Software Development Lifecycle

Rohan Mishra & Kiran Beniwal

Ask your Questions in the chat box

We will answer all of them in the end.

Let’s build a Software together.

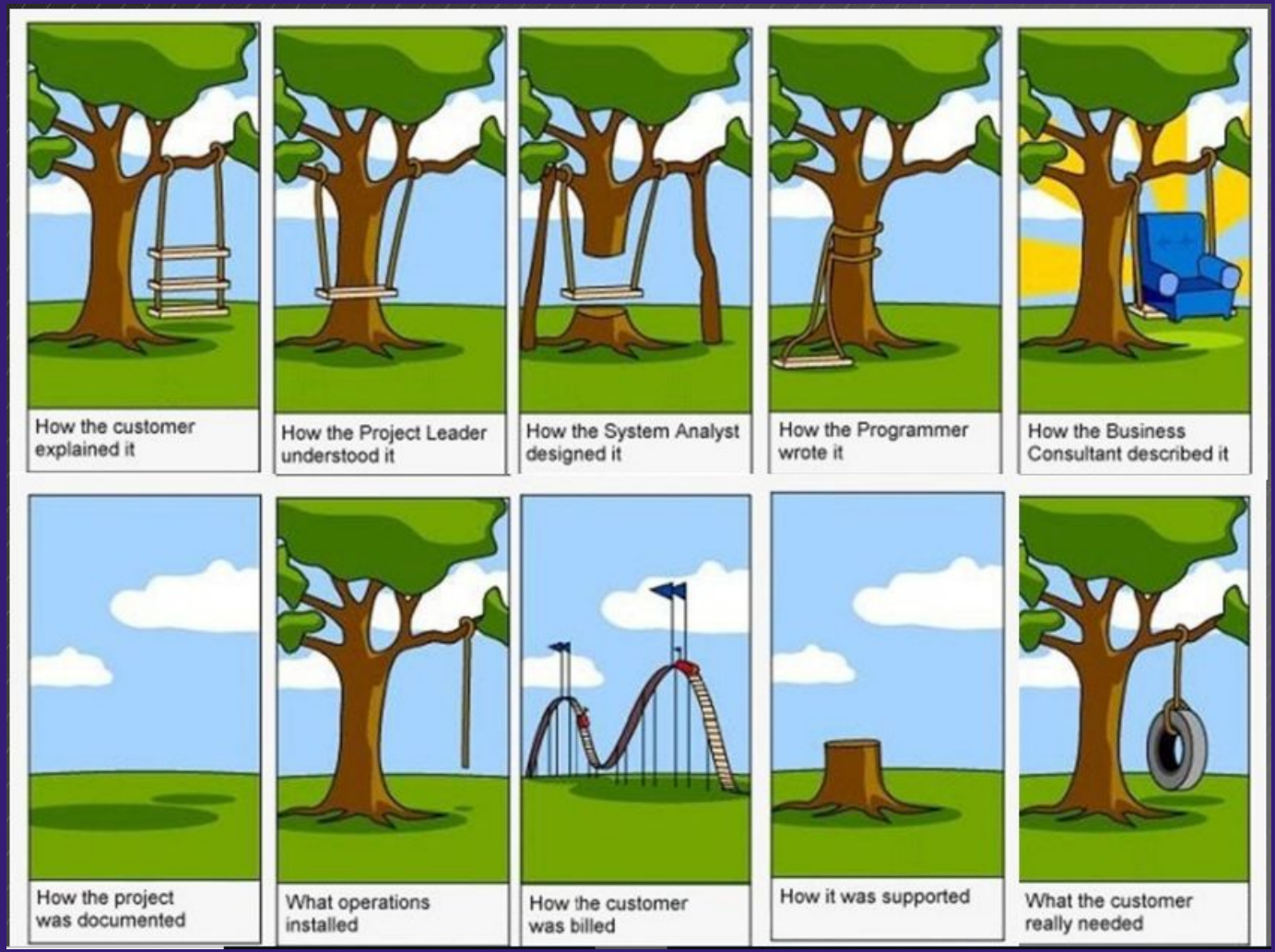
Think of an Idea.

