Date 2024
M T W T F S S

CH#2 SOFTWARE PROCESS

- Structure of set of activities required to develop a software system.

- Many different software processes but all involve:

  - Specification:- define what systems do
  - Design & implementi- organization of system, implenting system
  - Validation:- checking it does what customer want
  - Evolution:- changing system in response to changing customer needs

- Software process descriptions

  Specifying a data model, designing UI and ordering of these activities.

  Process description include:-

  Products:- outcomes of process activity
  Roles:- reflect responsibilities of people involved in the process.

  Pre & Post conditions:- statement that are true before and after a process activity has been enacted or a product produced.

Date 29
M T W T F S S

* PLAN DRIVEN :-

Processes where all process activities are planned in advance and progress is measured against this plan.

• Agile PROCESS :-

Planning is incremental, easier to change process to reflect changing customer requirements.

Most Practical processes include elements of both plan-driven and agile approaches. There are no right or wrong software processes

* SOFTWARE PROCESS MODELS :-

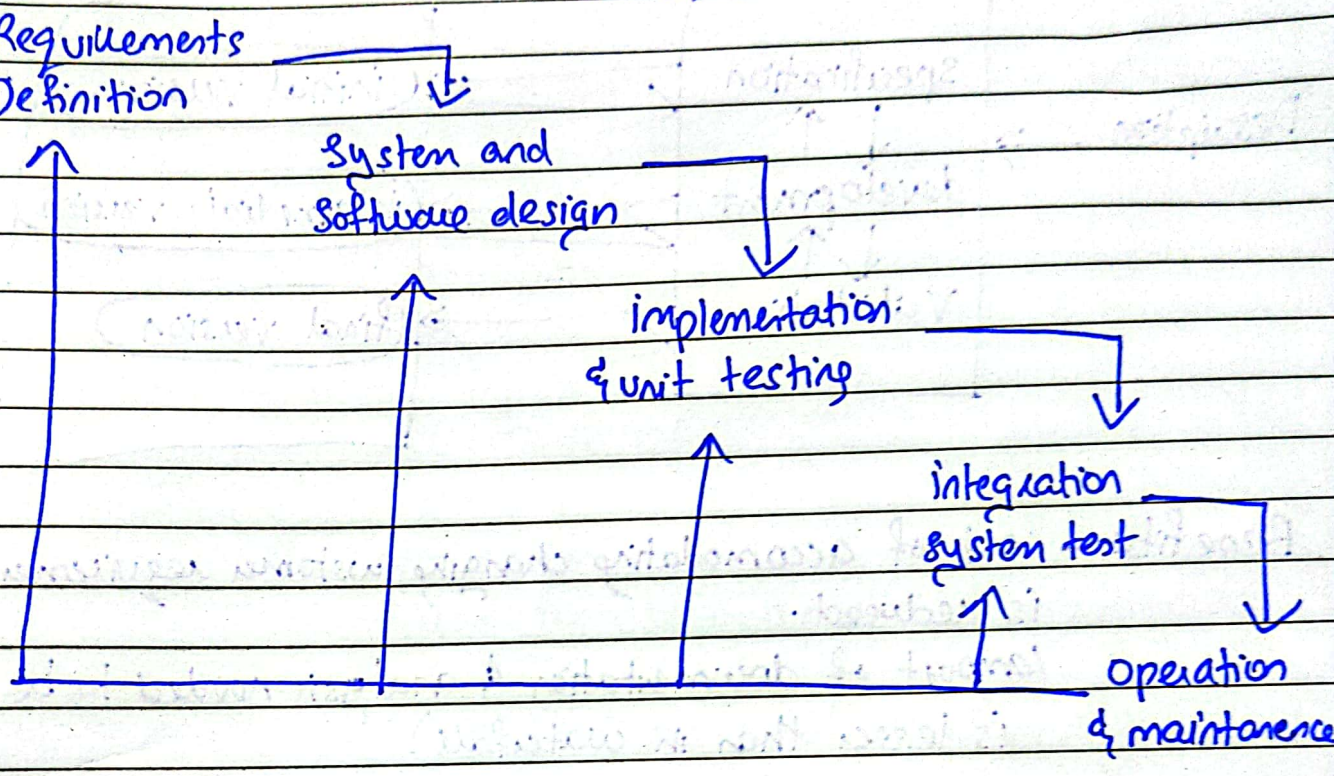
• WATERFALL MODEL :- PLAN DRIVEN MODEL
Seperate and distinct phases of specification & development.

