Software Design & Architecture
(Week-2)

Usama Musharaf

MS-CS (Software Engineering)

LECTURER *(Department of Computer Science)*

FAST-NUCES Peshawar

Agenda of Week # 2

- Overview of Software Process Models

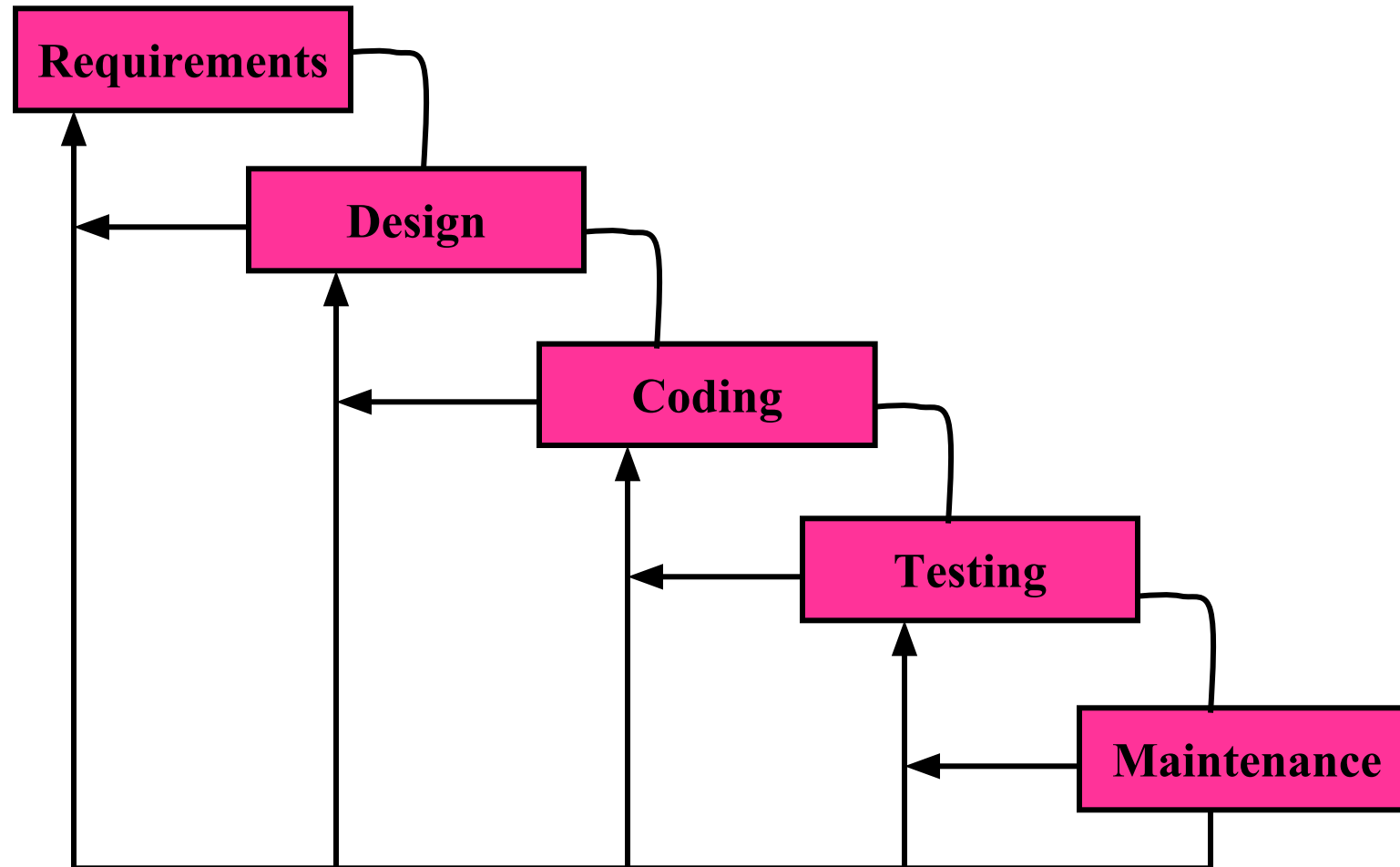- Agile Software Engineering

  - Extreme Programming
