



MODELLING I

Software Design Approach



ME

ME

- Ahmed Yossef
- Principle Software Engineer
- Skills:
 - Software Design
 - iOS / Android development
 - Java, Swift, C#, JS, Python, C
 - 3thic4l h4cker



CONTACTS

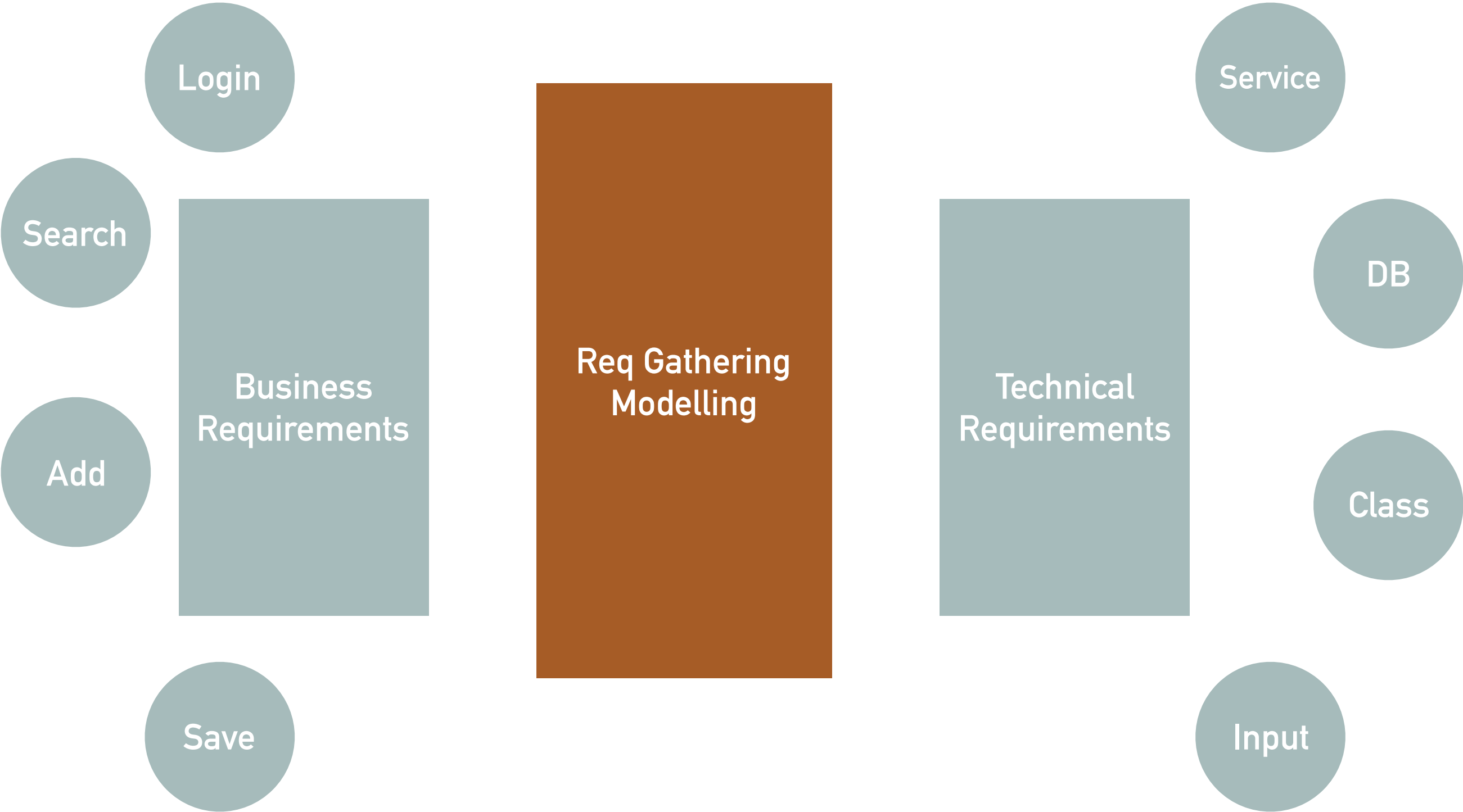
- Email: me@ayossef.net
- Email: ayossef@pitechnologies.net
- Mobile: 98756821
- Skype: ahmedyossef.21

