

Page No.			
Date			

Name :- Kausar Subhedar

SEPM

## UNIT - 1

### → Software.

- Computer Software is a product that software professional build and then support over a long term . It consists of programs that execute within a computer of different sizes , different architectures.
- The content that is presented inside the computer program is executed
- Descriptive information in hard copy , soft copy is generated may encompass electronic media.
- The Software engineering encompass a process a collection of methods , array of tools which allows professional to build high quality computer Software.
- Software engineer build and support software , virtually everyone in this world uses the software directly or indirectly
- Software is important because it effects nearly every aspect of our life . It has become important in our culture commerce and everyday activities software engineering is important because it enables to build complex system in a timely manner with high quality.

3-8-2092

### \* Development Scenario -

- professionals built the computer software like building any product by applying agile adaptable process which leads to high quality result that meet the needs of people that use the product.
- We apply a software engineering approach
- The work product is the set of program, data which is required to build the software. The work product is also the information that makes the users world better
- The process is to select the set of idea that are applicable to the software that we build and we apply them to the world

### \* Issues

- why does it takes so long complete the software?
- what are the development cost?

- why can't we find all errors before we give the software to the customer ?
- why do we spent so much time and effort in maintaining the existing program ?
- why do we continue to have difficulty in measuring the progress as the software is being developed and maintain ?

~~4-8-22~~

- A software is set of instructions i.e. computer program that when executed provide desired feature, function and performance.
- It consists of data structure that enable the program to adequately manipulate the information.
- The descriptive information is generated that describe the operation and use of the program.
- Software is developed, engineer it is not manufactured
- Software does not wear out

- Although, the industry is moving towards components based construction, most of the software continues to be custom build
- There are some similarity between software development & hardware manufacturing but the two activities are different
- In both activities high quality is achieve by good design
- Both the activities dependent on people.
- Both the activities require construction of product but the approach are different, the work accomplished is entirely different  
The software is not susceptible to environment that cause the hardware wearout.
- Undiscovered error will cause failures in early stage of program. The implication is clear, software does not wearout but get does deteriorate, software can undergo change. It is likely that errors will be introduce causing the failure.

- rate to rise.
- Another aspect of wearout is difference b/w hardware & software than a hardware component is wearout there are known Software Spare part. Every software failure indicates that an error in the design, an error in the process to which the design was transmitted through machine executable code
  - A software component should be designed and implemented so that it can be reused in many different programs
  - It enables the program to create new application using the reusable code. The data structure and processing detail required to build the interface are contained within a library of reusable components.
  - The reusability allows a programmer to truly focus on innovative elements of a design i.e., parts of the design representing something new.

## \* Software application domain (region)

→ The different application domains are System Software, Application Software, Engineering, Scientific Software, Embedded Software, product line software, Web application, artificial Intelligence

→ Advanced Scenarios.

1) open source.

2) Networking, Net Sourcing

3) ~~Net~~ sea open world computing

→ System Software is a collection of programs written to <sup>new</sup> service other programs

→ Application Software it is a stand alone program which solve a specific business need

→ Engineering / Scientific software has been characterised by complex number crunching algorithm.

→ Embedded software decides reside within a software product and it is used to implement and control feature and functions for the

end user and the system.

- product line software it is design to provide a specific capability or used by many different customers.
- Web application it is called web apps this are network centric software. The category spans a wide range of application.
- Artificial Intelligence software. it makes the use of non numerical Algorithm These are used to solve complex product problems.
- Open source :- It consist of IT deal with a going trend that results in distribution of open source software for system application so that many people can contribute to its development
- Networking / NetSourcing :- It deals with WWW which is becoming a computing engine and a content provider.
- Open world computing it deals with the rapid growth of wireless networking, distributed computing

8/08/22

Page No.	
Date	

## \* Legacy Software

- Generally legacy software systems were developed decade ago and have been continually modified to meet the changes and business and computing platform.
- The software system must be adopted to meet the needs of new computing environment. The software must be enhanced to implement new business requirement. The software must be extended to make it interoperable with other modern system.
- The software must be architected to make it viable within a network environment.