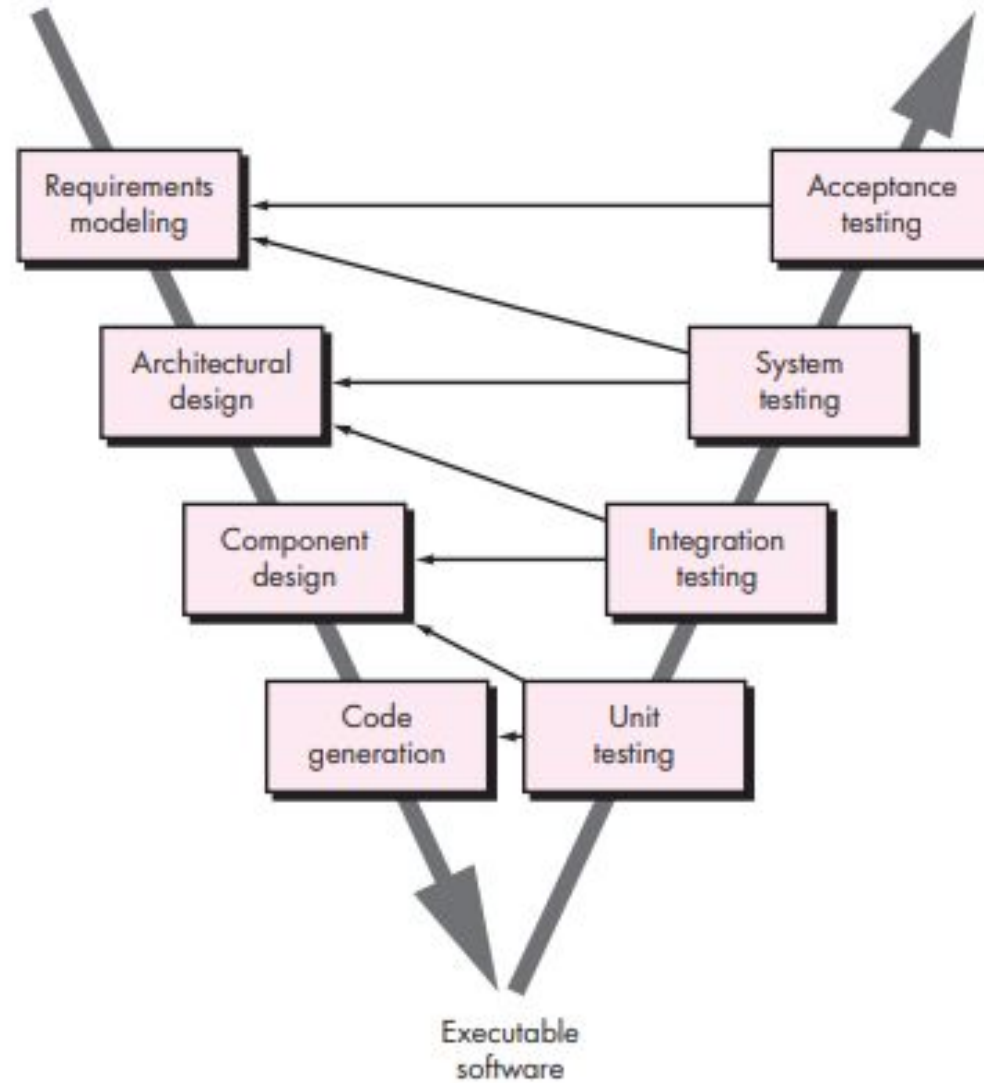
  - Scrum

Software process models

1. **Waterfall model**
2. **V Model**
3. **Rapid prototyping model**
4. **Evolutionary s/w process Model**
   - **Incremental model**
   - **Spiral Model**
5. **Agile Development**
   1. **Extreme Programming**
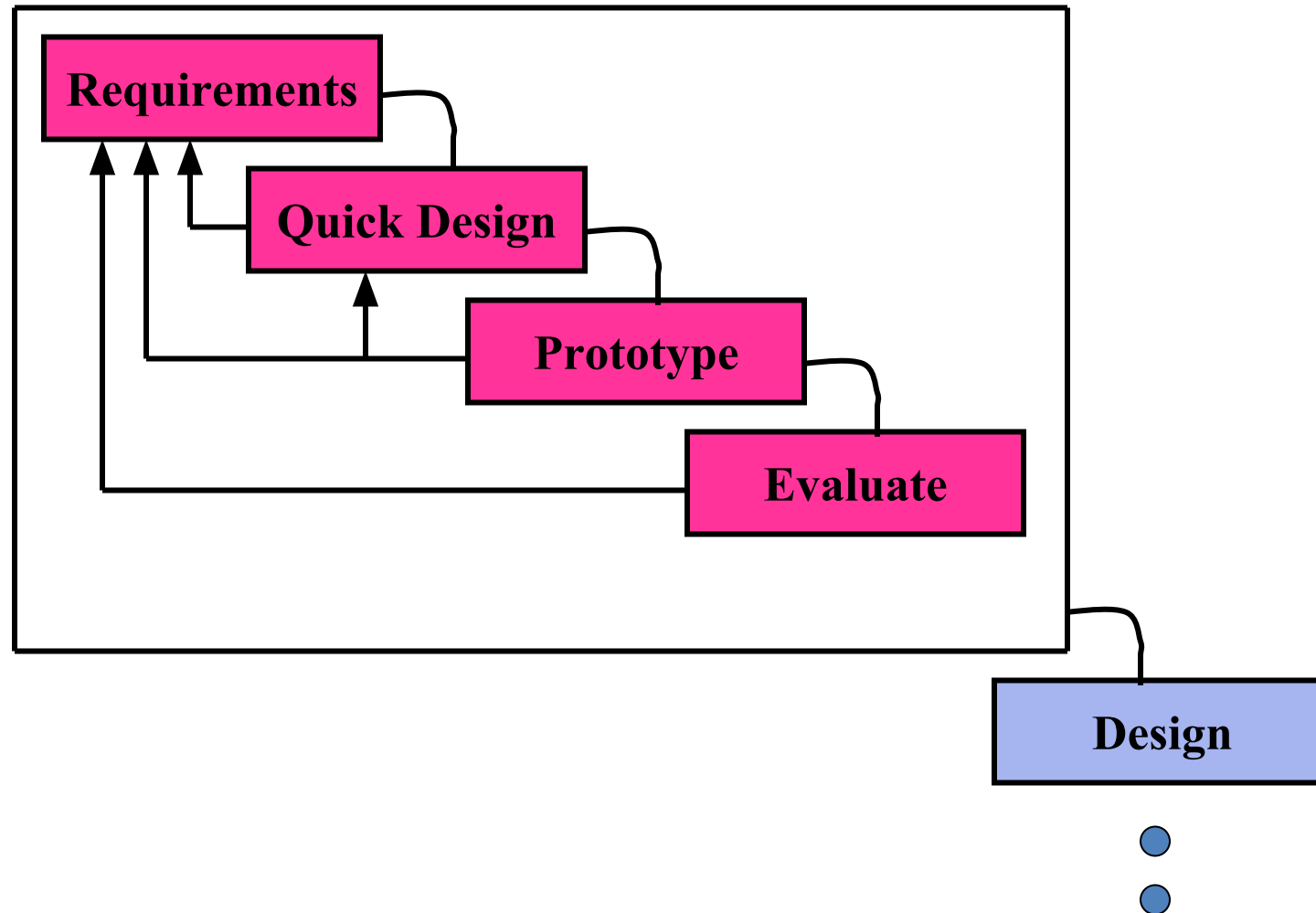   2. **Scrum**