Write the Code.

Ship it to the world.

Right?
What could go wrong?

Just Everything about it.


How the customer explained it
How the Project Leader understood it
How the System Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Hello My Name is Kiran Beniwal

Pythonist

Hello My Name is Rohan Mishra

Product and UX Designer

What is SDLC?

A framework that describes the processes followed at each stage of a software development project.

Every minute spent in planning saves as many as ten minutes in execution.

Stages of SDLC

Requirement gathering and analysis

Design

Development

Testing

Implementation

Maintenance and Evolution

Requirements Gathering and Analysis

Feasibility Study

Economic: Cost benefit analysis.

Technical: We need to check whether existing computer can support the system.

Behavioral: An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system.

Analysis

An in-depth analysis is performed to obtain a detailed understanding of the business needs.

Customers Need

Design

- It is the most crucial phase of SDLC.
- It defines the information architecture of the system.
- Users flows and task flows are defined.
- Two phases: Logical Design and Physical Design
- Logical Design: Specifies user needs.
- Physical Design: Tells the programmer about the flows and interactions.

Development

Implementing the design specified in design document to an executable programming language code.

Making the interactions and functionality come alive.

Integrating the software with a Database.

Setting up Database Interactions.

Testing

Testing is done as a comparison between Design documentation and Development build.

Test cases are defined.

Metrics are defined.

Implementation

Rolled out the software product to the users via App Stores, Web based Portals.

Maintenance & Evolution

Handling errors that may exist in software even after testing phase

Implementation of new requirements after software is deployed at customer location.

Example of a Development Process

New Software

Old Software

Let’s Build a New Product.

Requirement gathering and analysis

Design

Development

Testing

Implementation

Maintenance and Evolution

Let's Iterate an Old Product.

Requirement gathering and analysis

Design

Development

Testing

Implementation

Maintenance and Evolution-

Review is used to evaluate the effectiveness of the system.

The objectives are to determine if the system does what it is designed to do.

Does it support the user as required in an effective and efficient manner.

The review should assess how successful the system is in terms of functionality, performance, and cost versus benefits.

Some problems are defined to make sofware product even better.

To help understand and implement the SDLC phases various SDLC Models have been created by software development experts, universities, and standards organizations.

Why do we need SDLC Models?

Provides basis for project planning, estimating & scheduling.

Provides framework for standard set of terminologies, activities & deliverables.

Provides mechanism for project tracking & control.

Increases visibility of project progress to all stakeholders.

Advantages of using right SDLC Model:

Increased development speed.

Increased product quality.

Improved tracking & control.

Decreased project risk.

Advantages of using right SDLC Model:

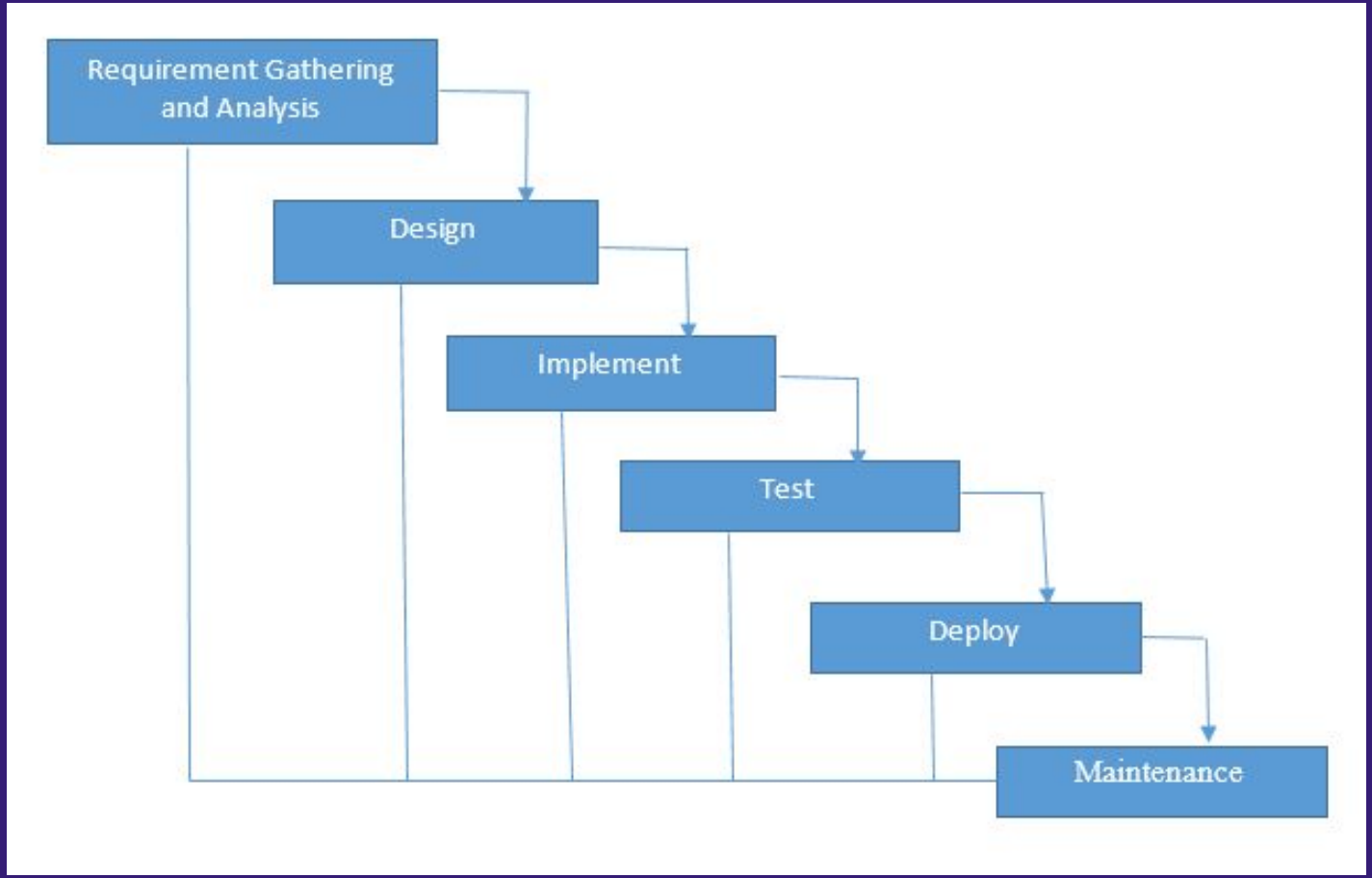
Increased development speed.

Increased product quality.

Improved tracking & control.

Decreased project risk.

Waterfall Model


Requirement Gathering and Analysis
Design
Implement
Test
Deploy
Maintenance

The waterfall Model is a linear sequential flow.

In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation.

