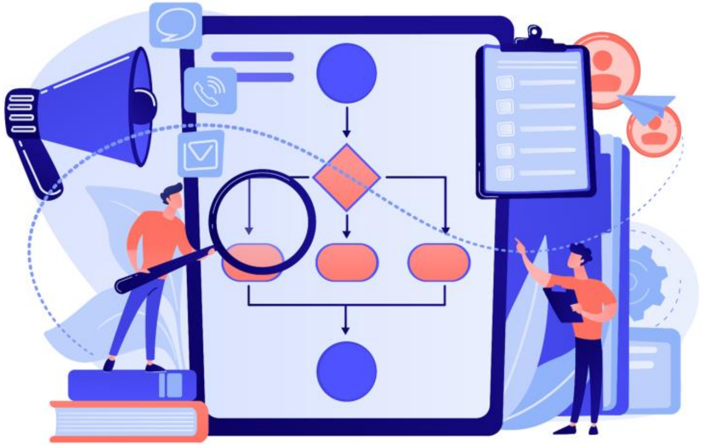
Sitemap

What are models and diagrams?

A model is an informative **representation** of an object, person or system. **Conceptual models** may be a representation of a system, concepts used to help people know, understand, or simulate a subject the model represents

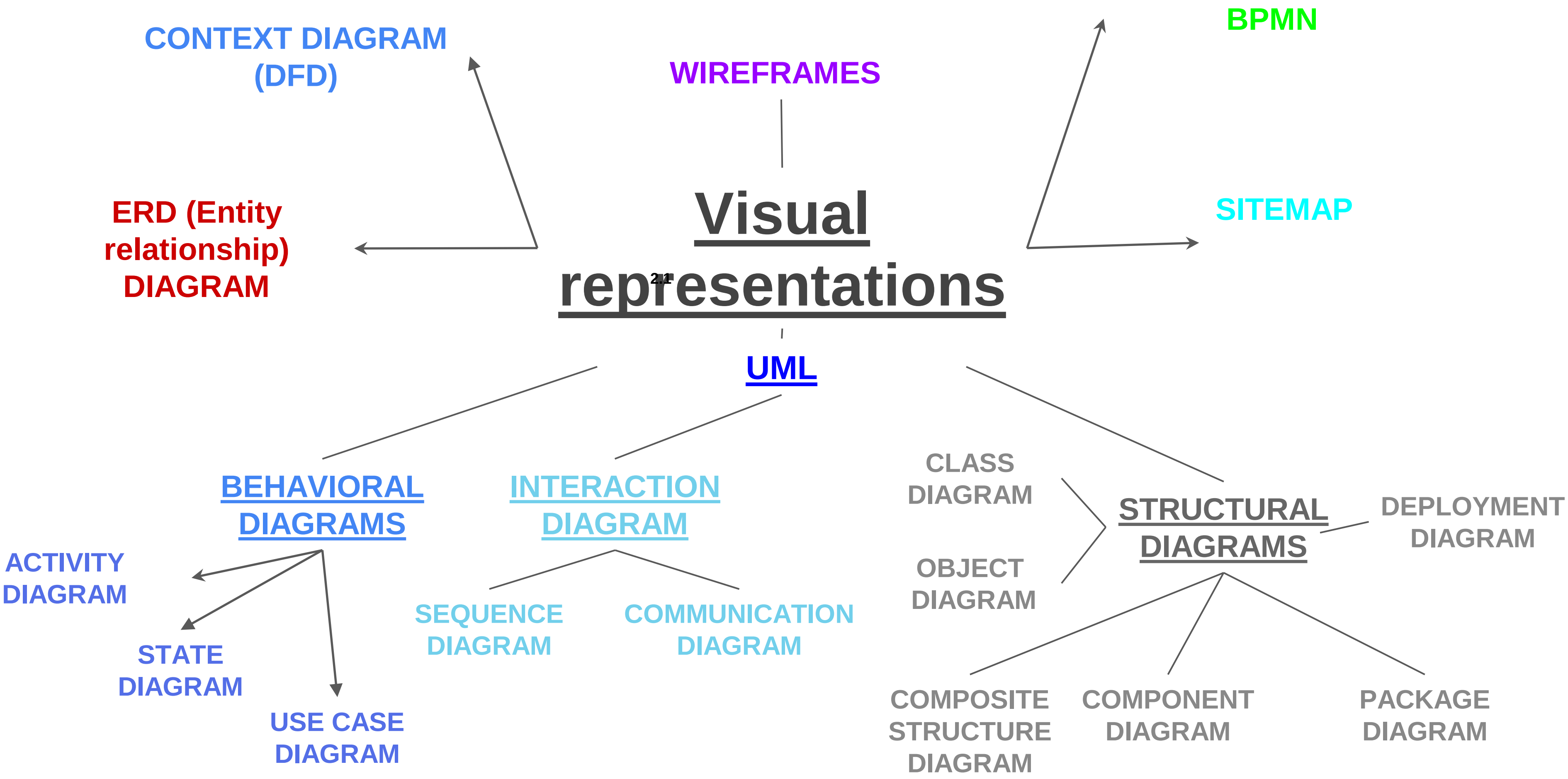
Diagrams are symbolic representations of information using visualization techniques.

