

INTRODUCTION

- ⇒ What is success of a software?
 - meets the customer's needs.
 - easy to modify & use.
 - perform flawless for long time.

⇒ What if not successful?

- customer dissatisfied.
- loss of money & human lives
- may face legal consequences -

⇒ Definition of SE :-

- Collection of techniques, methodologies & tools that help with the production of.

- high quality software system
- with a given budget
- before a given deadline while changes occur.

The intangible nature of software causes problems for management in planning, estimating, scheduling & budgeting for accounting purposes.
→ not physical appearance

⇒ Types of Software Products:-

① Generic Products-

- stand alone system

- sold to anyone who wants to buy.

Ex word processor, graphics processors, etc.

② Customized Products,

- customer oriented.

- commissioned by a specific customer to meet their own needs

Ex NADRA, air traffic system, etc.

⇒ Software Cycle:-

① Process ② Modeling

③ Quality Management

④ Project Management Techniques

⇒ Software Crisis

- Term used to describe the impact of rapid increases in computer power & the complexity of the problem that could be tackled.

- cost exceeded.

- never deliver or late deliver.

- substantially modified

- unmanageable & code difficult to maintain.

49

=> Why so many issues?

① Requirement of flexibility.

- so flexible to start working with it before fully understand what needs to be done.
- changes - policies, requirement, technology.

② Intangibility / Invisibility.

- Interfaces undecidable.
- Transient hardware, software errors, race condition, compatibility issues.
- Virtualization

③ Conformity

- system usually interact with outside system.

=> Software Categories

① legacy software.

② Data mining

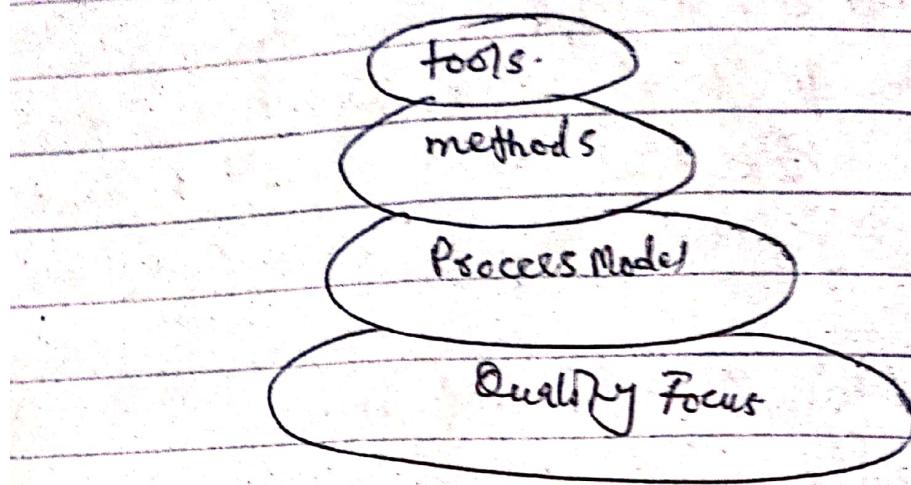
③ Grid computing

④ Content machines

⑤ Software for nanotechnologies -

70

⇒ Software Engineering Layers:



⇒ Types of Stakeholders:

① Primary Stakeholders

- internal stakeholders, are those that engage in economic transaction with the business.

Ex : Customer, supplier, creditors, employees.

② Secondary Stakeholders

- external.
- don't engage in direct economic exchange with the business.

Ex : General Public, communities, business support media, etc.

21

=> Why we need SE?

- ① Complexity ② Innovation ③ Flexibility
- ④ Invisible ⑤ Changeable.

=> Software Costs:

- more than a hardware cost
- costs more to maintain than develops.
- SE is concerned with cost-effective software development.

=> Essential attributes of good software:

- ① Reliability
- ② Dependability & Security.
- ③ Efficiency → responsiveness, processing time, memory utilization, etc
- ④ Acceptability -

=> Software Process activities

- ① Software specification
- ② Software development
- ③ Validation
- ④ Evolution. → modified to customer needs.

