

VV COLLEGE OF ENGINEERING

VV NAGAR

TISAYANVILAI



COURSE FILE

REGULATION 2017

B-Asha Devi

Designation: A P

Name of the staff:

Semester/Year:

IV / II

Branch: CSE

Subject Code/Name:

C58494

Software Engineering

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models Introduction to Agility Agile process Extreme programming-XP Process.

**UNIT II REQUIREMENTS ANALYSIS AND SPECIFICATION** 9

Software Requirements Functional and Non-Functional, User requirements, System requirements, Software Requirements Document – Requirement Engineering Process: Feasibility Studies, Requirements elicitation and analysis, requirements validation, requirements management, classical analysis: Structured system Analysis, Petri Nets- Data Dictionary.

**UNIT III SOFTWARE DESIGN** 9

Design process – Design Concepts-Design Model– Design Heuristic – Architectural Design - Architectural styles, Architectural Design, Architectural Mapping using Data Flow- User Interface design: Interface analysis, Interface Design –Component level Design: Designing Class based components, traditional Components

**UNIT IV TESTING AND MAINTENANCE** 9

Software testing fundamentals-Internal and external views of Testing-white box testing - basis testing with testing-control structure testing-black box testing- Regression Testing – Unit Testing – Integration Testing – Validation Testing – System Testing And Debugging –Software Implementation Techniques: Coding practices-Refactoring-Maintenance and Reengineering-BPFD-Model-Reengineering process model-Reverse and Forward Engineering.

**UNIT V PROJECT MANAGEMENT**

Software Project Management: Estimation – LOC, FP Based Estimation, Make/Buy Decision, COCOMO I & II Model – Project Scheduling – Scheduling, Earned Value Analysis Planning : Project Plan, Planning Process, RFP Risk Management – Identification, Projection - Risk management-Risk Identification-RMMM Plan-CASE TOOLS

**TOTAL :45 PERCENT**

**OUTCOMES:**

**On Completion of the course, the students should be able to:**

- Identify the key activities in managing a software project.
- Compare different process models.
- Concepts of requirements engineering and Analysis Modeling.
- Apply systematic procedure for software design and deployment.
- Compare and contrast the various testing and maintenance.
- Manage project schedule, estimate project cost and effort required.

**TEXT BOOKS:**

Roger S. Pressman, "Software Engineering – A Practitioner's Approach", Seventh Edition, Graw-Hill International Edition, 2010.  
Ian Sommerville, "Software Engineering", 9th Edition, Pearson Education Asia, 2011.

# CS8494 Software Engineering.

## Unit 1.

### Software Process and Agile

#### Development:

Introduction to Software Process

Engineering, Software Specialized

Perspective and Introduction to Process Models - Introduction to

Process - Extreme

Agility - Agile

- XP Process.

Programming -

# UNIT 1 Process and Software Development.

## INTRODUCTION:

What is Software?

computer programs and associated documentation

What is Software Engineering?

It is defined as a systematic approach to develop and maintain a software product in a cost effective and efficient way.

## SOFTWARE CHARACTERISTICS.

1. The Software is developed and engineered, it is not manufactured.

2. Software  
out.

3. Most software is currently built rather than being assembled from existing components.

## CATEGORIES OF SOFTWARE.

1. System Software
2. Engineering Software
3. Embedded Software
4. Application Software
5. Product Line Software
6. Web Applications
7. Artificial Intelligence Software.

1. Maintainability.
2. Software must evolve to meet changing needs.
3. Dependability
4. Software must be buildworthily
5. Efficiency.
6. Usability
7. Software must be usable by the users for which it was designed.

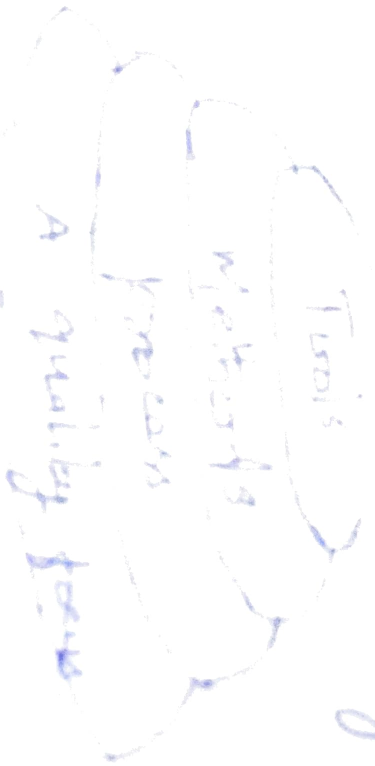
## CHALLENGES of Software

Engineering.