10/08/22

## \* A Software Process

- A process is a collection of activity, action & task that are performed when some work product is to be created.
- An activity strive to achieve a <sup>range</sup> <sub>bound</sub> of

Objective domains i.e. stakeholder, communication, size of the project, complexity of the efforts, degree of rigor, architectural design, etc.

→ An action consists of set of tasks which produce work product.

→ A task focus on small <sup>but</sup> well defined objective.

→ A process framework establishes the formation of complete software engineering process.

It also deals with ~~com~~

- 1) communication - Important to communicate & collaborate
- 2) modeling - It creates a sketch to understand a big picture
- 3) construction - It deals with <sup>program &</sup> code generation, testing
- 4) deployment - It is dealing with the product delivery to the customer
- 5) planning - It creates a map to guide the team

→ The Software engineering framework activities are supported by

- 1) Software project tracking → To control <sup>To assess progress of</sup> software development
- 2) Risk Management → To analyze the risk during the software development
- 3) Software quality assurance <sup>Technical</sup> → To ensure software quality
- 4) Practical review → To remove the technical issues
- 5) Measurement → To analyze process, product measures, needs of customer
- 6) Software configuration management → It manages the configuration effect of change

- 7) Reusability Management  $\rightarrow$  reuse  
8) total product preparation , production.  
 $\Rightarrow$  It deals with creating models, documents, lists, forms, etc

19/10/08

### \* Professional's Myth Software Myth

Myth 1 :- Software engineering will make us to make create unnecessary documentation & slope us down

Reality 1 :- Software engineering is not about creating documentation it is about creating quality product , to reduce rework , faster delivery time

Myth 2 :- The only deliverable for product for a successful project is the working program

Reality 2 :- A working program is only one part of a software configuration.

A variety of work product include document, model , plan, etc.

Page No.	
Date	

Myth 3 Until we found program running there is no way to access its reality [one of the most effective way for Software quality assurance can be applied using the technical review] Reality 3

Myth 4 Once we write the program & get into work our job is done

Reality : It is said that the sooner you begin writing code the longer it will take to get done, 60-80% of efforts done during software development are expanded after it is delivered to the customer.

19/8/22

Page No.		
Date		

## Customers Myth

Myth A :- A general statement of objective is sufficient to begin writing a program

Reality A :- While the software is developed only through effective and continuous communication bet" customer and developer so

Myth B :- Software Requirement continuously changed, but the change can be easily accommodated

Reality B :- The impact of change varies with the time  
If the requirement are requested early the cost involved is small, If they are requested late the cost impact grows rapidly

The change can cause additional resources & major design modification to be done & added.

## \* Management Myth

Myth I :- We have a books i.e. full of standards, procedures for building a software.

Reality I :- The books of standard may exist but is it used ?, Are software programmers aware of it ?, does it reflect modern Software engineering ?, Is it complete ?, Is it adaptable?

Does it focus on quality?

Myth II :- If we get behind the schedule, we cannot add programmer.

Reality II :- The software development is not a mechanical process like manufacturing. Adding more programmer to a late software project makes it more latter. The people who were working should spent more time educating the new team, reducing the productive time.

People can be added but in a plan and well coordinated manner.

Myth III :- If I decide to outsource the software to a 3rd party. I can just relax.

Reality III :- If an organisation does not understand how to manage & control software, they will finally struggle during the management of the project.

22/05/22

Page No.

