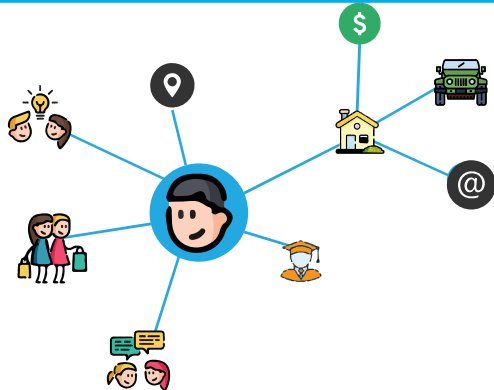




Introduction

Introduction



Our Modern life is full of complexity

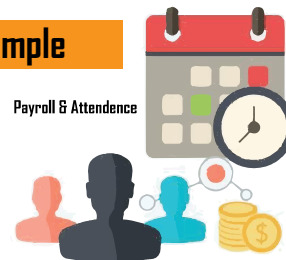


Almost all the time we have to deal with some complexities

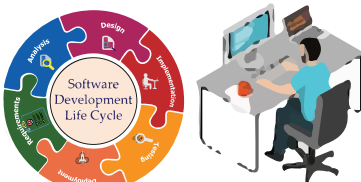


We often use software to deal with some complexities

Example



A corporate house uses Payroll & Attendance System solution for deal with employees attendance & payroll related complexities



Software is the end result of a process and continuous efforts of related peoples who develop that.



That end result must be perfect.

Introduction

Example

Continues with payroll & attendance system solution software- the software uses for-



Tracks employees
attendance



Generate Employees
Salary based on attendance
data and company policy



Generate festival,
performance, attendance
bonus for employees



If the software is perfect then all will done accurate

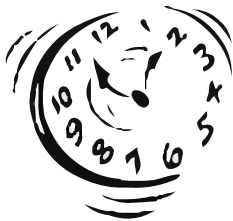


If the software is imperfect then track wrong attendance, generate wrong salary, bonus



Result: Wrong salary = employee satisfaction loss, company revenue loss

So the bottom line is - software need to perect & useful- else it will be waste of our valuable



Time



Money



Effort



Clients become disappointed



Developers become frustrated

Introduction

A good software will



Solve business problems



Reflect the real
business domain



Meet clients needs



Make clients happy

Challenging

Making such software is not so easy- it's very challenging.



THE
STANDISH
GROUP

But developing successful software is very challenging- according to Standish Group only 16% softwares are considered as successful and rest 84% software somehow due to different reasons



■ Passed = 16% ■ Failed = 31% ■ Challenged = 53%

Reference: <https://www.javiergarzas.com>

Some important factors behind such massive failures are-



Incomplete
requirements



Lack of user
involvement



Unrealistic
Expectation



Change of requirements



Requirements change



Software doesn't
needed anymore

Introduction



If we look at those factor's of failure then we'll see that all failures are happened due to lack of collaboration between development team and domain experts.



How can we be sure that only collaboration between business peoples and developers are act as key reason behind all those factors?



Okay let me explain in details-



Incomplete requirements



Why requirements remain incomplete?- Because actually business peoples are not sure about what their needs and wants within the soft-ware- that's are one and only reason for the incompleteness of requirements.



If the development team had collaboration with business team and both team jointly explore the business process of the domain then the vision related to problem domain were clear the business team - result they had the precise requirements.



Lack of users involvement



This point clearly saying that users needed to involve during development time to make software succeessful- I don't think further clarification needed for this point.

In particular, users will not intended to get involved - in such case they need to be convinced that they will be the beneficiaries if they are involved. Still if they don't want to get involved, they can be involved with the help of influential people in the company.

Introduction



Unrealistic expectations

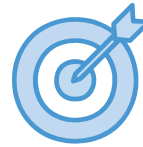
Business peoples may have unrealistic expectations- it's very normal.
If development team has good relation with business peoples then development team will help them to understand the-



Technological Barriers



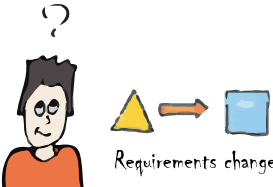
Business constraints



Goal of Developing Software

- +. Automate the business proces,
- +. Reduce complexity
- +. Get competitive advantages

Once the business peoples have proper understanding then their expectations will also become realistic



Change of requirements

Change of clients requirements are also very normal in software development.

Now the question is- why requirements are changes?

Answer is very easy- business peoples doesn't have clear vision about what they need

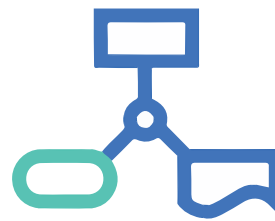
So highly recommended to-



Work in collaboration
between dev team &
domain experts



Explore the business
process in collaboration
with development team



Identify rules & logics
related to the business



Clear the vision
of developing
the software

Once they have clear vision about the product- change of requirements will no or very less