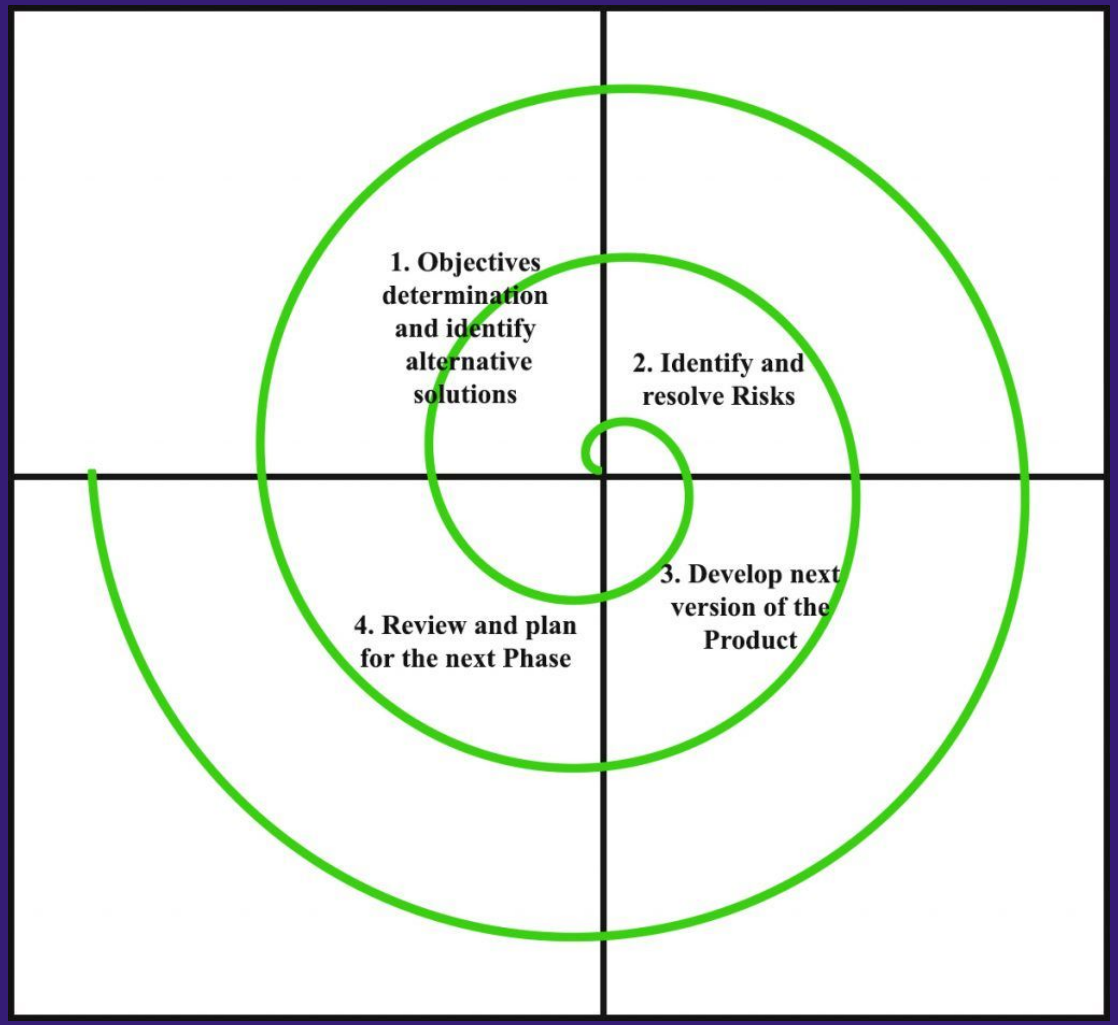
This means that any phase in the development process begins only if the previous phase is complete.

The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement.

The waterfall approach is the earliest approach that was used for software development

Advantages	Disadvantages
Easy to be explained	inflexible,
Stages and activities are well defined	Adjusting scope is difficult and expensive
Milestones are well understood	Model doesn't provide a clear path for problems found during testing phases
Each phase has specific deliverables	Costly and required more time
Works well for smaller projects where requirements are very well understood.	Needs very proper and detailed planning
Process and results are well documented	No continuous customer involvement to review the output

Spiral Model


1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

It is combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts.