Date

## \* Software Process Model

- 1) When we work to a build a product it is important go through a series of steps
- 2) A roadmap that helps to create a timely high quality result. The Roadmap that follow is called as software process
- 3) Software Engineering & their managers adopt the process according to their needs & follow it
- 4) The people who have requested a Software have a role in the process of defining, building and testing.
- 5) It provides stability, control & organisation to an activity that can if uncontrolled become unstable.
- 6) The modern software engineering process must be agile. It must demand only those activities, control, work product that are appropriate for the project and the product which is to be produced.

8/10/14

7) In a detailed level, a process that you adopted depends upon the software which you are building. One process might be appropriate for creating a software.

eg Software for Aircraft system

8) A different must be indicated, required for creation of airlines, website.

eg airlines, website.

9) from the point of software engineering the work product one program, document, data, etc.

10) There are different mechanism of software process ~~as~~ assessment, mechanism that enables the organisation to maturity of software & process.

11) The quality of time blindness, long term liability, ~~flex~~ flexibility are best indicators of efficiency of the process.

24/08/22

Page No.

Date

## \* Task set

- 1) Computer perform the many task <sup>software</sup> inside the computer
- 2) When the programmer create any software they faces many problems.

How the Software perform & after completion?

Is there any error is occur?

- A task set defines the actual works to be done
- It is used to accomplished the objective of software development
- The role of requirements and the collection, provides understanding of various stakeholders
- For a small relatively simple project the task set for the requirements might look like
- 1) Make a list of stakeholder for the project.
  - 2) Invite all the stakeholder for the discussion.
  - 3) Ask each stakeholder to make a list of features and functional requirements.
  - 4) Discuss the requirement and built a final list.
  - 5) Give priority to the requirement.
  - 6) Note the challenging areas.

- For a larger complex product different task set would be required
- Following might incorporate following

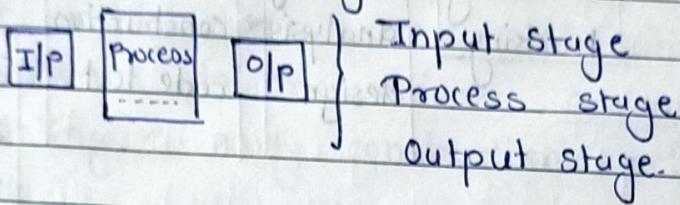
- 1) Make a list of stakeholders for the project
- 2) Discuss with each stakeholder to determine the requirements
- 3) Make a list of functions & features based on the requirements
- 4) Schedule a series of meetings regarding the application
- 5) Generate the user scenario of each meeting
- 6) Refine the user scenario based on the stakeholder feedback
- 7) Built a revise. list of stakeholder requirements
- 8) Use the quality function deployment technique to give the priority of the requirements
- 9) Package the requirement so that they can deliver on time.
- 10) Note the constraints / condition & distinction that will be place on a system
- 11) finalize the method to validate the system.

## ★ Process Pattern

- Every software team encounters problem throughout the software process
- It would be useful if solution to this problem are available so that the problems could be addressed and resolved quickly.
- A process pattern described a process-related problem which is encountered during the software engineering work, it also identifies the environment in which problem has been encountered, it also provides solution to the problem which has been encountered.
- Pattern can be defined at any level
- Pattern might be used to describe the problem associated with process model. It is also used to describe a problem associated with framework, activity (planning, project estimation).

## 2) Stage Pattern.

- It defines a problem associated with framework activity.
- Stage pattern in corporates multiple task pattern relevant to the stage.



## 2) Task Pattern.

- It define a problem associated with software engineering action for successful software engineering practice.

## 3) Phase Pattern.

- It define a sequence of framework activities that occurs within the process steps.

## \* The Waterfall Model

communication

Project initiation

Requirement gathering

Planning  
estimating  
scheduling  
tracking

Modeling  
analysis

construction

design

code test

Deployment

Delivery  
Support

feedback

### Introduction

→ During the software development there is a situation when the requirement for the problem are well understood when the work flow from the communication to the deployment.

→ The waterfall model is called classic life cycle model.