Embedded systems

Software + hardware,

System Processes

- A structured set of activities required to develop a software system.
- System Process involves
 - ① Specification → what system should do.
 - ② Design & Implementation → organization & implementation
 - ③ Validation → check it fulfill the customer's needs.
 - ④ Evolution → modification w.r.t customer needs.
- A software process model is an abstract representation of a process. It represents a description of a process from some particular perspective

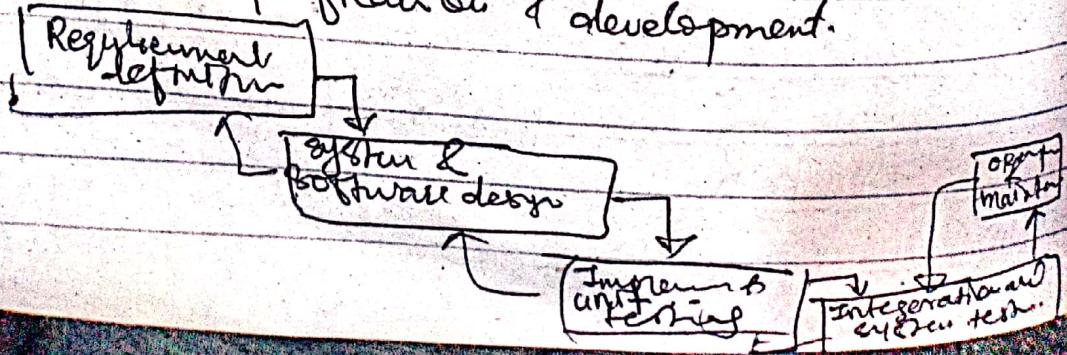
⇒ Plan Driven & Agile Processes:

- ↓
- Plan in advance.
 - Incremental.
 - easier to change.
 - reflect the customer changes requirement

⇒ Software Process Models:

① Waterfall model.

- Plan driven
- Separate & distinct Phases of specification & development.



Drawback

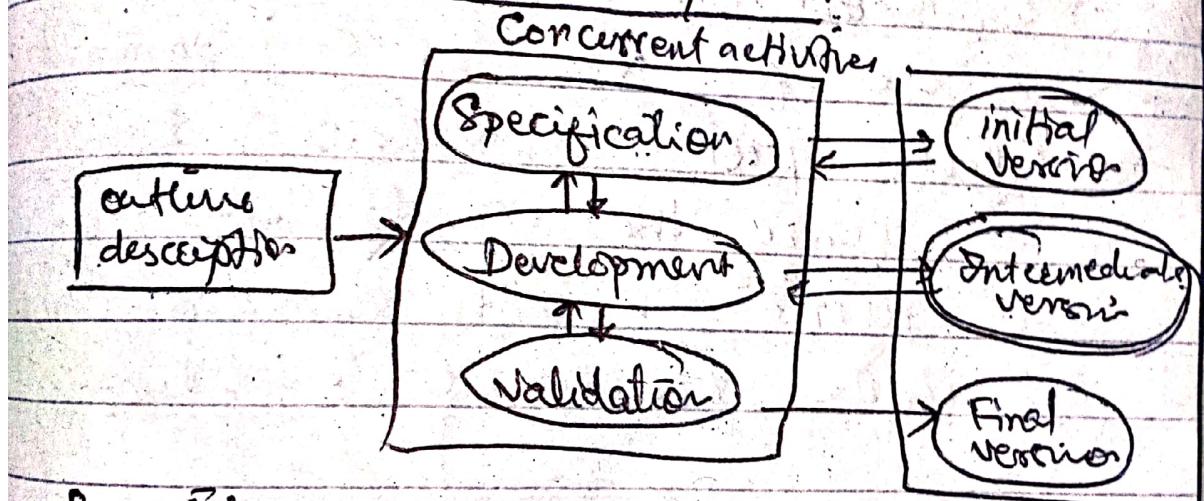
- difficult of accommodating change after the process is underway.

