UNIVERSITATEA DIN BUCUREȘTI

VIRTUTE ET SAPIENTIA

tremend

A company of publicis sapient

Course 7
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

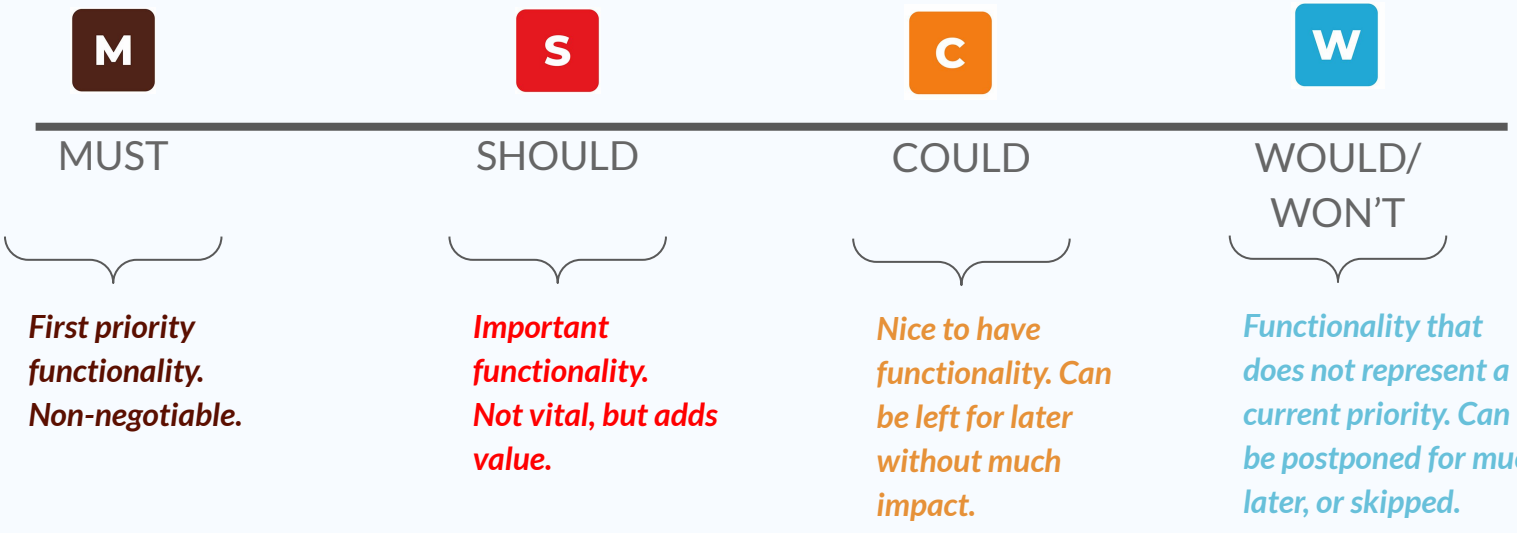
Delivering just enough: MVP & MMP

Visual representation

Requirement traceability and management

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.



Roadmap

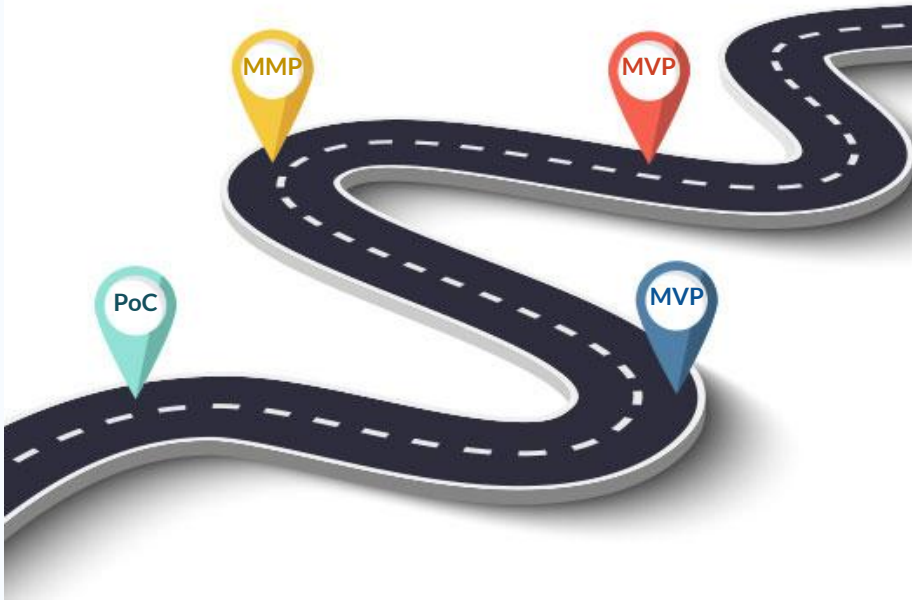
The product strategy describes how the long-term goal is attained; it includes the product's value proposition, market, key features, and business goals. The **product roadmap** shows how the product strategy is put into **action by stating specific releases with dates, goals, and features.**

Roman Pichler

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.

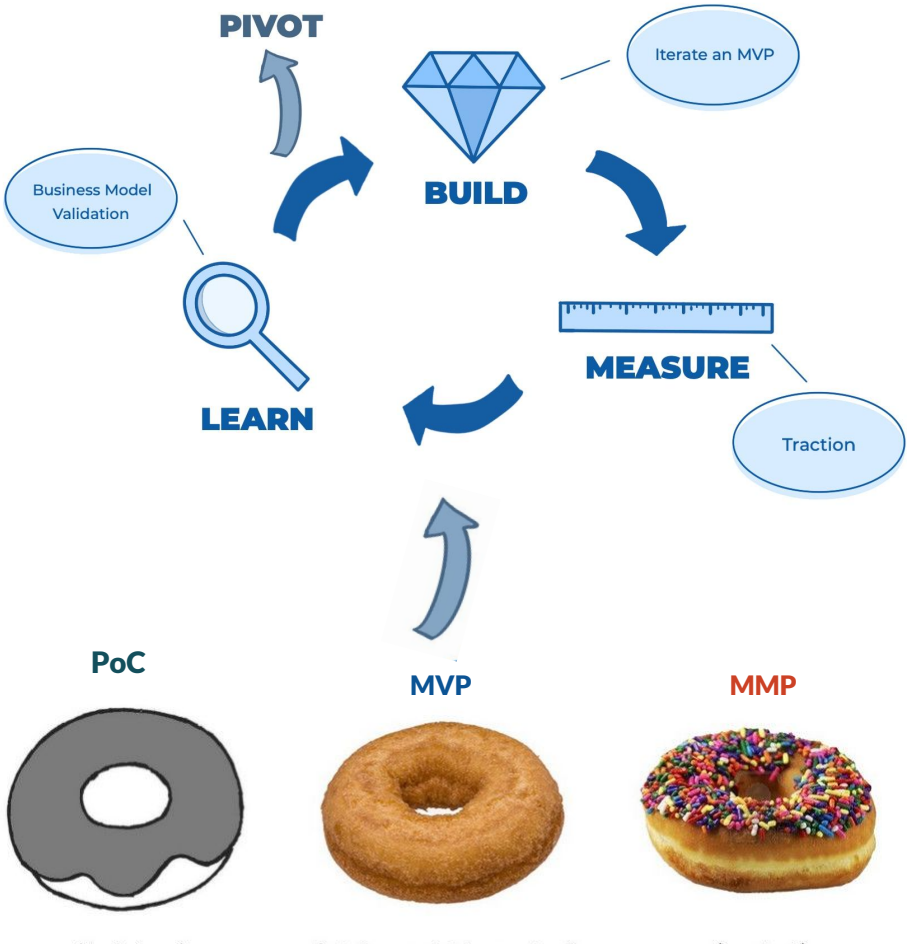


Why define an MVP?