- Used to validate solutions
- Facilitate communication with stakeholders
- Accompanies the textual form of specifications



Credits to: Free Vector | Businessmen with magnifier looking at business process flow chart. business rules and regulation, main company policy, it business analysis concept illustration

Visual representations



2.1

CONTEXT DIAGRAM (DFD)

ERD (Entity relationship) DIAGRAM

WIREFRAMES

BPMN

SITEMAP

UML

BEHAVIORAL DIAGRAMS

ACTIVITY DIAGRAM

STATE DIAGRAM

USE CASE DIAGRAM

INTERACTION DIAGRAM

SEQUENCE DIAGRAM

COMMUNICATION DIAGRAM

STRUCTURAL DIAGRAMS

CLASS DIAGRAM

OBJECT DIAGRAM

DEPLOYMENT DIAGRAM

COMPOSITE STRUCTURE DIAGRAM

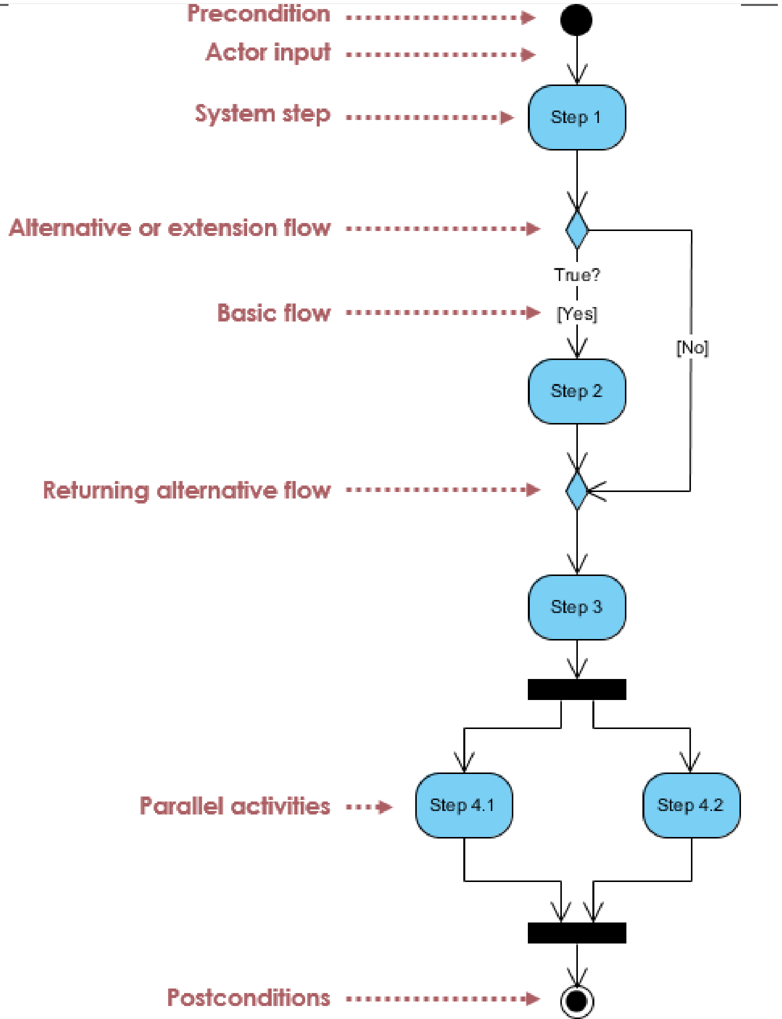
COMPONENT DIAGRAM

PACKAGE DIAGRAM

Activity diagram

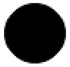
Is a behavioral diagram used to represent workflows