Problems

VO.

- ② mostly used for large systems.

2) Incremental Development



Benefit

- ① Change welcoming
- ② easier to get customer feedback.
- ③ Rapid delivery.

Problems

- ① Process is not visible
- ② System structure tends to degrade as new increments are added

Q4

COTS → Commercial off the shelf

③ Integration & Configuration:

- Based on software reuse, where system are integrated from existing component or app-system. → (COTS).
- Reused elements may be configured to adapt their behaviour & functionality to a user's requirement.

→ Types of reusable software:

- ① Stand alone app system (COTS) that are configured for particular environment.
- ② Collection of objects in package → .NET -
- ③ web services that are developed according to service standards and which are available for remote invocation

→ Key process stages

- ① Req. Specification
- ② Software discovery & evaluation
- ③ Req. Refinement.
- ④ App system configuration
- ⑤ Component selection & integration

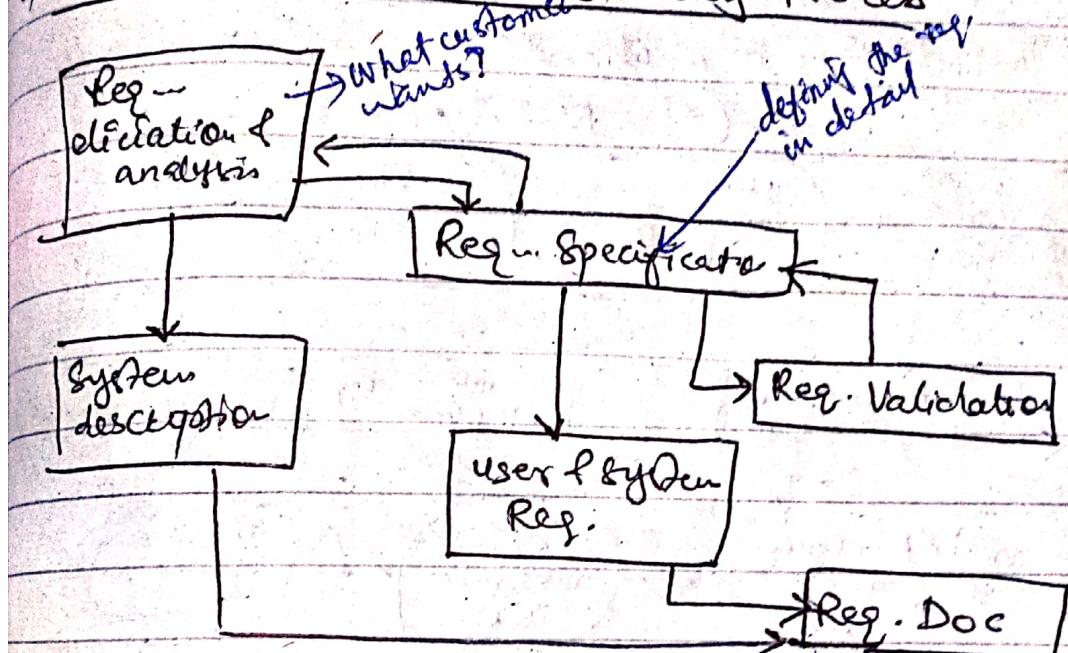
→ Advantages & Disadvantages

- ① Reduced cost → Not develop from scratch.
- ② Faster delivery
- ③ But req. compromises are inevitable so sys. may not meet real needs of user.
- ④ Cots of control over evolution of focused system elements.