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.

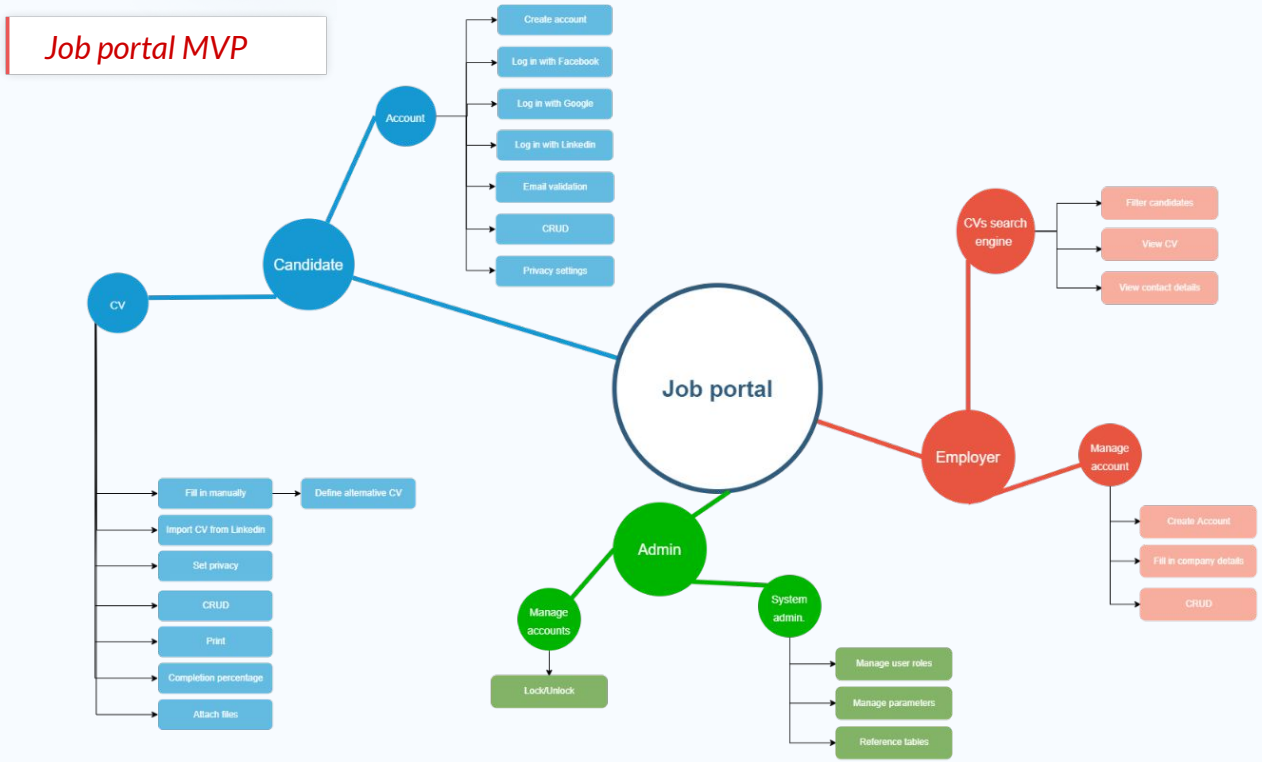


MValuableP

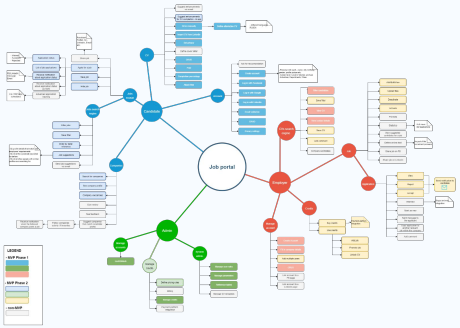
Mind that while the MVP will require the minimum effort, the effort must be invested in valuable functionalities. They may not be fully baked, but they must be valuable. Otherwise, the effort, whether minimum or not is wasted.

Job portal MVP - example

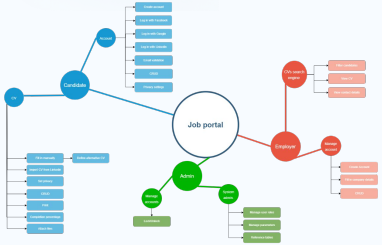
Job portal MVP



Job portal - full functionalities list



Job portal MVP - example



Phase no	Phase scope	Oct	Nov	Dec	Jan	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**						▼ ▼	▼	▼ ▼	▼ ▼	▼	▼ ▼	▼

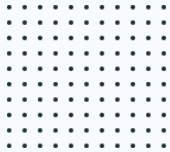
Visual representations

UML diagrams (activity/state diagram)

Context diagram

Sitemap

Wireframes

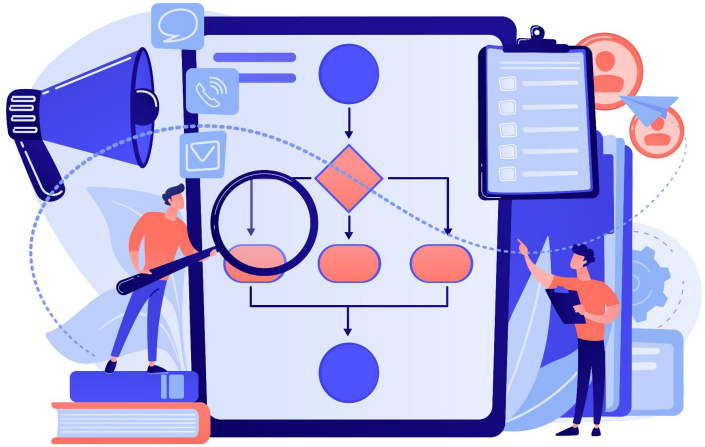


What are models and diagrams?

