

# Importance of Software Engineering

1. Society rely on SW advancement

- reliable
- trust worth
- economical

2 Cheaper in long run

- methods
- techniques.

SW Process  $\leftarrow$  Systematic approach

4 functional SW process activities

1 SW Specification - user requirements

2 SW Development - type of lang / specification

3 SW Validation - check for proper data

4 SW Evolution - updation.

Issues that may affect any SW

1. Heterogeneity :

- Diff types of devices used
- Diff versions of SW
- Diff programming lang.

2. Business and social challenge :

changes must be reliable, economical, which can adapt new technologies

### 3. Security and trust

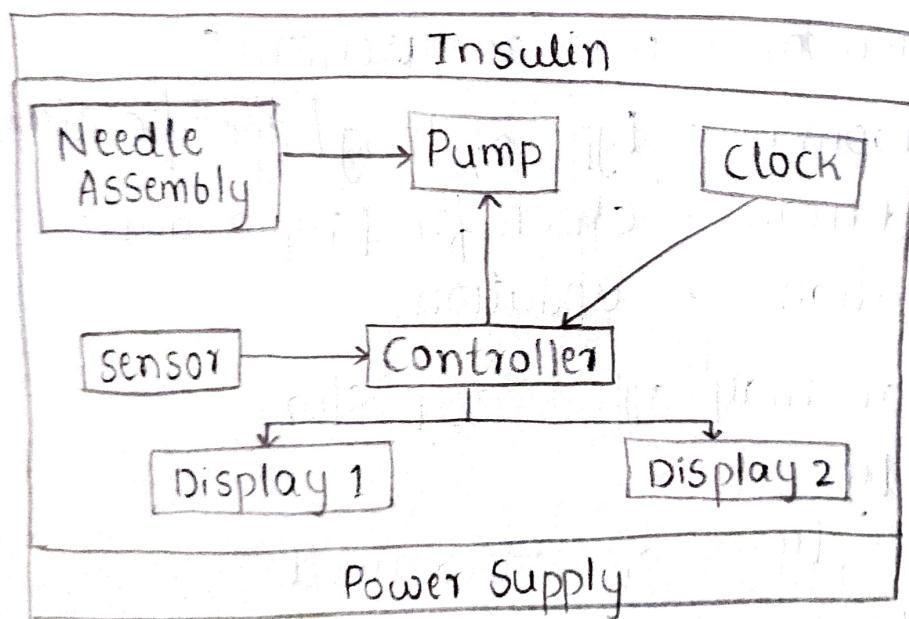
- trust worthy s/w

### Software Engineering Ethics:

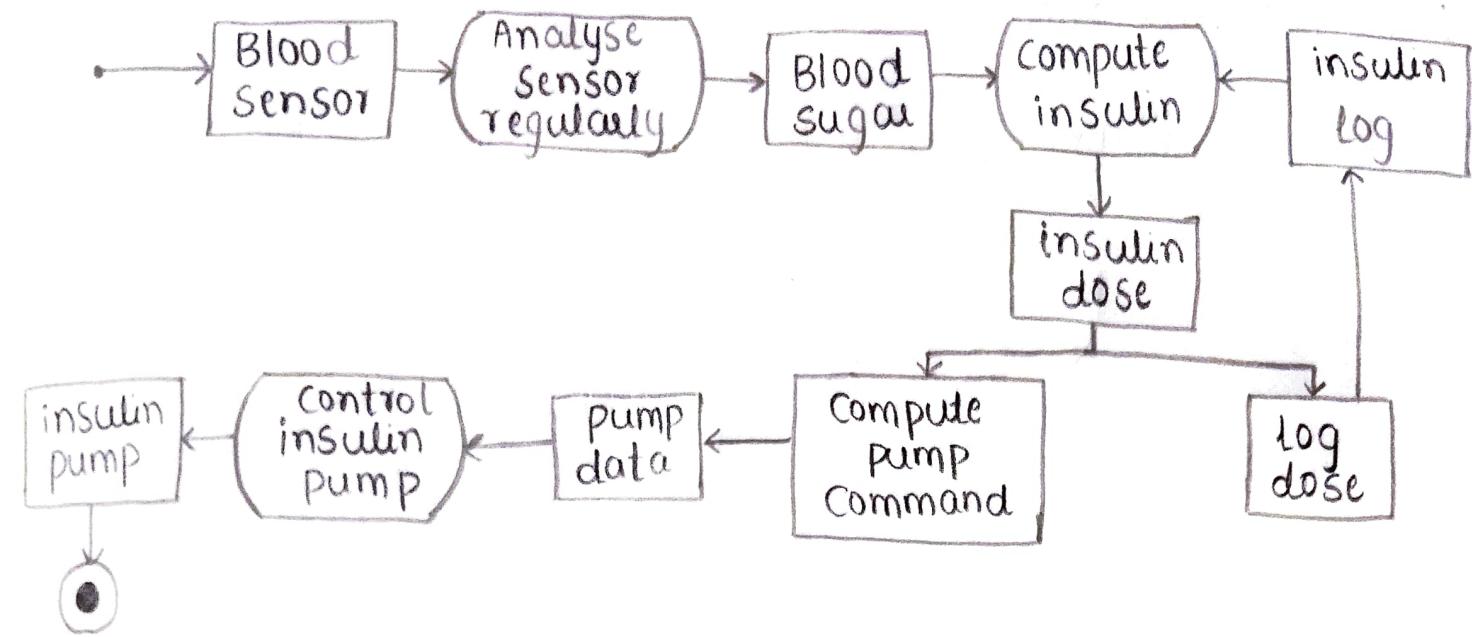
1. Confidentiality.
2. Completeness - misrepresent your level
3. Intellectual property rights
  - copyright and patents
4. Computer misuse.

### Case Study:

An insulin pump system:



Activity model of the insulin pump.



11/1/2020

Four fundamental SE activities.