AGENDA

- What is Modelling ?
- Why Modelling ?
- Software Design Approach
- Team, Meetings and Tools

WHAT IS MODELLING

REQUIREMENTS



MODELING

Modelling

Describing

Representing

“

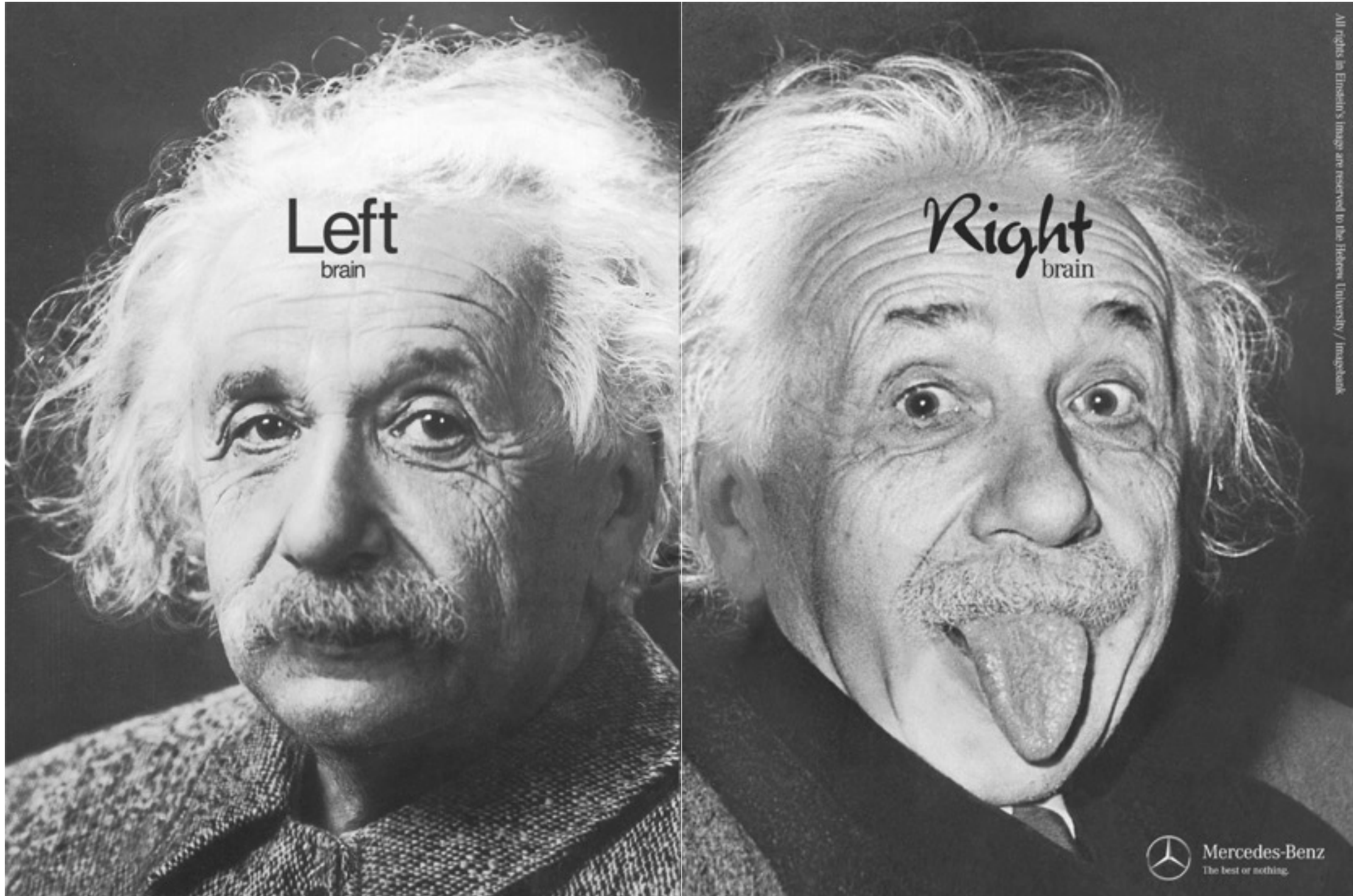
You can't exactly model what you
don't exactly know