Waterfall Process Model
AKA
Linear sequential model
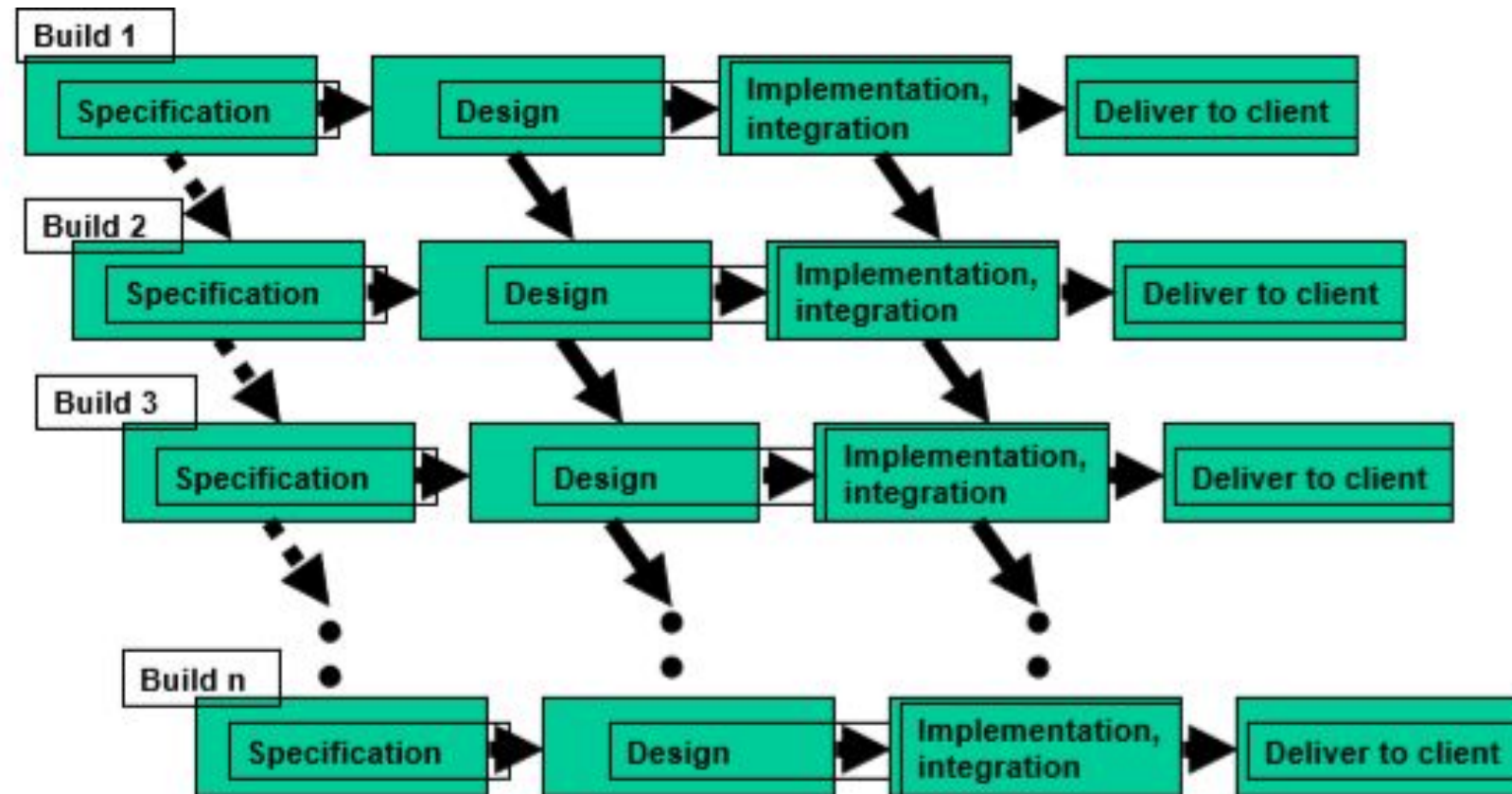
V Model