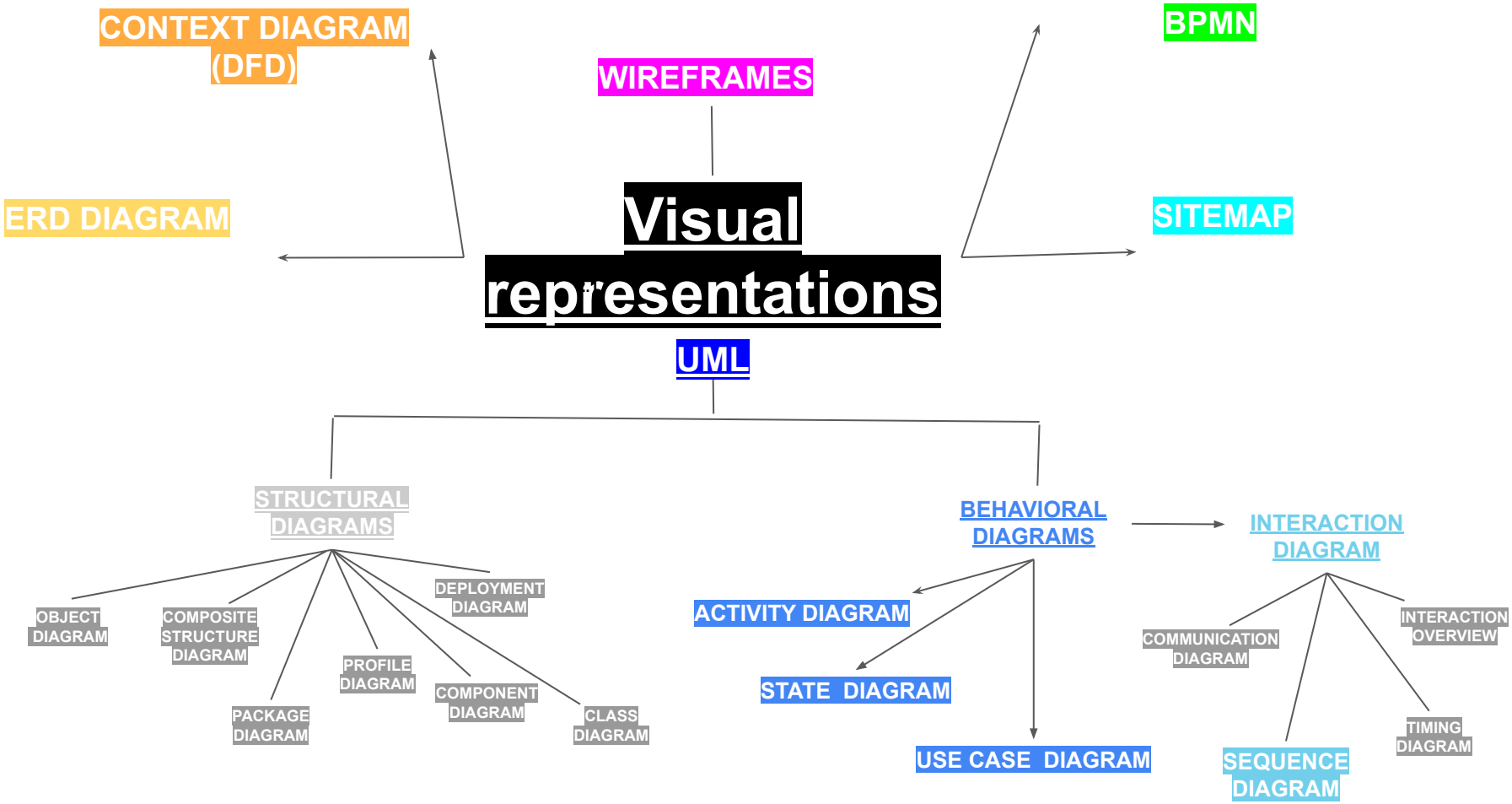
A model is an informative **representation** of an object, person or system. **Conceptual models** may be a representation of a system, concepts used to help people know, understand, or simulate a subject the model represents

Diagrams are symbolic representations of information using visualization techniques.

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



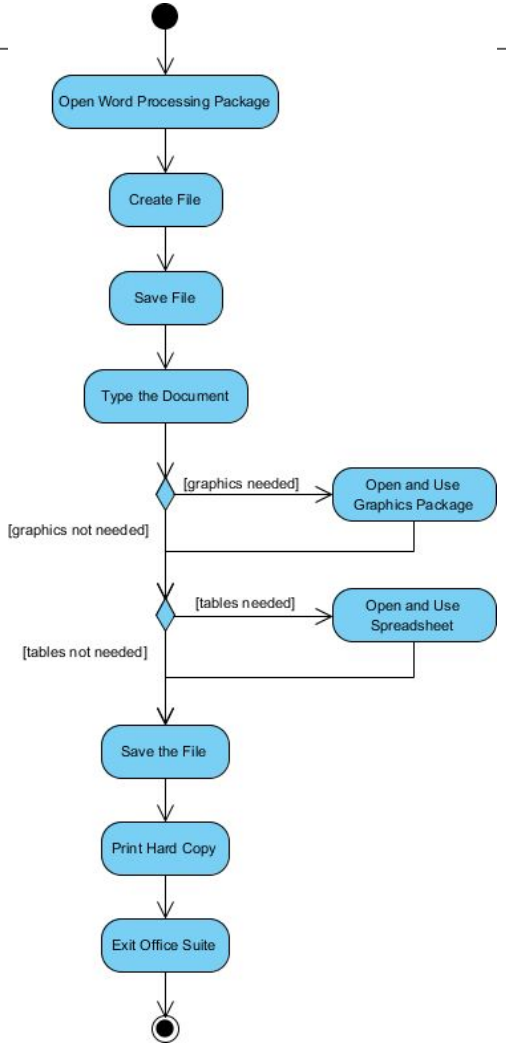
Credits to: Free Vector | Businessmen with magnifier looking at business process flow chart, business rules and regulation, main company policy, it business analysis concept illustration


CONTEXT DIAGRAM (DFD)
WIREFRAMES
BPMN
ERD DIAGRAM
Visual representations
SITEMAP
UML
STRUCTURAL DIAGRAMS
BEHAVIORAL DIAGRAMS
INTERACTION DIAGRAM
OBJECT DIAGRAM
COMPOSITE STRUCTURE DIAGRAM
DEPLOYMENT DIAGRAM
PROFILE DIAGRAM
PACKAGE DIAGRAM
COMPONENT DIAGRAM
CLASS DIAGRAM
ACTIVITY DIAGRAM
STATE DIAGRAM
USE CASE DIAGRAM
COMMUNICATION DIAGRAM
INTERACTION OVERVIEW
TIMING DIAGRAM
SEQUENCE 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



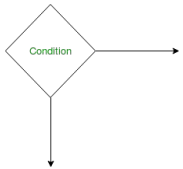
Notations

- **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

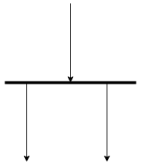
Initial State – The starting state.

Action - execution of an action.

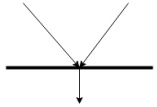
Action flow - transition from one activity to another.



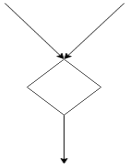
Decision node - The point of decision in the flow.



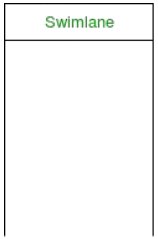
Fork - support concurrent activities.



Join – concurrent activities converging into one.