Ahmed Yossef

WHY MODELLING ?

BRAIN

.....



BRAIN

Left brain

I am the left brain.
I am a scientist. I am a mathematician.
I love the familiar. I categorize. I am accurate. I am
analytical. Strategic. I am practical.
I am in control. A master of words and language.
Realistic. I calculate equations and play with numbers.
I am order. I am logic.
I know exactly who I am.

$$E=mc^2$$

Right brain

I am the right brain.
I am creative. A free spirit. I am passion.
Yearning. Sensuality. I am the sound of roaring laughter.
I am taste. The feeling of sand beneath bare feet.
I am movement. Vivid colors.
I am the urge to paint on an empty canvas.
I am boundless imagination. Art. Poetry. I sense. I feel.
I am everything I wanted to be.

“

It is all about correct choice !!

AhmedYossef

UNDERSTAND VS MEMORY

Processor
Quad Core 4 GHZ

RAM
100MB

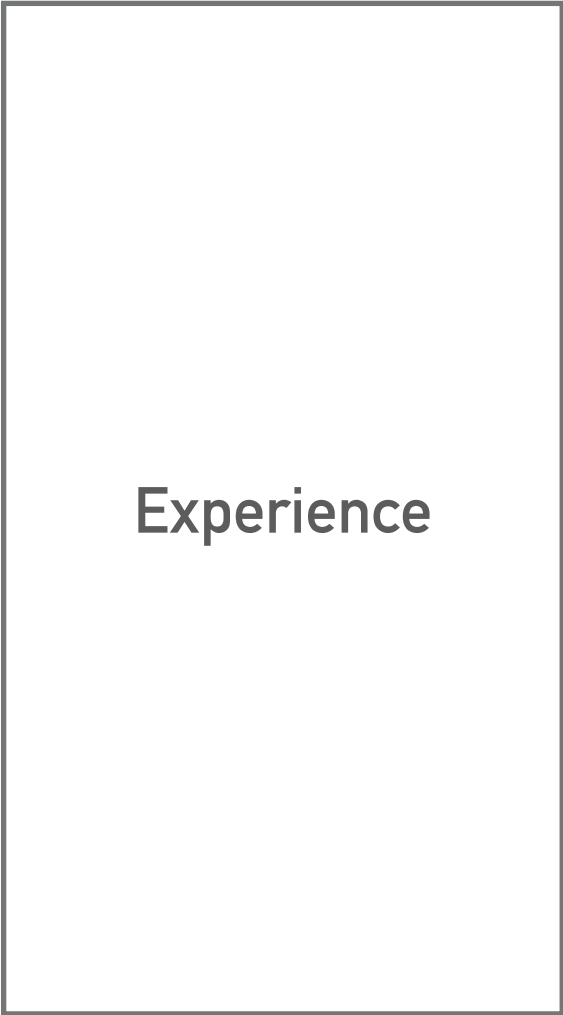
Processor
Dual Core 2 GHZ

RAM
32GB

Processor
Single Core 0.5GHZ

RAM
128GB

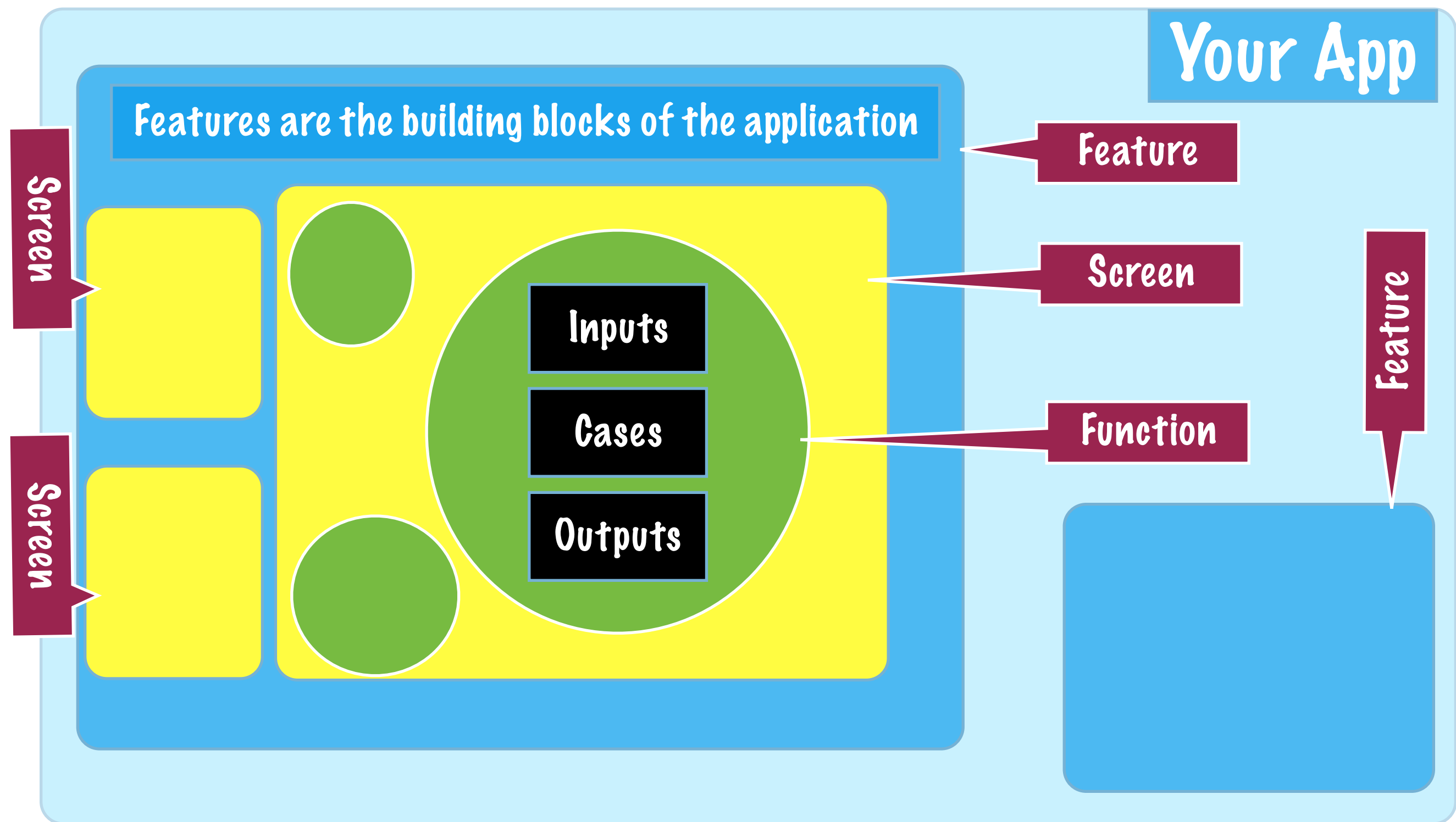
MIND POWER



DESIGN APPROACH

DESIGN TIPS

.....



