Systems Analysis And Design

Chapter Six:

Current trends in System Development

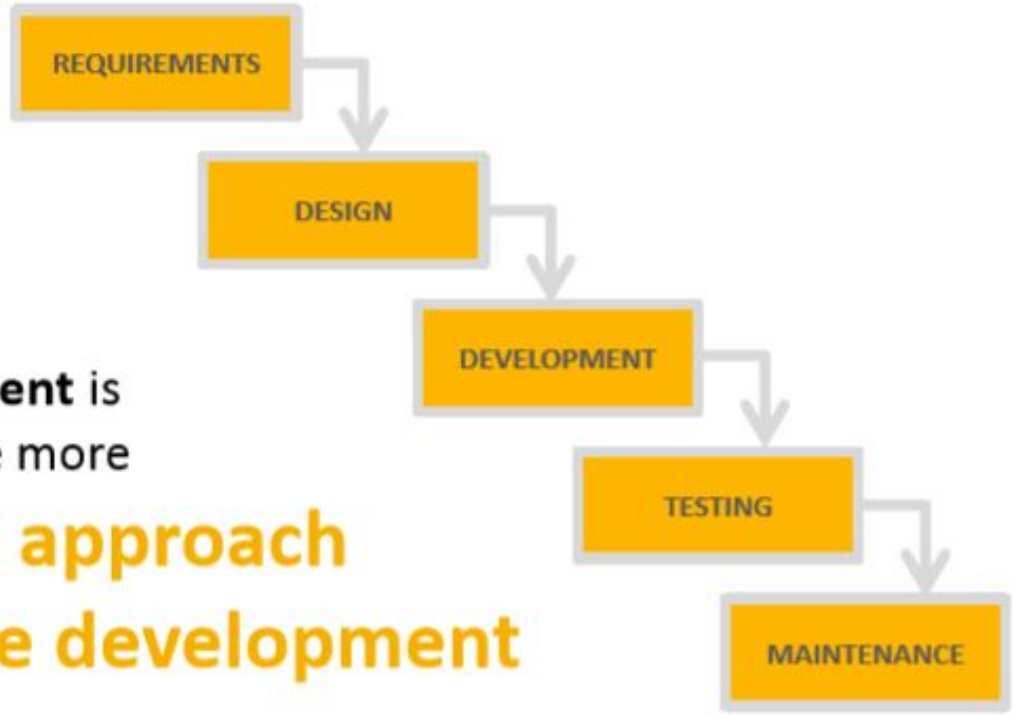
Learning Objectives

- Define Agile Methodologies.
- Discuss the value of Agile Vs Waterfall methods
- Explain when to use Agile Methodologies and when to use engineering based approaches to systems development.
- Define eXtreme Programming.
- Discuss the Agile Methodologies approach to systems requirements determination, design specifications, and the combination of coding and testing.

The Trend to Agile Methodologies

- Systems analysis and design, or systems development, has gone through three major phases since its inception
  - Undisciplined
  - Engineering approach
  - Agile Methodologies