1. Legacy Systems.
2. Old valuable systems must be maintained and updated
3. Heterogeneity.
4. Delivery

# SOFTWARE

Software Layer



## PROCESS

## SOFTWARE

It is a software

and associated resources produces a software product

These activities are

1. Software Requirements
2. Software Development
3. Software Validation
4. Software Evolution

## Commonly Used Activities

1. Communication
2. Planning
3. Modeling
4. Construction
5. Deployment

## Umbrella Activities

1. Supplement project tracking and control.
2. Risk management.
3. Software quality assurance.
4. Formal technical reviews.
5. Software configuration management.
6. Work product preparation and production.

Productivity Management



PROCEEDS

APPROVAL

EMM1 APPROV.

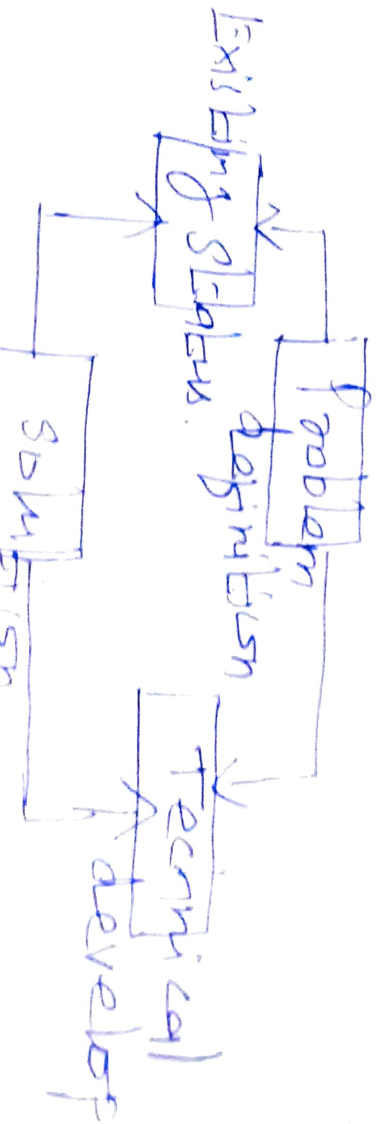
FOR

1. Standard Improvement for PROCEEDS
2. CMM based Appraisal of Internal Process Improvement
3. SPICE - The defines a set of requirements for software process ANSWER
4. ISO 9001-RISSE for Software

SOFTWARE ENGINEERING PARADIGM

The Phase of a Problem

Solving a Problem.



NEED FOR THE STATE MODEL

The primary goal of the life cycle model is to develop a software system in a systematic and disciplined manner.

Different life cycle models

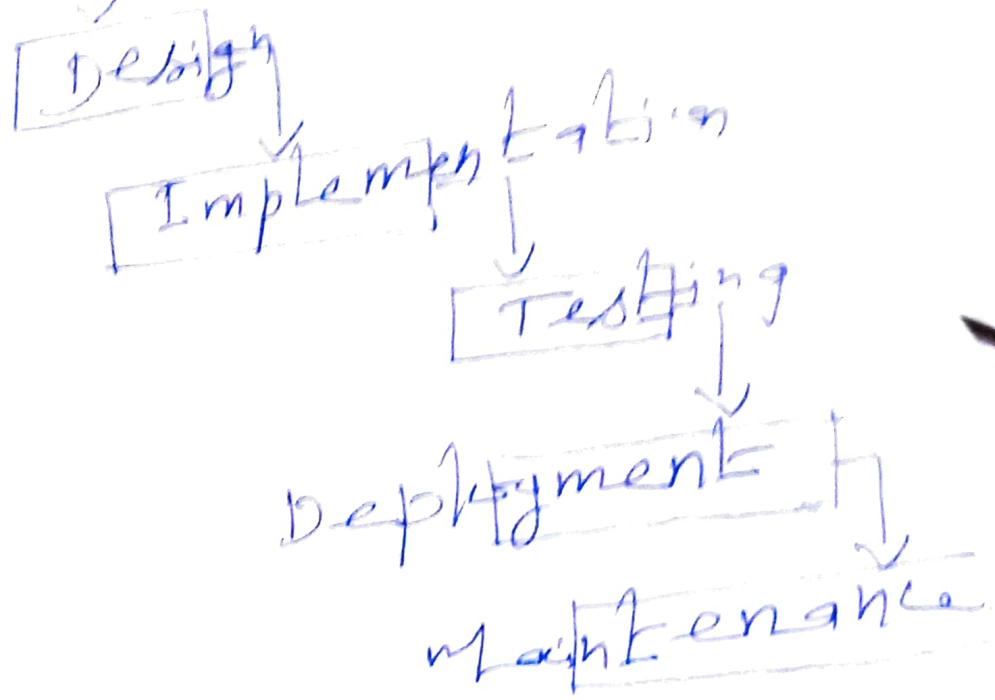
- i) Waterfall model.
- ii) Prototyping model.
- iii) Incremental model.
- iv) Spiral model.
- v) WIN WIN spiral model.
- vi) Evolutionary model.
- vii) Object-Oriented model.