• INCREMENTAL DEV :- MAY BE PLAN DRIVEN OR AGILE
Specification, validation and validation are interleaved.

Most large systems developed using process that incorporates elements from all of these models.

Date 20
M T W T F S

* WATERFALL MODEL



DRAWBACK:-

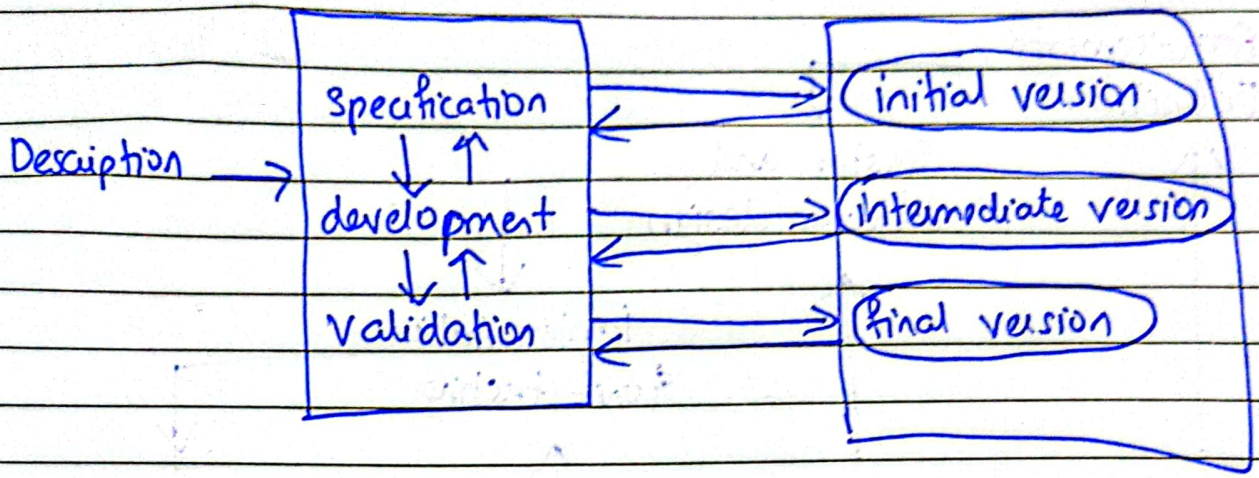
Difficulty of accomodating change after the process is underway (phase has to be complete before moving onto next phase)

Dividing the project into fixed stages that can't be easily adjusted makes it difficult to respond to changing customer requirements.

Waterfall is mostly used for large systems engineering projects where a system is developed at several sites.

Date 20
M T W T F S

* INCREMENTAL DEVELOPMENT:-



Benefits:- cost of accomodating changing customer requirement is reduced.
(amount of documentation & analysis needed to be redone is lesser than in waterfall)

- easier to get customer feedback on development work that has already be done.
- More rapid delivery and deployment of useful software to the customer is possible.

Drawbacks/Problems:-

Process is not visible

System structure tends to degrade as new increment added. (unless time and money is spent on refactoring to improve software process)

Date 20
M T W T F S

* INTEGRATION AND CONFIGURATION

Reused elements may be configured to adapt their behaviour and functionality to a users requirements.

Reuse is now the standard approach for building many types of buisiness system.