Waterfall Development



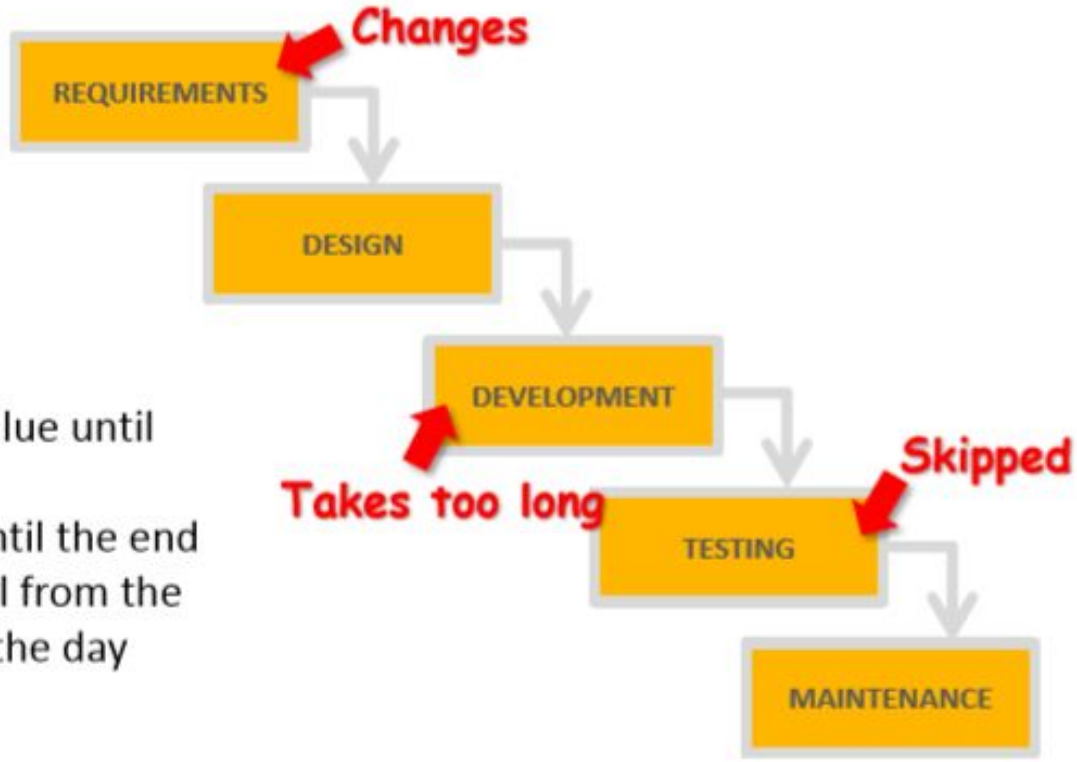
Waterfall Development is another name for the more

traditional approach to software development

Problems with Waterfall Method

- Difficult to accommodate change once a process is underway
- Phases must be completed in a sequential order
- Difficult to respond to changing customer requirements
- Few business systems have stable requirements

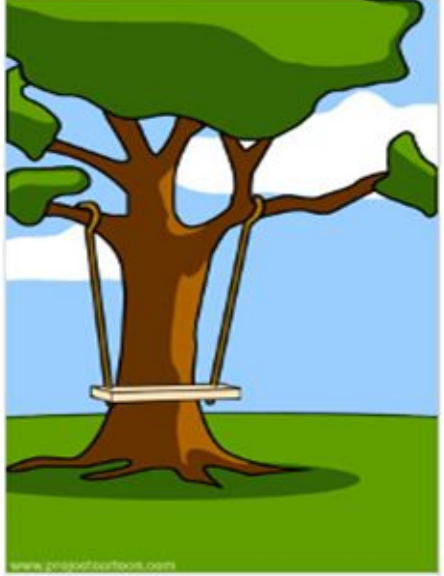
But…



- You don't realize any value until the end of the project
- You leave the testing until the end
- You don't seek approval from the stakeholders until late in the day

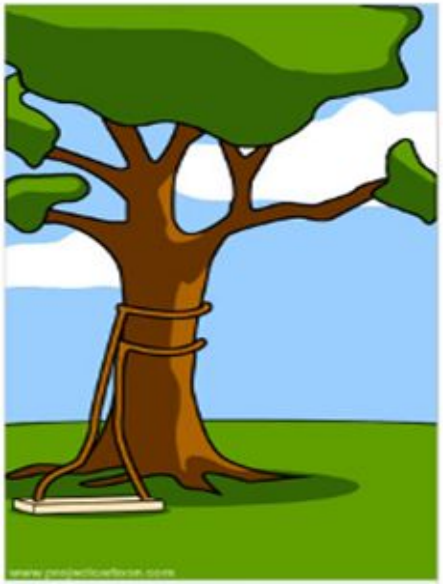
This approach is **highly risky**, often more **costly** and generally **less efficient** than **Agile** approaches

