Introduction of Software Engineering
Larina Wang

School of Engineering and Mathematical Science
Department of CSIT

What is software?

- Computer software, or simply software, is a collection of data or computer instructions that tell the computer how to work. This is in contrast to physical hardware, from which the system is built and actually performs the work. In computer science and software engineering, computer software is all information processed by computer systems, programs and data. Computer software includes computer programs, libraries and related non-excutable data, such as online documentation or digital media. Computer hardware and software require each other and neither can be realistically used on its own.

Where is software?

Where is software?

Some popular software

Some popular software

Some popular software

Some popular software

Some popular software

Some popular software

Even on mobile devices

Importance of Software

- Software is everywhere!

- The economies of all developed nations are dependent on software

- Software costs often dominate system costs

- Software costs more to maintain than to develop!

Software Engineering

Definitions: Software Engineering

- Engineering:

  o Action of defining and sequencing necessary tasks for the creation of a new product or service

- Software Engineering:

  o Engineering discipline concerned with all aspects of software production

  o Captures the customers' business needs and specify the blueprints for the system, so that programmers can implement it

Definitions: Software Engineering

Software Engineering is the application of a **systematic**, **disciplined**, **quantifiable** approach to the development, operation and maintenance of software

Definitions: Software Engineering

- Software Engineers must:
  - o Adopt a systematic and organized approach to their work
  - o Use appropriate tools and techniques

… depending on:

  - o The problem to be solved
  - o The development constraints
  - o The available resources

… with a given **budget**

… before a given **deadline**

… while **changes** occur!

What's the problem?

Some problems encountered during software development:

- The software does not suit the users' needs.

- Difficult to expand after deployment.

- Poor documentation.

- Poor quality.

- Process of development costs more money and time than excepted.

A good software …

- delivers the required **functionality** and **performance** to the user

- is **maintainable** : evolves to meet changing needs

- is **dependable** : must be trustworthy and secure

- is **usable** by its users

- is **efficient** : should not make wasteful use of system resources

- is **reusable**: can be reused in another form or for other purposes

Skills of a Software Engineer

- A software engineer is not (only) a developer!
    - " It is not enough to do your best: you must **KNOW what to do, THEN do your best**"

    W. Edwards Deming

- Aside from the technical skills, a software engineer must have other skills :
    - ✓ Communication
    - ✓ Feedback
    - ✓ Motivation
    - ✓ Analytical
    - ✓ Planning
    - ✓ Risk assessment

SDLC : Software Development Lifecycle

SDLC : Definition and Actors

- **SDLC :** Process of creating and altering systems, and the models and methodologies that people use to develop these system

- **Actors**: Mainly (but not only)

SDLC: Steps

Step 1: Requirement Analysis

- Work closely with the customer

    o Customer can be

        ▪ A client

        ▪ Your employer

        ▪ Or yourself

- Focus on the "What", not the "How"

- End Result: **Functional Specification**

- Reviewed by :

    o Customer, Designer.
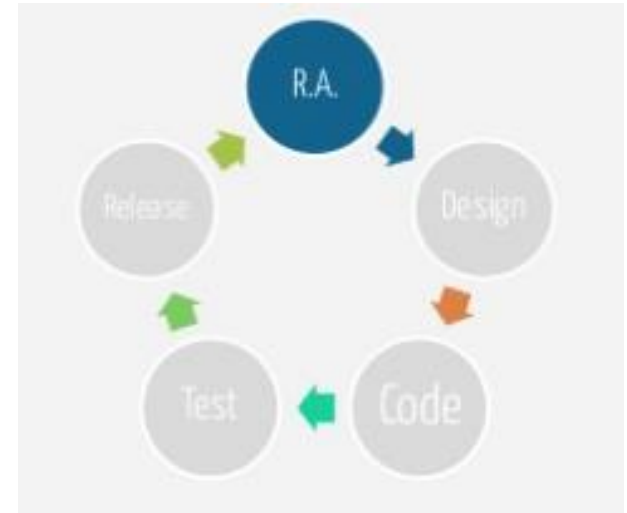
Step 2: Design

- Focus on the "How"

- Break the problem into smaller pieces

  - Components

- End Result: **Design Specification**

- Reviewed by :