- ❑ Simple, understood by end-users
- ❑ Easy to learn
- ❑ Used at different levels of abstraction
- ❑ Focus on sequence of steps (the roles can be represented as partitions)
- ❑ Represents alternative flows and parallel flows

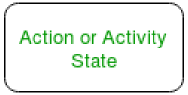


Notations

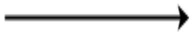
- **actions are always verbs**
- always use the initial state
- one or more final states are allowed
- from an activity there is only one action flow leaving, and one or more action flow entering
- parallel flows need to merge at some point
- action flows starting from a decision point need to include the condition for that flow to happen
- usually after each condition, 2 disjunctive flows are represented, yet you mai have a compound condition and more than 2 flows
- think of pairs Decision- Merge, Fork-Join



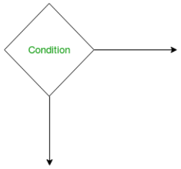
Initial State – The starting state.



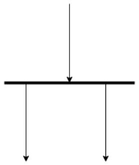
Action - execution of an action.



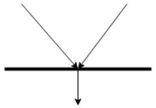
Action flow - transition from one activity to another.



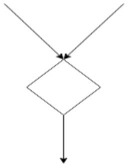
Decision node - The point of decision in the flow.



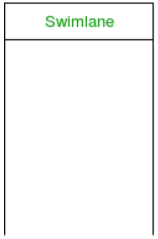
Fork - support concurrent activities.



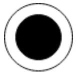
Join – concurrent activities converging into one.



Merge – activities which are not being executed concurrently have to be merged.