EXAMPLE

APPLICATION

Application Summery

Application Description

List of features

Feature 1

Feature 2

Feature 3

FEATURE 1

About Feature

Feature Description

List of Screens

Screen 1

Screen 2

Screen 3

S11 [SCREEN 1 FEATURE1]

About Screen

List of Functions

Function 1

Function 2

Function 3

S11 [SCREEN 1 FEATURE1]

Screen Design

F111 [FUNCTION 1 SCREEN 1 FEATURE 1]

About Function

Inputs

List of cases

Case 1 - Output

Case 2 - Output

TEAM, MEETINGS AND TOOLS




GP TEAM

TEAM HOMOGENEITY

- All A+ team
- All F- team
- All developers team
- A+ and F- team

TEAM HOMOGENEITY



Team should be miscellaneous
Different skills
Same goal

TEAM WORK

➤ $1 + 1 = 3$

Expected

➤ $1 + 1 = 2$

Accepted

➤ $1 + 1 = 0$

Actually

$$1 + 1 = 3$$

- Different tasks require different skills
- Different players .. different minds .. new solutions
- More motivation
- Work review
- Focus more on similar tasks

$$1 + 1 = 0$$

- D I will work on all tasks .. me too
- D i Think .. you work
- M More arguments .. More problems
- W You discover my errors .. i will discover yours
- F I am working on all tasks .. again

SOLUTIONS

- Changing the previous attitudes
- It is business not personal
 - I don't like your idea .. but i respect you
 - You are right your idea is better than mine
 - I don't like team decision .. i will execute it
 - I discuss only when it is good for my team not to prove my point of view



MEETINGS

MEETINGS

- When ?
- Why ?
- Meetings are toxic

REASONS

- Discuss plans (Limited #)
- Make decisions ()
- Follow up
- Measure progress

AGENDA

- Leader should announce the agenda before the meeting
- Members are free to add any topics before the meeting
- Leader should state the agenda as check list of
 - tasks
 - decisions
 - deliverables

APOLOGY

- If you will be delayed, you should report before the meeting starts.
- If you will not be able to come, you should delegate your vote and work to some one else.

DURING MEETING

- No argument
- Limited discussion
- If A and B has different point of views, each one would separately explain his, then leader should terminate discussion to start voting.
- In dark situations, leader can make a decision, and you should accept it.

AFTER THE MEETING

- During the meeting, leader should have written notes about the discussions and the decisions.
- Leader should send MoMs to the team
- Leader should make the agenda and the time of the next meeting

APOLOGY FOR DELIVERABLES

- You should report any errors or blocking breaks just when they appear.
- Always report and make your leader updated
- Don't change your task without confirmation



TOOLS – VERSION CONTROL

WHAT IS VC

- Version control - Source control - Revision control
- It is a software enables you to keep history of changing in documents, code files, ..
- It can be centralized or distributed

WHY VERSION CONTROL

- No conflict
- No project_last, project_final, project_finaaaaal
- A change in a single file within the project, generates a new version with number, timestamp, creator and comment
- No need to merge copies and move code on flash memory

CHECK OUT

- It is an operation to request a copy of the project to work on
- You can check out the latest version or any older version you like