How the customer explained it



How the project leader understood it

How the analyst designed it



How the programmer wrote it

How patches were applied

What the customer really needed

Not a process, it's a **philosophy** or **set of values**

Agile Manifesto

Individuals and interactions over **processes and tools**

Working software over **comprehensive documentation**

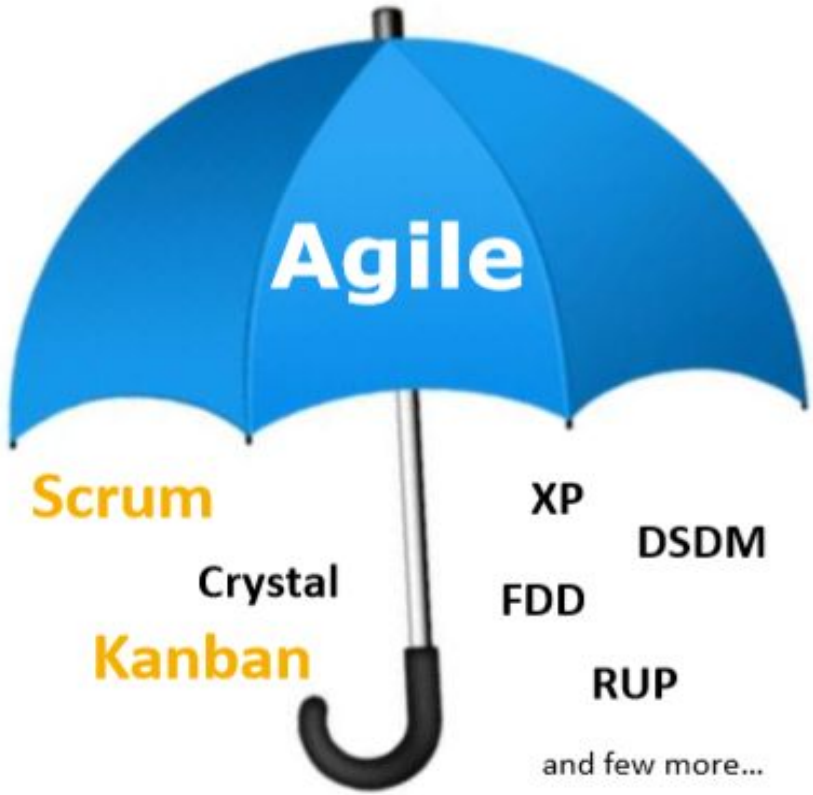
Customer collaboration over **contract negotiation**

Responding to change over **following a plan**

Agile Methodologies

- Agile Methodology is an umbrella of methodologies including:
  - Crystal family of methodologies
  - Adaptive Software Development
  - Scrum
  - Feature Driven Development (FDD)
  - eXtreme Programming.(XP)
  - Dynamic Systems Development Method(DSDM) and more…

Agile Umbrella



More Prescriptive
more rules to follow

- RUP (120+)
  RUP has over 30 roles, over 20 activities, and over 70 artifacts
- XP (13)
- Scrum (9)
- Kanban (3)
- Do Whatever!! (0)