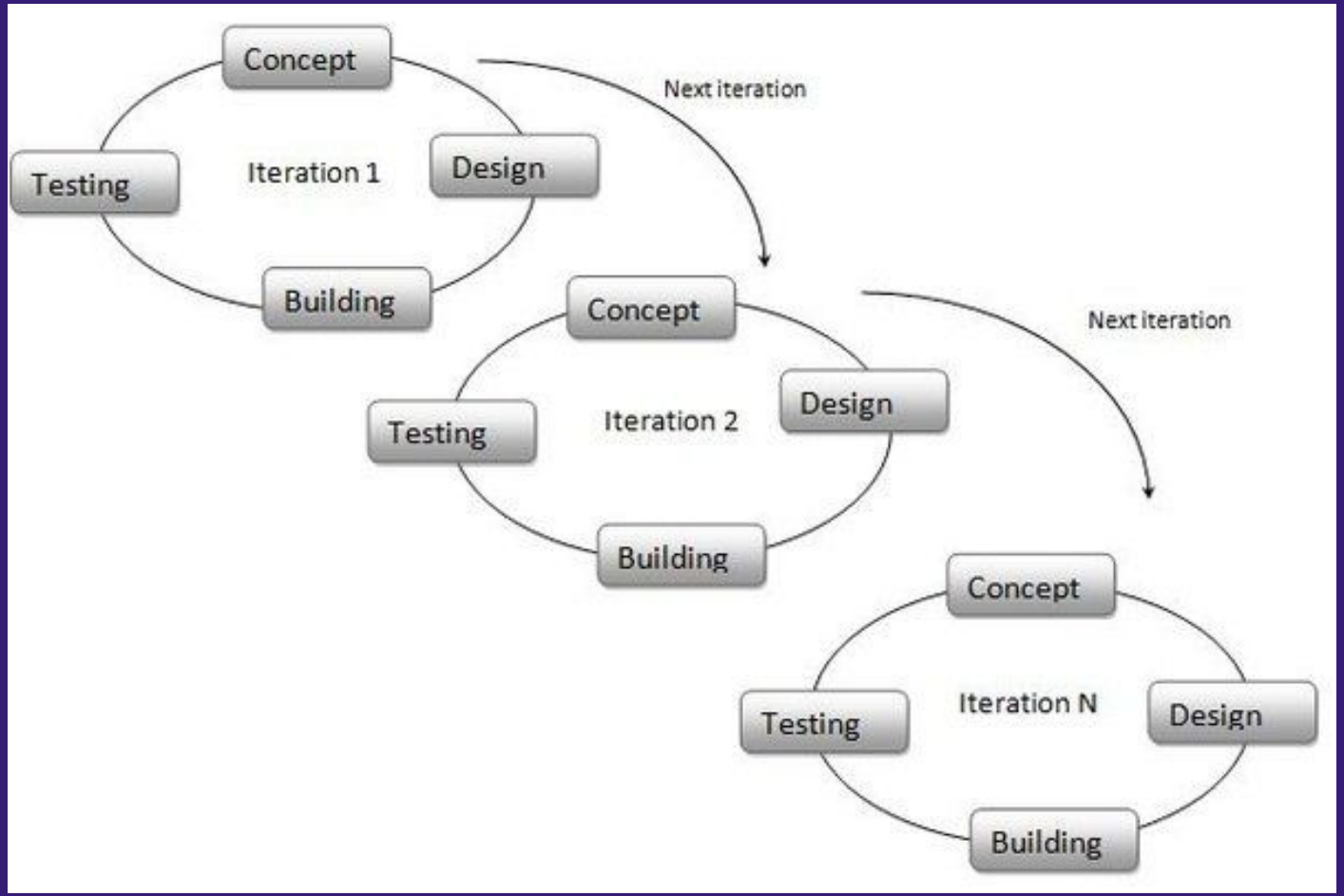
This model of development combines the features of the prototyping model and the waterfall model.

The spiral model is favored for large, expensive, and complicated projects.

This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.

Advantages	Disadvantages
Estimates (i.e. budget, schedule, etc.) become more realistic as work progresses	High cost and time to reach the final product
Changing requirements can be accommodated	Needs special skills to evaluate the risks and assumptions
Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management	Highly customized limiting re-usability
Users see the system early and give corrective feedback	Time spent planning, setting objectives, doing risk analysis and prototyping will be huge
A more accurate end product	

Agile Model


Concept
Next iteration
Testing
Iteration 1
Design
Building
Concept
Next iteration
Testing
Iteration 2
Design
Building
Concept
Testing
Iteration N
Design
Building

It is based on iterative and incremental development, where requirements and solutions evolve through collaboration between cross-functional teams.

Usually less formal and reduced scope.

Used for time-critical applications.

Advantages	Disadvantages
Decrease the time required to avail some system features	Scalability
Face to face communication and continuous inputs from customer representative leaves no space for guesswork	Skill of the software developers
The end result is the high quality software in least possible time duration and satisfied customer	Ability of customer to express user needs
Produces business value early in the development life cycle (Do you know why)	Documentation is done at later stages
	Reduce the usability of components
	Needs special skills for the team

We are here for your Questions

Sneak peek into future

We are coming back soon with “Moving Agile”.

Summary what we talked about.

What is SDLC?

Stages of SDLC.

Models of SDLC.

Thank you for Joining us.