i) Waterfall model.

It is the earliest.

one approach for software development.

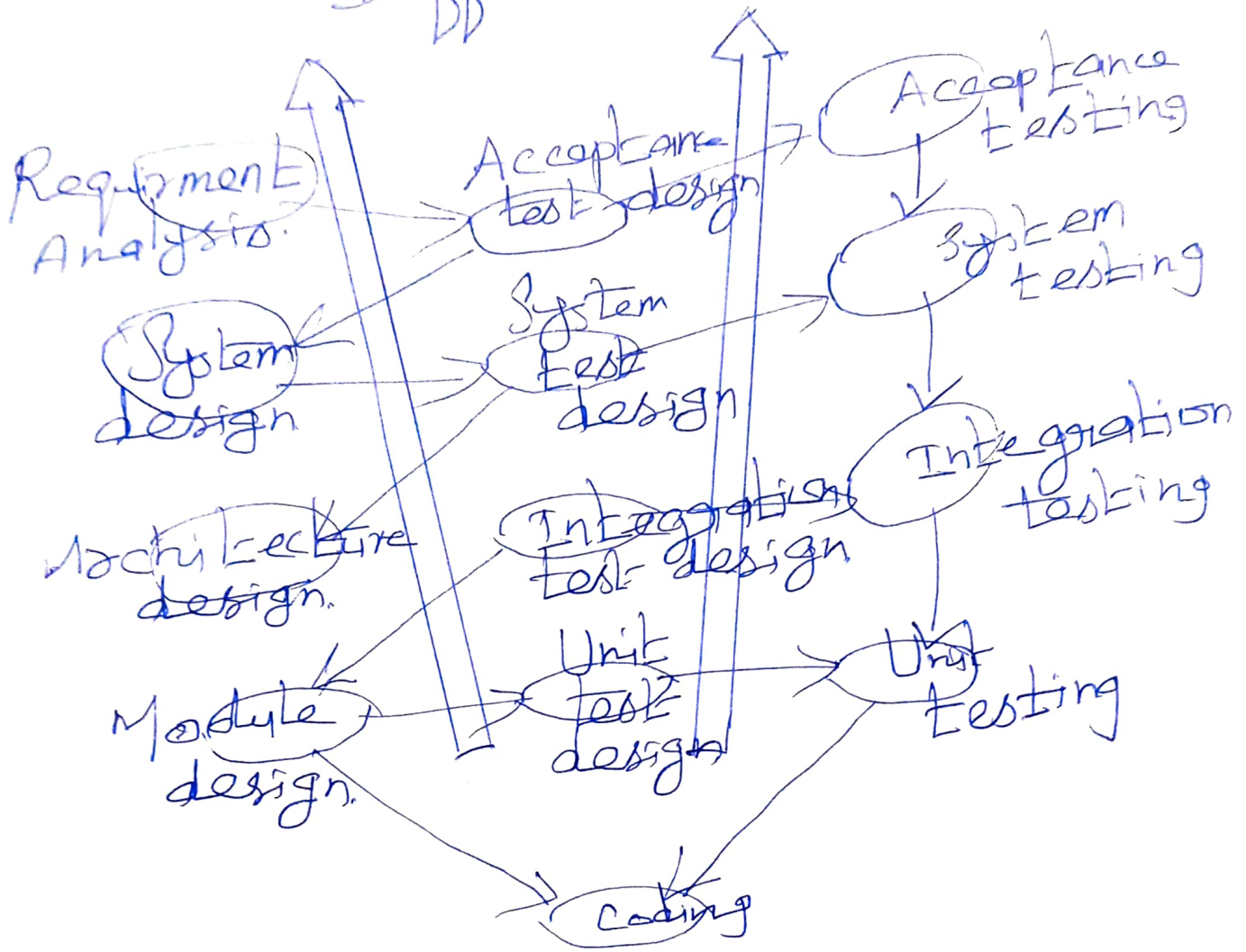
used for software development.