→ It suggest systematic sequential approach

→ It starts from

- It starts from customer specification, requirement gathering, planning, modeling, construction, up to the deployment.
- A variation in the above representation is called as 'V'-model.
- The model depicts the relationship of assurance to the actions, communication, modeling & early construction activity.
- The waterfall model is the oldest model.
  - a) The real project based upon waterfall model may follow sequential flow, it may sometimes cause diversion as the project team leads.
  - b) It may be difficult for a customer to give all the requirements.
  - c) The customer should be also have patience, the working version of program will not be available. then it is tested & made error free.

## 1. Communication

⇒ In the communication phase, the major task performed is requirement gathering which helps in finding out the exact need of the customer. Once all the needs of the customer are gathered the next step is planning

## 2. Planning

⇒ In planning major activities like planning for scheduled, keeping tracks on the processes and the estimation related to the project are done. Planning is even used to find the types of risks involved throughout the projects. Planning describes how technical tasks are going to take place & what resources are needed & how to use them.

## 3. Modeling

⇒ This is one of the important phases of the architecture of the system is designed in the phase. An analysis is carried out & depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the first phase & the planning done in

the second phase.

#### 4. Construction

⇒ The actual coding of the software is done in this phase. This coding is done based on the model designed in the modeling phase. So, in this phase software is developed & tested.

#### 5. Deployment

⇒ In this last phase, the product is rolled out or delivered & installed at the customer's end & support is given if required. Feedback is taken from the customer to ensure the quality of the product.

→ Today the software work in past phased. It subjects to never ending stream of changes.

→ The waterfall model is often inappropriate as it suppose linear flow of software development

→ It serve as useful process model where the situation requirement are fixed.

21/09/22

Page No.

Date

## \* Incremental Model

- There are many situations in which the software requirements are well defined. The overall process may be a linear process.
- There may be a need to provide limited set of software functionality, which can be further defined & expanded in later stages.
- We can select a process model i.e. designed to the use of the software in increments.
- The incremental model combines elements of linear & parallel processes.
- The model applies linear sequence as calendar time progresses (as schedule time).
- Each linear sequence produces deliverable increments in a manner. i.e. similar to increments produced by an evolutionary process.

- When an incremental model is used the first increment is often a basic product.
- The basic requirements are addressed.
- Many supplementary features remains undelivered.
- A plan is made for the next increment. The plan addresses the modification of the basic product to leads the needs of the customers and delivery of additional features & functionality. This process repeated following each increment until the complete product is developed.
- The incremental process model focuses on delivery of operational product of each increment. The early increments are limited version of final product but have the capability that serve the user & provide a platform for evolution.
- The incremental development is particularly useful when the staffing is unavailable for the complete implementation. early increment can be implemented with fewer people.

if the core product is well received then additional staff can be added to implement the next implement.

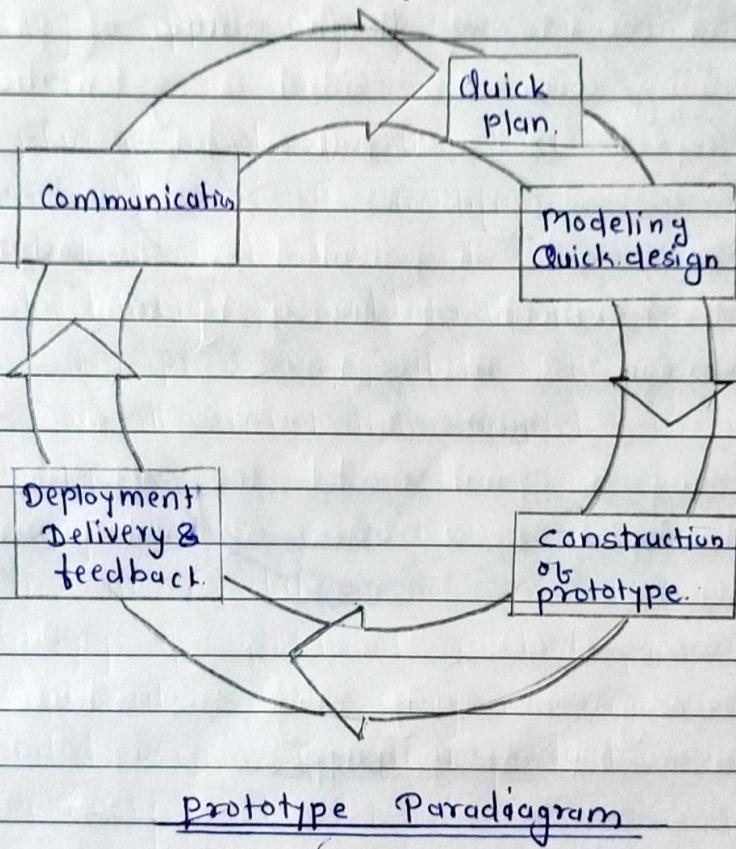
- Increment can be planned to manage technical race.

