THE SOFTWARE DEVELOPMENT PROCESS

Software Development Models – Waterfall model

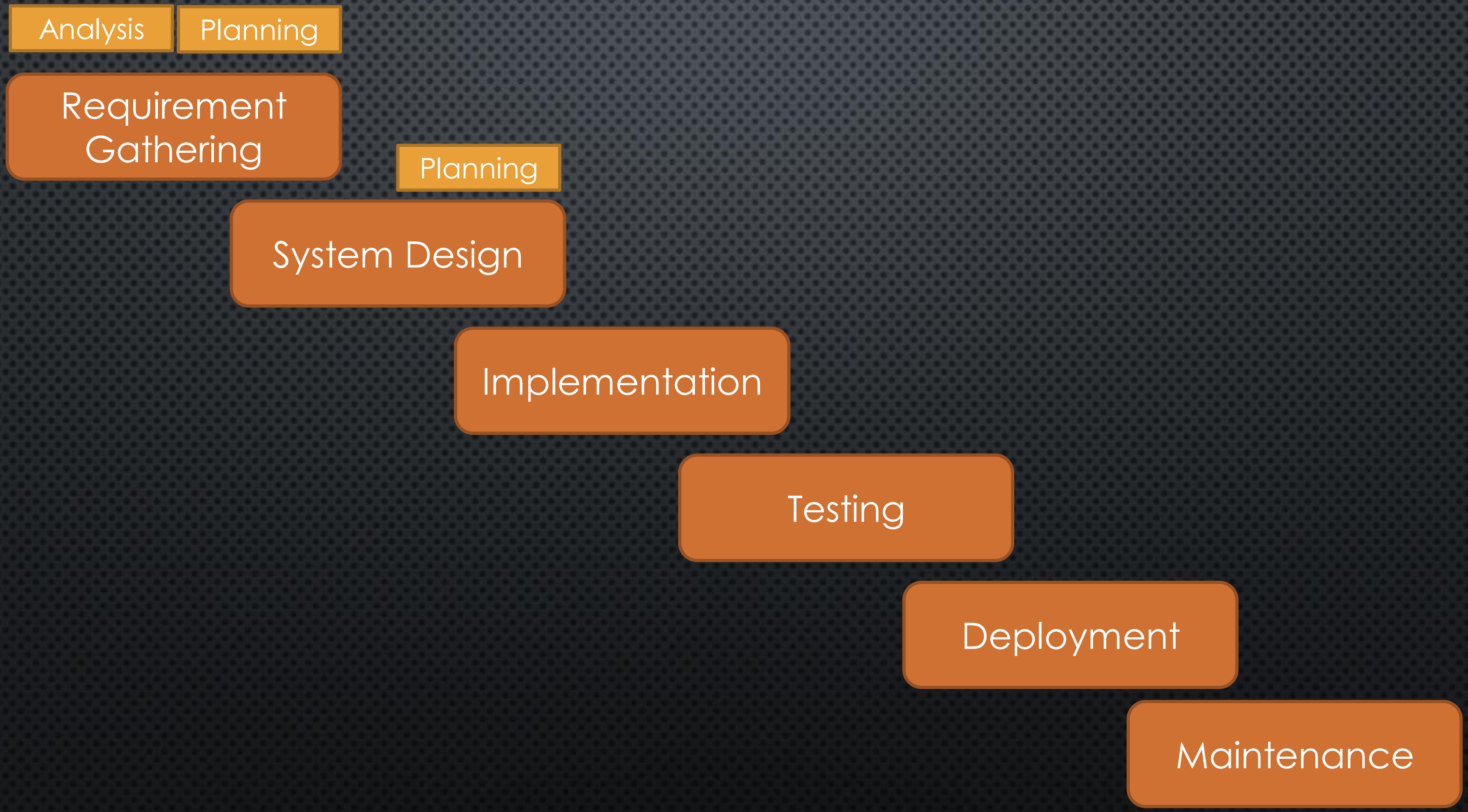
WATERFALL MODEL

- The ‘Waterfall Model‘ is the very old model
- It is also referred to as a **linear-sequential life cycle model**
- In ‘Waterfall Model’ testing starts only after the development is completed
- Because of which there are many defects and failures which are reported at the end
- So, the cost of fixing these issues are high
- Hence, these days people are preferring ‘Agile Model’
- In ‘Agile Model’ after every sprint there is a demo-able feature to the customer
- Hence customer can see the features whether they are satisfying their need or not