The sequential phases in the waterfall model are

1. Requirement Gathering and analysis.
2. System design.
3. Implementation.
4. Integration and testing.
5. Deployment of the system
6. Maintenance

(ii) Verification Different phases

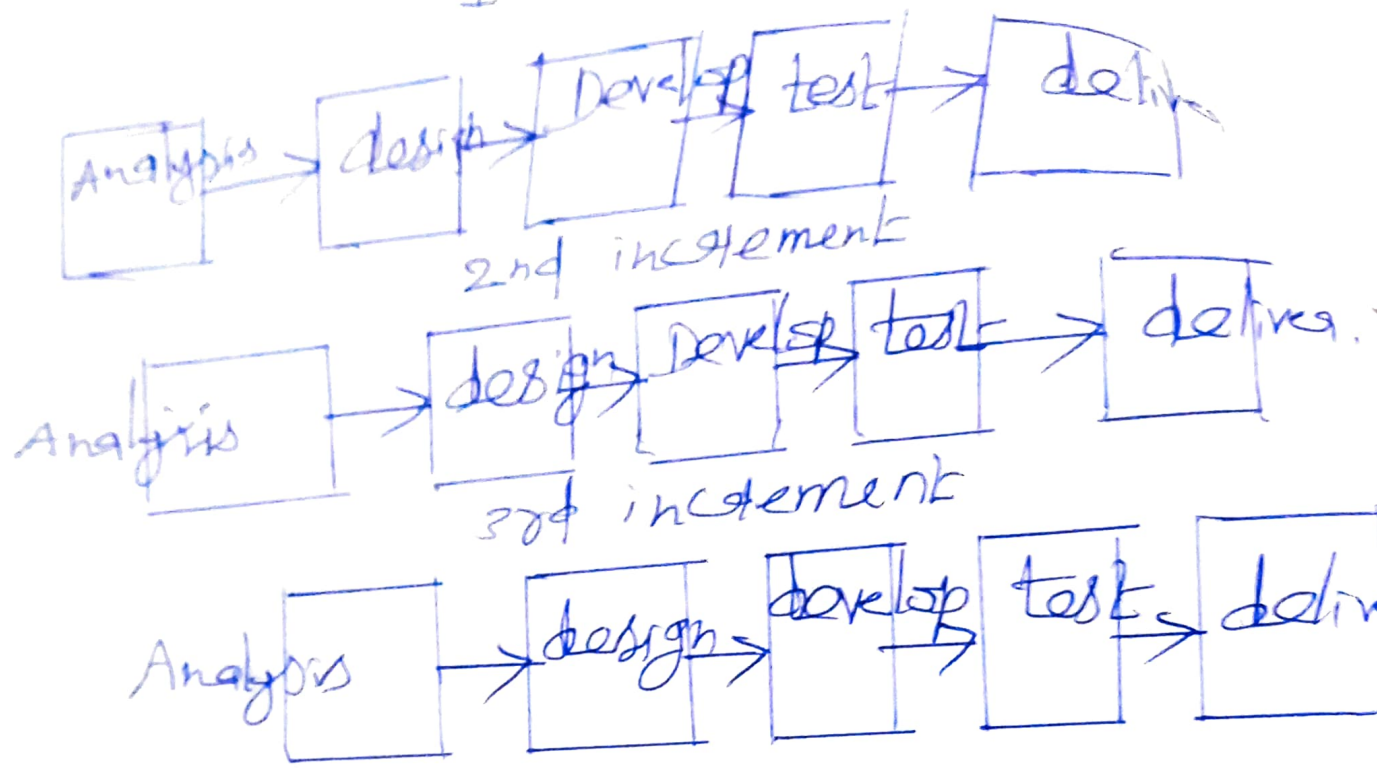


### Verification Phases.

1. System design.
2. Architecture design.
3. Module design.