1. SW specification

2. SW design : how to implement the system

3. SW development : writing the program

4. SW maintenance : fix bugs

These activities must include:

1. Products : outcomes of process activity

2. Roles : responsibility of people

3. Pre and post condition of process activity.

SW Process Model.

Basically 3 process model.

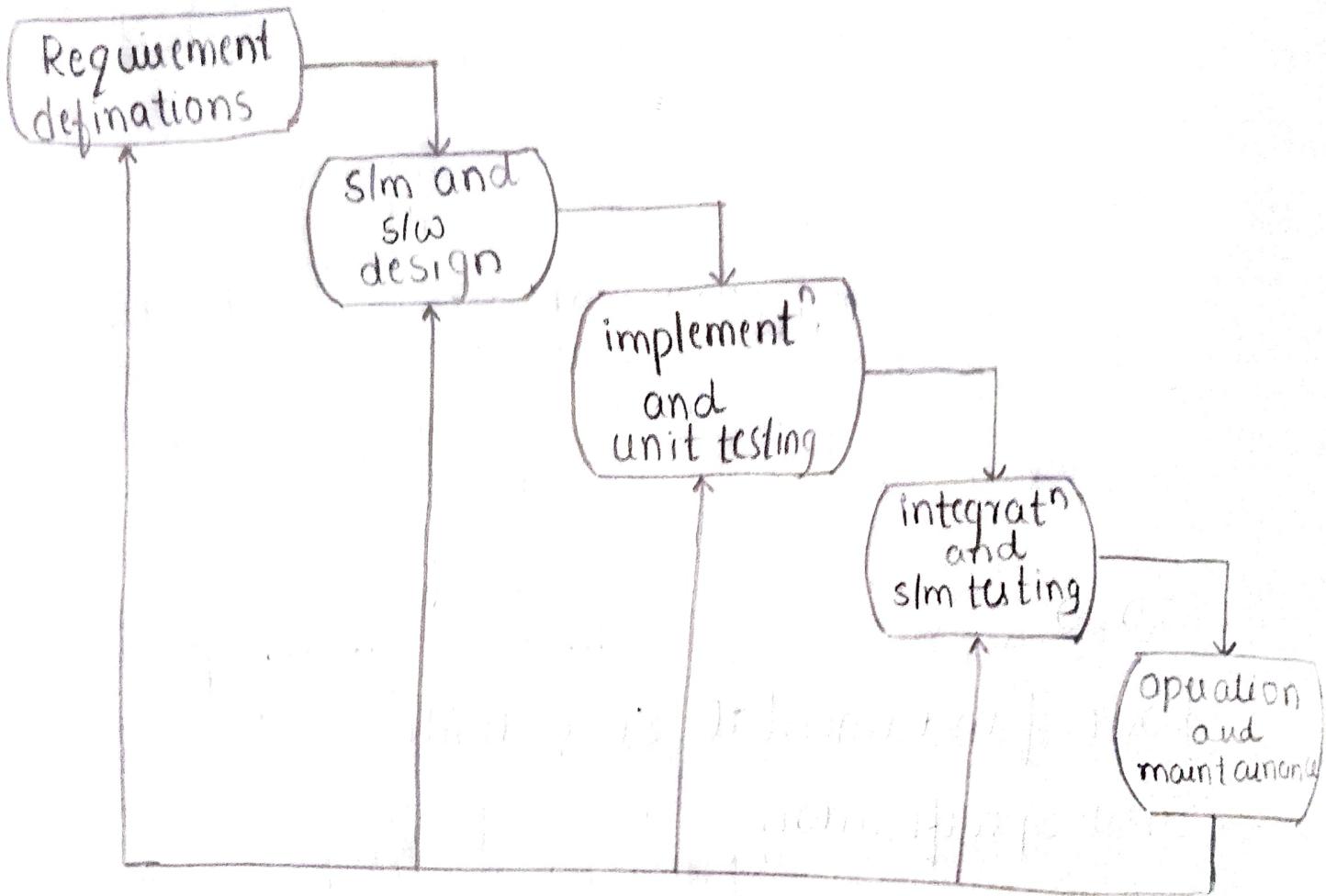
1. Waterfall model

2. Incremental model

3. Re-use - Oriented SE

# M 1. Waterfall model

or  
12M



## 1. Requirement Analysis and definition :

- constraints and goals are established
- user requirements to be specific.
- developers constraints

## 2. System and software design

- hardware requirements
- software requirements - fundamental software.

## 3 Implementation and unit testing

- involves verify each unit meet its specification (module testing)