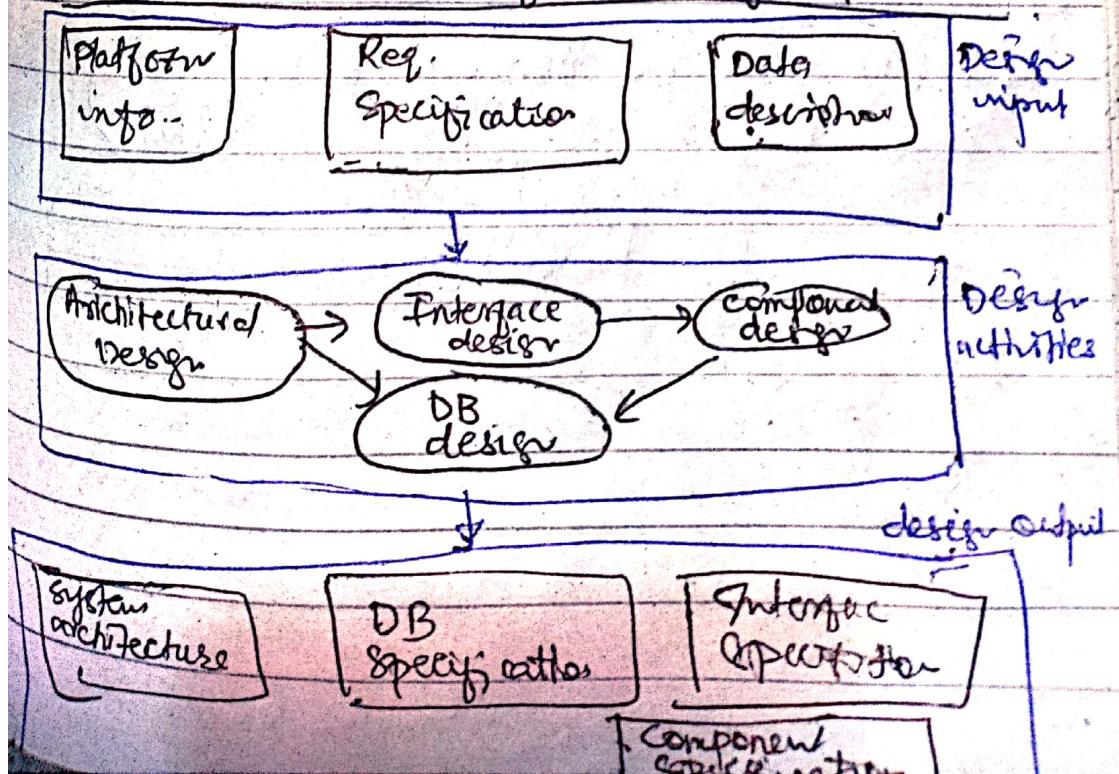
⇒ Process Activities

- 4 basic activities
- ① Specification
 - ② development
 - ③ Validation
 - ④ Evolution.