Rapid Prototyping Process Model

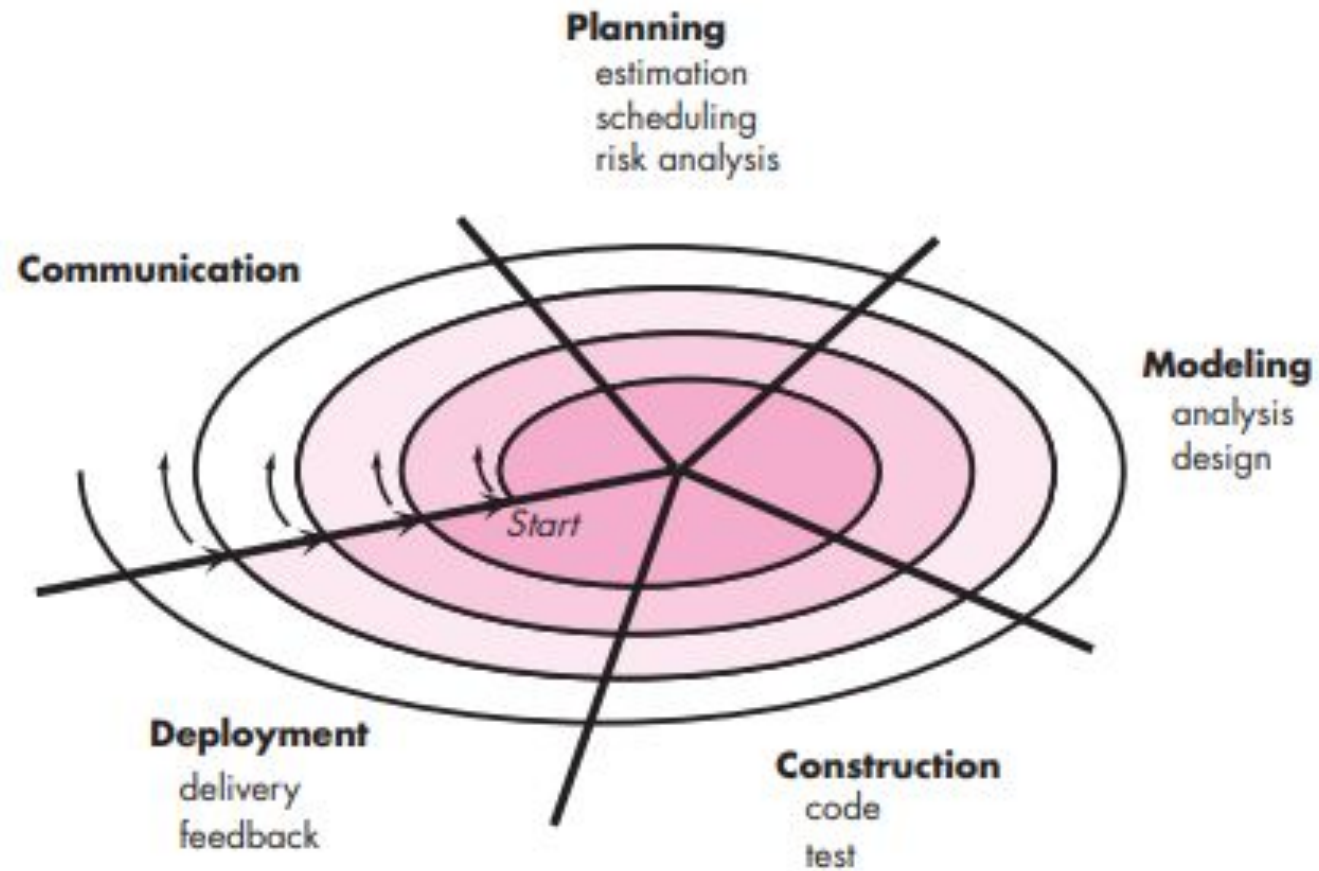# Evolutionary s/w process Model

1. **Incremental model**
2. **Spiral Model**

Incremental or Iterative Process Model

Spiral Process Model

Agile Development

Common Fears for Developers

- The project will produce the wrong product.

- The project will produce a product of inferior quality.

- The project will be late.

- We'll have to work 80 hour weeks.

What is "Agility"?

- Effective (rapid and adaptive) response to change.
- Effective communication among all stakeholders.

Yielding …

- Rapid, incremental delivery of software

An Agile Process

 Is driven by customer descriptions of what is required (scenarios)

 Develops software iteratively with a heavy emphasis on construction activities

 Delivers multiple 'software increments'

 Adapts as changes occur

Principles of agile methods

Principle	Description
Customer involvement	The customer should be closely involved throughout the development process. Their role is provide and prioritise new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognised and exploited. The team should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and design the system so that it can accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process used. Wherever possible, actively work to eliminate complexity from the system.

Agile process models

- Extreme Programming (XP)
- Scrum
- Adaptive Software Development
- Dynamic System Development Method (DSDM)
- Crystal
- Feature Driven Development
- Agile Modeling (AM)

Extreme Programming (XP)

- Perhaps the best-known and most widely used agile method.

- Extreme Programming (XP) takes an 'extreme' approach to iterative development.

  - New versions may be built several times per day;
  - Increments are delivered to customers every 2 weeks;
  - All tests must be run for every build and the build is only accepted if tests run successfully.