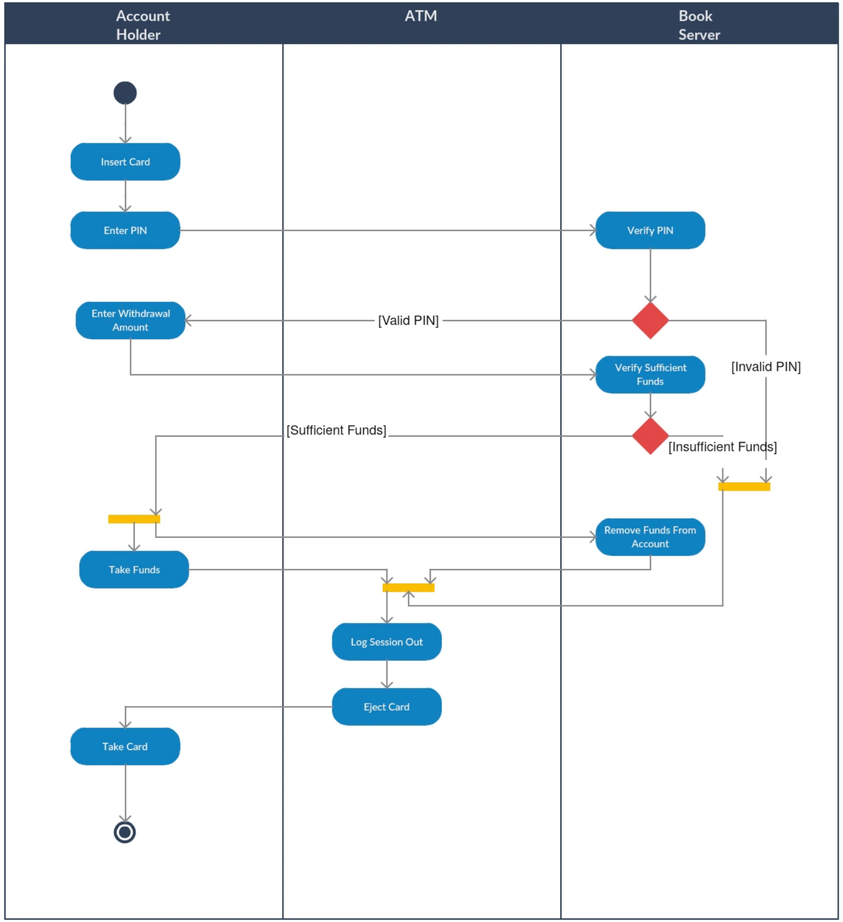


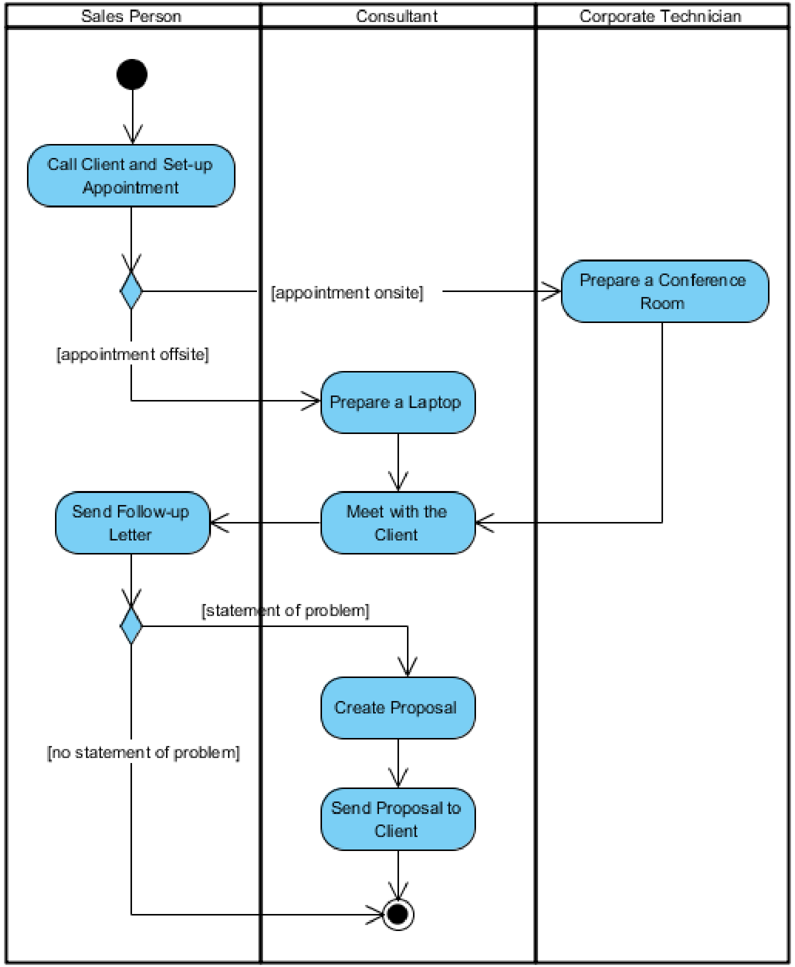
Swimlanes – swimlanes for group related activities in one column.



Final State – the state in which the activity ends.

ATM SYSTEM for ABC BANK

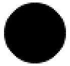




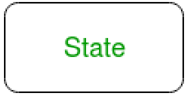
State diagram

A **state diagram** is used to represent the condition of the system or part of the system at given instances of time.

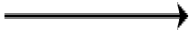
- model the dynamic behavior of an entity in response to time and external triggering factor
- used for 3 or more states
- similar notations as activity diagrams
- the states are adjectives
- the events include verbs and name the transitions
- there are no conditions represented
- forks and joins can be used



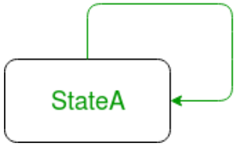
Initial State – The starting state.



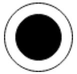
Action - the conditions or circumstances of an object of a class at an instant of time.



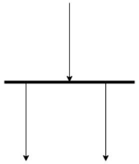
Transition – change of control from one state to another. The arrow is labelled with the event which causes the change in state.



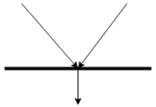
Self transition - when the state of the object does not change upon the occurrence of an event.



Final State – the state in which the activity ends.