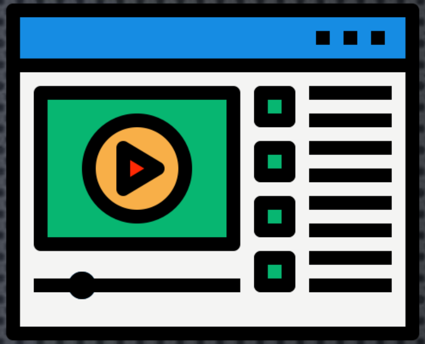
WATERFALL MODEL

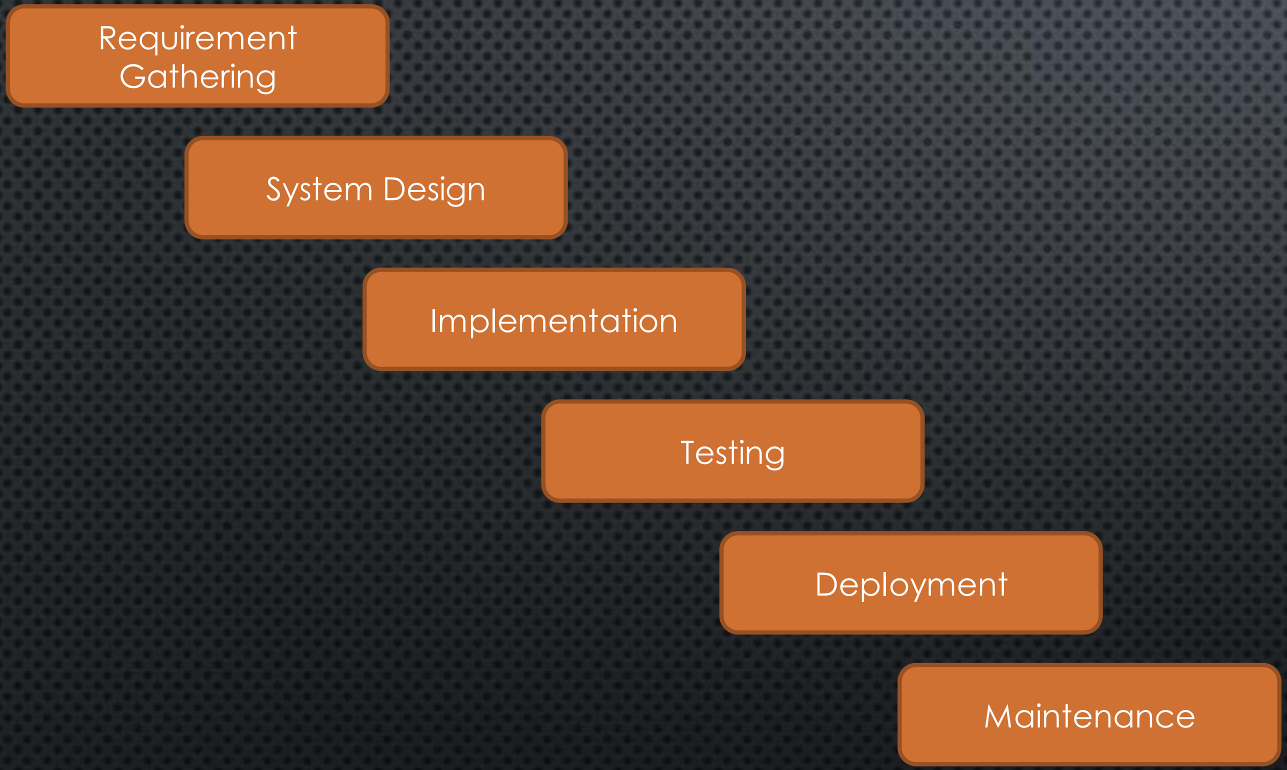
- In a waterfall model, each phase must be completed fully before the next phase can begin
- This type of software development model is basically used for the project which is small and there are no uncertain requirements
- Each phase must be completed fully before the next phase can begin
- At the end of each phase, a review takes place to determine if the project is on the right path and whether to continue or discard the project
- This means that software testing starts only after the development is complete
- This type of software development model is basically used for the project which is small and there are no uncertain requirements