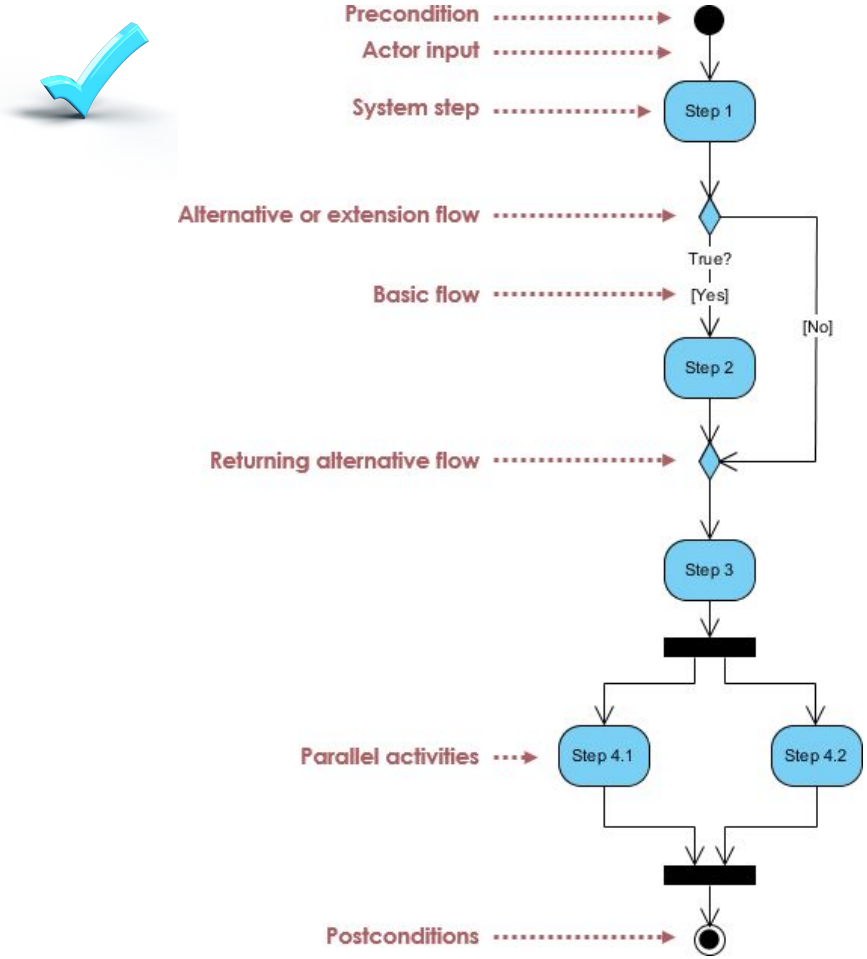


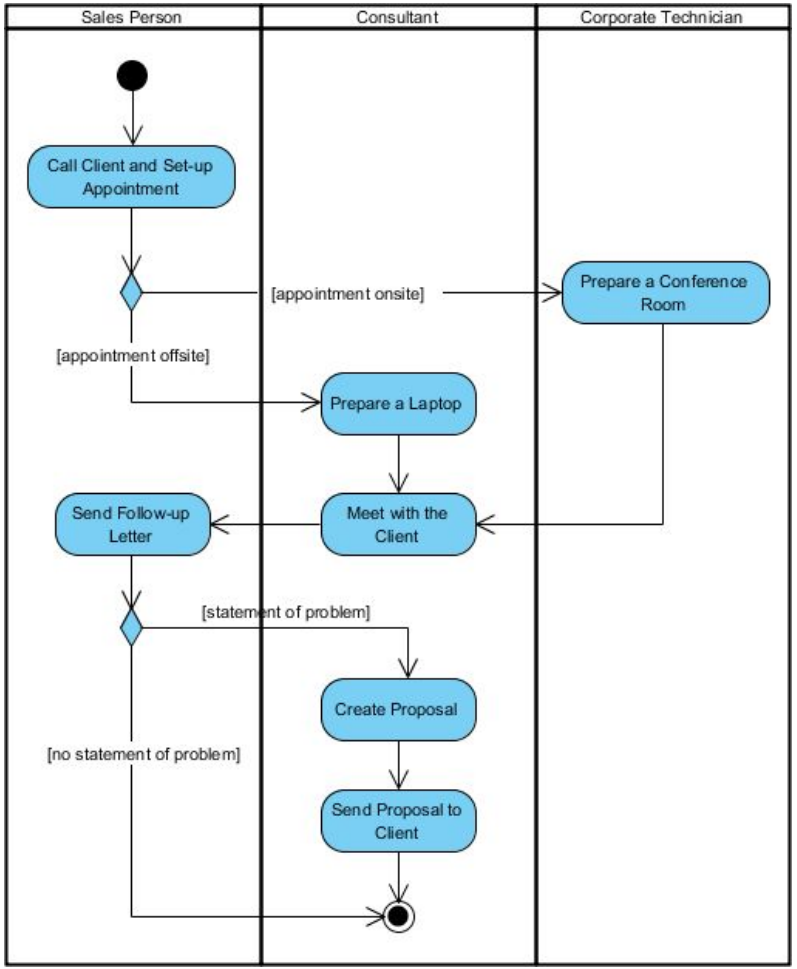
Merge – activities which are not being executed concurrently have to be merged.



Swimlanes – swimlanes for group related activities in one column.

Final State – the state in which the activity ends.


Precondition
Actor input
System step
Step 1
Alternative or extension flow
True?
Basic flow
[Yes]
[No]
Step 2
Returning alternative flow
Step 3
Parallel activities
Step 4.1
Step 4.2
Postconditions


Sales Person
Consultant
Corporate Technician
Call Client and Set-up Appointment
[appointment onsite]
Prepare a Conference Room
[appointment offsite]
Prepare a Laptop
Send Follow-up Letter
Meet with the Client
[statement of problem]
Create Proposal
[no statement of problem]
Send Proposal to Client

State diagram

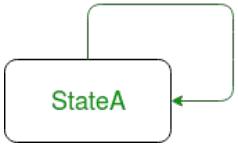
A **state diagram** is used to represent the condition of the system or part of the system at given instances of time.

- 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.

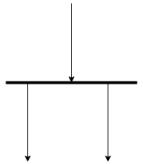
Action - the conditions or circumstances of an object of a class at an instant of time.

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

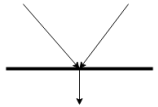


Self transition - when the state of the object does not change upon the occurrence of an event.

Final State – the state in which the activity ends.

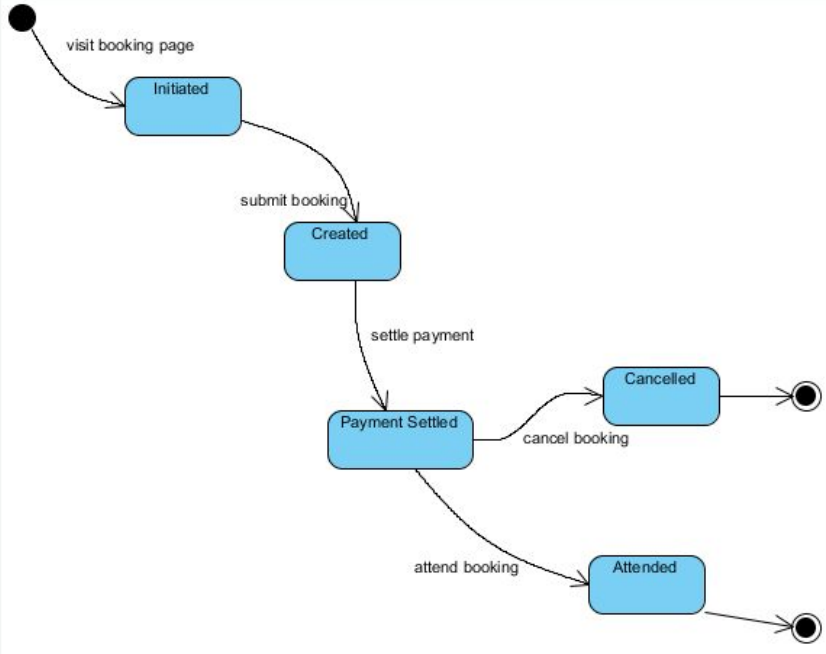


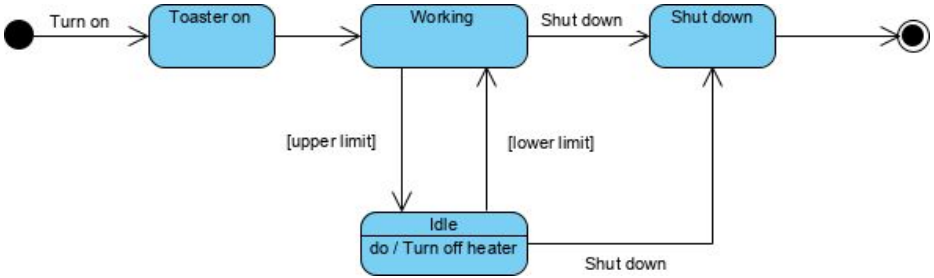
Fork - support concurrent states.