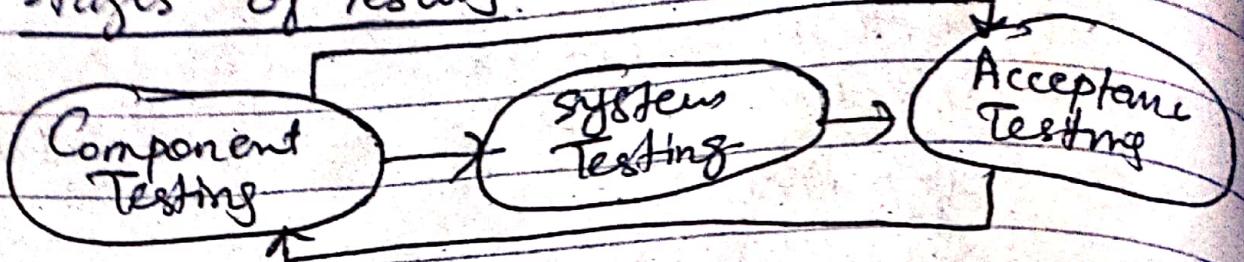
→ The Requirement Engineering Process



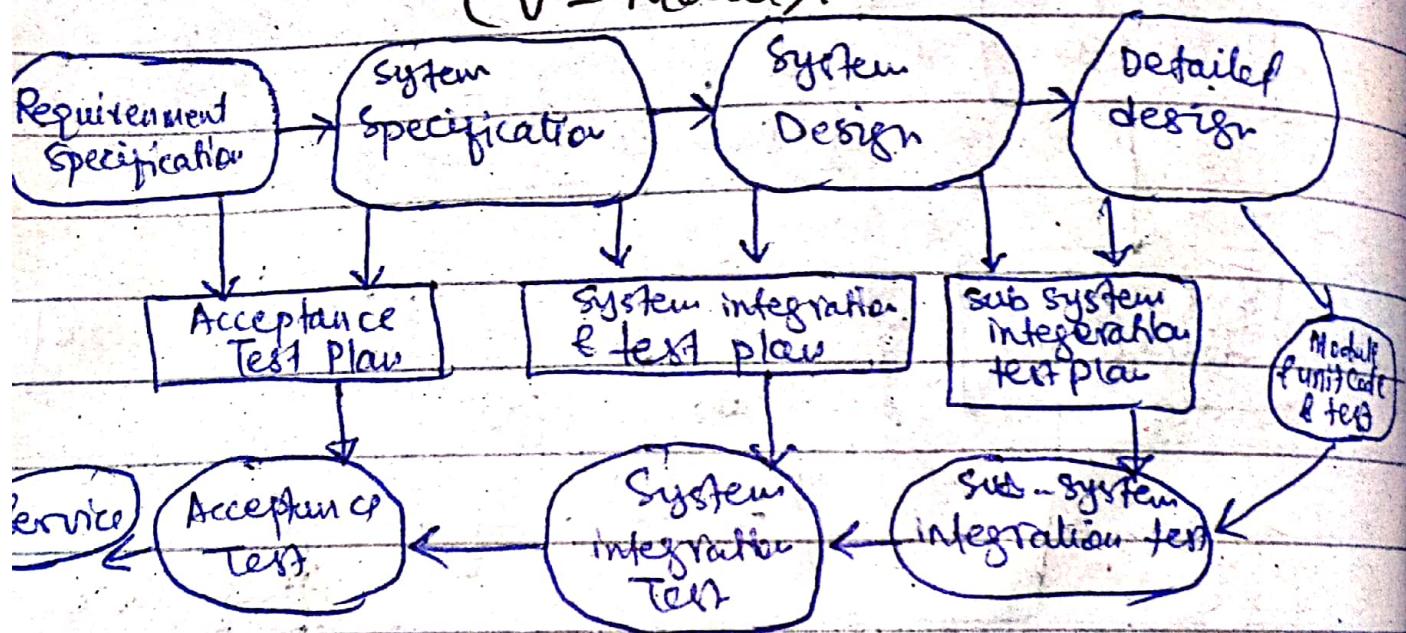
→ A General model of the design process.