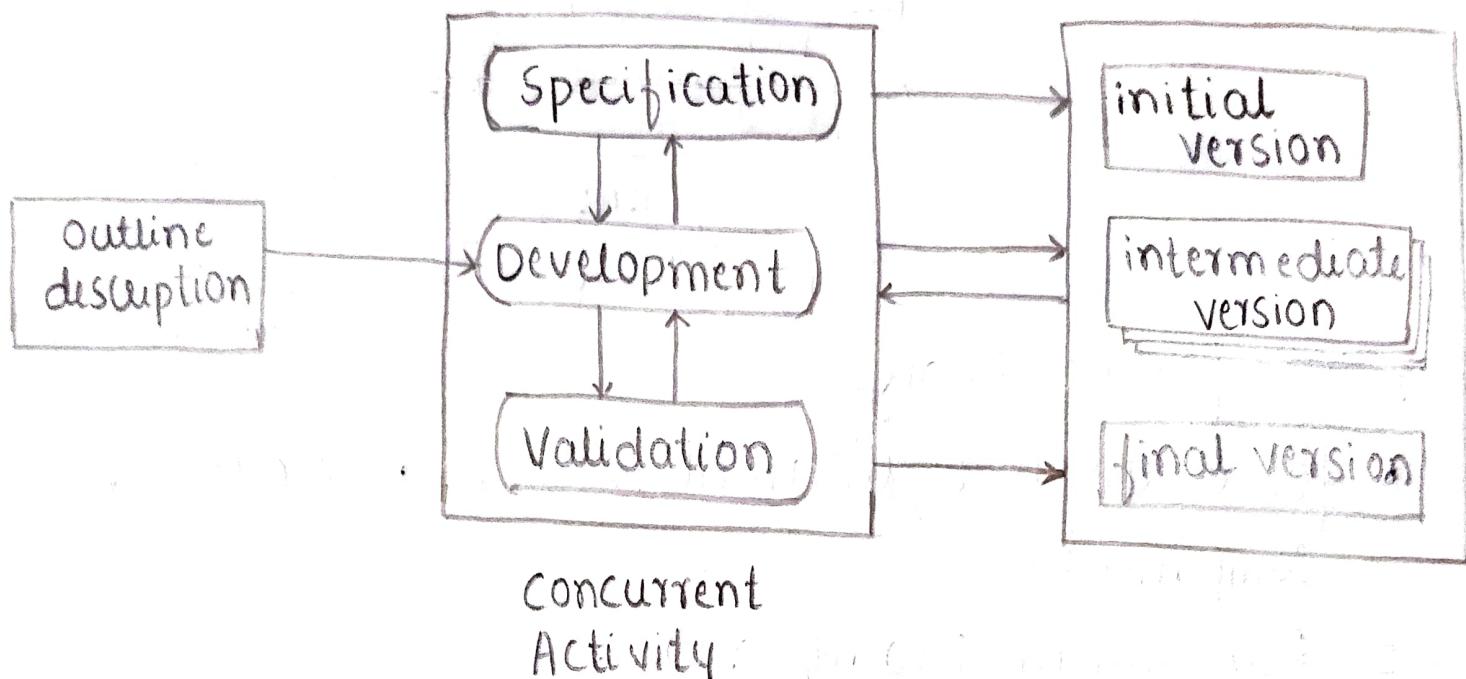
## 4. Integration and System testing

- complete system is tested.

## 5. Operation and maintenance

- This is the longest life cycle phase.

## 2. Incremental Development



Benefits compared to waterfall model.

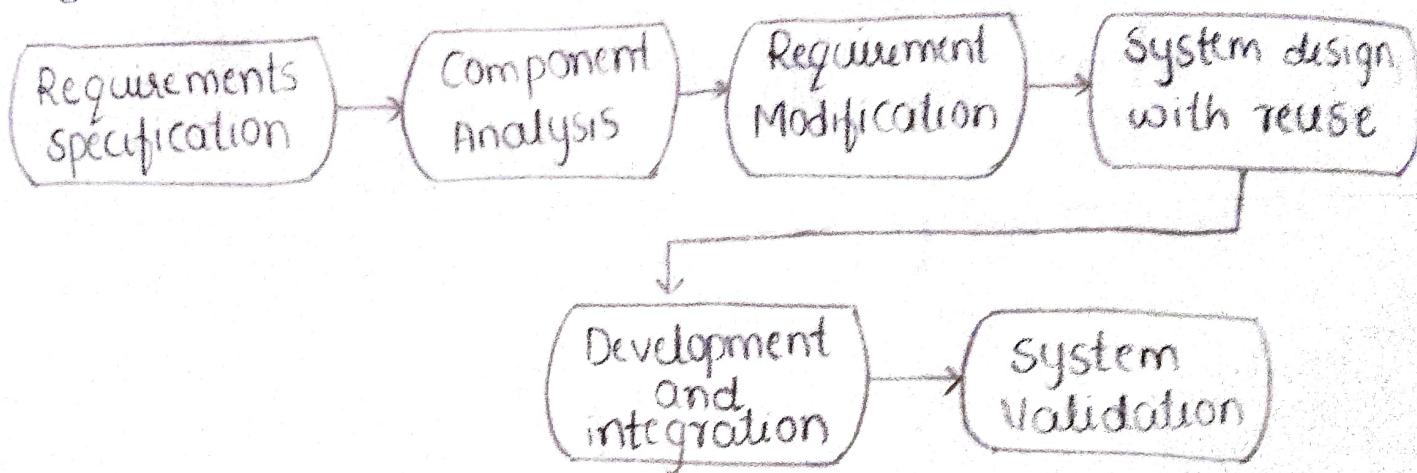
1. Cost of accommodating changing customer requirements is less.
2. Easier to get customer feed back.
3. Rapid delivery and development.

Management perspective: the incremental model has 2 problems:

1. The process is not visible.
2. System structure tends to degrade as new increments are added.

13/1/2020

### 3. Reuse - Oriented SE



#### 1 Component Analysis :

- for a given requirement , search is made for components.

#### 2 Requirement Modification :

- analysis of the requirements are made about the component.

#### 3 System design with reuse :

- frame work of the system is designed or if an existing system reuse.

#### 4. Development and integration :

- part of development
- integrate or join the diff. components.