### Validation Phases.

1. Unit testing.
2. Integration testing.
3. System testing.



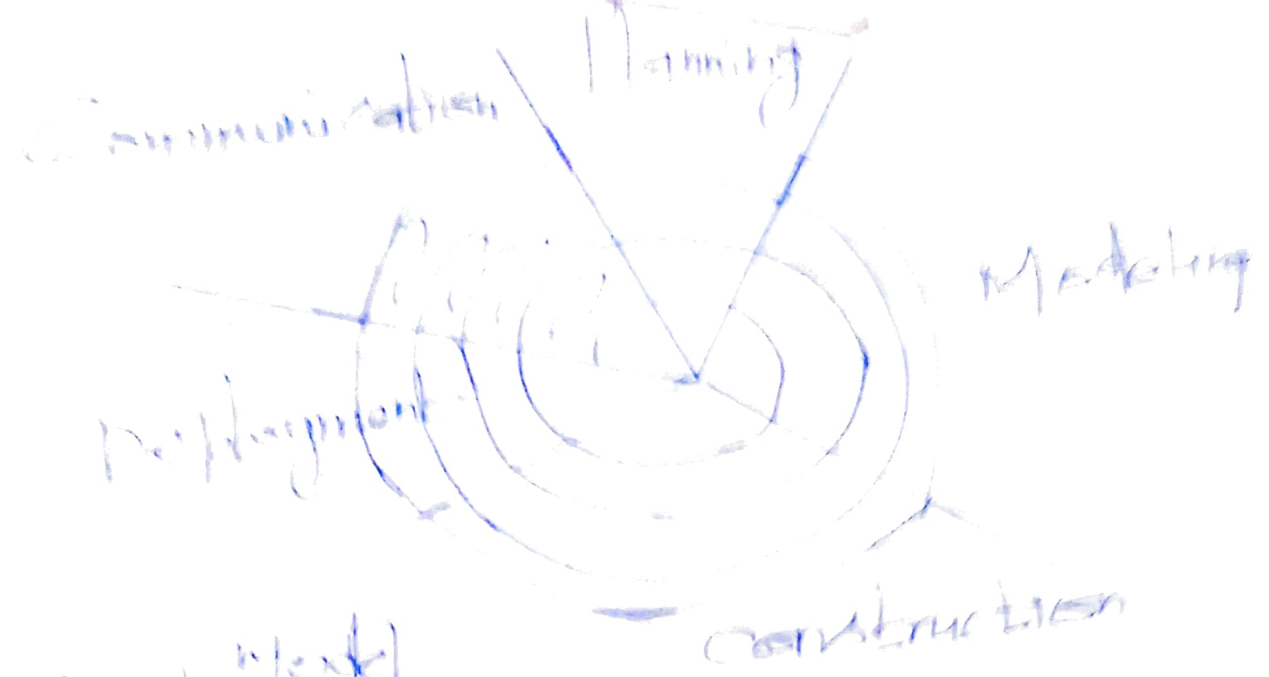
---

## Iterative and Incremental Model.

1. Risk analysis is better.
2. It supports changing requirements.
3. Initial operating time is less.
4. During life cycle software is produced early which facilitates customer evaluation and feedback.