  - Designer, Developer
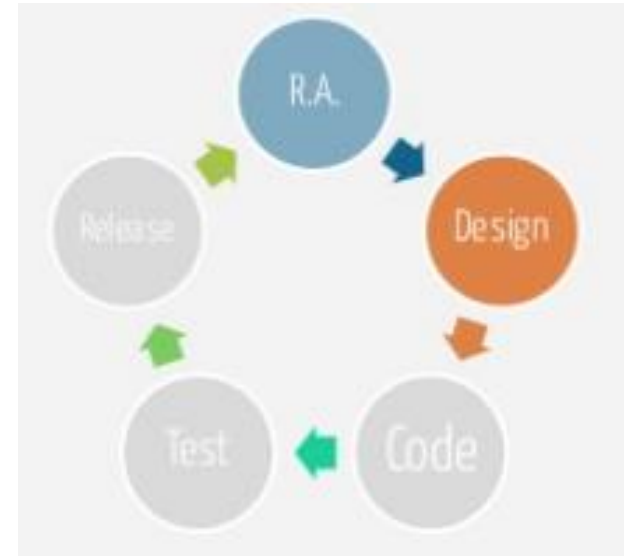
Step 3: Implementation

- Write the code according to design

- In some cases(oddly) the shortest phase

- End Result: **Executable* Application**

- Reviewed by :

  - Designer, Developer, Tester.
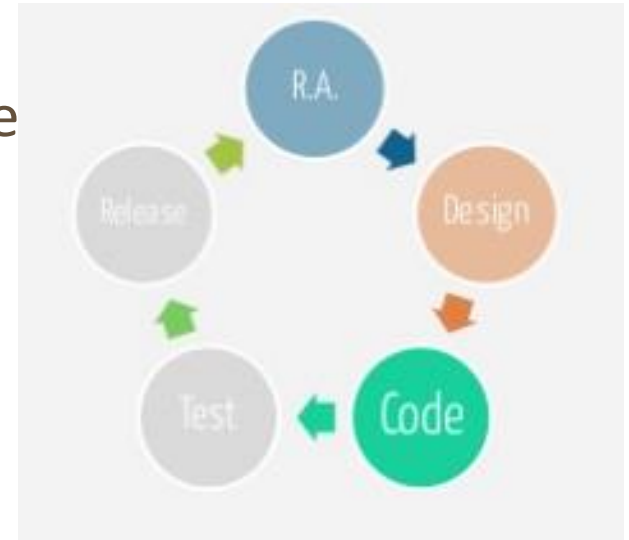
* or not …

Step 4: Testing

- Verify the code works according to the functional specification

- Look for corner cases

- End Result: **Test Plan**,

    A program that **works***

- Reviewed by :

    o Designer, Developer, Tester.
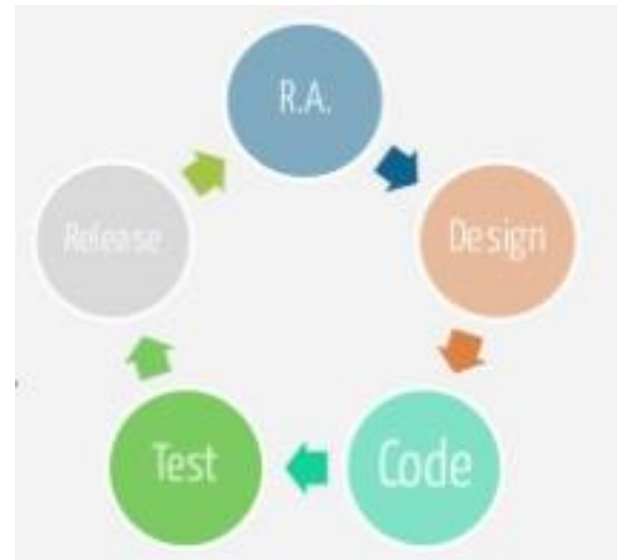
 * hopefully …

Step 5: Release

- Close to the customers(again)