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

Examples

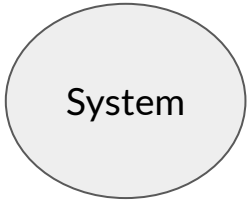

visit booking page
Initiated
submit booking
Created
settle payment
Payment Settled
Cancelled
cancel booking
attend booking
Attended


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

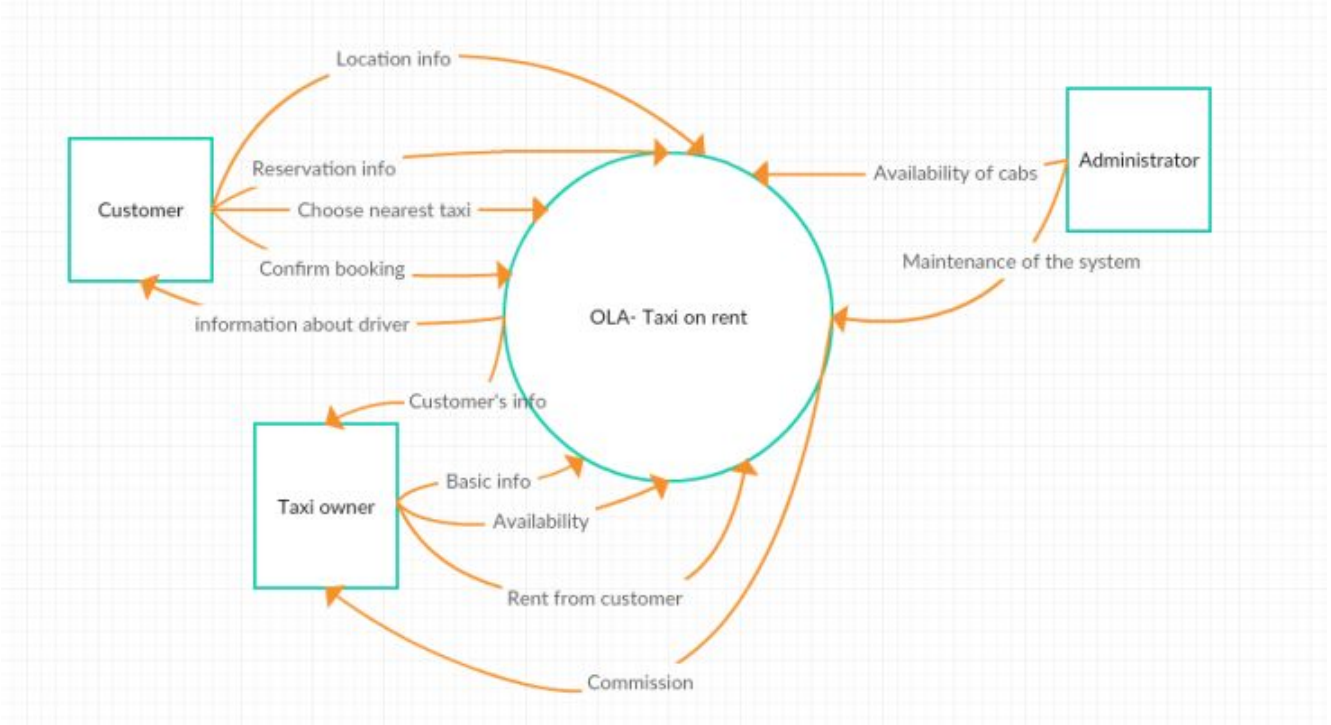


Process (in DFD 0 is the system)

Terminator - external entities that interact with your system.

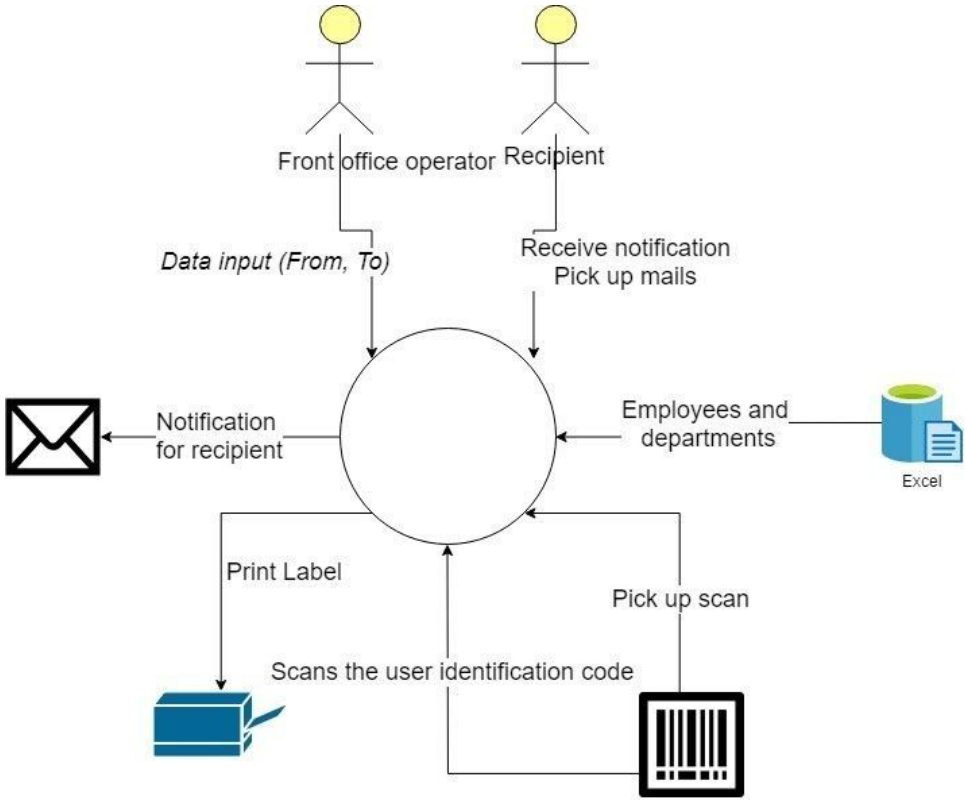
Data Flow - the information transferring between different parts of the systems, arrows representing direction of data