DIAGRAM OF WATERFALL MODEL



PHASES OF WATERFALL MODEL





Requirement Gathering

System Design

Implementation

Testing

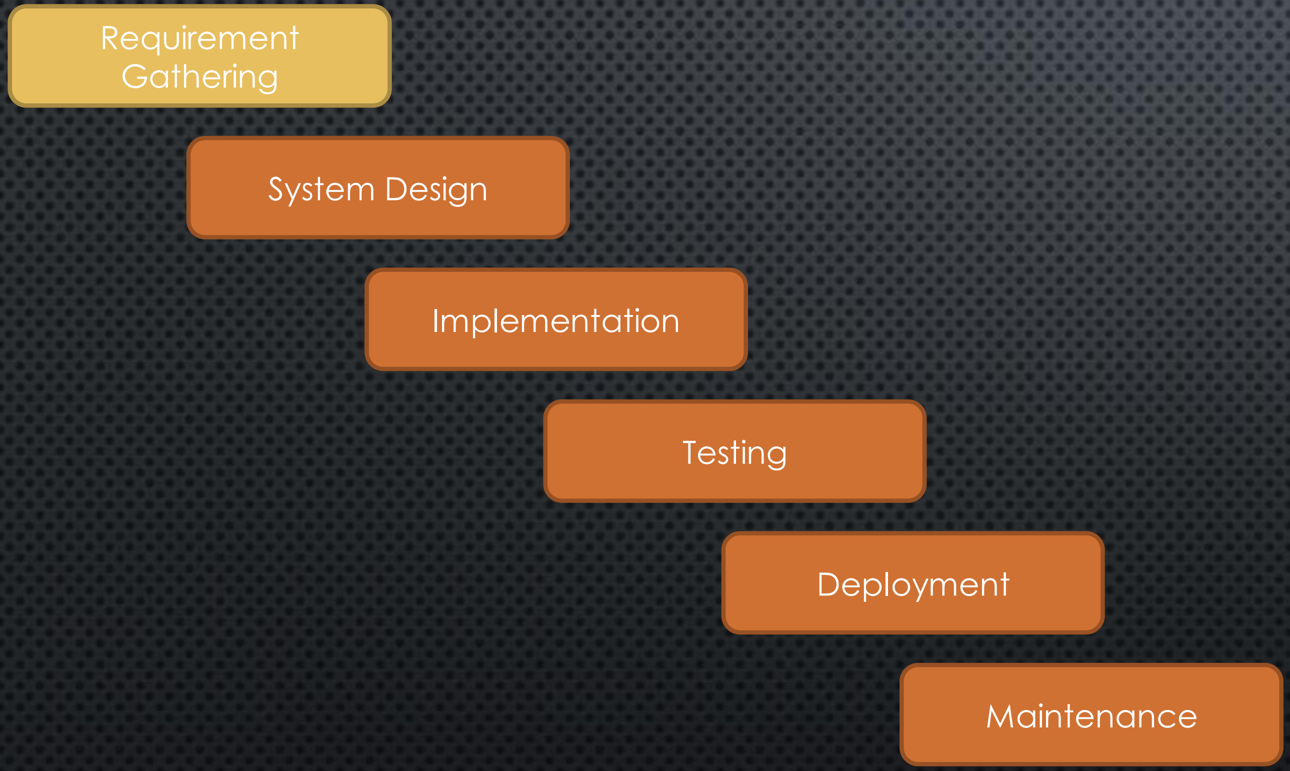
Deployment

Maintenance

Let’s use as an example to illustrate this model an eLearning Platform

A membership website where students register and can watch videos or download resources for educational purposes

