

DAY 1

Introduction to Software Engineering

Welcome to Software Fellowship

But what do we learn?

Is it the course on tutorial of programming language? No

Is it a course on tutorial on full stack web development? No

Is it a course on software development taking example of Web Dev? Yes



Overview

- Complexity of the Software
- Software Models
- Development Diagrams(Use Case Diagram)
- Paradigms of Software



Software Development: Deceptively Complex

Once built a building, a civil engineer need not change the position of its basement. Isn't it?

But, once the real software is made, now the requirements changes.

And, Programmer has a strong tendency to dive into coding before proper planning.

Building is constructed but Software is developed.



Software Development: Deceptively Complex

- **The Complexity of the Problem Domain**
 - Actual Domain of the software might be other major fields.
- **The Difficulty of Managing the Development Process**
 - Large number of manpower(Developer, Q/A)
- **The Flexibility Possible through Software**
 - Clients learn through the developing software.



Software Development: Deceptively Complex



How the customer explained it



How the project leader understood it



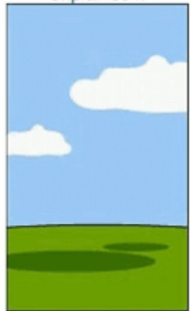
How the engineer designed it



How the programmer wrote it



How the sales executive described it



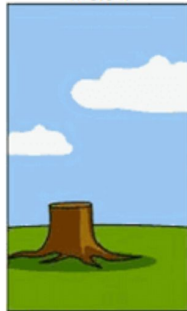
How the project was documented



What operations installed



How the customer was billed



How the helpdesk supported it



What the customer really needed



World's quest::Can we manufacture Software?

Manufacture/Construction:

- Blessed with pre-defined techniques, models, and specifications
- Low risk and high success in product.

Development:

- No or very little pre-defined techniques, models and specifications
- Every product has some research component linked with it.
- Software Engineering is the process of making the software capable of being manufactured. World is striving towards the goal.



Software Models

A spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.