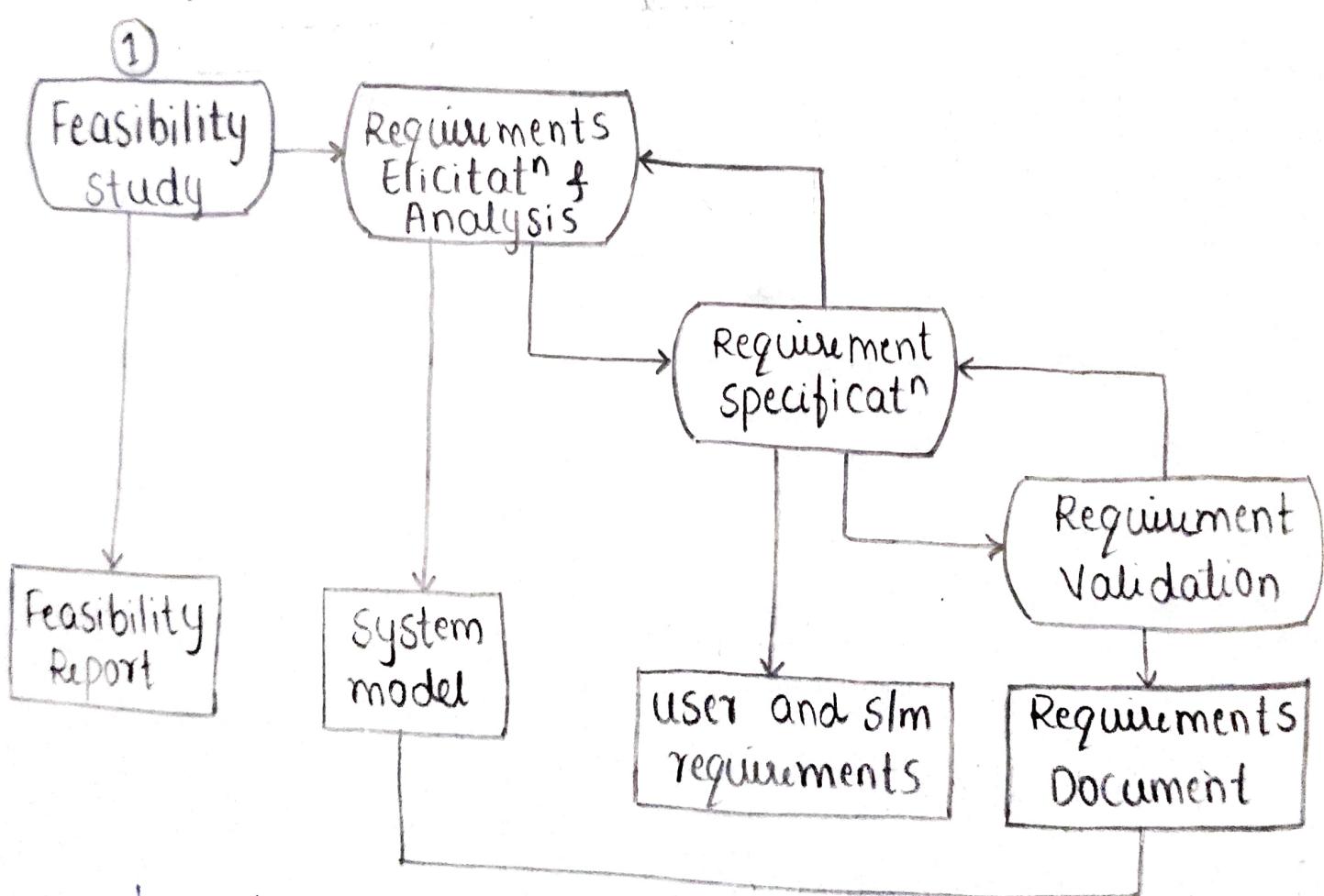
### Process Activity :

#### 1 SLW specification :

- Requirements Engineering process aims to produce and agreed requirements documents.

- There are 4 main activities in the requirements eng. process.

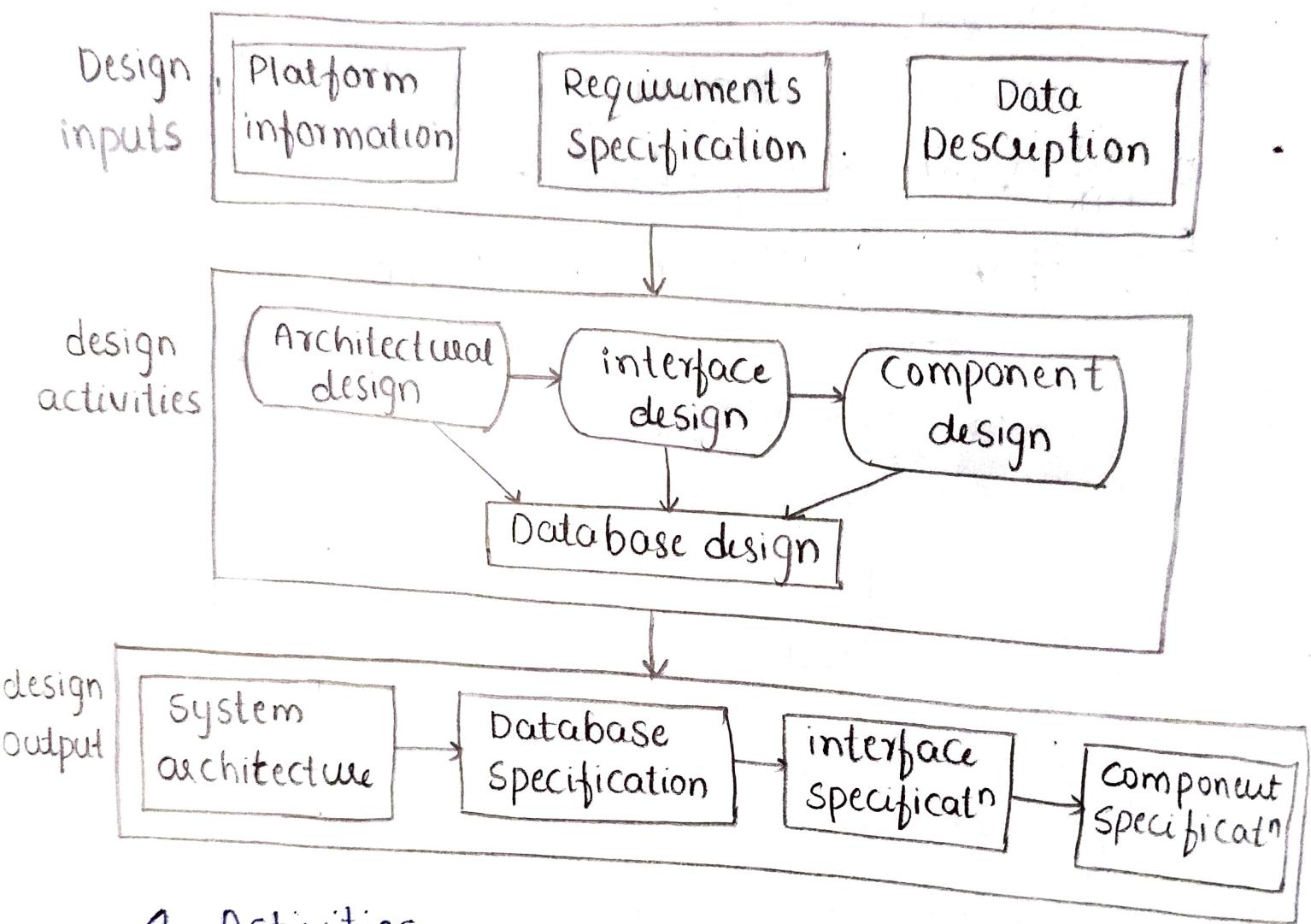
1. Feasibility study
- 2 Requirements Elicitation and Analysis
3. Requirement Specification
4. Requirements Validation.