- Validate their expectations

- Get feedback for

  o Next version

  o Maintenance

- End Result: A **happy** customer

  and lots of **$$$**

Development Disasters

Product development from an IT failures perspective

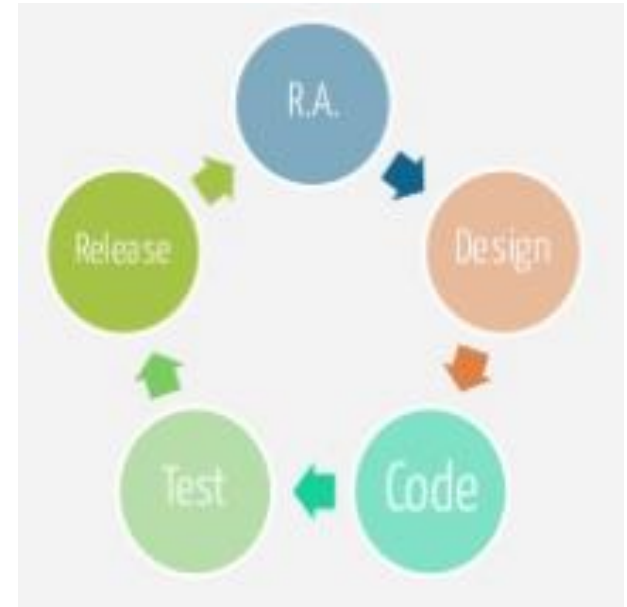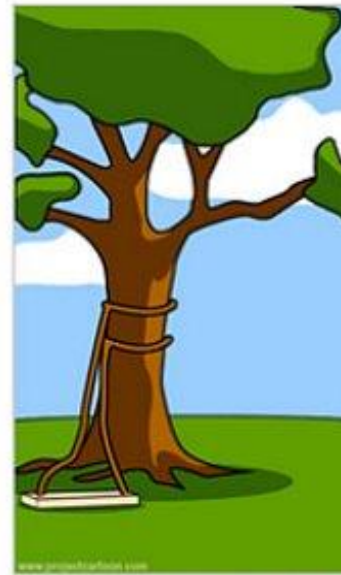|							
How the customer explained it	How the project leader understood it	How the analyst designed it	What operations installed	How it was documented	How it was supported	How they advertised the open source version	How they applied open source patches
How the business consultant described it	How the programmer wrote it	What the beta testers received	What marketing advertised	What the customer really needed	How it performed under load	The disaster recover plan	

Disasters related to Designers

How the customer explained it | How the project leader understood it | How the analyst designed it | How the programmer wrote it | What the customer really needed

How can Software Engineering help?

Software engineering aims to:

- Guarantee the timeline of production, prevents missing the deadlines.

- Guarantee the quality of product, prevents the crash of a system.

- Guarantee the integrity of software, with full documentation and suits the needs of the users.

What is a Software Process Model?

There are many different software processes, but all involve:

- Specification – defining what the system should do;

- Design and implementation – defining the organization of the system and implementing the system;

- Validation – checking that it does what the customer wants;