REQUIREMENT GATHERING



Requirement Gathering

System Design

Implementation

Testing

Deployment

Maintenance

In this phase the requirements are gathered by the business analyst and they are analysed by the project team

Requirements are documented during this phase and clarifications are conducted

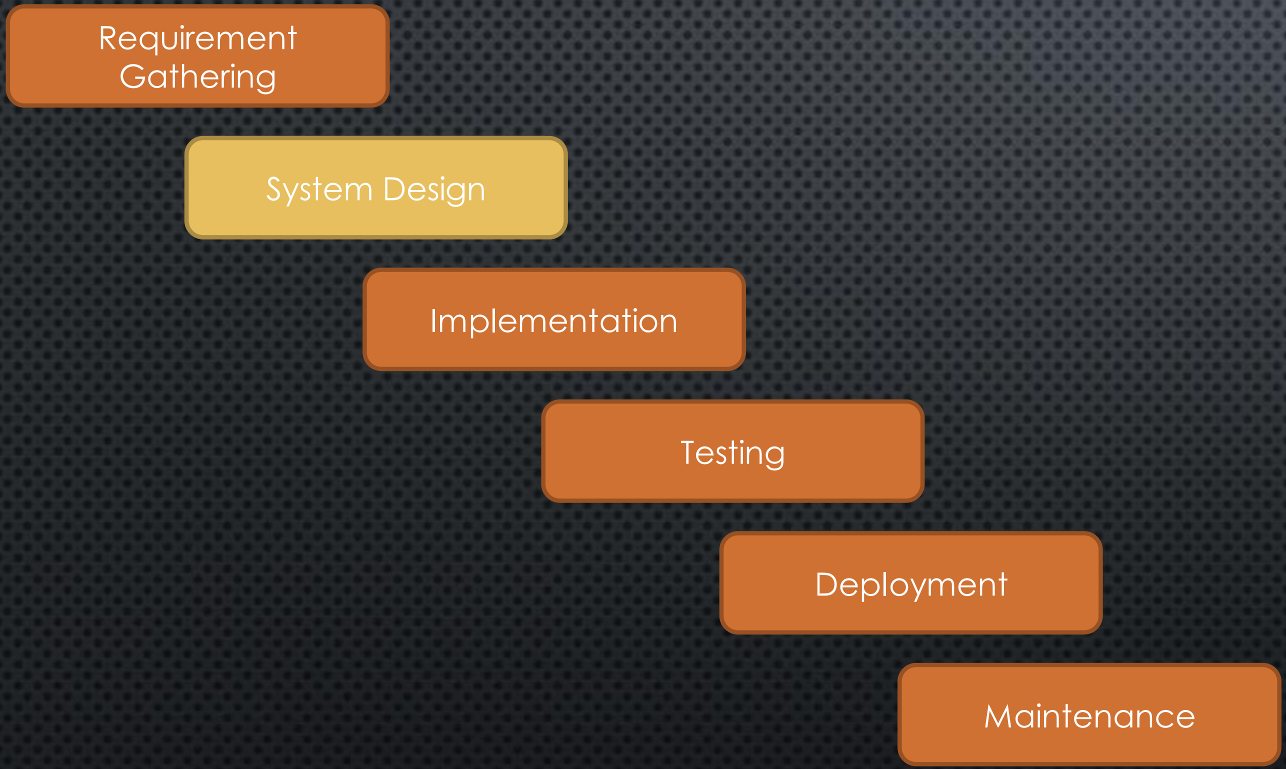
The Business Analysts document the requirement based on their discussion with the customer

REQUIREMENT GATHERING

Going through the requirements and analysing them has revealed that the project team needs answers to the following questions which were not covered in the requirements document:

- Do we have to support multiple languages?
- How many users are expected to use the application?
- Which video formats should we use?
- Do we need to use a Global Content Delivery (CDN)

SYSTEM DESIGN

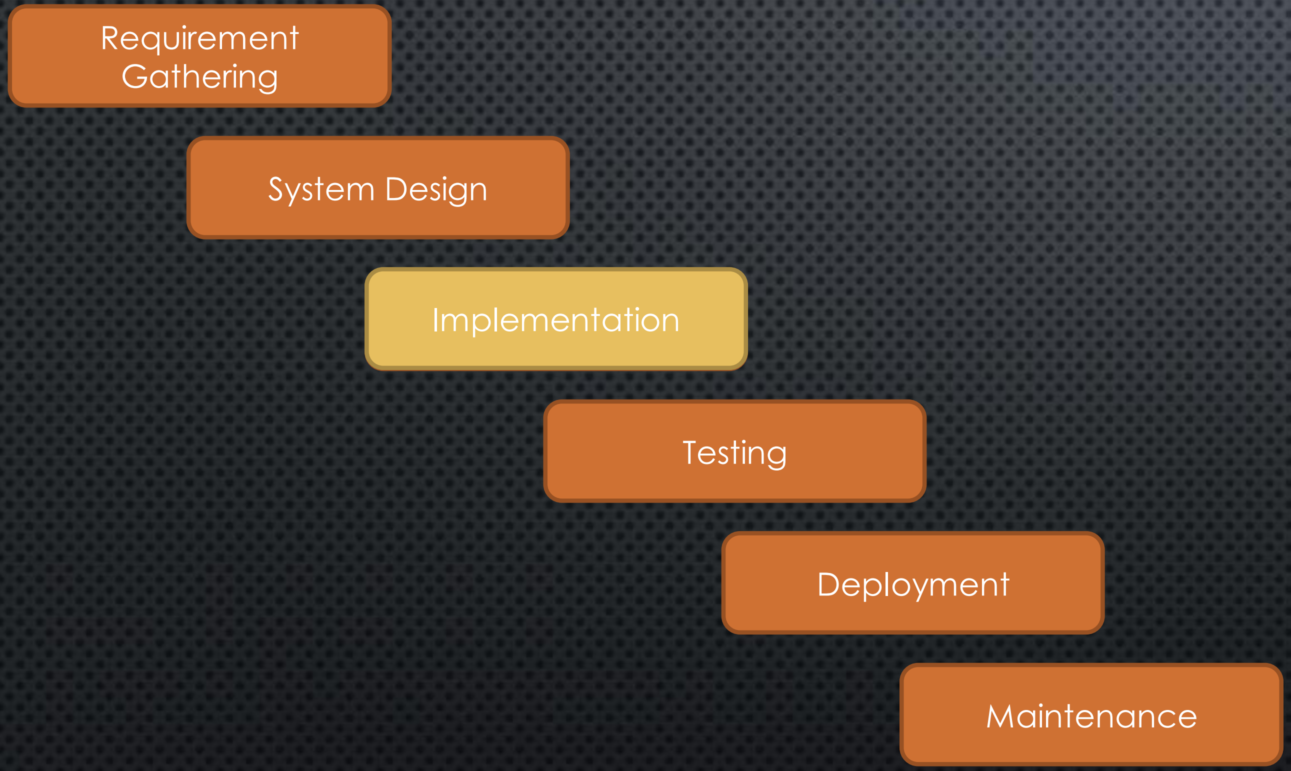


The architect and senior members of the team work on the software architecture, high level and low level design for the project

The decision is that the eLearning platform needs to have redundant backup and failover capabilities such that system is accessible at all times

The architect creates the Architecture diagrams and high level / low level design documents (mock-ups, wireframes, flow charts,…)