CHECK IN

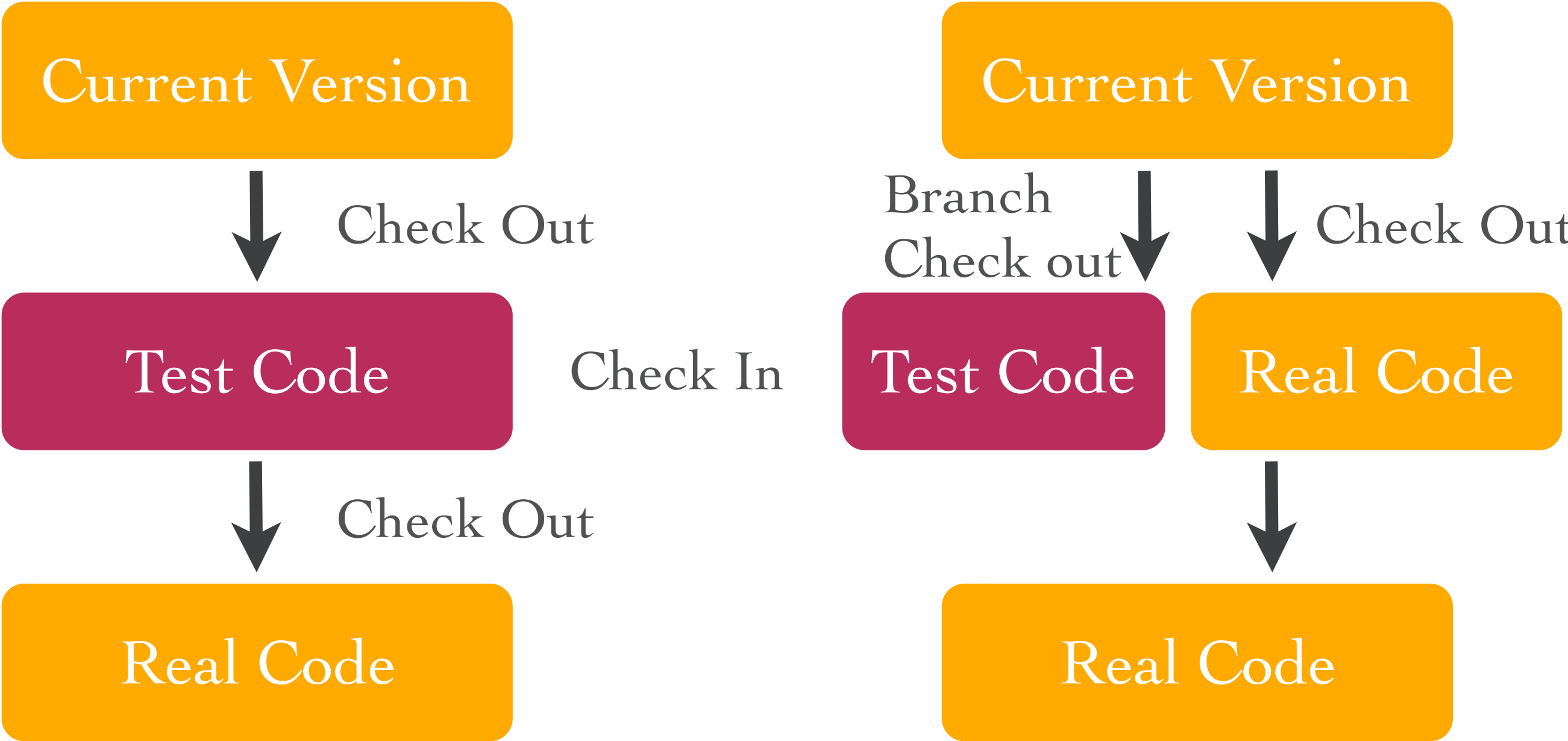
- Check in - Commit -Submit
- It is an operation done after checking out and making some modifications
- You commit your changes to save them as a new version with timestamp and comment