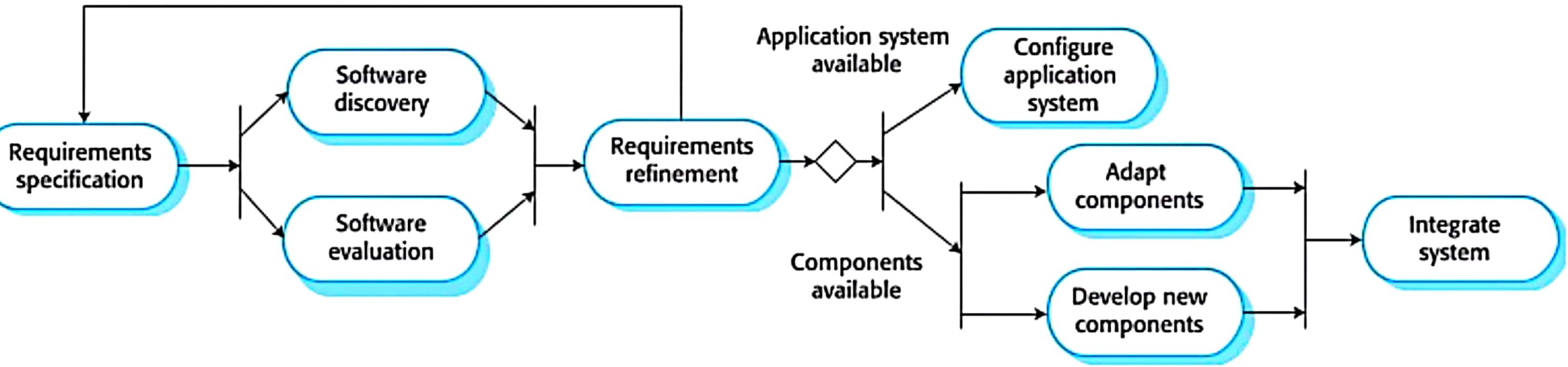
- TYPES OF REUSABLE SOFTWARE

- Stand-alone app system:-
  configured for use in a particular environment.

- Collections of objects that are ~~used~~ developed as a package to be integrated with a component framework such as .NET or J2EE.

- Web services that are developed according to service standards and which are available for remote invocation

Reuse-oriented software engineering



Key process stages

- Requirements specification
- Software discovery and evaluation
- Requirements refinement
- Application system configuration
- Component adaptation and integration

Advantages and disadvantages

- Reduced costs and risks as less software is developed from scratch
- Faster delivery and deployment of system
- But requirements compromises are inevitable so system may not meet real needs of users
- Loss of control over evolution of reused system elements