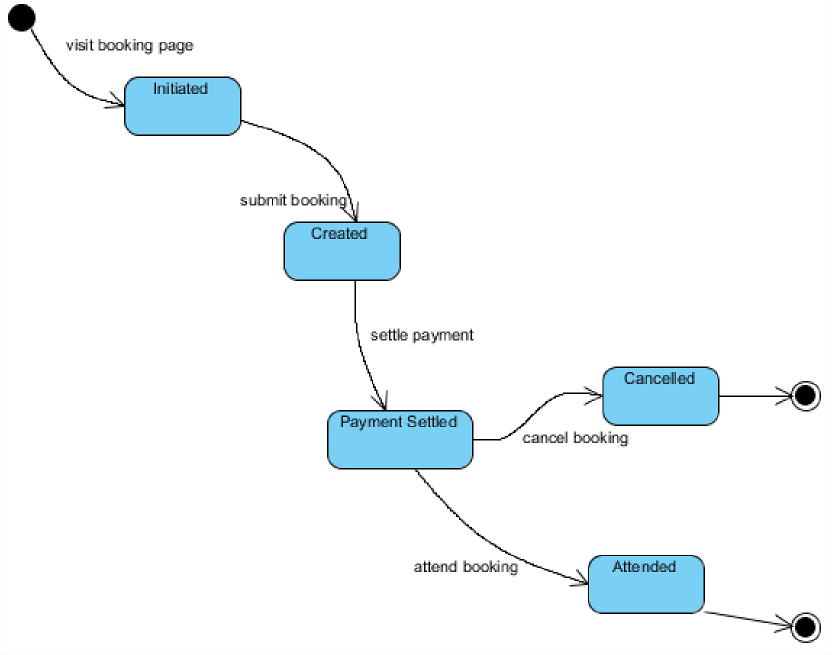


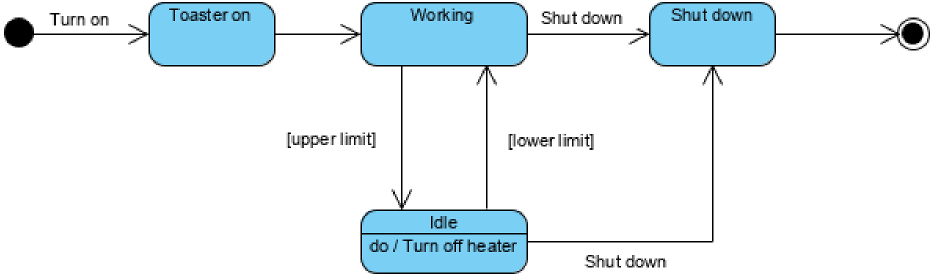
Fork - support concurrent states.



Join – two or more states concurrently converge into one on the occurrence of an event or events.

Examples


visit booking page
Initiated
submit booking
Created
settle payment
Cancelled
Payment Settled
cancel booking
attend booking
Attended


Turn on
Toaster on
Working
Shut down
Shut down
[upper limit]
[lower limit]
Idle
do / Turn off heater
Shut down

Context diagram

Is an abstraction view, showing the system as a single process with its relationship to external entities: people, organizations, and/or systems.

- ❑ Is level 0 data flow diagram
- ❑ the system as a single bubble with input and output data indicated by incoming/outgoing arrows
- ❑ defines the scope of your system
- ❑ created in early stage of the project (after 1-3 workshops)
- ❑ update it when new interactions/interfaces are added to the product

System

Process (in DFD 0 is the system)