2. sl/w design and implementation

- a description of the structure of the sl/w to be implemented.
- different stages of the design
- a feed back from one stage to another and consequent design rework.

Most s/w interface with other softwares like database s/w



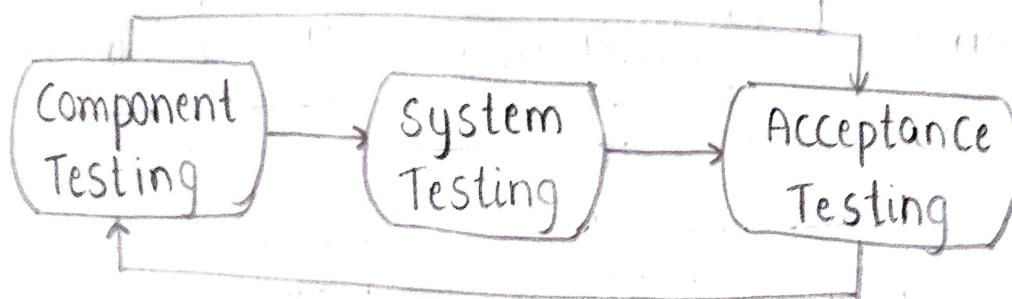
#### 4 Activities :

1. Architectural design
2. Interface design
3. Component design
4. Database design.

14/1/2020.

### 3. Software Validation.

- intended to show that a system confirms to specifications and requirements of a customer



Stages in Testing.

#### 1. Development or component testing

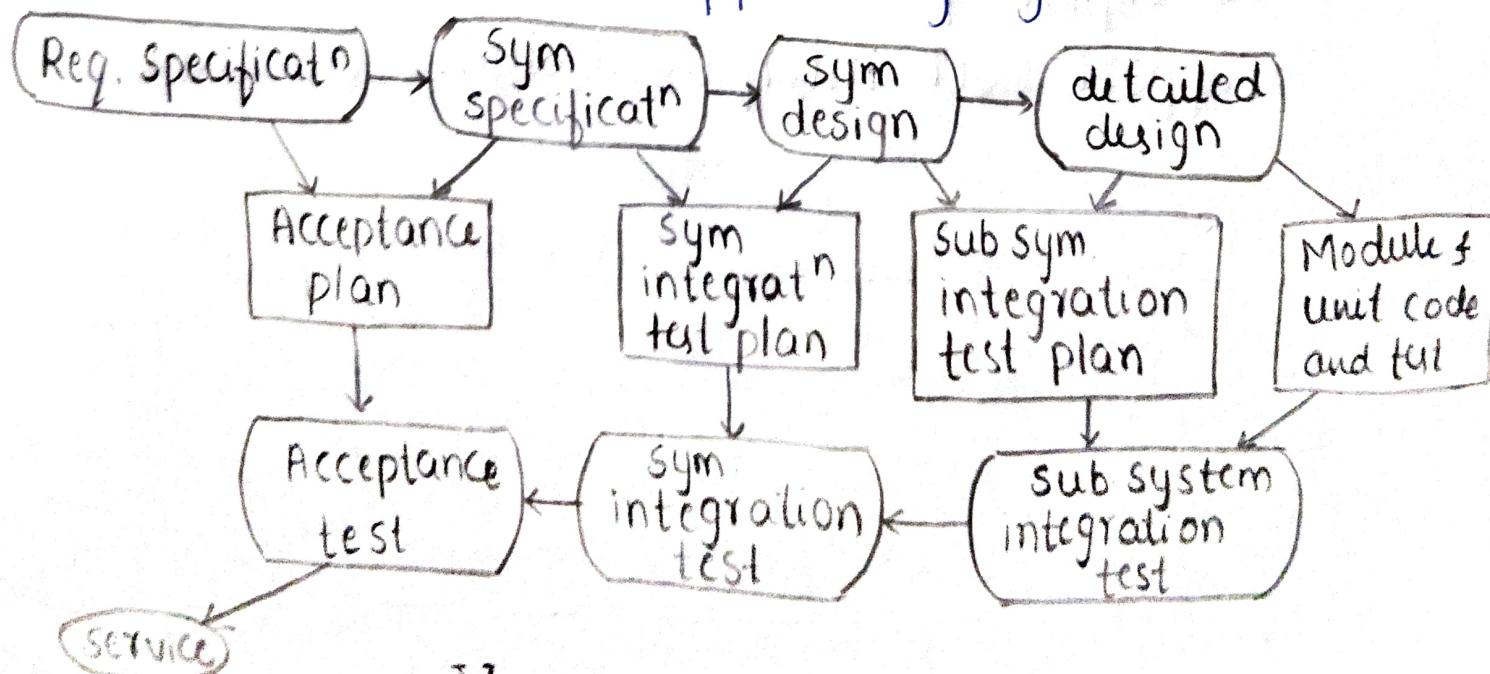
- testing for components and their working