Standard example



“Enhanced” example

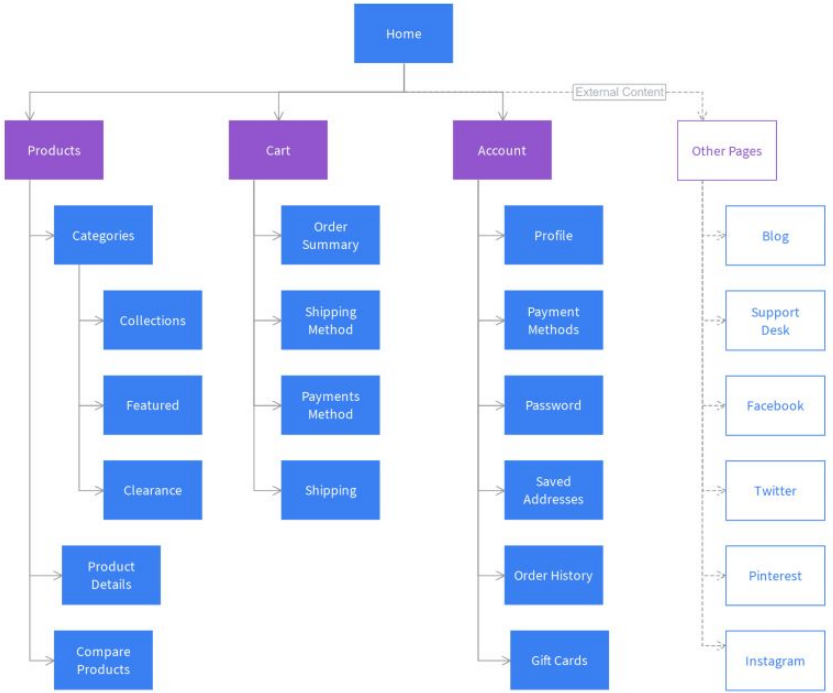
Context diagrams with customised representation, which evolved to include also the integrations within the system.



Sitemap

Organized lists or diagrams that shows connections between web pages, web page trees, and website content.

- ❏ Used to define scope of work and prioritization
- ❏ Used to validate user journey
- ❏ Easy to build and understand
- ❏ Pages are represented as blocks or cells connected by lines that represent a link or path by the user

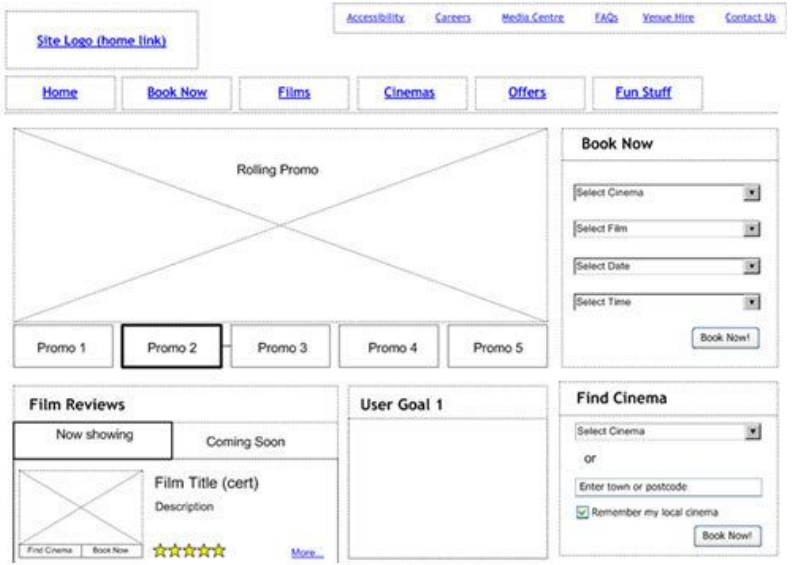


Wireframe

A wireframe is a schematic or blueprint which displays the functional elements of a website or page, typically used for planning a site's structure and functionality.

Used

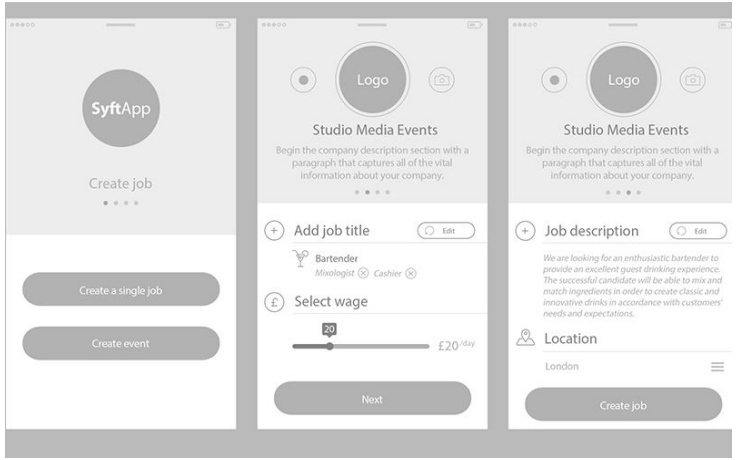
- to layout content and functionality on a page which takes into account user needs and user journeys
- in early in the development process to establish the basic structure of a page before visual design and content is added



Types of wireframes

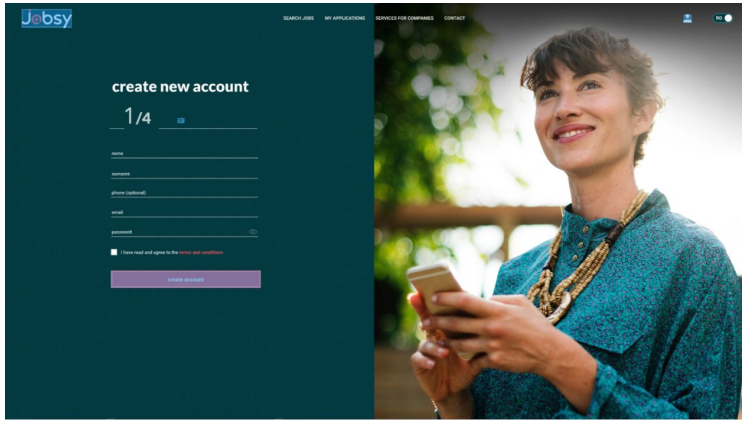
Low fidelity wireframes

- include the most basic content and visuals
- usually static (not interactive)
- sketches, the skeleton of the interface used in early communication with stakeholders
- visualize and test early concepts, requirements and design assumptions



High fidelity wireframe

- medium to high fidelity wireframes, mockups or prototypes are more complete representations of the end product
- a taste of what real UI elements might look like
- include besides the brand identity and style guide elements (logo, color palette, fonts etc.)
- are interactive
- used to communicate functionality to developers



Requirements traceability

Requirement traceability: the ability to describe and follow the life of a requirement, in both a forward and backward direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases).

Requirements traceability matrix: is a table type document that is used in the development of software application to trace requirements.