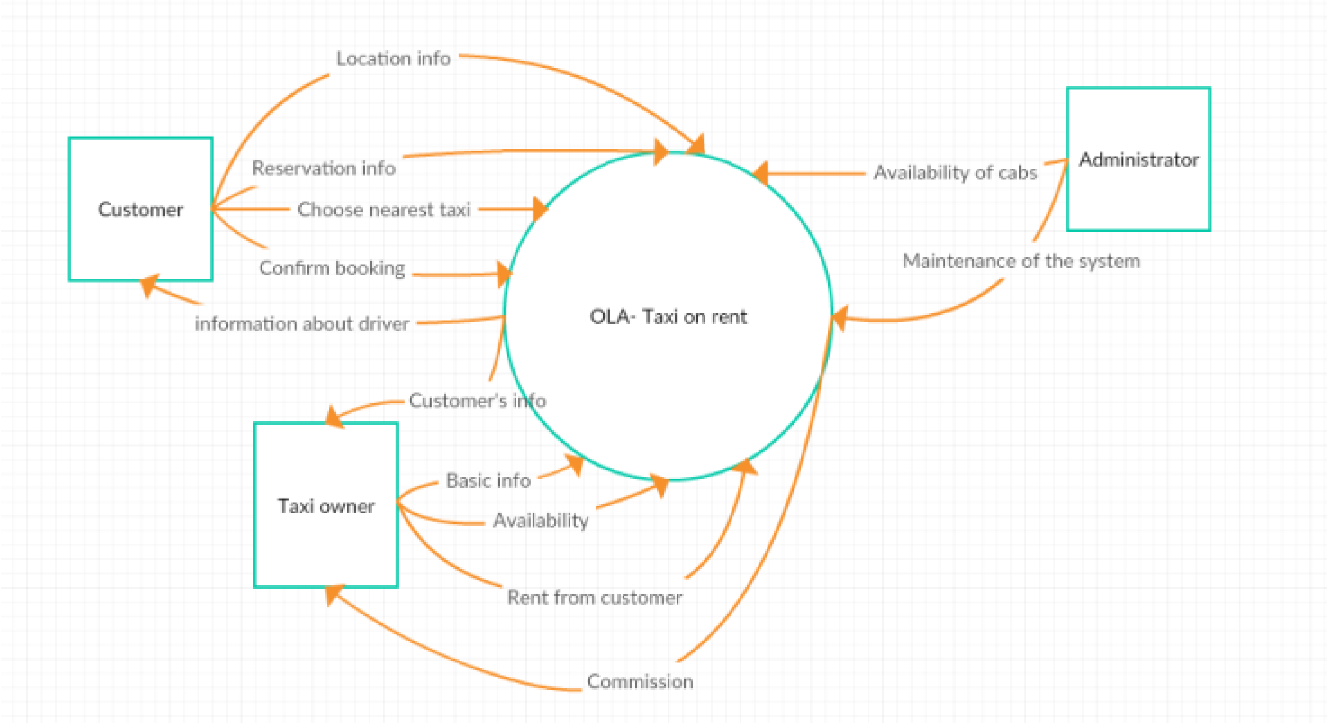
Entity

Terminator - external entities that interact with your system.

⟶

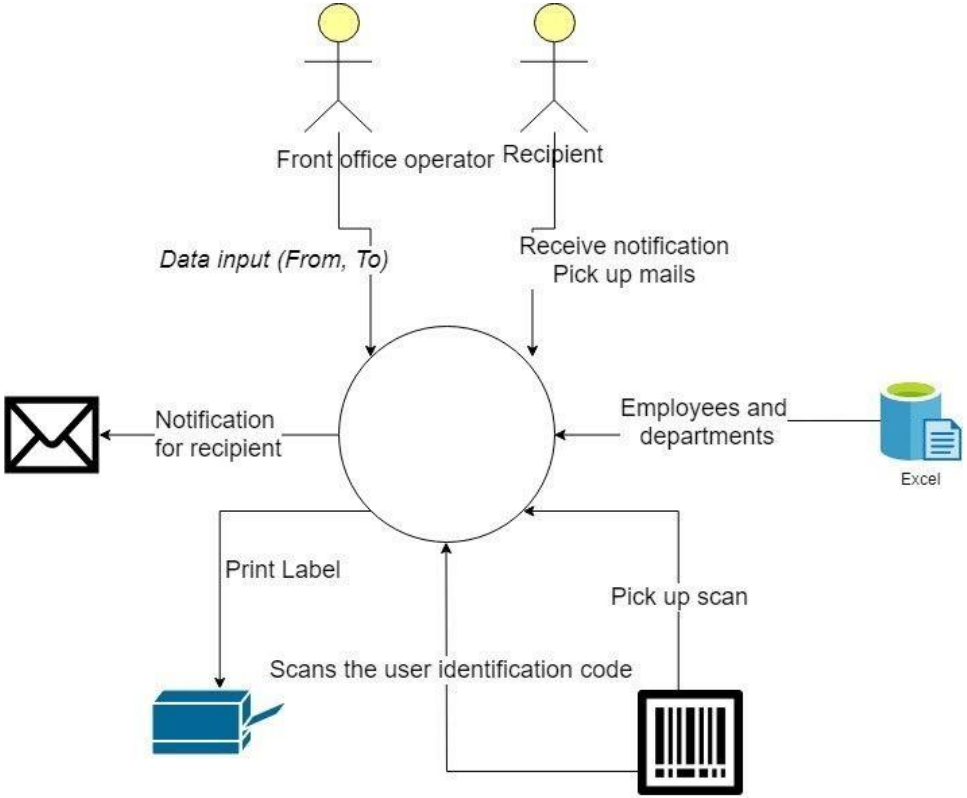
Data Flow - the information transferring between different parts of the systems, arrows representing direction of data