### \* Evolutionary model

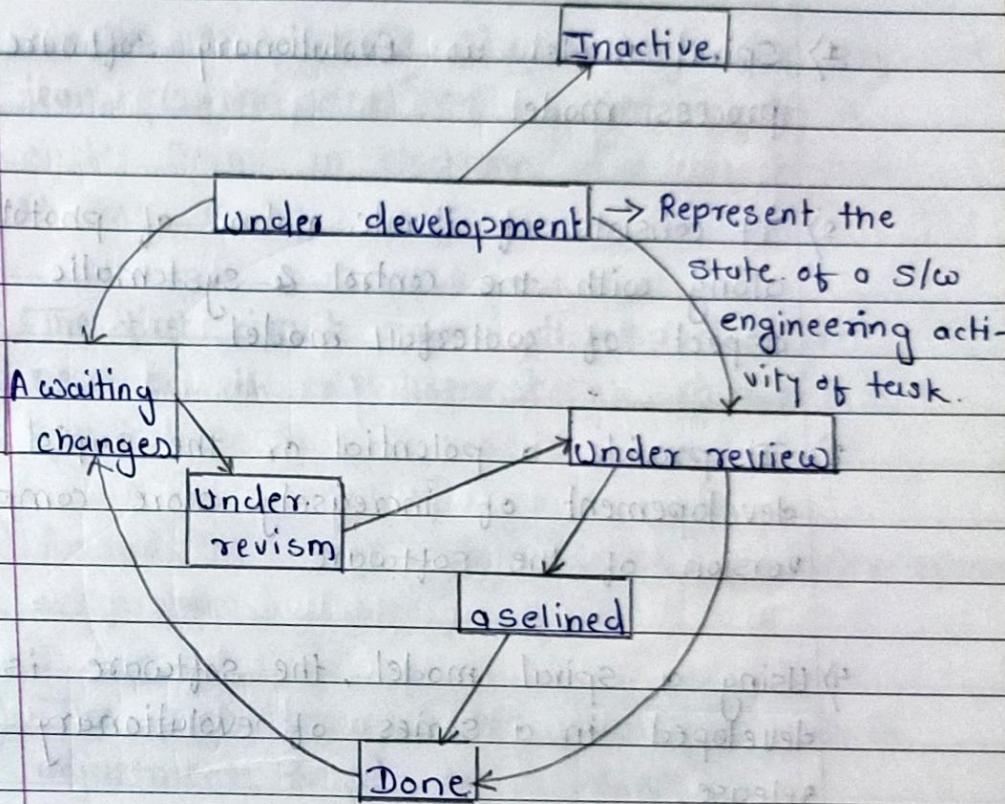
- The software over a period of time get evolve. business requirement often a change during a development process, making a straight line path to a product unrealistic, tight market deadline make competition and completion of software product impossible.
- A limited version must be introduced to meet competitive business pressure, A set of product, system requirement is well understood but the details of the product, system extension have yet to be defined.
- In this situation, we need a process model that has been designed to accommodate a

product which evolve over time.