- Waterfall model.
- Incremental model.
- Spiral model
- Agile model



Waterfall Model

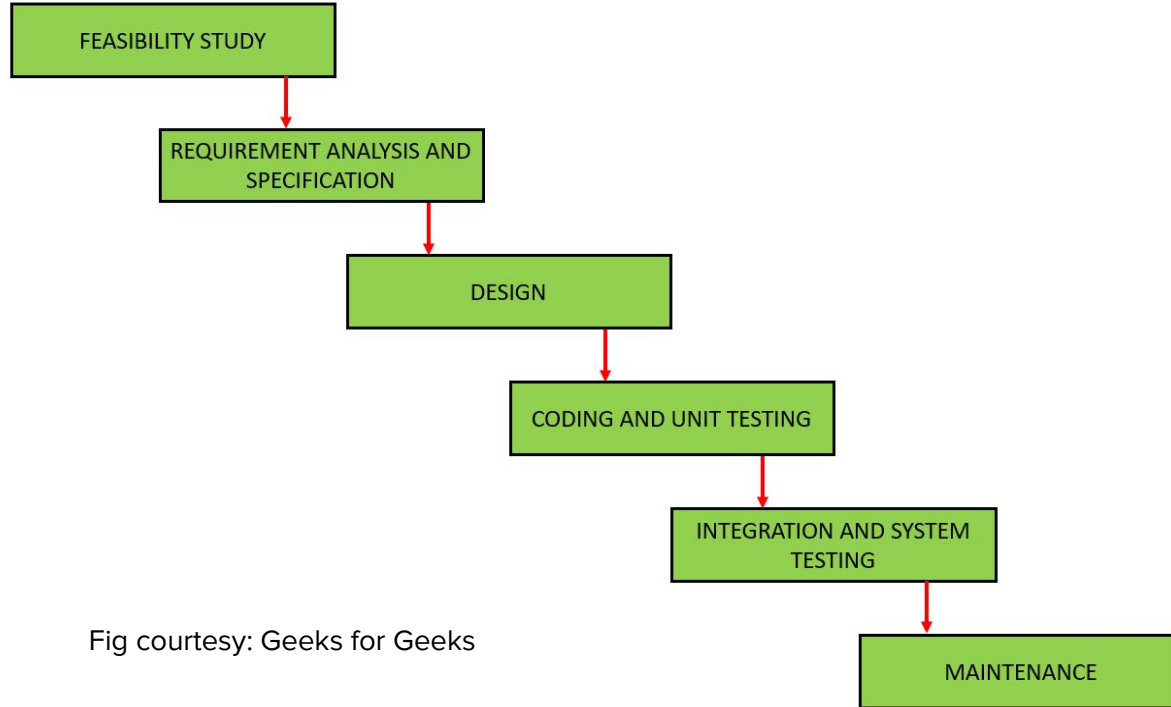


Fig courtesy: Geeks for Geeks



Waterfall Model::Pros and Cons

Pros:

- Simple model to understand and implement
- Easy to take progress

Cons:

- Not suitable for complex projects where requirements are frequently changed.



Incremental model

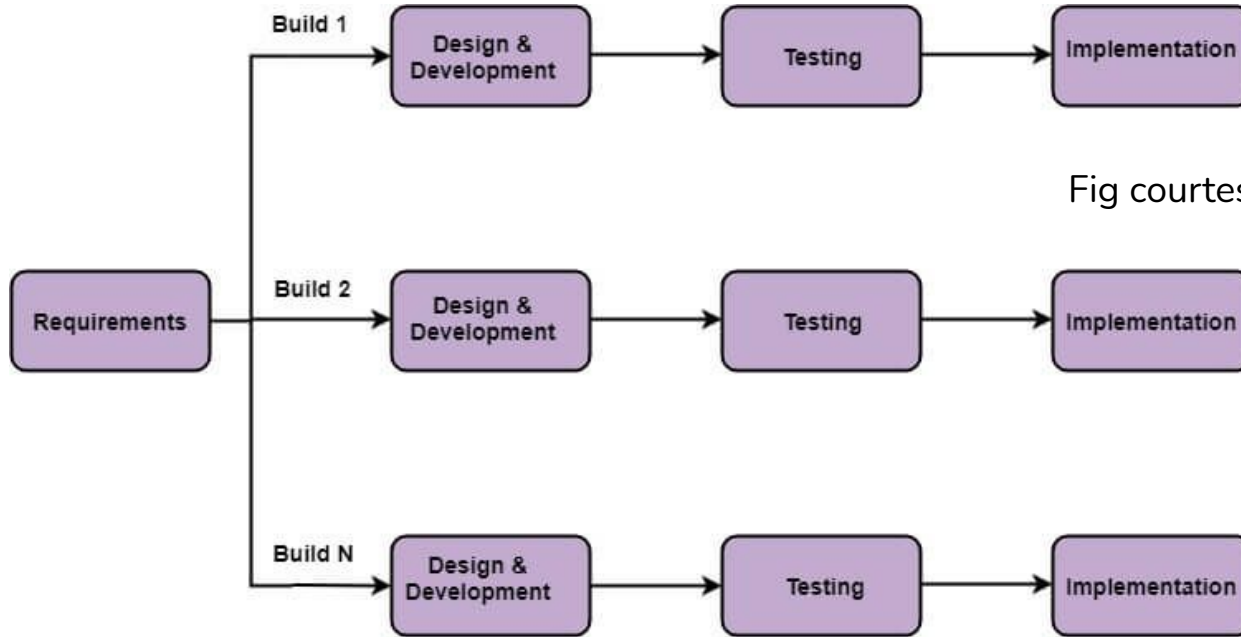


Fig courtesy: Javatpoint

Fig: Incremental Model



Incremental Model::Pros and Cons

Pros:

- Error Reduction (core modules are used by the customer from the beginning of the phase and then these are tested thoroughly)
- Uses divide and conquer for breakdown of tasks.

Cons:

- Requires good planning and design.
- Complex



Design Diagrams

- Use Case Diagram
- Domain Model Diagram
- Class Diagram
- Object Diagram
- Interaction Diagram
- Deployment Diagram



Use case diagram

The different ways that a user might interact with a system.

This is the most important document of the requirement elicitation phase.