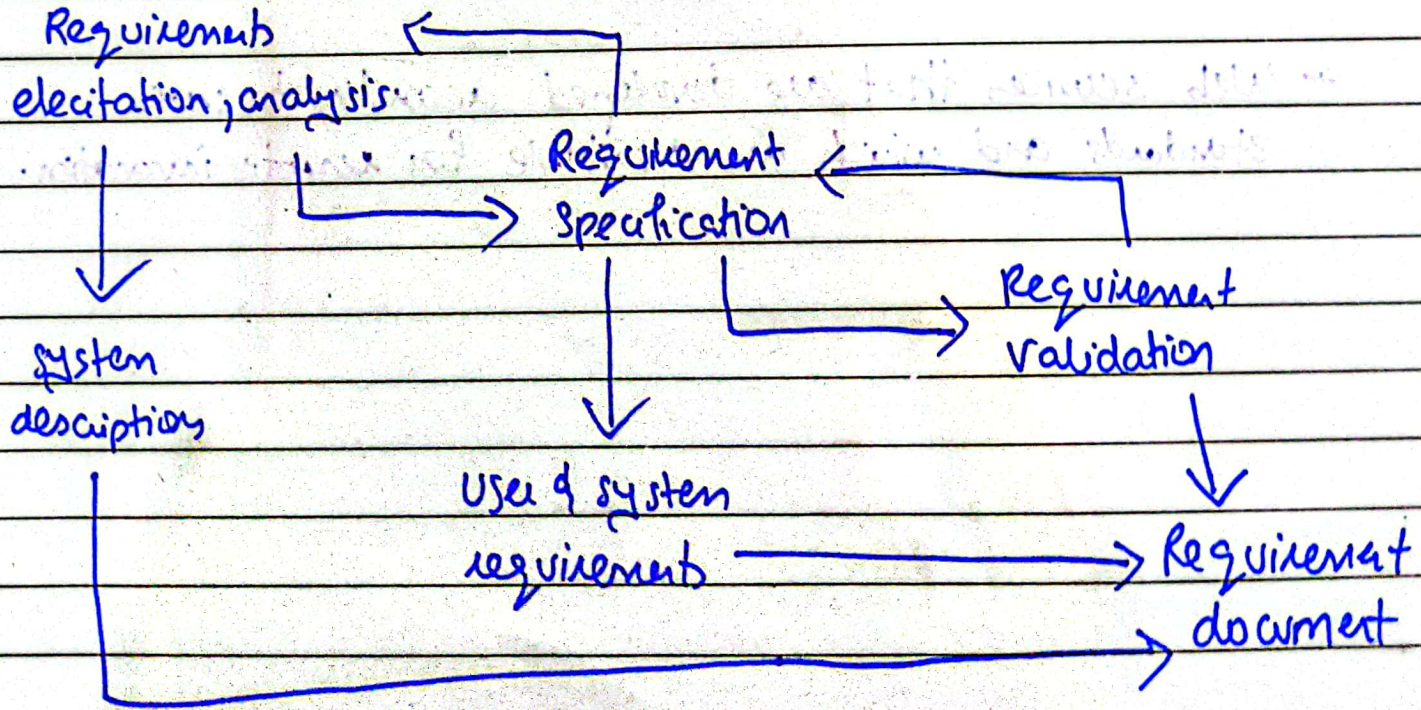
- PROCESS ACTIVITIES

Real software processes are interleaved sequences of technical, collaborative and managerial activities with the overall goal of specifying, designing, implementing and testing a software system

4 basic process activities of specification, development, validation and evolution are organized differently in different development processes.

Eg:- In waterfall → organized in sequence.
In Incremental → interleaved.

* Requirement Engineering Process



Date 20
M T W T F S

- Software Specification

Process of establishing what services are required and constraints on the system's operation and development.

- Requirement Engineering process:-

Elicitation & analysis:-
what the system require from the system?

Specification:-
Defining requirements in detail

Validation:-
checking validity of the requirements.

- Software design and implementation.

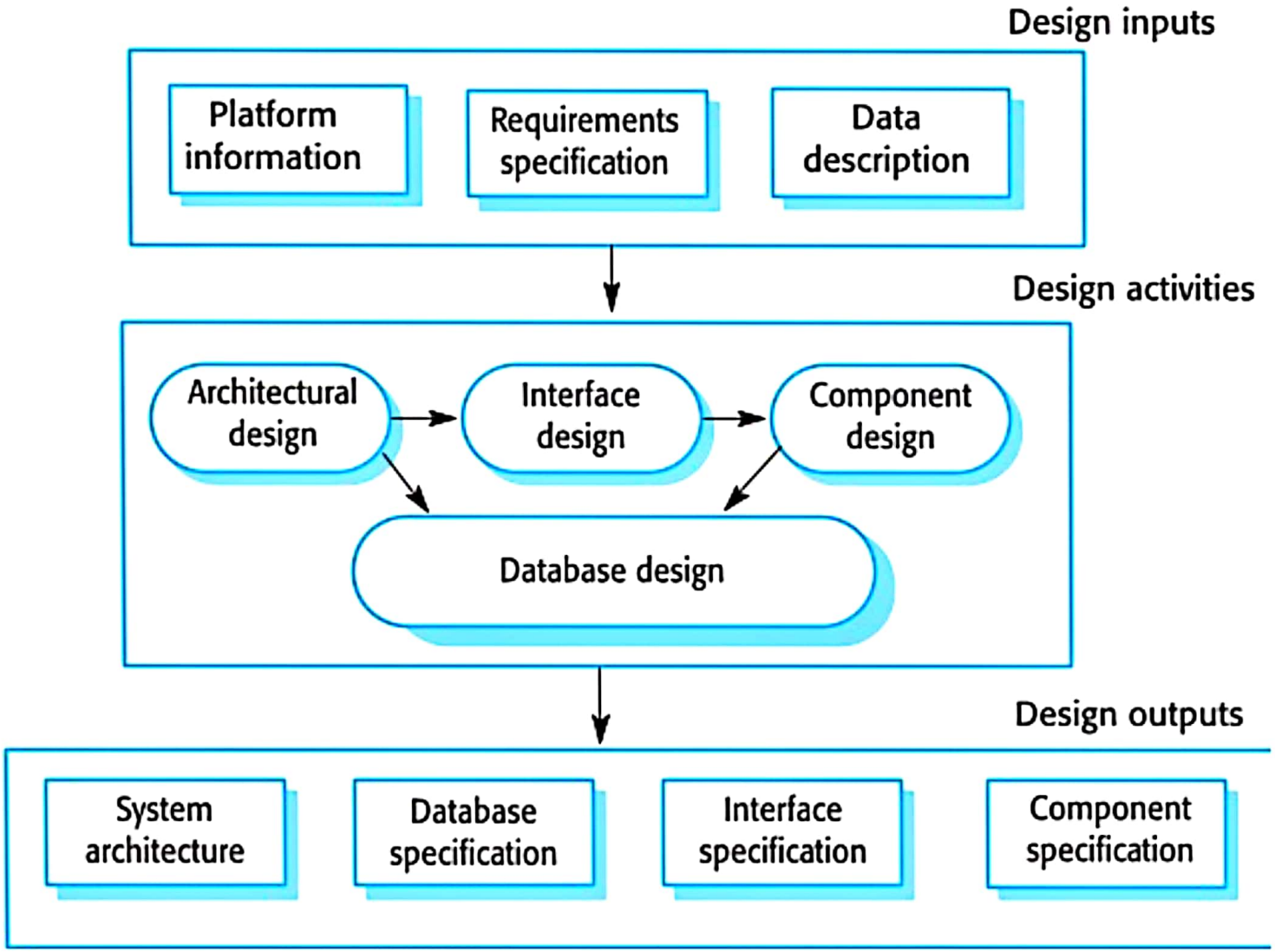
Process of converting system specs to executable system