- The evolutionary models are iterative. They enables you to develop increasingly more complex version of software.



## \* Modeling activity :-



### Current Model

## # Evolutionary process model / Spiral Model

- 1) Spiral model is Evolutionary Software process model
- 2) It consists of iterative nature of prototyping along with the control & systematic aspect of waterfall model
- 3) It provides a potential or the rapid development of increasingly more compact version of the software
- 4) Using a spiral model, the software is developed in a series of evolutionary release.
- 5) During the early stage the release might a model or prototype.
- 6) During the later stage increasingly more complex version of engineered system are produced.
- 7) The spiral model is divided into the set of framework activities defined by software

## engineering team

- 8) As the evolutionary process begin the software team perform the activity implied by a circle, spiral in clockwise direction beginning at the center.
- 9) The first circuit around the spiral might reserved in development of product specification. Subsequent passes.
- 10) Around the spiral might be used to development a prototype.
- 11) The pass through the planning result in adjustment to the project plan.
- 12) cost and schedule are adjusted based upon a feedback from customer.
- 13) The project manger adjust the planned number of interations required to completed the software
- 14) The spiral model can be adopted to apply throughout the life of computer software

26/09/22

Page No.

Date

## \* Agile Software Engineering

- Agile software engineering combines a philosophy and set of development guidelines
- It encourages customers satisfaction and early development of software, small compact size, highly motivated team, formal and informal methods and development simplicity.
- The development guideline is the focus on delivery core analysis, continuous communication between developer & customer.
- The software engineer, project stakeholders work together as a team i.e. self organizing it promotes (promotes), communication, collaboration, all stakeholders
- The modern business environment based on computer system is fast paced and ever changing.

Page No.	
Date	

- Agile software engineering represents a reasonable alternative to conventional software engineering.
- It has been demonstrated to deliver successful systems very quickly.
- Agile development may be termed as software engineering, a version having basic framework activities, communication, planning, modeling, construction and deployment.
- In agile development what the customers and software engines have similar view.
- If the team agrees that the process works and the team reduces it deliverable software, that satisfies the customer, we have achieve agility.

28/09/22

Page No.