Extreme Programming (XP)
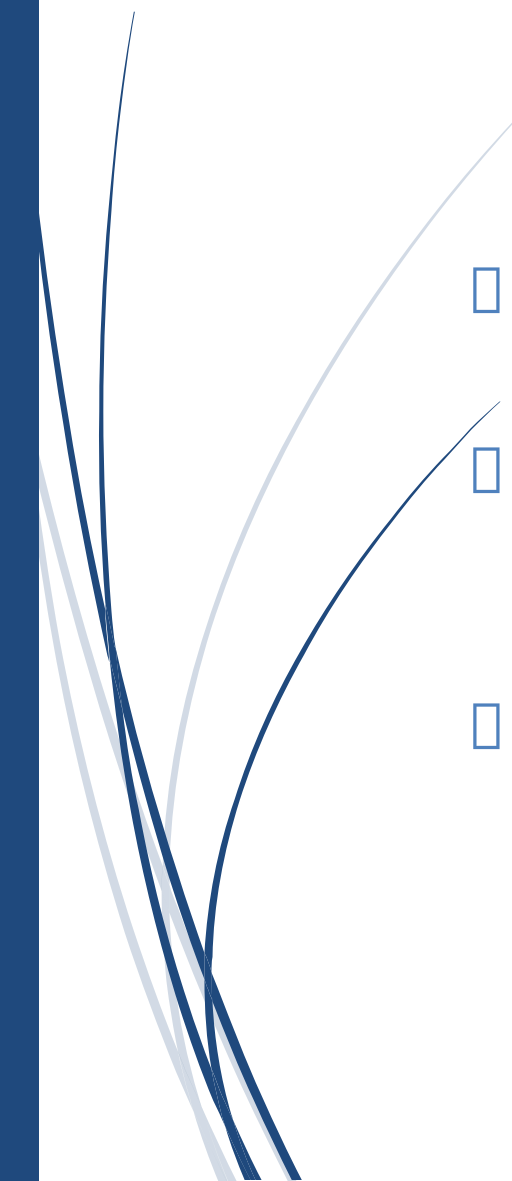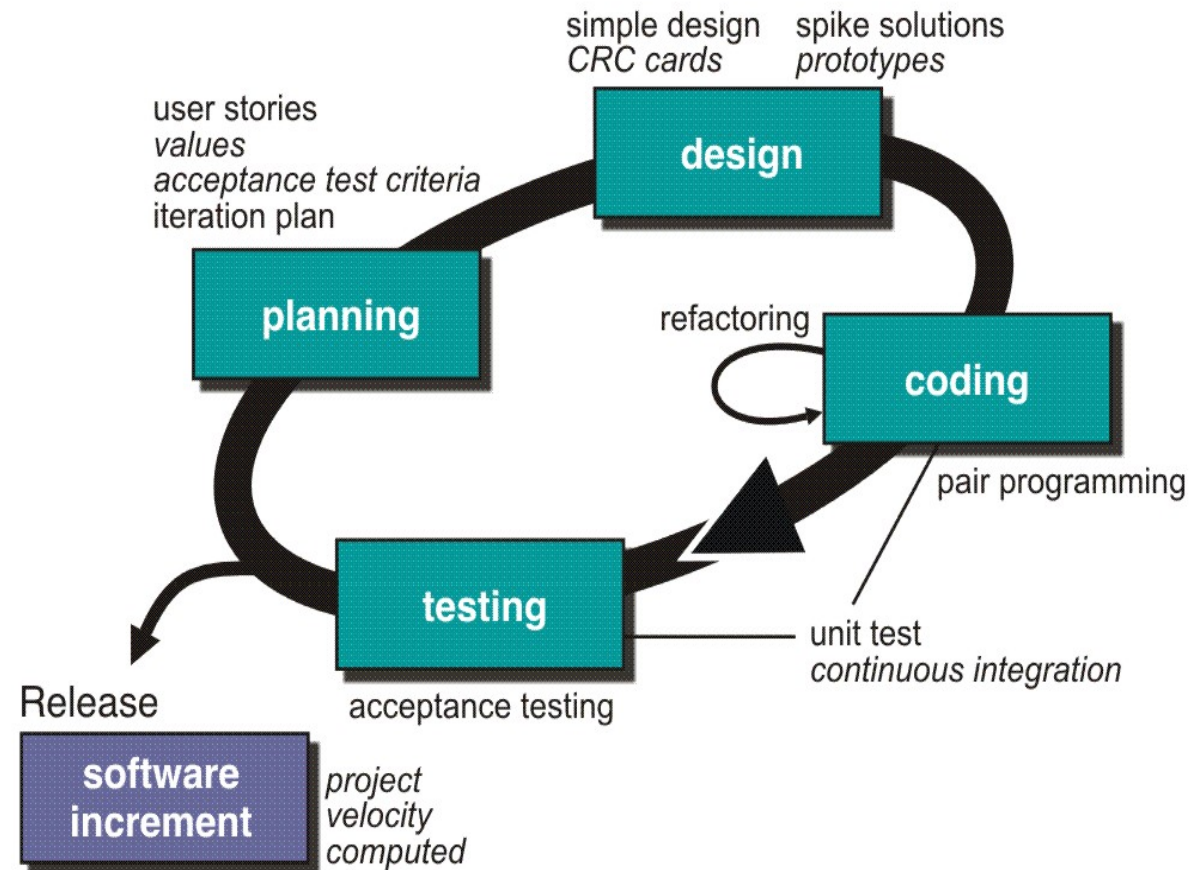
Extreme Programming (XP)

XP Planning

- Begins with the creation of user stories

- Agile team assesses each story and assigns a cost

- Stories are grouped to for a deliverable increment

- A commitment is made on delivery date

Requirements scenarios

- In XP, user requirements are expressed as scenarios or user stories.

- These are written on cards and the development team break them down into implementation tasks. These tasks are the basis of schedule and cost estimates.

- The customer chooses the stories for inclusion in the next release based on their priorities and the schedule estimates.

Story card for document downloading

Downloading and printing an article

First, you select the article that you want from a displayed list. You then have to tell the system how you will pay for it - this can either be through a subscription, through a company account or by credit card.

After this, you get a copyright form from the system to fill in and, when you have submitted this, the article you want is downloaded onto your computer.

You then choose a printer and a copy of the article is printed.   You tell the system if printing has been successful.

If the article is a print-only article, you canÕt keep the PDF version so it is automatically deleted from your computer .

Extreme Programming (XP)

XP Design

- Follows the KIS (keep it simple) principle

- Encourage the use of CRC (class-responsibility-cards) cards

- For difficult design problems, suggests the creation of spike solutions — a design prototype

- Encourages refactoring — an iterative refinement of the internal program design

Extreme Programming (XP)

CRC Cards:

Class-responsibility-collaboration (CRC) cards are a tool used in the design of object-oriented software.