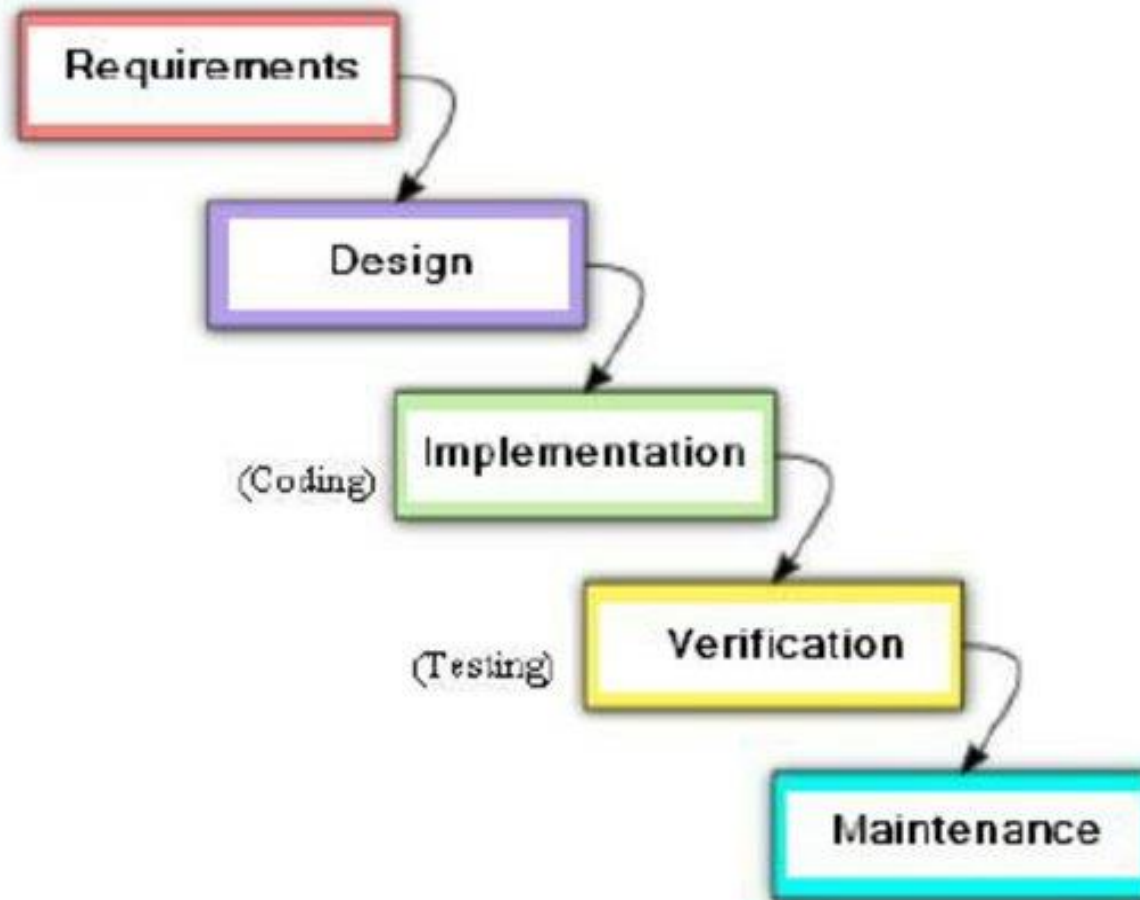- Evolution – changing the system in response to changing customer needs.

Software Development Lifecycle Model

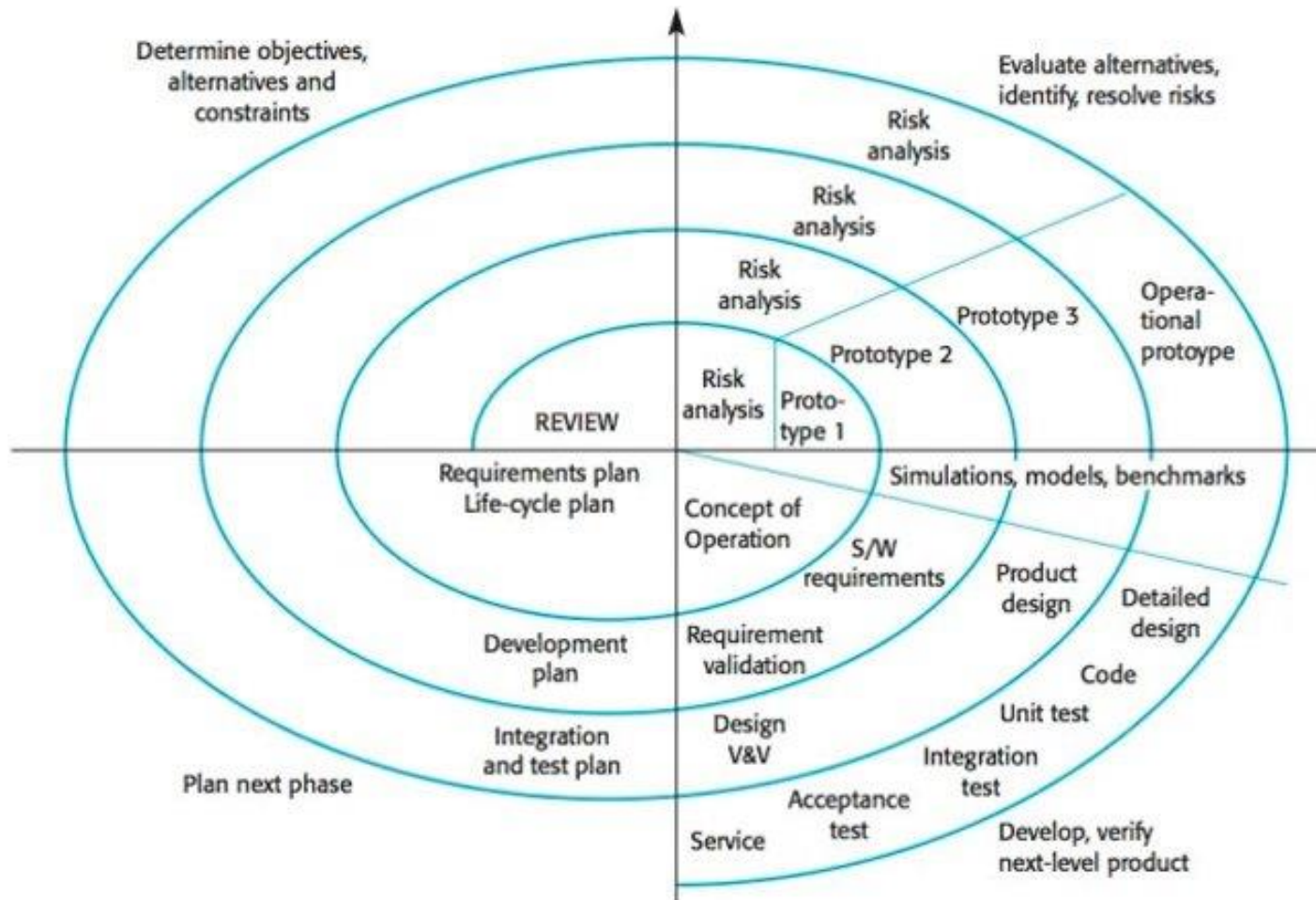The most used, popular and important SDLC models:

- Waterfall model

- V model

- Incremental model

- RAD model

- Agile model

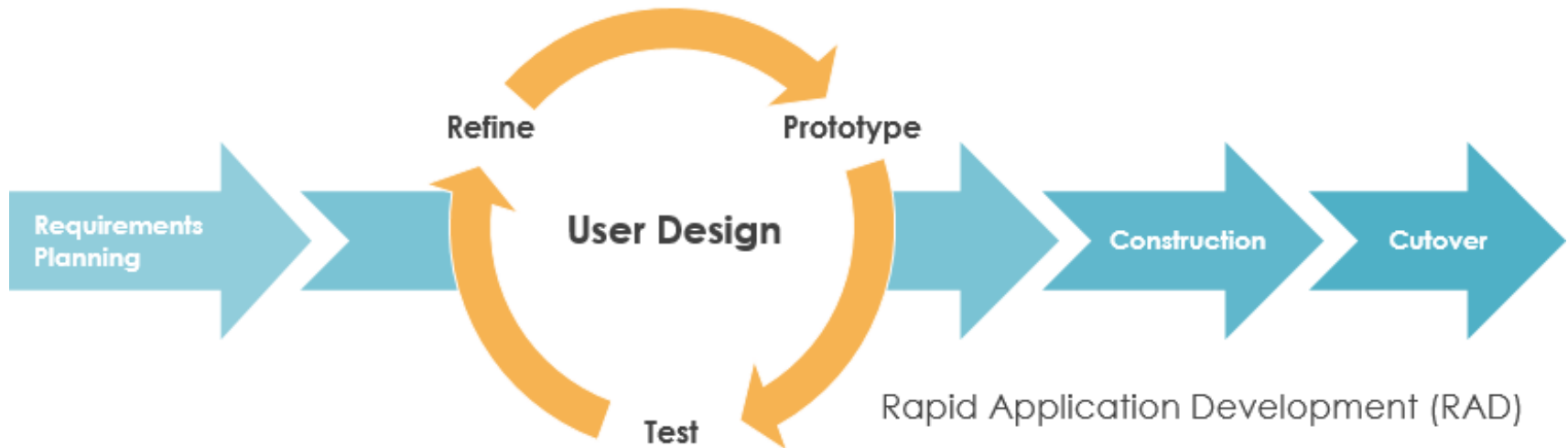- Iterative model

- Spiral model

- Prototype model

Waterfall Model

Spiral Model

RAD Model

Rapid Application Development (RAD)

Thank you