More Adaptive
fewer rules to follow

* Check wikipedia for list of all Agile methods

Agile Methodologies Cont'd…

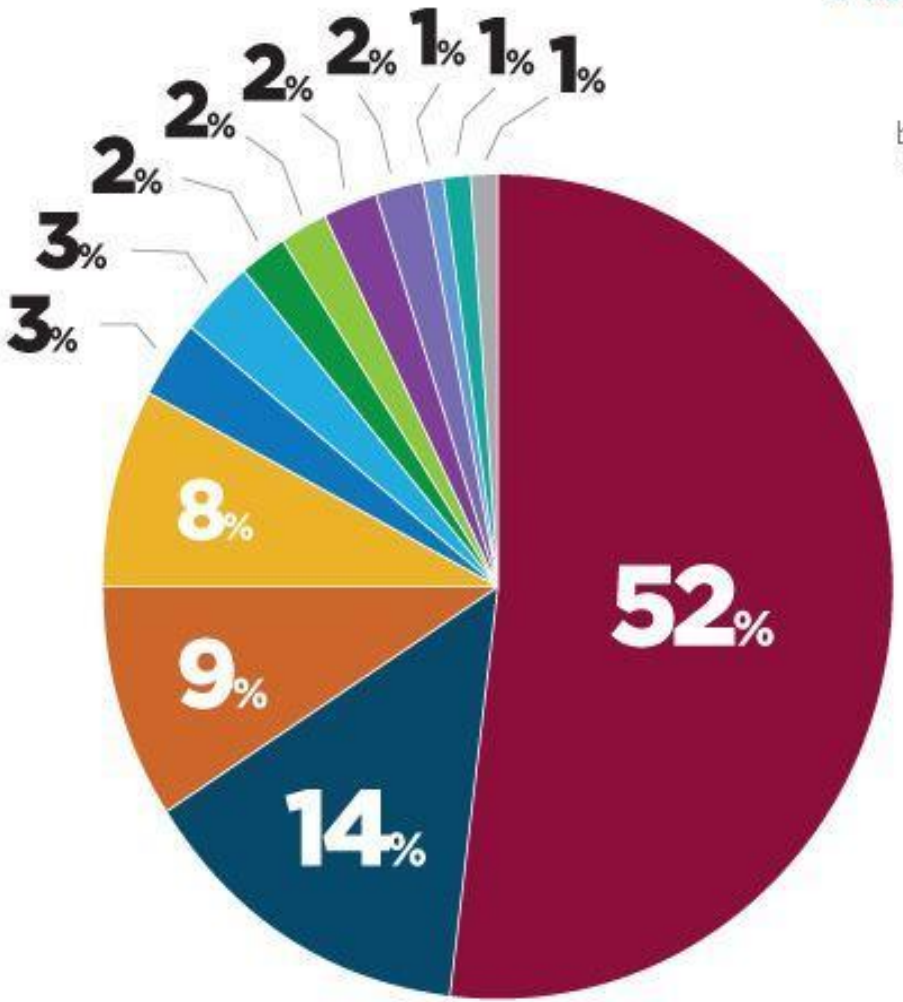
- The Agile Methodologies share three key principles:
  - a focus on adaptive rather than predictive methodologies,
  - a focus on people rather than roles,
  - a self-adaptive process.

Agile Methodologies Cont'd…

- Agile Methodologies are not for every project. (it is not "a one size fit all" thing)
- An Agile or adaptive process is recommended if the project involves:
  - Unpredictable or dynamic requirements
  - Responsible and motivated developers
  - Customers who understand and will get involved

AGILE METHODOLOGY USED

Scrum or Scrum variants continue to make up more than two-thirds of the methodologies being used, while Kanban has entered the scene this year as a meager player. The only category that saw growth this year was Custom Hybrids (9% up from 5%).



- Scrum
- Scrum/XP Hybrid
- Custom Hybrid
- Don't Know
- Kanban
- Scrumban
- Feature-Driven Development
- Extreme Programming XP
- Lean
- Other
- Agile Unified Process (AgileUP)
- Agile Modeling
- Dynamic Systems Development Method

eXtreme Programming- an example

- eXtreme Programming is one of the most written-about Agile Methodologies and was one of the best.
- It illustrates many of the central ***philosophies*** of the Agile Methodologies approach to systems development
- *Recently, **Scrum** has been the most widely used agile method as per the data in the previous slide (2013)*