Extreme Programming (XP)

CRC Cards:

The card is partitioned into three areas:

I.   On top of the card, the class name

II.   On the left, the responsibilities of the class

III.   On the right, collaborators (other classes) with which this class interacts to fulfill its responsibilities.

Extreme Programming (XP)

CRC Cards:

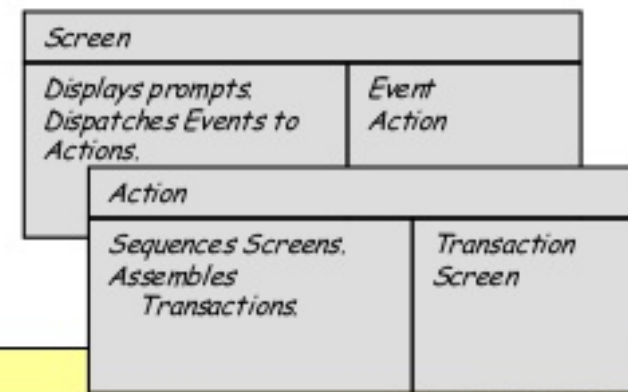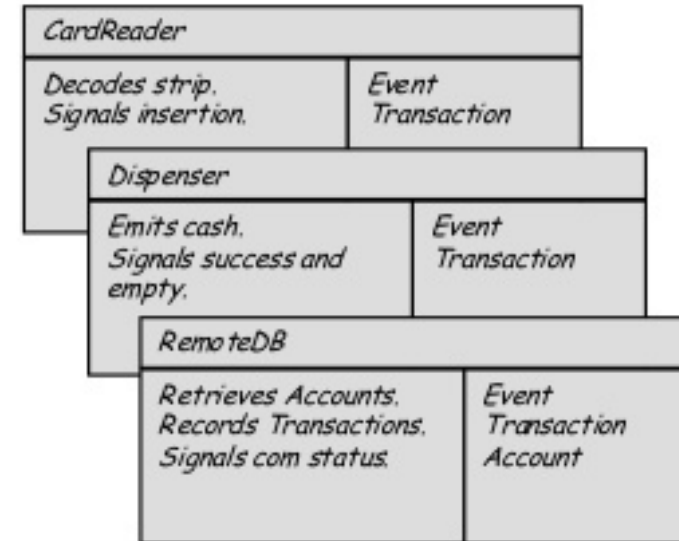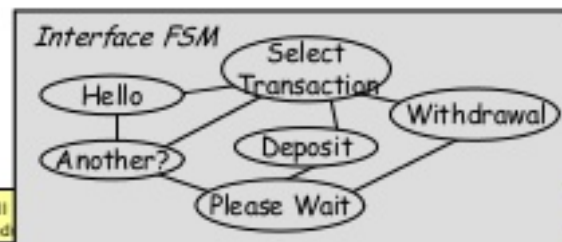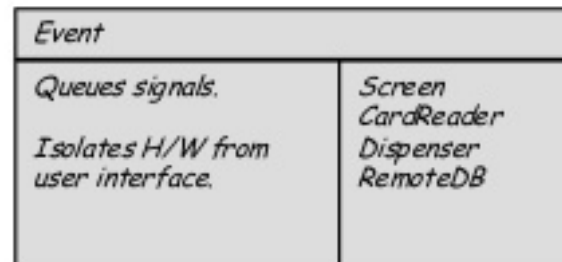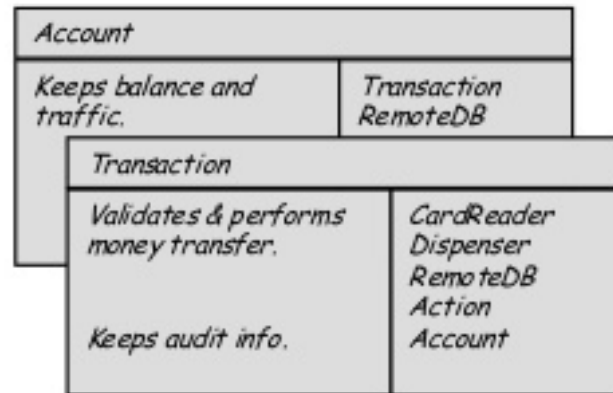| Class Name	
Responsibilities	Collaborators

Extreme Programming (XP)

CRC Cards:

Student	
Student number Name Address Phone number Enroll in a seminar Drop a seminar Request transcripts	Seminar

CRC example: ATM software

[Beck, Cunningham]

Account	
Keeps balance and traffic.	Transaction RemoteDB

Transaction	
Validates & performs money transfer. Keeps audit info.	CardReader Dispenser RemoteDB Action Account

Event	
Queues signals. Isolates H/W from user interface.	Screen CardReader Dispenser RemoteDB

CardReader	
Decodes strip. Signals insertion.	Event Transaction

Dispenser	
Emits cash. Signals success and empty.	Event Transaction

RemoteDB	
Retrieves Accounts. Records Transactions. Signals com status.	Event Transaction Account

Screen	
Displays prompts. Dispatches Events to Actions.	Event Action

Action	
Sequences Screens. Assembles Transactions.	Transaction Screen

Interface FSM

- Hello
- Select Transaction
- Another?
- Deposit
- Withdrawal
- Please Wait

Extreme Programming (XP)

XP Coding

- Recommends the construction of a unit test for a store *before* coding commences

- Encourages pair programming

XP Testing

- All unit tests are executed daily

- Acceptance tests are defined by the customer and executed to assess customer visible functionality

Testing in XP

 Test-first development.