#### 2. System Testing:

- process to find errors between components and interfacing

#### 3. Acceptance Testing:

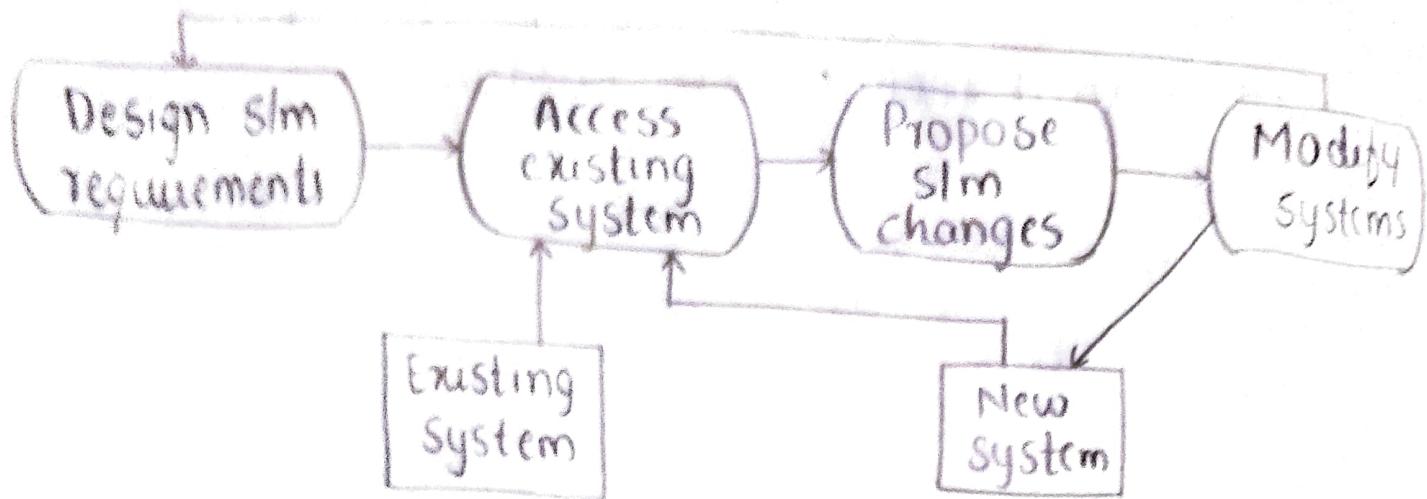
- test with data supplied by system customer.



V-model for Testing

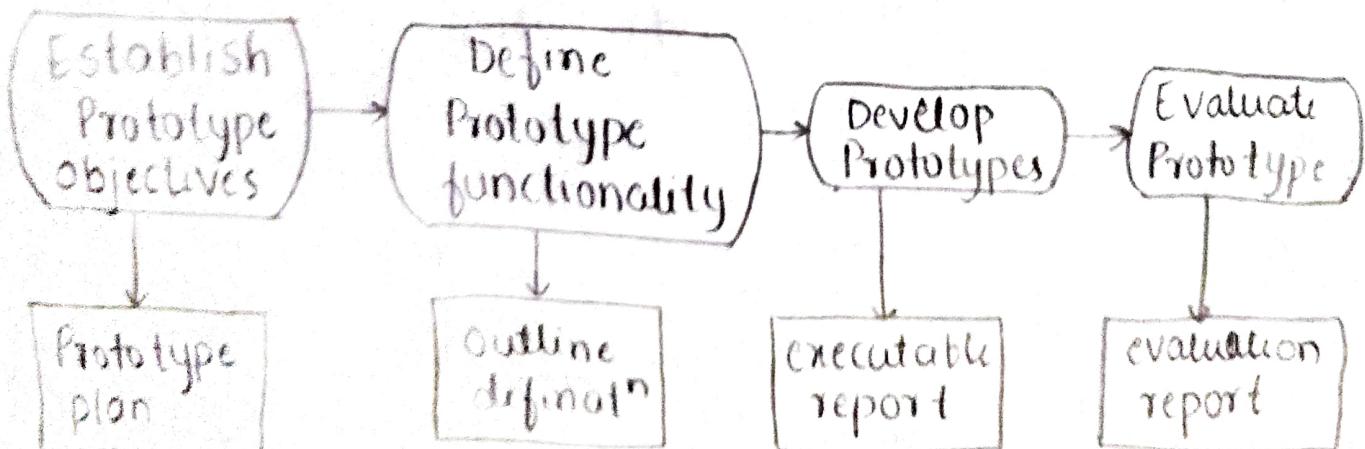
# 1. System Evolution

Why evolution?



## Prototyping:

- an initial version of a system
- A slw prototype can be used in a slw development process to help anticipate changes that may require:
  - i. in elicitation and validation of a system requirements
  - ii. to explore particular slw solution and to support user interface design.



16/1/2020.

## Reducing the cost of work:

- Evolution / changes made in a system may add up the cost.
- When changes are to be made, it may require ~~ment~~ to redo the entire work, also called as rework.

### 2 Types of reducing in the changes.

#### 1. change avoidance

- process includes activities that can anticipate possible changes

Eg: prototype of a system.