IMPLEMENTATION

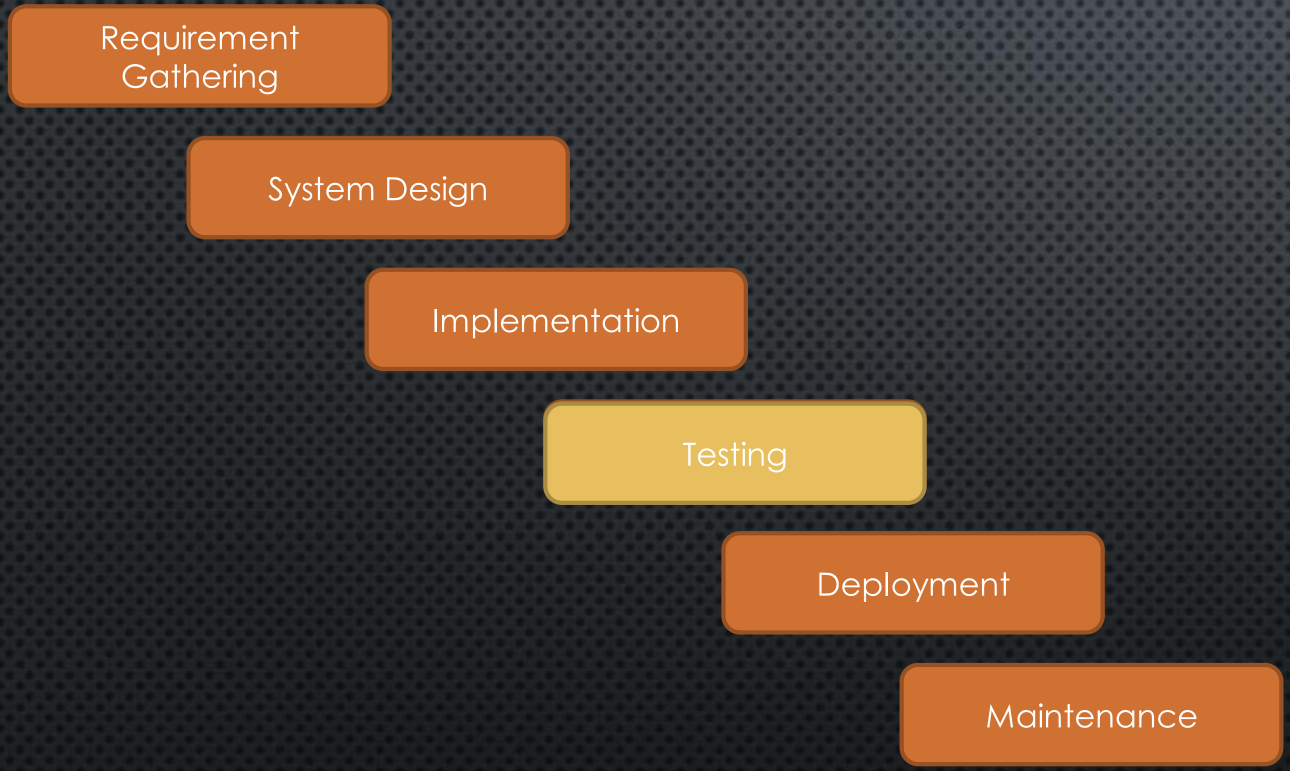


The development team works on coding the project, taking the design documents and ensure that their solution follows the design finalized by the architect

Since the application is a eLearning platform with videos and video delivery was a high priority in the application requirements, they implement several check of video streaming to verify the videos are played on any platform and conditions

They also perform several other activities like a senior developer reviewing the other developers code for any issues. Some developers perform static analysis of the code