 Incremental test development from scenarios.

 User involvement in test development and validation.

Task cards for document downloading

Task 1: Implement principal workflow

Task 2: Implement article catalog and select

Task 3: Implement payment collection

Payment may be made in 3 dif ferent ways. The user selects which way they wish to pay . If the user has a library subscription, then they can input the subscriber key which should be checked by the system. Alternatively, they can input an or ganisational account number . If this is valid, a debit of the cost of the article is posted to this account. Finally , they may input a 16 digit credit card number and expiry date. This should be checked for validity and, if valid a debit is posted to that credit card account.

Test case description

Test 4: Test credit card validity

Input:
A string representing the credit card number and two integers representin
the month and year when the card expires
Tests:
Check that all bytes in the string are digits
Check that the month lies between 1 and 12 and the
year is greater than or equal to the current year .
Using the first 4 digits of the credit card number ,
check that the card issuer is valid by looking up the
card issuer table. Check credit card validity by submitting the card
number and expiry date information to the card
issuer
Output:
OK or error message indicating that the card is invalid

Significance of Test-first development

 Writing tests before code clarifies the requirements to be implemented.

 Tests are written as programs rather than data so that they can be executed automatically. The test includes a check that it has executed correctly.

 All previous and new tests are automatically run when new functionality is added. Thus checking that the new functionality has not introduced errors.
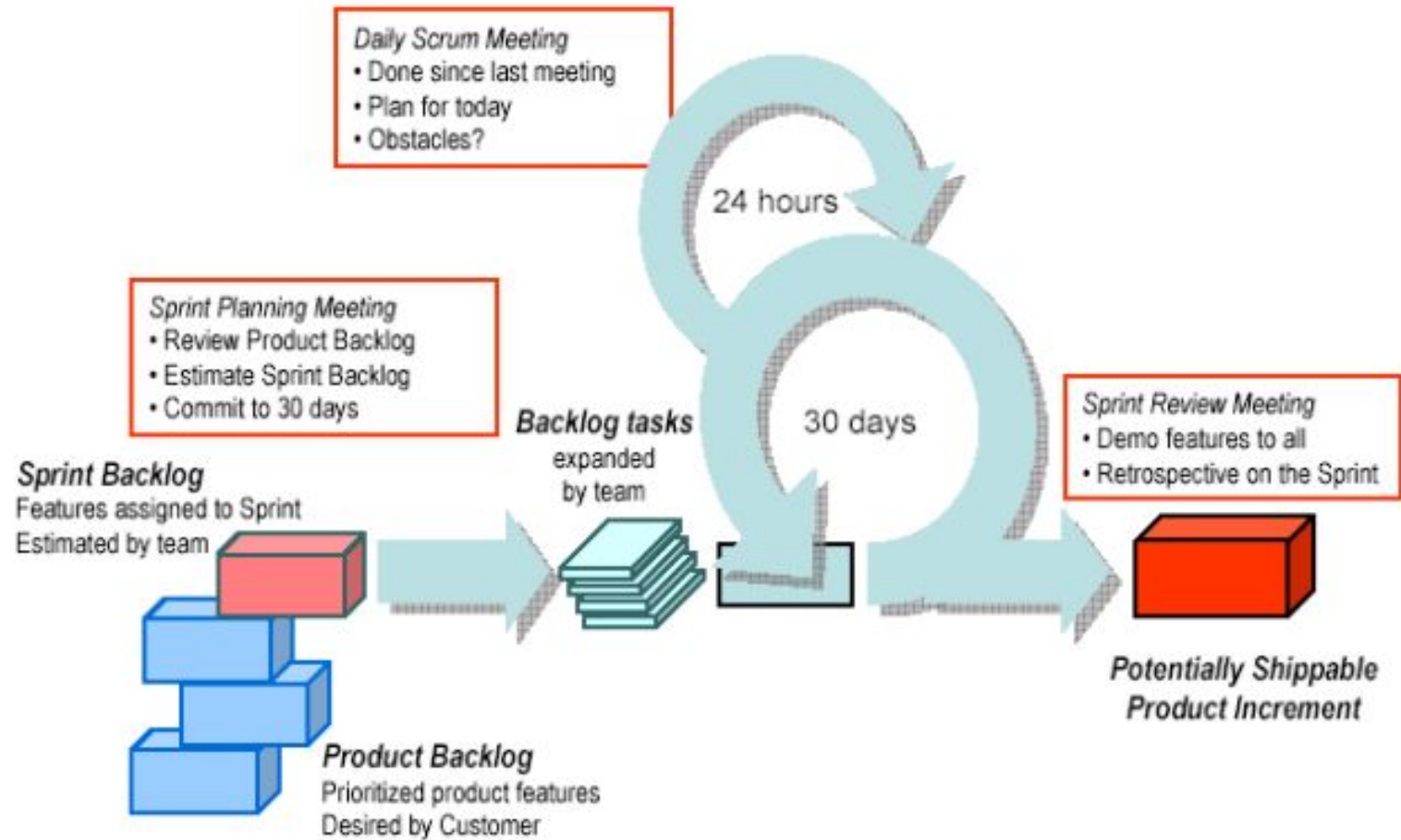
Cost of Change in Agile
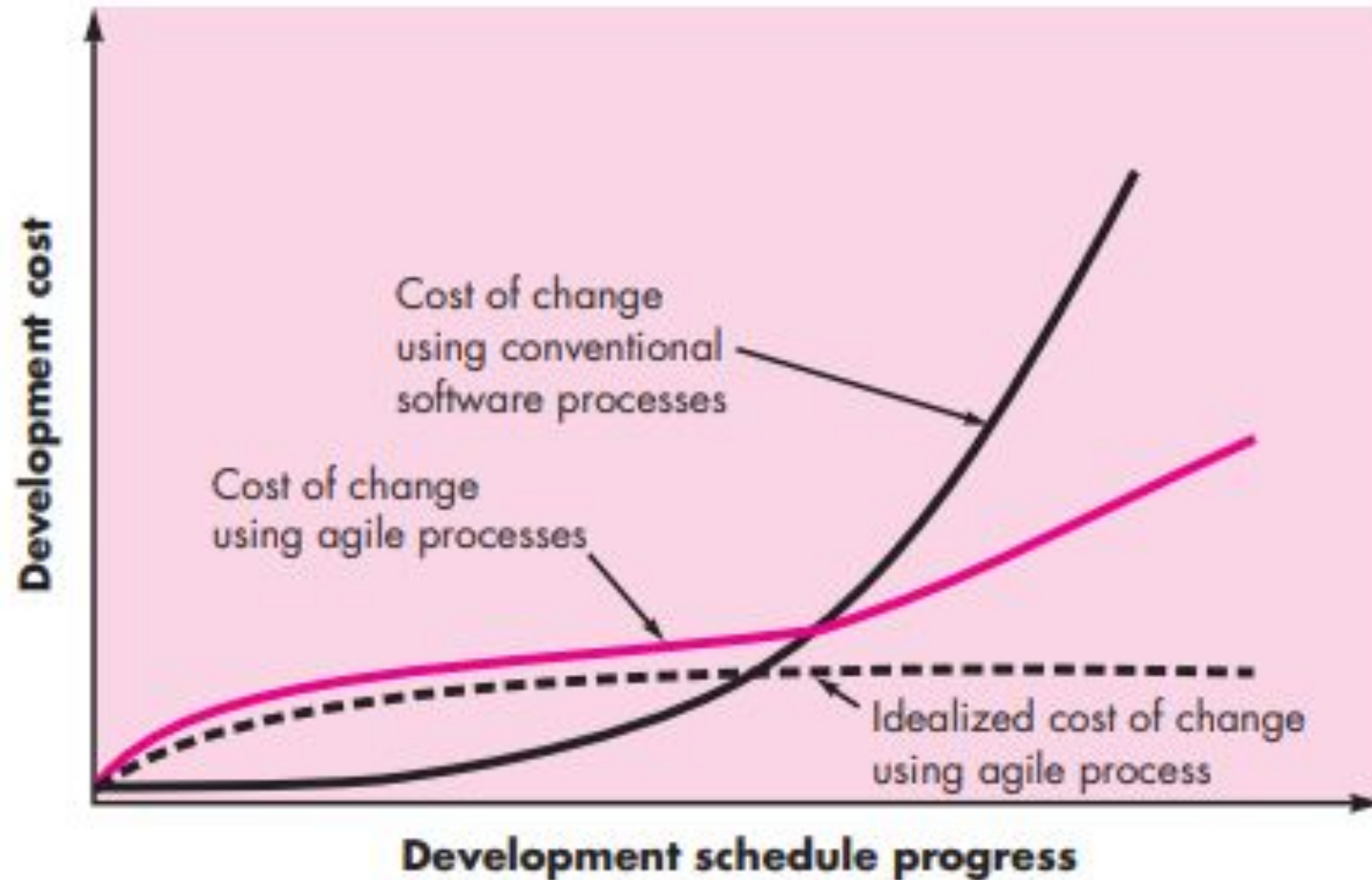
SCRUM