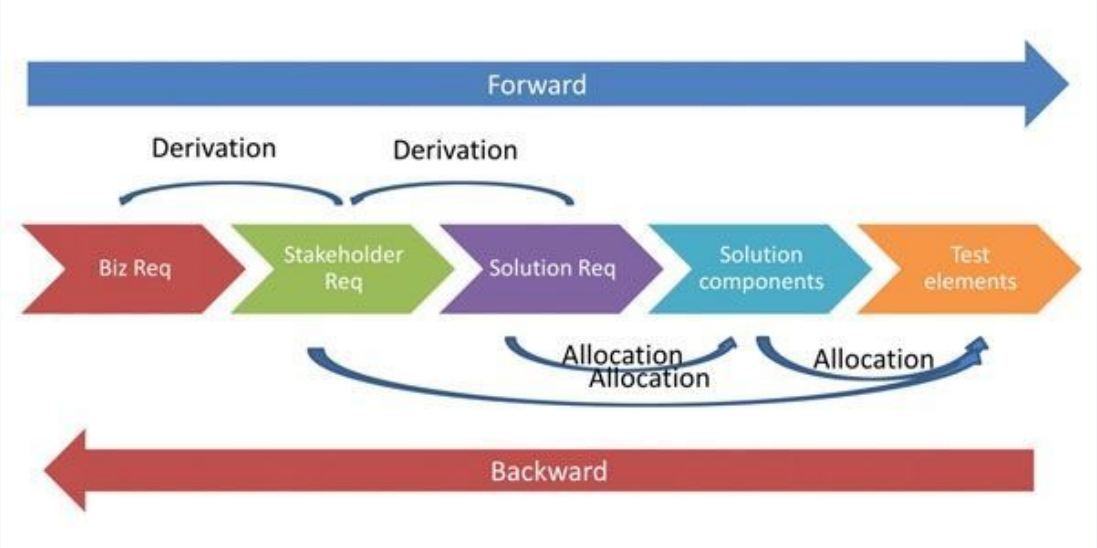
Relationships

Derive

Depends

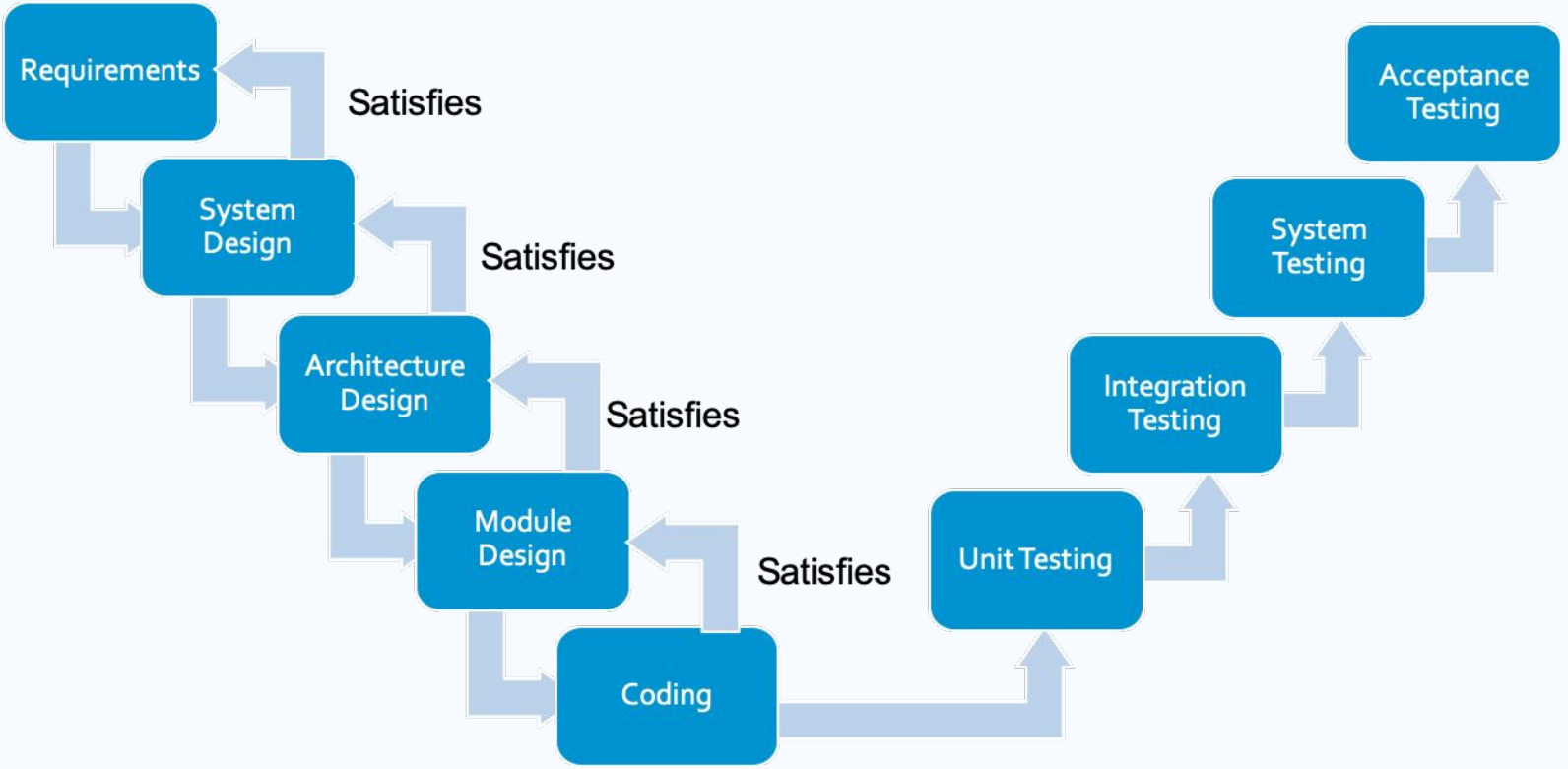
Satisfy

Validate



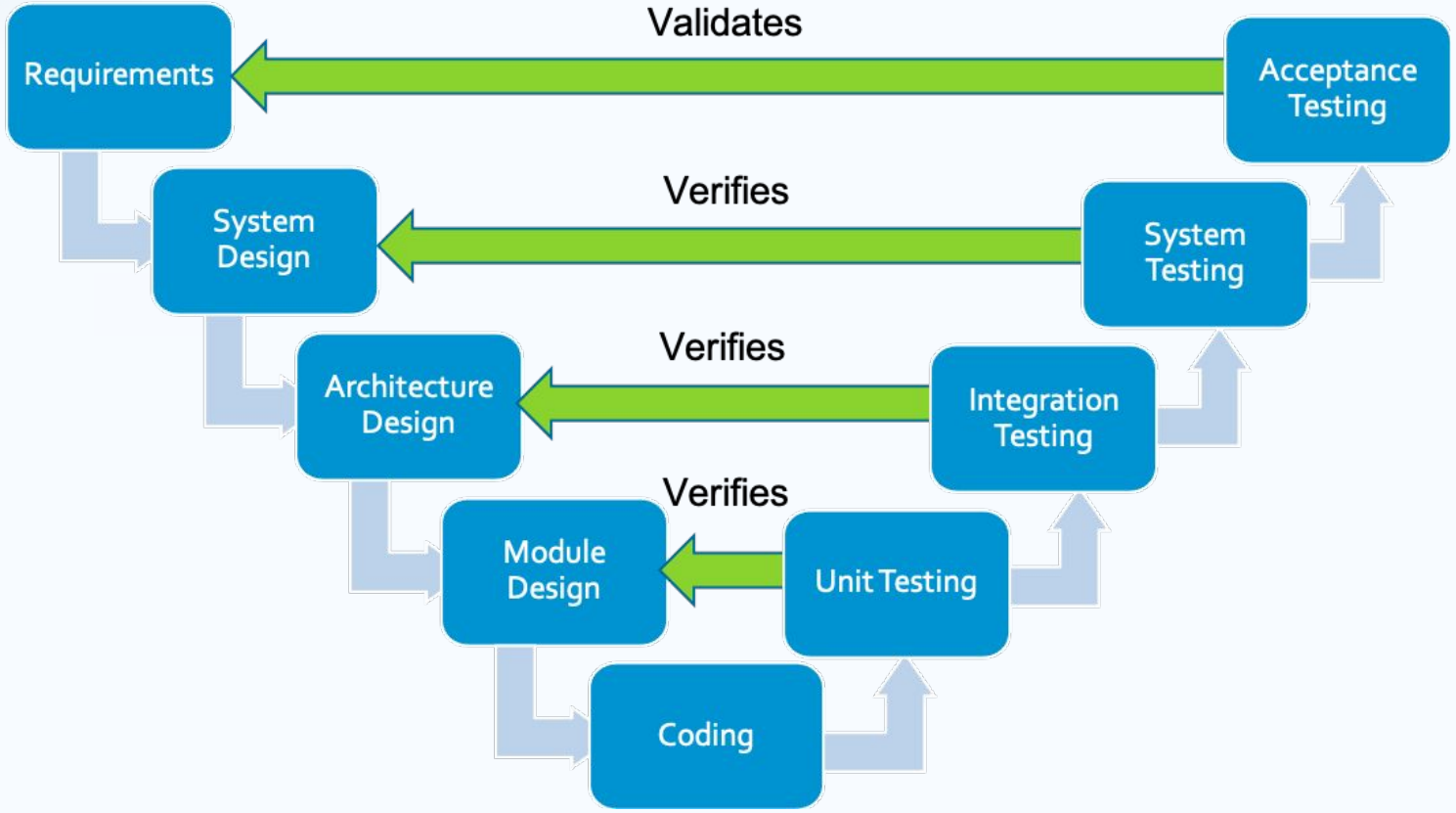
https://www.parasoft.com/blog/requirements-management-and-the-traceability-matrix/