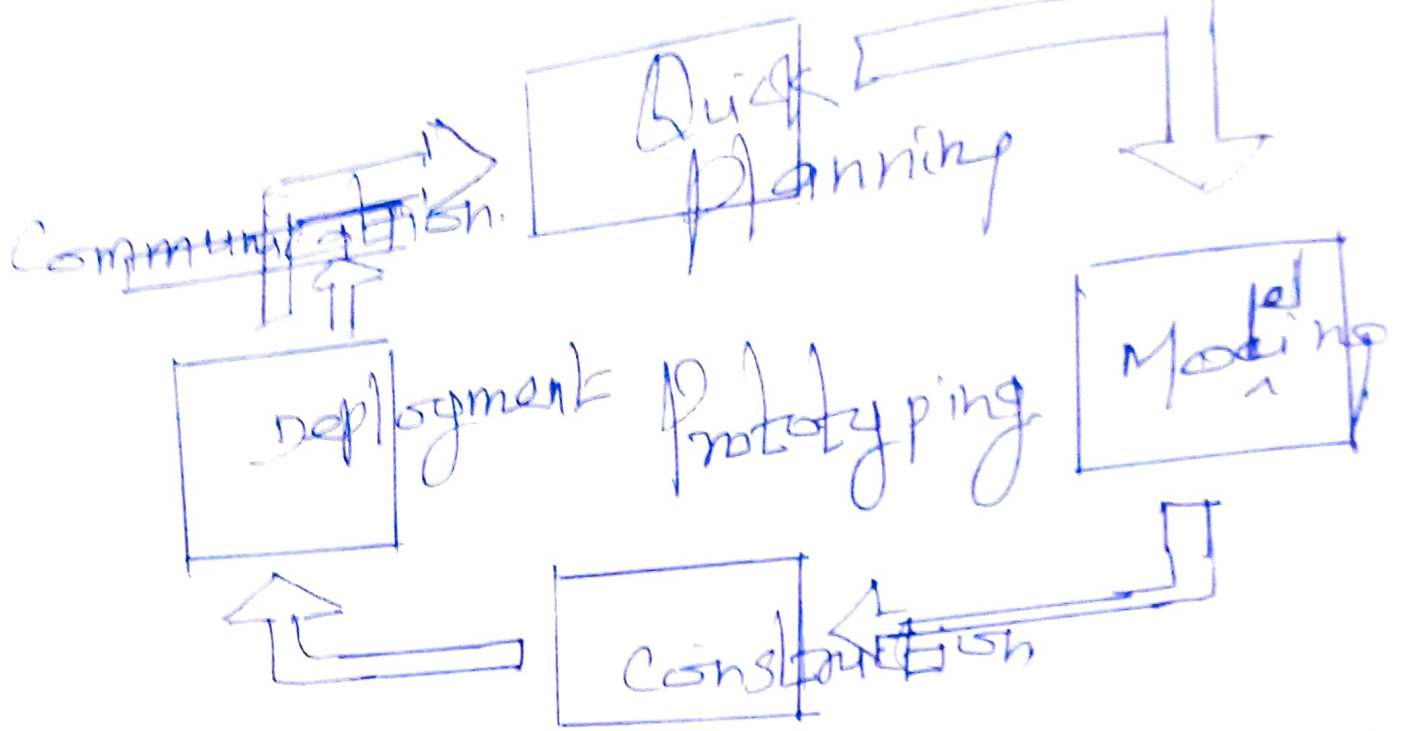
# Software Engineering Models

## Spiral Model



### Spiral Design

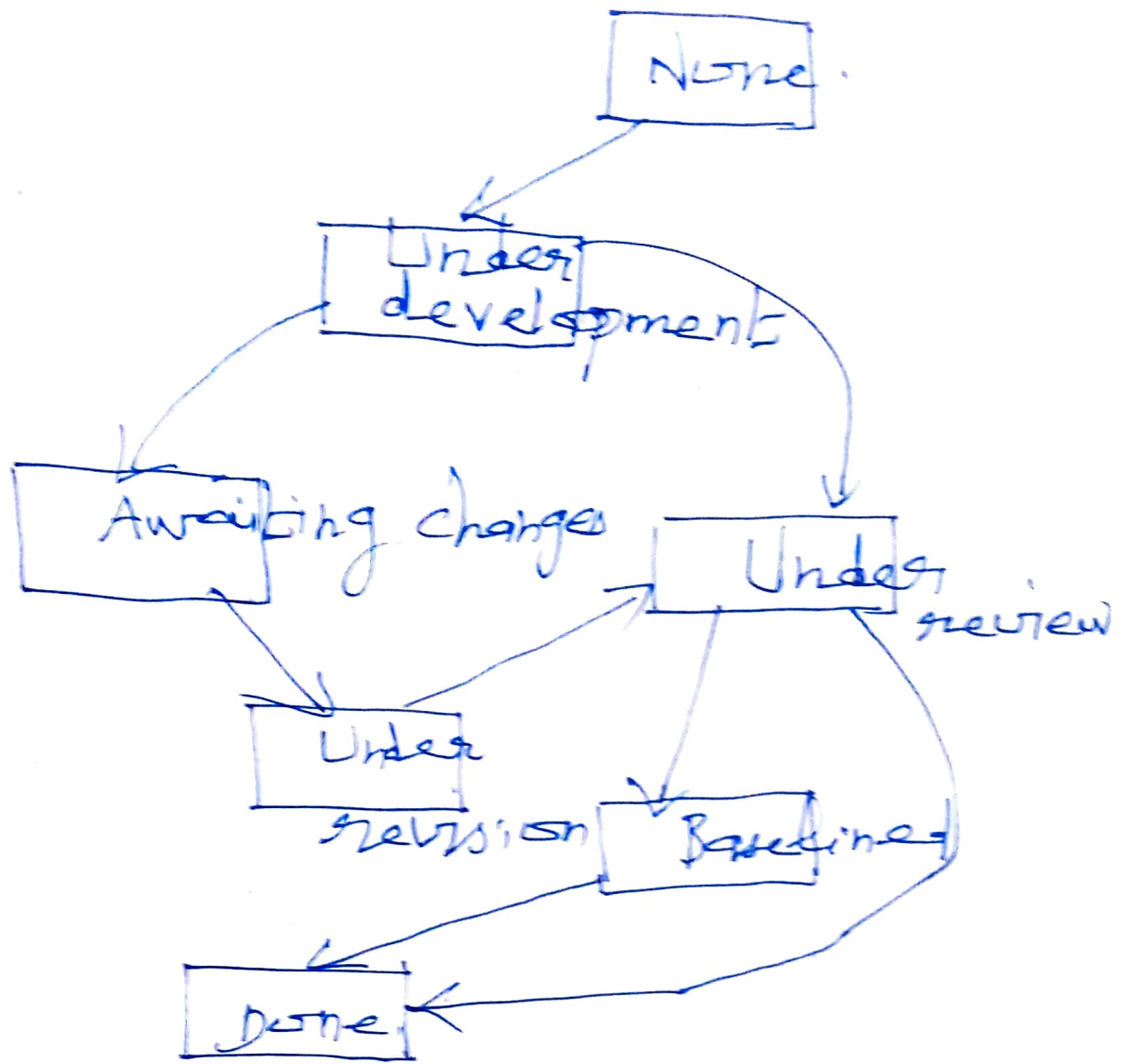
1. Identification
2. Design
3. Construct or Build
4. Evaluation and Risk Analysis.



The Prototyping Model is a system development in which a prototype is built, tested and then reworked.

### Advantages.

1. Errors can be detected much earlier as the system.
2. User feedback is available leading to better solutions.



It is sometimes called concurrent engineering can be represented frame work activities.

Software Engineering actions of tasks and their associated states.



- Spec.
1. Component based
  2. The Formal Methods Mo.
  3. Aspect-oriented software development.

## Introduction to AGILE development.

1. program specification design and implementation are interleaved.

## What is Agility?

- Effective communication among all stakeholders
- Organizing a team that it is in control of the work

What is an Agile

An Agile Software process must adapt incrementally.  
Agile process models.

1. Delivers multiple software increments.
2. Adapts as changes occur.

Agile principles.

The highest priority of this process is to satisfy the customer.

Acceptance of changing requirement even late in development.