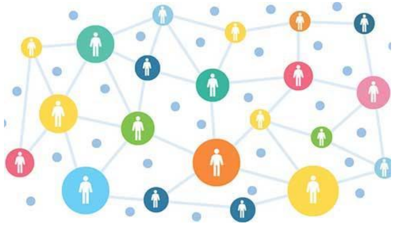
Standard example



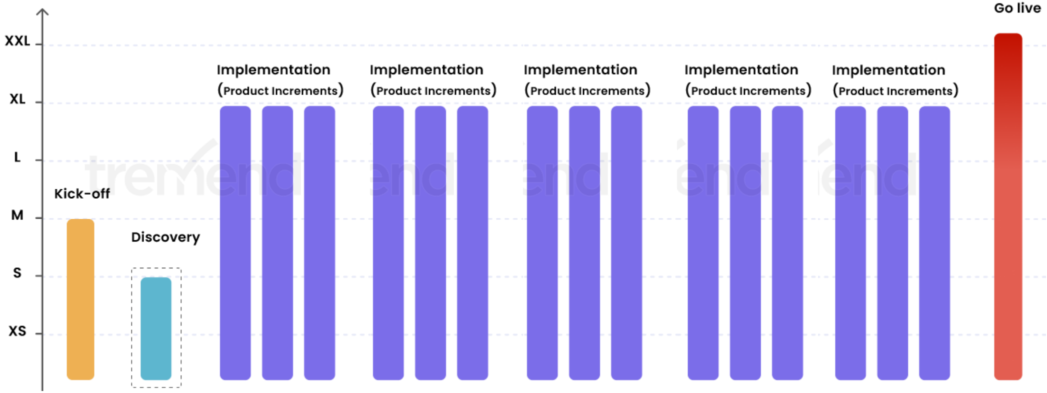
“Enhanced” example

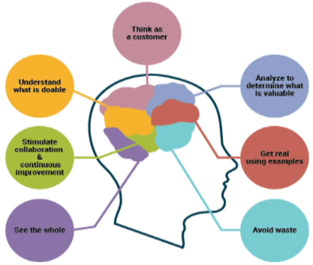
Context diagrams with customised representation, which evolved to include also the integrations within the system.



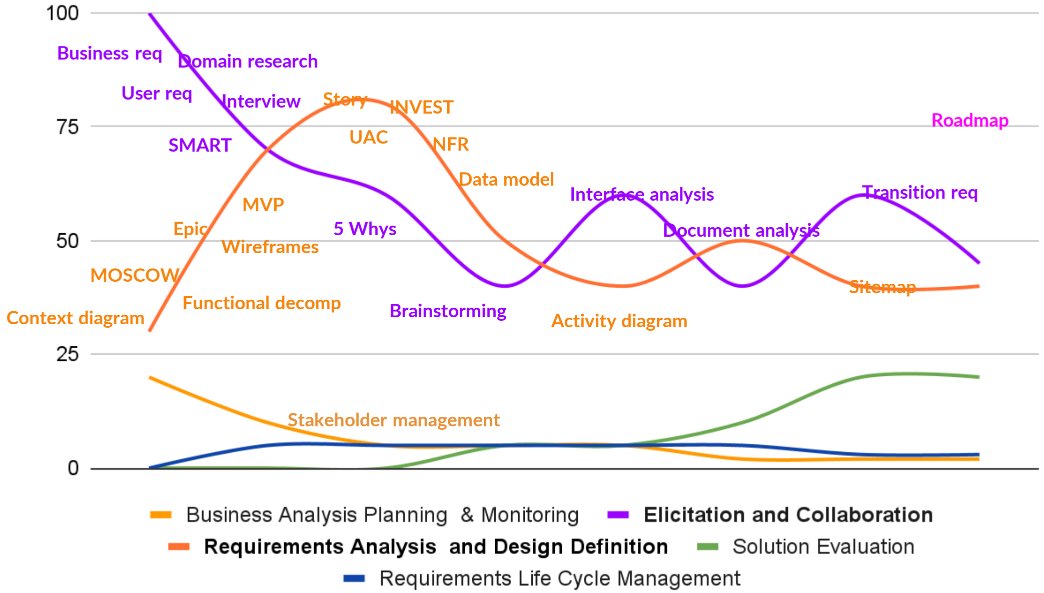


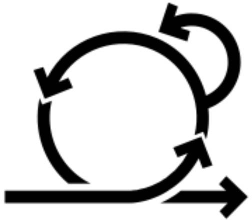
Requirements & Stakeholders




Think as a customer
Understand what is doable
Analyze to determine what is valuable
Stimulate collaboration & continuous improvement
Get real using examples
See the whole
Avoid waste

Mindset





Agile