TESTING



Requirement Gathering

System Design

Implementation

Testing

Deployment

Maintenance

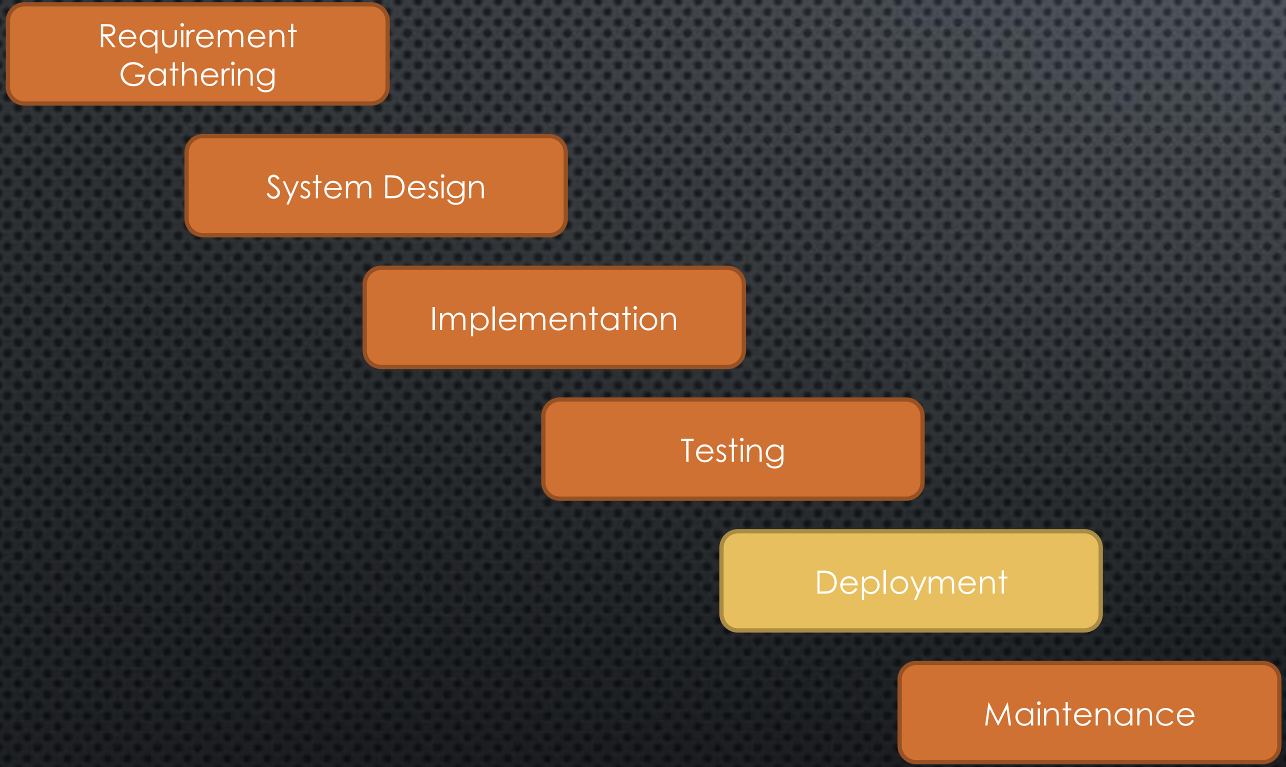
The testing team **tests/verifies** the complete application and identifies any defects in the software solution

These **defects** are fixed by the developers and the testing team tests the fixes to ensure that the defect is fixed

They also perform **regression testing** of the application to see if any new defects were introduced

Testers with **video streaming** programming knowledge were also hired for the project so that they could test the application based on the domain perspective.

DEPLOYMENT



Requirement Gathering

System Design

Implementation

Testing

Deployment

Maintenance

The team builds and installs the application on the servers which were procured for the eLearning platform application