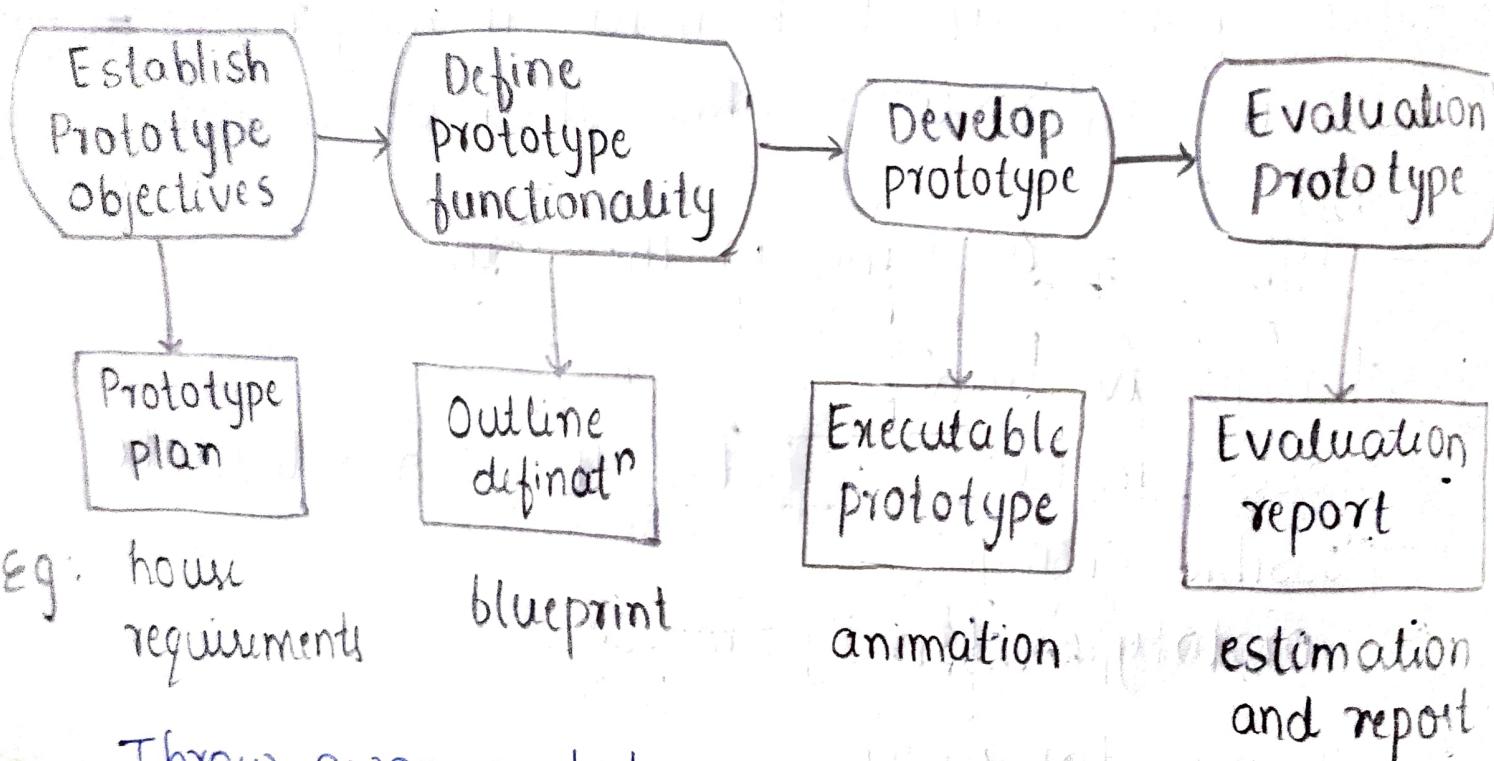
#### 2. change tolerance

- process is designed so that changes can be accommodated at relatively low cost
- normally involves incremental development
- development may be done that is not yet developed

#### 1. Software Prototype

Def<sup>n</sup>: "A prototype is an initial version of a system used to demonstrate concepts and try out design options".

- \* Can be used in :
  - requirement engineering process
  - in design to explore options
  - in testing process to run back-to-back tests



Throw away prototypes :

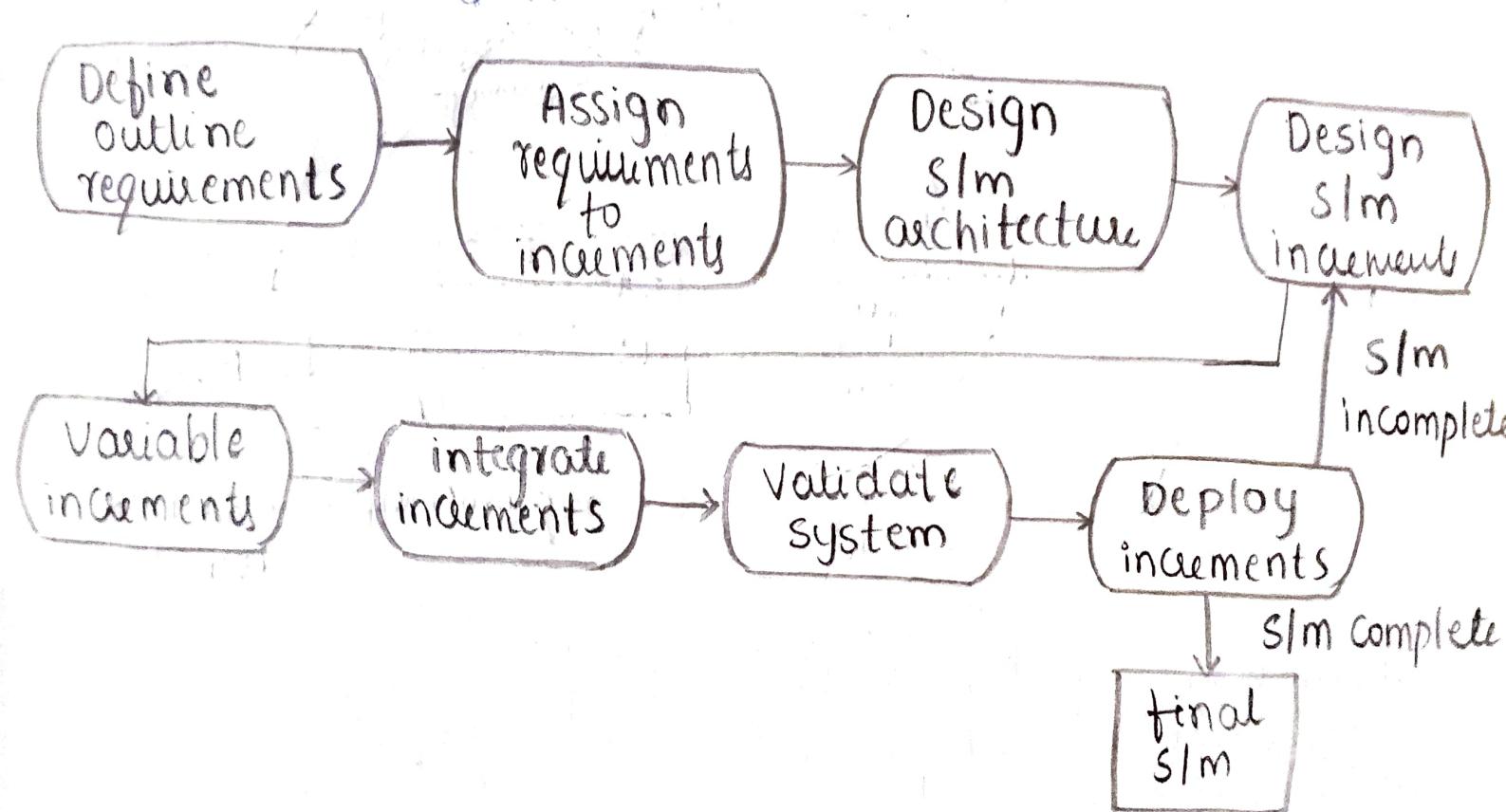
- They are not a good basis for a product
- Not documented
- will not meet organisational requirements
- degrades with rapid changes