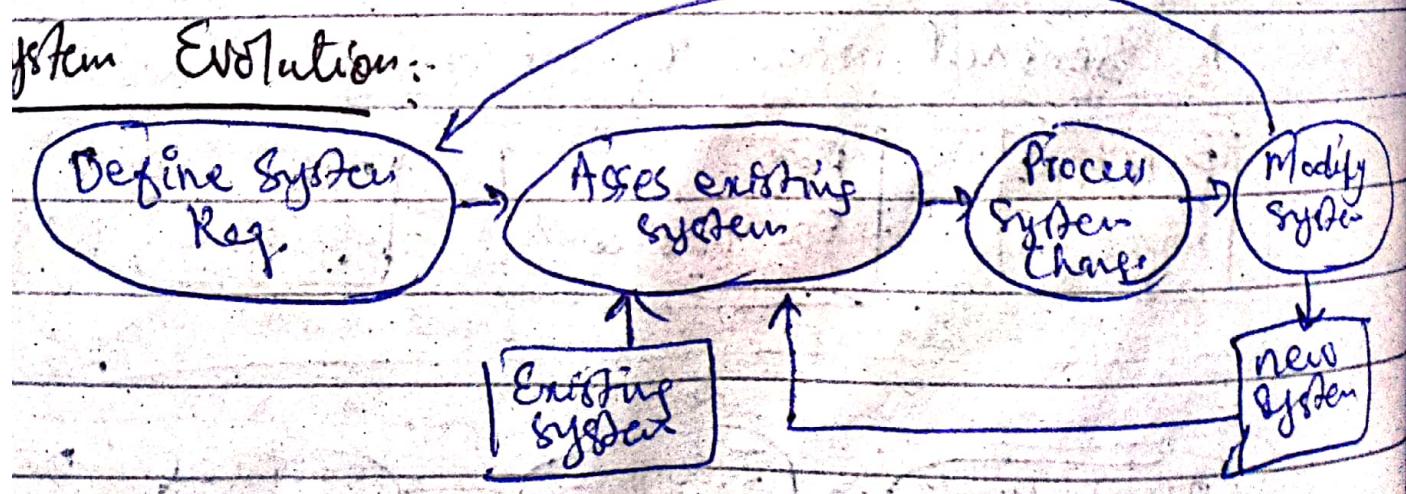
→ Stages of Testing



→ Testing Phases in a plan-driven Software Process (V-Model):-



System Evolution:-



(27)

⇒ Copying With Change

- change is unavoidable in all large software projects
- change leads to rework. so the cost of change include both rework (re-analyzing) as well as the costs of implementing new functionality

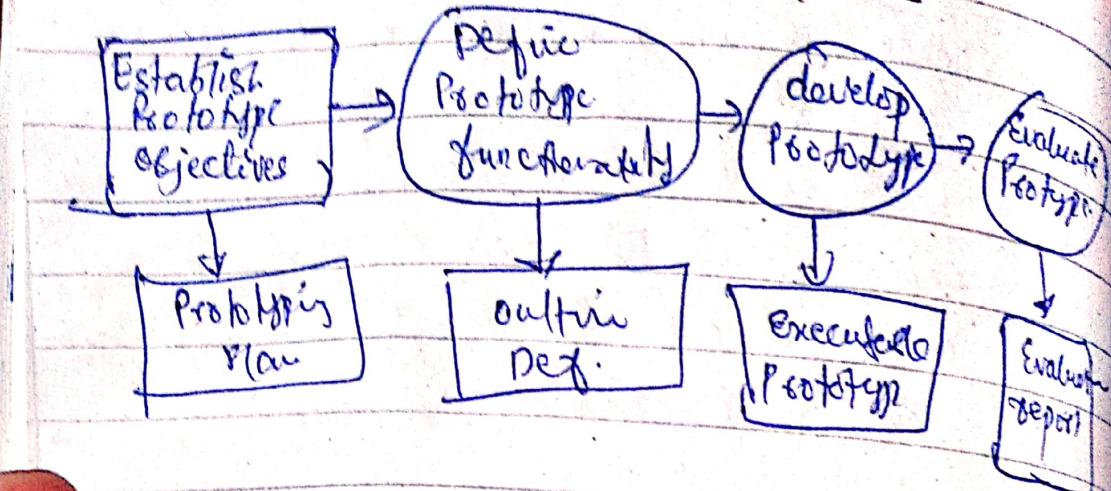
⇒ Software Prototyping

- A prototype is an initial version of a system used to demonstrate concepts & try out design options.
- used in
 - req. engineering process, to help w/ req elicitation & validation
 - in design process. to explore the option & develop a UI design
 - In testing process to run back - back tests.