- Scrum is an Agile framework for completing complex projects.

- Scrum originally was formalized for software development projects, but it works well for any complex, innovative scope of work.

- Scrum is a team-based approach, to iteratively, incrementally develop systems and products.

- when requirements are rapidly changing .

How does SCRUM work?

- Small teams (< 10 people)
- A series of Sprints (1-4 weeks)
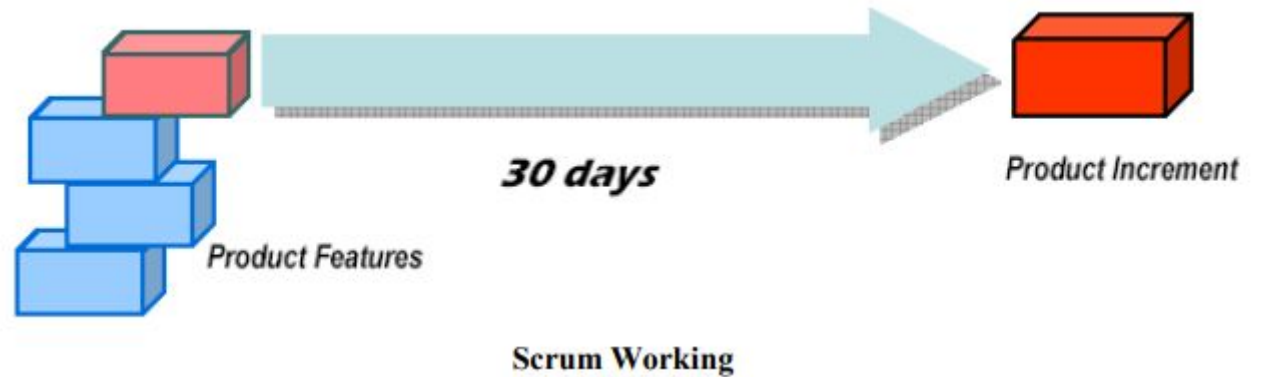
How does SCRUM work?

How Scrum work

HAVE A GOOD DAY !