Requirements traceability

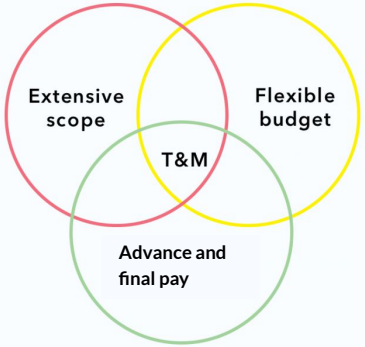


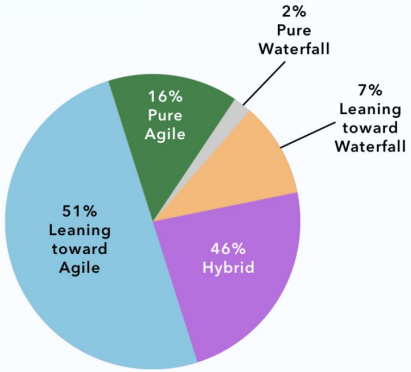
https://www.parasoft.com/blog/requirements-management-and-the-traceability-matrix/

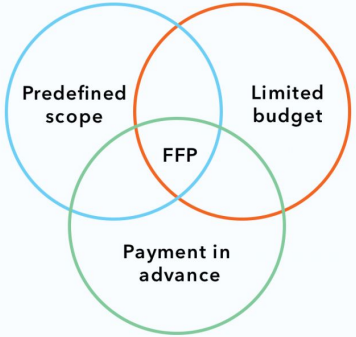
Requirements traceability



Type of project engagements







Time & material model

- More focus on business value
- More focus on prioritization, dependencies, slicing feature
- Easily adopt changes
- Closer collaboration with stakeholders
- Very transparent

Fixed price model

- Predefined scope, yet not necessarily clear, SMART, nor fixed
- Formal change management process, formal documentation
- Longer discovery phase or more towards waterfall
- More effort for requirements traceability
- Less transparency
- Less customer engagement on solution, more demands in scope
- Not flexible to change, less value delivered

https://justcoded.com/blog/time-and-material-vs-fixed-price-what-to-choose-for-your-project/

Scope management

Constant need of negotiation of scope or time!

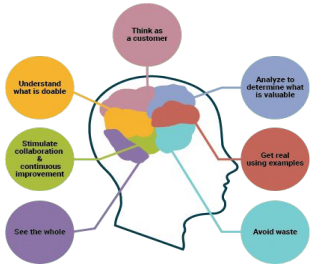
Scope
Product scope - the functions and features that characterize a product or a service
Project scope - is the work that must be done in order to deliver a product according to the product's scope (required functions and features)

Change request
Any client request that has an impact on the previously agreed scope, effort, delivery dates or budget.

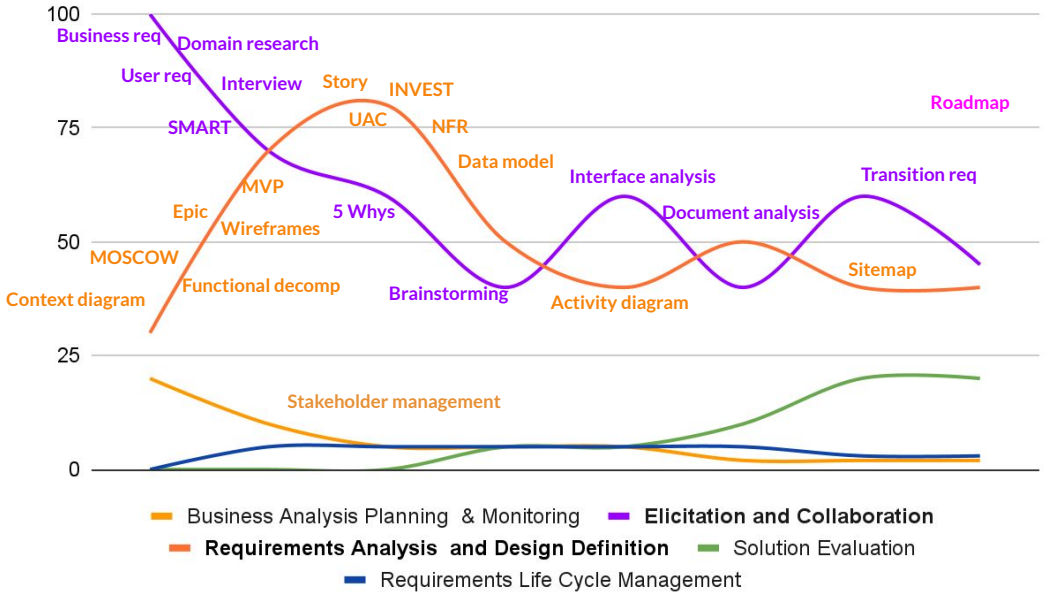
Tips to handle it

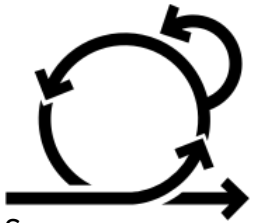
- Challenge the need and the urgency
- Impact assessment
- Trade of priorities
- Prioritize for later (FP)
- Do not commit on the spot

Requirements & Stakeholders

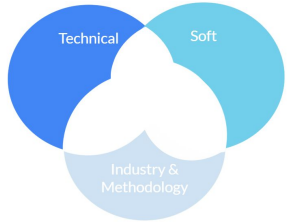


Mindset





Scrum



Skills