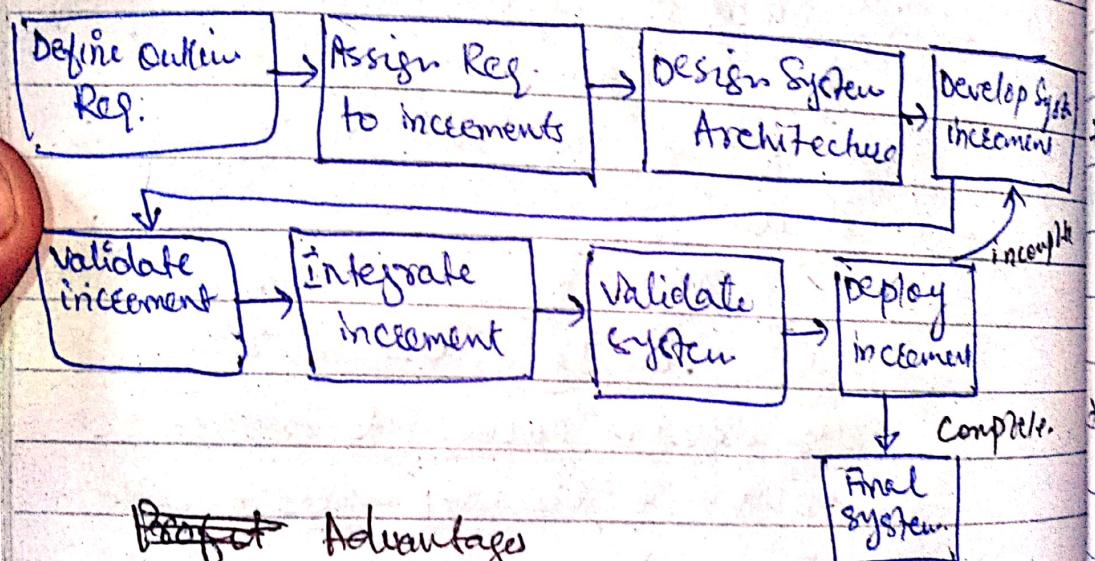
Benefits

- Improved system reusability
- A closer match to user's real needs.
- Improved design quality
- " maintainability
- Reduced development effort.

→ The Process of Prototype development



→ Incremental Delivery



Benefit Advantages

- ① Customer value can be delivered with each increment so system functionality is available earlier.
- ② Early increments act as prototype to help elicit req. for later increments.
- ③ Lower risk of overall project failure.
- ④ The highest priority system services tends to be built.

Problems

- (1) As req are not fully defined therefore difficult to implement all basic functionality while can go used by all other increments.

⇒ Process Improvement

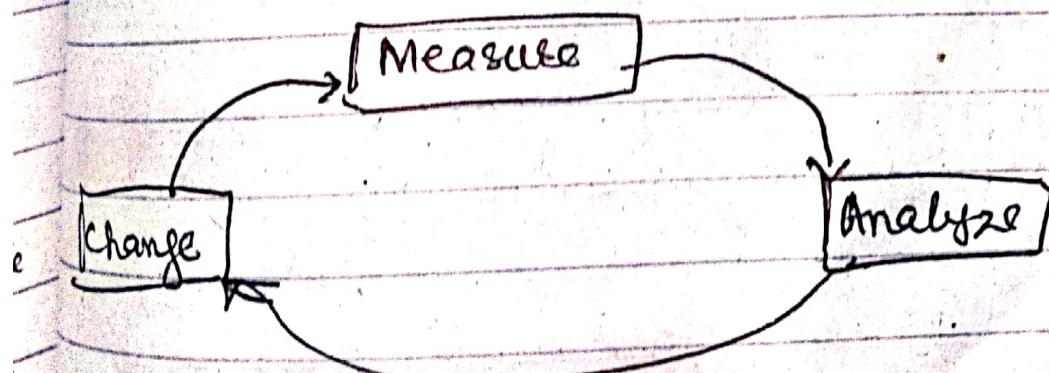
- enhance quality of software.
- reducing costs.
- accelerating the development process.

Approaches to improvements-

(1) ^{Process} Maturity approach

(2) Agile Approach.

The Process Improvement Cycle-



Process Metrics -

- Time taken -
- Resource req -
- No of occurrence of a particular event

⇒ Capacity Maturity Levels