eXtreme Programming

- eXtreme Programming is distinguished by:
  - short development cycles
  - incremental planning approach
  - ***automated tests*** written by programmers and customers to monitor the process of development, and
  - reliance on an ***evolutionary approach*** to development that lasts throughout the lifetime of the system

eXtreme Programming

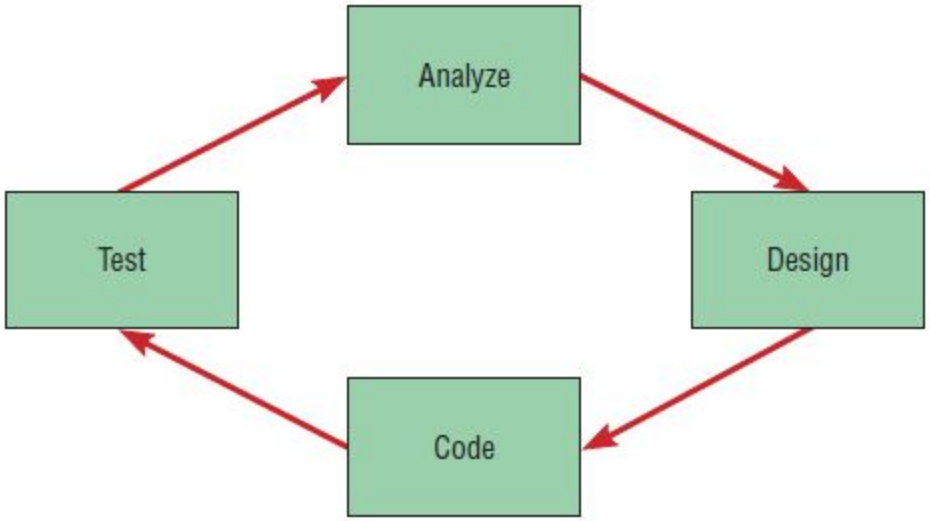
- ***Coding and testing*** are intimately related parts of the same process.
- The programmers who write the code also write the tests.
- The emphasis is on testing those things that can break or go wrong, not on testing everything
- The overall philosophy behind eXtreme Programming is that **code will be integrated into** the system it is being developed for and tested ***within a few hours after it has been*** written

eXtreme Programming

- Another feature of XP is pair programming.
- The advantages of pair programming include:
  - more (and better) communication among developers,
  - higher levels of productivity,
  - higher-quality code, and
  - reinforcement of the other practices in eXtreme Programming, such as the *code-and-test discipline*.

The heart of systems development process

- In the Agile Methodologies, the activities traditionally thought of as belonging to analysis, design, and implementation are combined into a single process(iteration).



Requirement Determination in Agile Methodologies

- The requirement determination techniques are:
  - Continuous User Involvement:
    - involvement of users in the analysis and design process
  - Agile User-Centered Design
    - experts are gathered together and work with the help of a facilitator
  - The Planning Game
    - Structured communication of Business (customer) and Development (builder) personnel

Design Specifications in Agile Methodologies

- Design specification **is captured as code itself**.
- eXtreme Programming in particular employs two techniques for improving the quality of the design:
  - **simple design** and
  - **refactoring**

Implementation in Agile Methodologies

- Coding and testing are intimately related parts of the same process, and the programmers who write the code also write the tests.
- The general idea is that code is tested soon after it is written.
  - If the code passes the tests, then it is integrated into the system.
  - If it does not pass, the code is reworked until it does pass.

Trends in Coding and Development

- Two new trends in development
  - TDD –Test Driven Development
  - BDD – Behavior Driven Development

- What is TDD/BDD?
  - Express expected behaviour before writing code
- Why is TDD/BDD a good thing?
  - Enjoy more efficient and predictable course of development
  - Find and fix bugs faster
  - Prevent bugs from reappearing
  - Improve the design of our software
  - Reliable documentation
- How do we do TDD/BDD?
  - Write test programs
  - Run the tests automatically

Workflow of TDD/BDD