Software Design: Design a software structure that realises the specification.

Implementation:- Translate this structure into executable progr

Activities of design and implementation are closely related and may be interleaved.

A general model of the design process



Date ___/___/___
M T W T F S

- Design Activities:-

Architectural Design:- identify overall structure of system, principle components, relationship, distribution.

Database Design:- design system data structure and representation in database.

Interface Design:- define interfaces b/w system and components.

Component selection & design:- search for reusable components.

- System Implementation:-

- Software implemented either by developing a program or by configuring an app system.
- Design and implementation are interleaved activities for most types of software system.
- Programming is individual activity with no standard process
- Debugging is activity of finding program faults and correcting these faults.

Date 20
M T W T F S

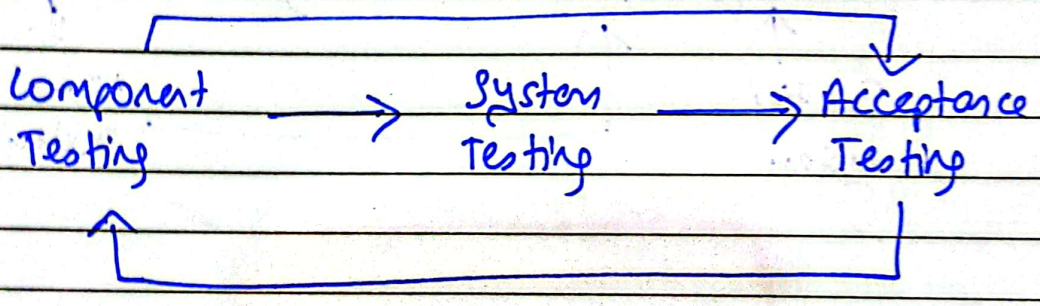
- Software Validation

intended to show that system confirms to its specification and meets the requirements of the system customer.

involves checking and review processes and system testing.

Testing is the most commonly used Verification and validation (V & V) activity.