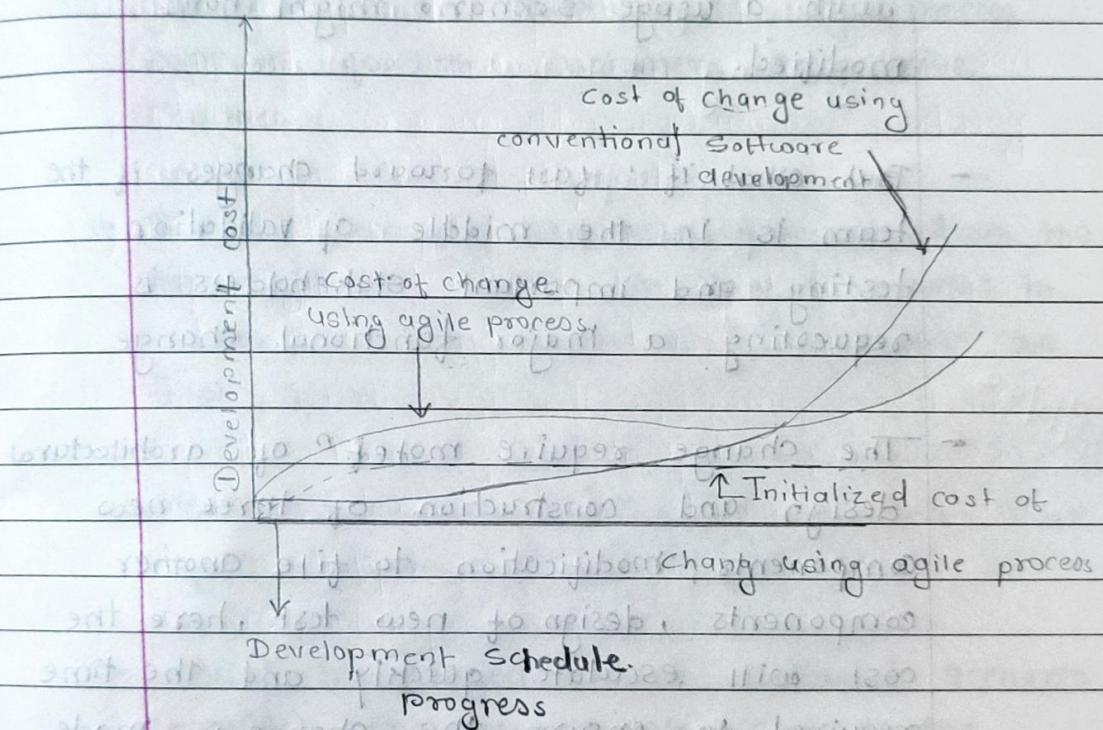
Date

## \* Agility

- It has become today's hot topic, when describing a modern software process.
- Every software should be agile. (apart from software. It is applied to every scenario)
- An agile team is a team which is able to respond to changes.
- Change is all about software development process.
- changes in the software developed, changes to the team member, changes because of new technologies.
- changes of all kinds that may have an impact on the product they build or the product they create.
- Support for changes should be built in everything we do in software, i.e. the thing

which is the heart and soul of the software.

- An agile team recognizes that the software is developed by the engineering individuals working in a team and skills of these people, their ability to collaborate, is at the core for the success of project.



- The conventional approach gains software development. Hence the cost of change increases non-linearly as the project progresses.

- It is relatively easy to accommodate a change when the software team gathering requirements.
- The cost of doing this work are minimal and the time required did not addressly effect the outer project i.e. it may deal with a usage scenario might have modified.
- But what if ' fast forward changes if the team is in the middle of validation, testing and important stakeholders is requesting a major functional change.
- The change require. modif." of architectural design and construction of three new components , modification to file another components , design of new test , here the cost will escalate quickly and the time required to ensure the change is made unintended which will create major side effect .

8/10/27

Page No.	
Date	

## A Agile Principles

- The highest priority i.e. to satisfies customer request through early and continuous delivery of software.
- To welcome the changing requirement even getting take in developments agile process supports for the customers competitive changes
- Delivery working software frequently from the time to time , weeks to weeks , months to months with a preference to shorter the time span.
- Business people and developer must work together throughout the project.
- Build the project around motivated scenario provides the requirements support the request trust them to get job done
- The most efficient and effective method of converting information with in a development team is face to face conversation.

- Working software is the primary measure of progress
- Agile process is sustainable process the sponsors developer should be able to maintain and use the project.
- Agility focus on continuous attention regarding technical excellence
- The team reflects on how to become more effective and adjust its behaviours accordingly.

12/10/22

## \* Problem Solving Ability

- Software managers must recognize that the agile team will continually have to deal with ambiguity. And will continually be changed
- In some cases the team must accept the fact that the problem we have solving today may not be the problem that needs to be solved tomorrow

## \* Mutual Trust

An agile team must become a group that exhibit trust ~~that~~ and respect ~~that~~ make them so strongly bonded as a large group is better than the <sup>individual</sup> part.

## Self Organization.

- It implies three things.
  1. The agile team organizes itself for a work
  2. The team organizes the process to best accommodate its local environment
  3. The team organizes the work schedule to best achieve delivery of software on

time.

- self organization serves to improve collaboration and boost team model
- The team select how much work it believes it can perform within the given time and the team commits to the work. nothing demotivates a team as someone as making commitments for it