# Competence

2. Common focus

3. Collaboration

4. Decision making ability

5. Fuzzy problem solving

ability

6. Mutual trust and respect

7. Self organization

2. Extreme programming (\*)

1. Communication
2. Simplicity
3. Courage
4. Feedback
5. Respect.

### 3. The XP process?

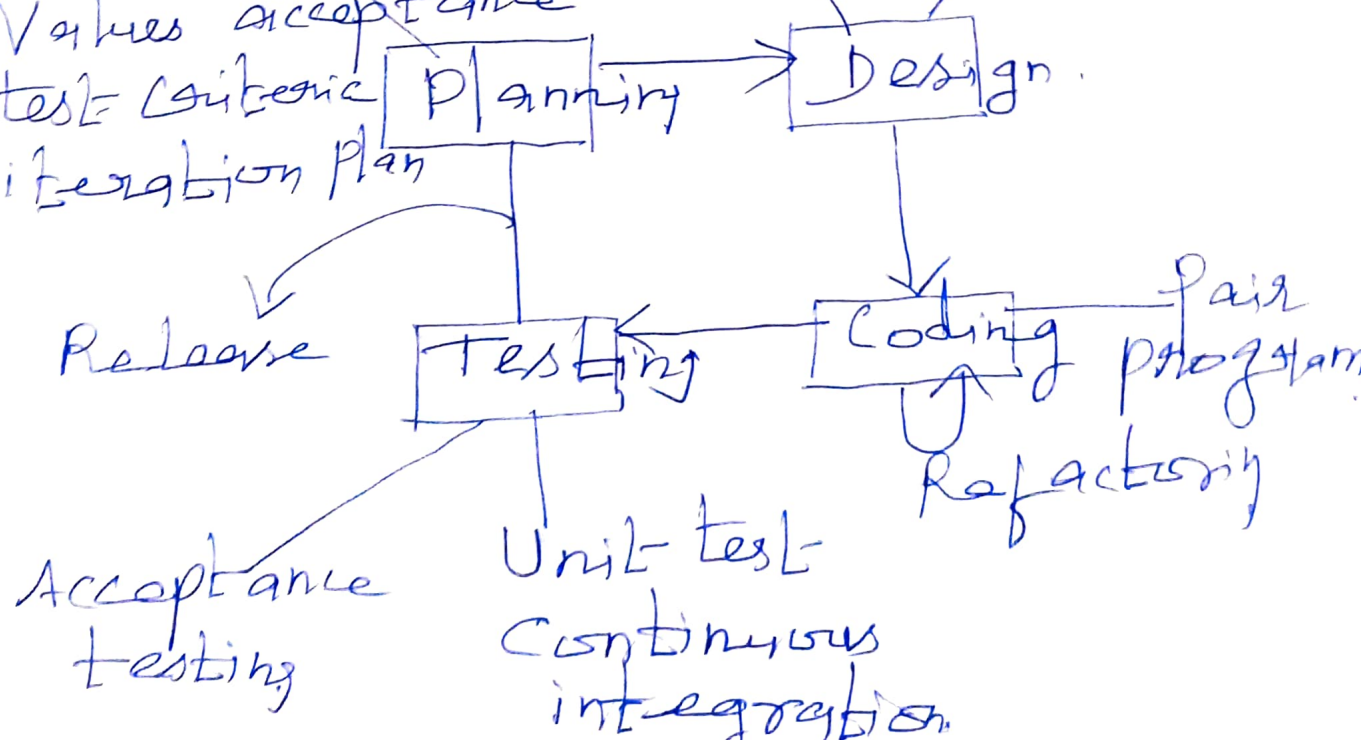
#### The XP process

Comprises four framework

activities.

user stories  
 Values acceptance  
 test criteria  
 iteration plan

Simple design  
 CRC cards  
 Spice Solutions  
 prototypes



## 1. Planning

- Planning starts with the requirements gathering which enables xp team to understand the rules for the software.
- The customer and developers work together for the final requirements.

## 2. Design.

- The xp design follows the 'keep it simple' principle
- A simple design always prefers the more difficult representation.

Encourage refactoring - an  
iterative refinement of the  
overall program design  
that revisits the code modification  
of implementing small design  
changes that may improve the  
design.

### 3. Coding

- The coding is started  
after the initial design work  
is over.

- After the initial  
design work is done, the team  
creates a set of unit tests  
which can test each distributed

that should be a part of  
the release.

## Testing

- Validation testing of the system occurs on a daily basis. It gives the XP team a regular indication of the progress.
- XP Acceptance tests are known as the customer tests.
- Acceptance tests are defined by the customer user stories and executed to assess customer visible functionality.

Users who don't have formal  
Knowledge  
→ Understandable by system  
→ natural language  
→ tables, diagrams

Problems

① Lack of clarity

② Confusion

③ Analogization

guidelines

Invent a standard format  
& ensure all fit to requirements

Definitions

Use blue language consistently  
& avoid modality

distinguish with

Text

high light for  
bold.

words

like

Avoid

Use of

foreign.