- Stages of Testing:

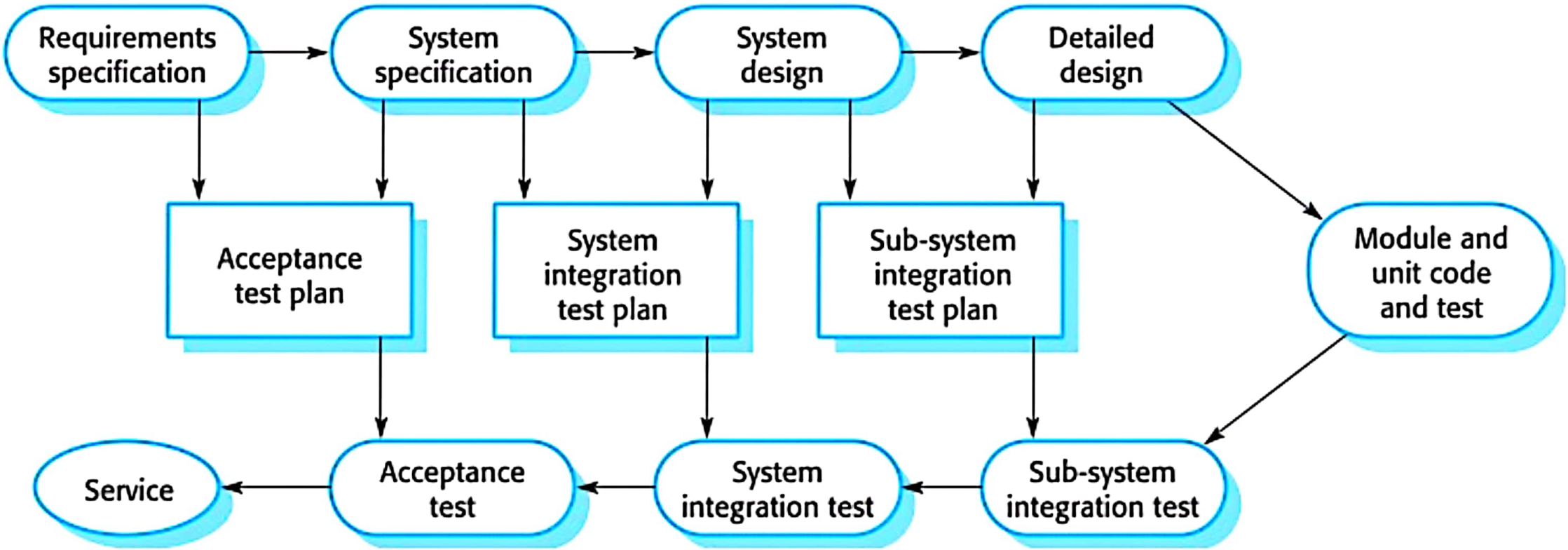


Component Testing:- individual components are tested independently
- components maybe function or objects of these entities.

System Testing:- Testing of the system as whole.

Customer Testing:- Testing with customer data to check that system meets customer needs.

Testing phases in a plan-driven software process (V-model)



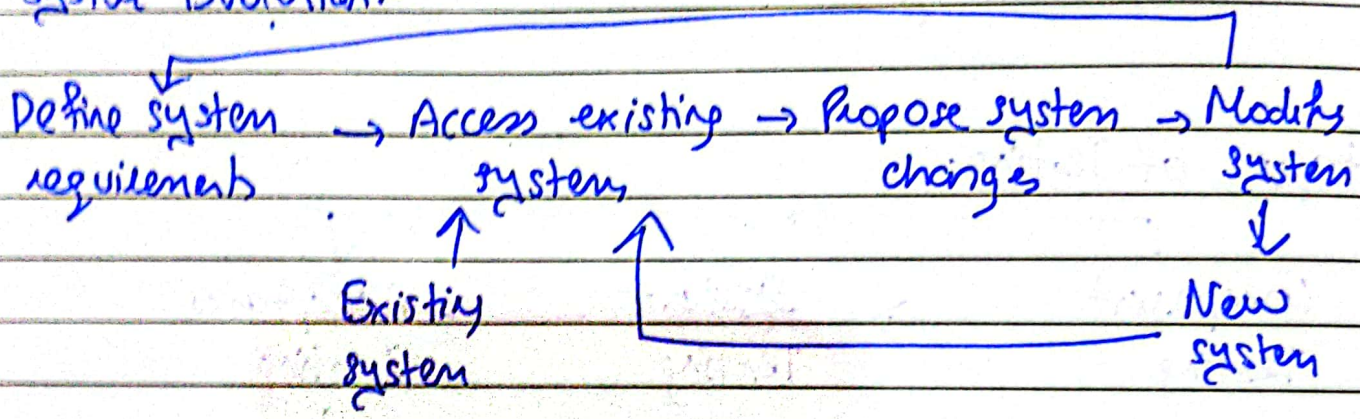
Date _____ 20__
M T W T F S

- Software evolution

  Software is inherently flexible and can change.