Benefits :

- cost effectively
- Faster
- changes are cheaper
- can be close to real system

## 2. Incremental Model:

- Rather than delivering whole system, delivers in part by part.
- involves functionality like
  - i. focus on area of the product
  - ii. checking for errors.



Disadvantages:

1. Require a set of basic facilities
2. Development is conjunction with the software
3. Iterative development is difficult.

3/2/2020

## UNIT-3

### Agile Software Development

Different approach for rapid software development, they have some characteristics.

1. The process of specification design and implementation are interleaved / inter-related
  - There no detailed sm specification and no detailed design document
  - User req's document only defines the most sm characteristics
2. Sm is developed in a series of versions
  - End users and stakeholders are involved
  - They may propose changes to sm in later versions.

3. slm user interfaces are often developed using an interactive development slm that allows the interface design to be quickly created by drawing and placing icons.

### Issues with Rapid slw development

1. Since it is inter-related we cannot directly go the other step.
2. No proper documentation
3. Difficult with interactive process

### Solution for this:

#### Agile slw Development

- Agile method universally rely on an incremental approach
- Best suited to application development where req's change rapidly
- They intend to deliver working slw quickly
- Aim is to cut down on process by avoiding working that has long-term value and eliminating documents

\* Agile System has been successful for some types of slm development

1. Product development : where a slw company is developing small or medium-sized product for sale.
2. Customer slm development within an organisat<sup>n</sup> where there is clear commitment from the customer

### The Principles of Agile Methods.

1. Customer involvement
2. Incremental delivery
3. People not process
4. Embrace changes
5. Maintain simplicity.

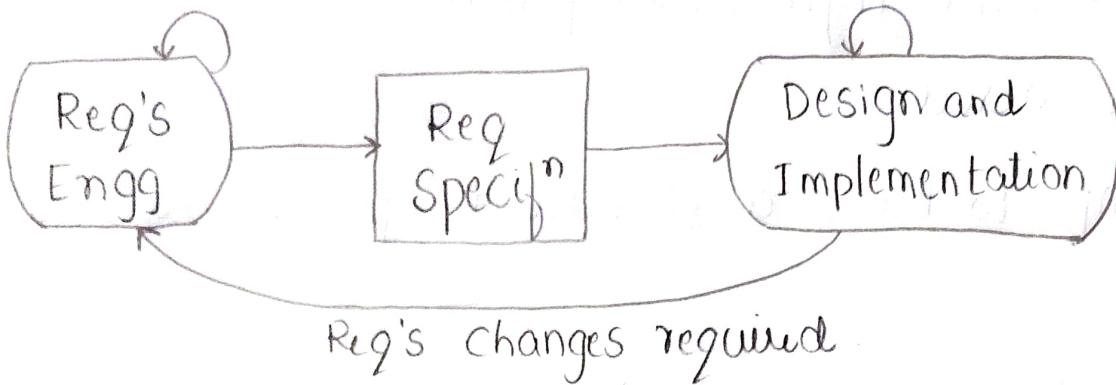
6/2/2020

- \* Another non-technical problem:
  - general problem with incremental development and delivery occurs when the slm customers uses an outside organisation for slm development.

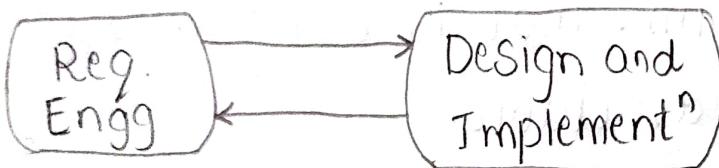
There are 2 Q's that should be considered when considering agile method and maintenance.

1. Agile slm's that are developed using an agile approach maintainable?
  2. Can Agile methods be used effectively for evolving a slm in response to customer change requests.
- \* Main difficulty after slow delivery is likely to be keeping customers involved in the process.
  - \* Other problems.
    - maintaining continuity of the development team
    - as team rely on team members, understanding aspect.
  - \* Plan - Driven & Agile development.
    - \* Agile approaches to slow development consider design and implementation to be central activity
    - \* Other activities are
      - req's elicitation and testing into design and implementation.

# 1 Plan Based Development :



# 2 Agile Development :



Plan Based Development	Agile Development
1. Iteration occurs with <del>activities</del> activities with formal documents used to communicate	1. Iteration occurs across activities, Req's and design are developed together.
2. Process cannot support incremental development and delivery.	2. Process is not inevitably code-focused and may produce some design
3. Feasible to allocate req's and plan the design and development	3. Mostly unplanned

To decide to balance between :

Plan based and Agile development , few Q's have to be answered.

1. Is it imp to have a detailed specification and design?
2. Is incremental delivery strategy where delivery of a slm and rapid feedback realistic
3. How large is the slm . that is being developed.
4. What technologies available?
5. Are there any cultural issues.
6. How good are designers?