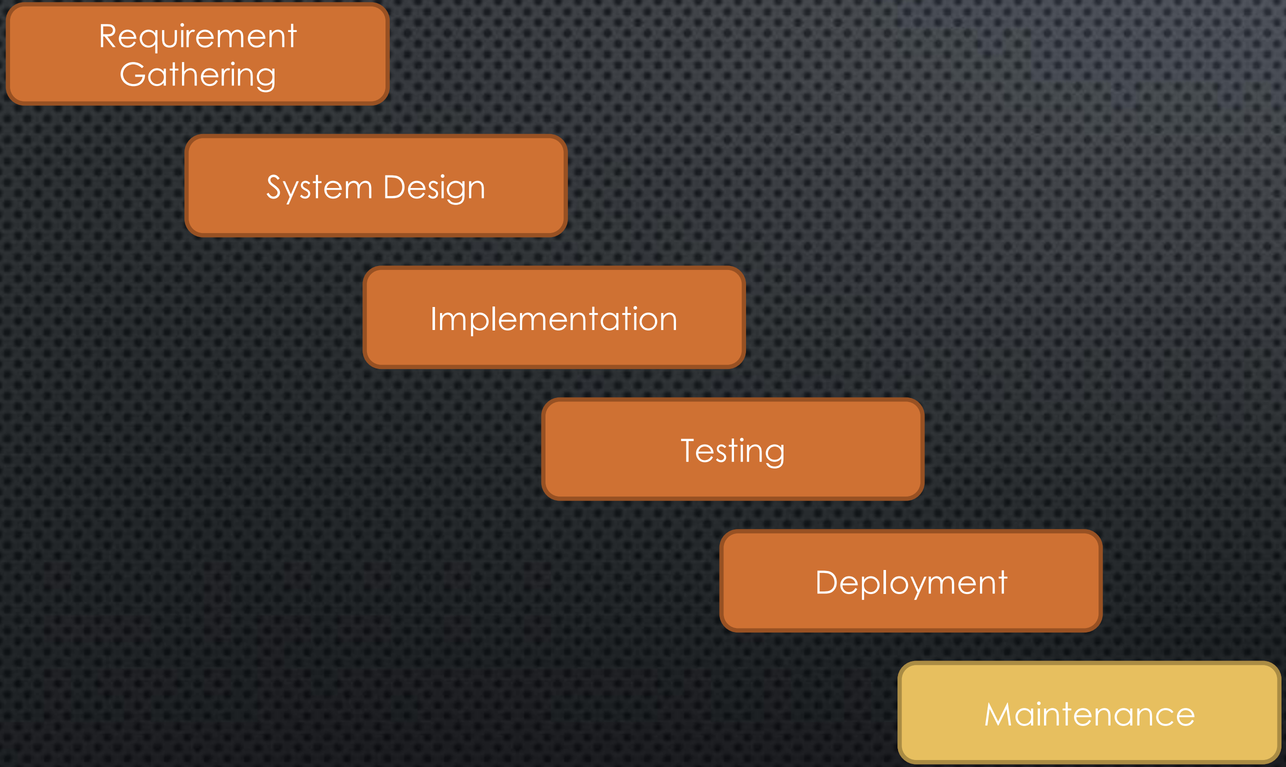
Some of the high level activities include installing the OS on the servers, installing security patches, hardening the servers, installing web servers and application servers, installing the database etc.

They also co-ordinate with network and IT administrative teams etc to finally get the application up and running on the production servers

MAINTENANCE

During the maintenance phase, the team ensures that the application is running smoothly on the servers without any downtime

Issues that are reported after going live are fixed by the team and tested by the testing team



Requirement Gathering

System Design

Implementation

Testing

Deployment

Maintenance

ADVANTAGES OF WATERFALL MODEL

- This model is simple and easy to understand and use
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process
- In this model phases are processed and completed one at a time. Phases do not overlap
- Waterfall model works well for smaller projects where requirements are clearly defined and very well understood

DISADVANTAGES OF WATERFALL MODEL

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage
- No working software is produced until late during the life cycle
- High amounts of risk and uncertainty
- Not a good model for complex and object-oriented projects
- Poor model for long and ongoing projects
- Not suitable for the projects where requirements are at a moderate to high risk of changing

WHEN TO USE THE WATERFALL MODEL

- This model is used only when the requirements are very well known, clear and fixed
- Product definition is stable
- Technology is understood
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short

- In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.
- Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everything from document till the logic
- In today’s world, Waterfall model has been replaced by other models like iterative, agile etc.