  As requirement change, software should change too.
  (increasingly irrelevant as fewer systems are new)

- Syster Evolution:-



- Coping with change

  - change is inevitable in all large software projects.
    - Business changes, new technologies,
  - change leads to rework so the cost of change include rework as well as cost of adding new functionality.

Date _____ 20__
M T W T F S

- Reduce cost of rework.

Change anticipation → activities that can anticipate possible changes before significant rework is required

Change tolerance → process is designed so that changes can be accomodated at low cost relatively.

- Software Prototyping:-

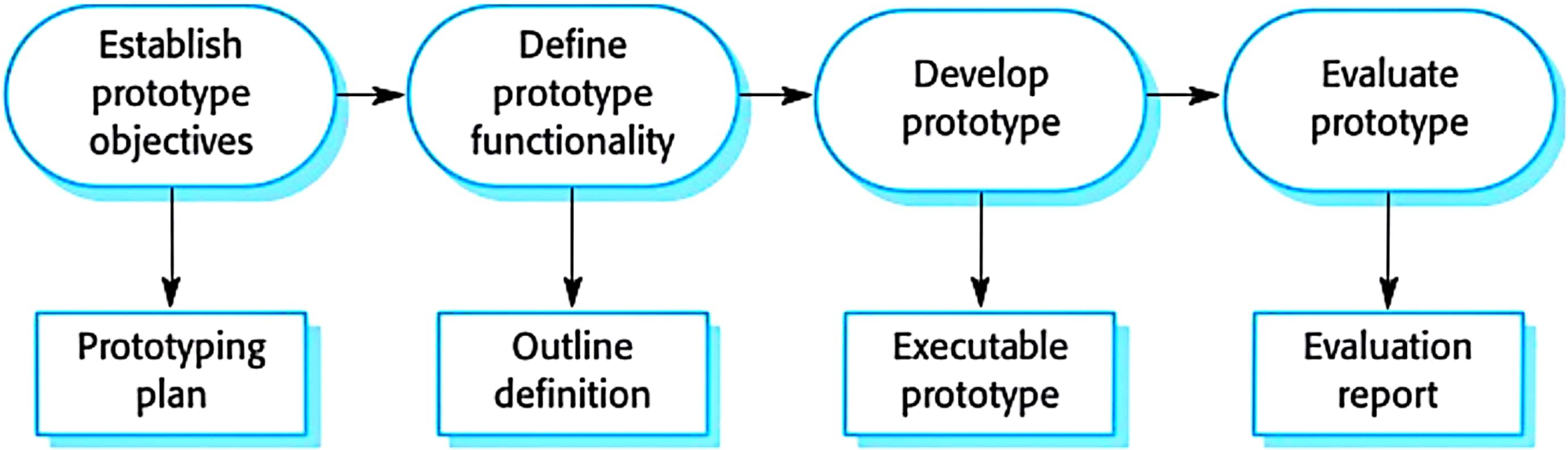
Prototype:- initial version of a system used to demonstrate concepts and try out design options.

Prototyping can be used in:-

- Requirement Engineering :- help with requirement elicitation & validation
- Design process:- develop UI design.
- Testing process:- run back-to-back tests.

Benefits:- - Improved system usability.
- Improved design quality.
- Improved maintainability.
- Reduced development effort.

The process of prototype development



Date 20
M T W T F S

- Prototype development

-May be based on rapid prototyping languages or tools.
-May involve leave out functionality.

Prototypes should focus on area of product not well understood.

Error checking and recovery may not be included in prototype.