BRANCHING

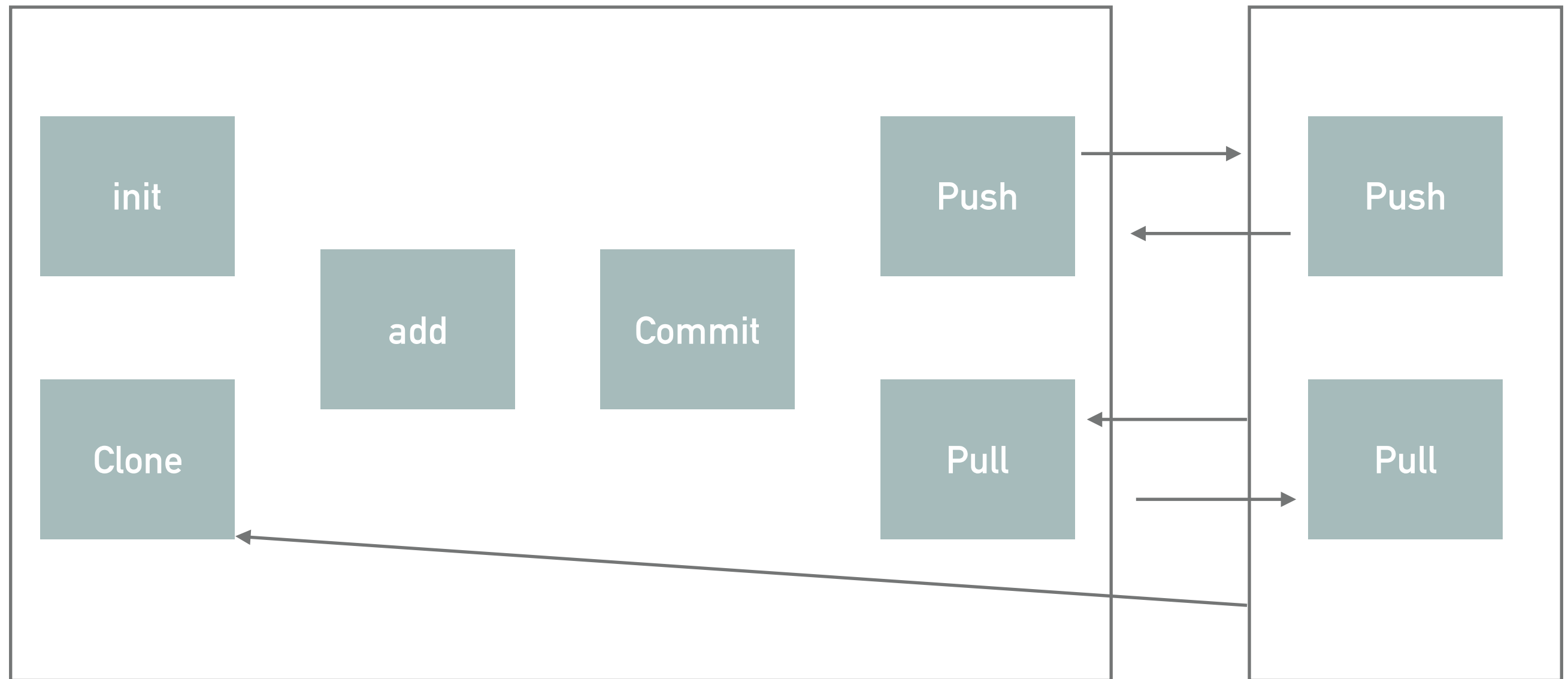
- It is an operation to make a sub-project/parallel project
- You make this operation when you want to test a new feature.

TEST SCENARIO

.....



MORE GIT



EXAMPLES

- SVN
- CVS
- <http://assembla.com/> (online)
- Git
- <https://github.com/> (online)



TOOLS – PROJECT MANAGEMENT

WHAT IS PM ?

- It is the process of
 - Diving work into phases and tasks
 - Identifying execution percentage
 - Finding critical issues
 - Finding alternative solutions

Project

Phase

Task

Task

Task

Task

Phase

Task

Task

Task

Task

Phase

Task

Task

Task

Task

PHASES

- You should divide your project into phases
- Each phase represents a percentage of total project
- Phase is a milestone, with defined date you should measure your progress

TASK

- It is the building block of the project
- Each task should have the following
 - Major phase
 - Type
 - Dead line
 - Assigned player(s)
 - Reviewer and/or supervisor

Task the basic element in the project..

If you could manage tasks .. you will manage your project .. otherwise !

TASK TYPES

Planning

- To define tasks
- Involves a lot of estimation and lack of info
- Usually involves learning tasks

Learning

- To know about something
- Undefined time
- Should has a deadline

Execution

- To get something done.
- Defined goal
- Estimated time

TASKS BOARD

	Not Started	In Progress	Done
Member 1	<div></div> <div></div>	<div></div>	<div></div>
Member 2		<div></div>	<div></div> <div></div> <div></div>
Member 3	<div></div> <div></div> <div></div>	<div></div>	

THE END ..