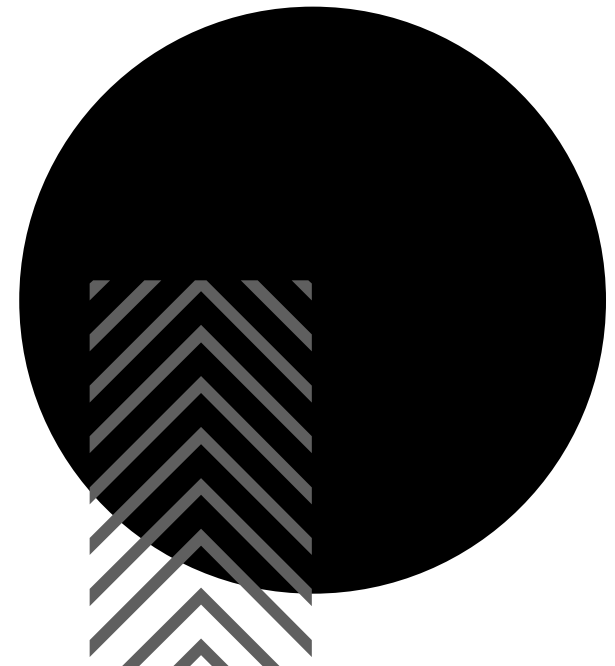




# **SOFTWARE DEVELOPMENT MODEL**



Software Engineering Teaching Team



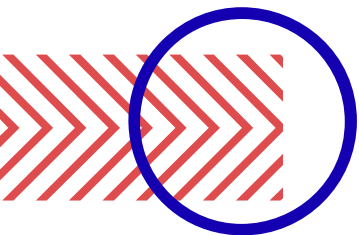
# STUDY MATERIALS

01

Software  
Development  
Cycle

02

Software  
Development  
Model

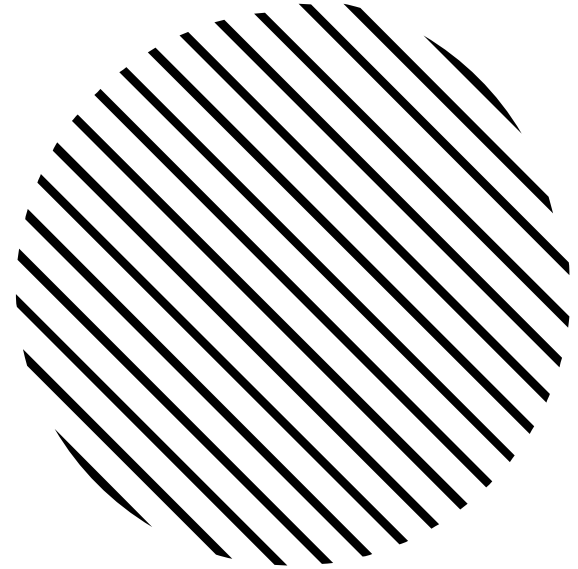




**01**

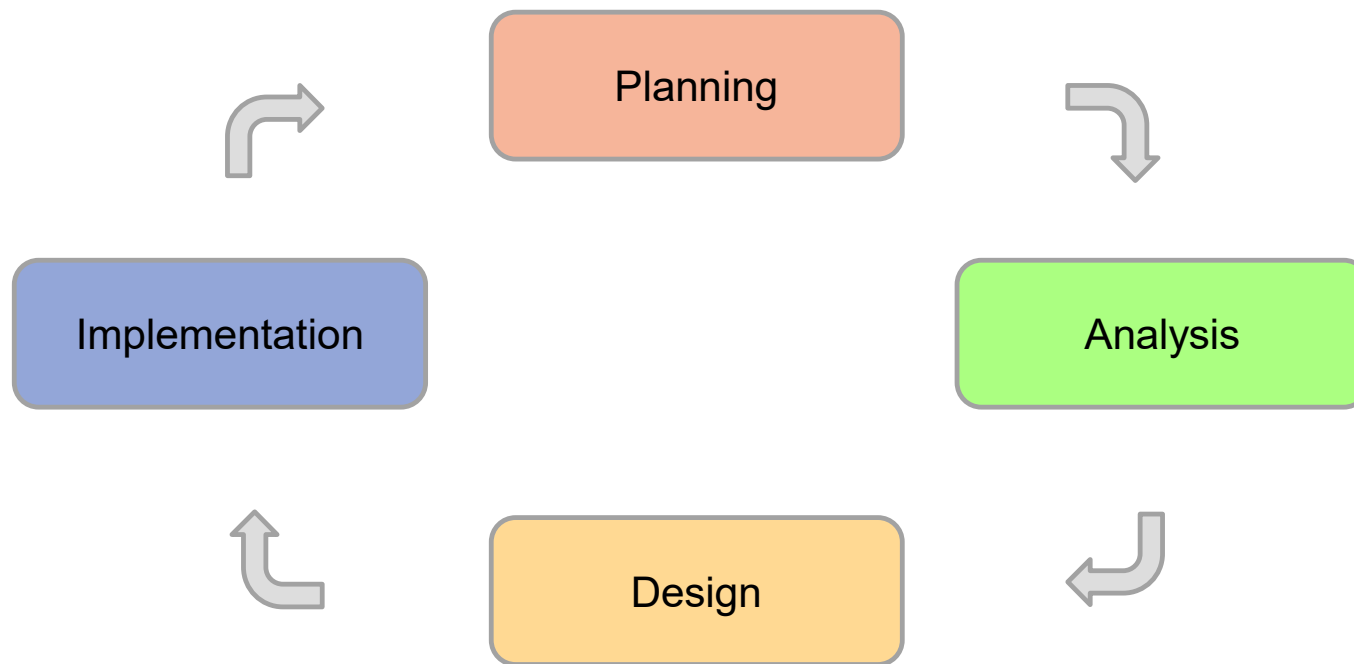


# Software Development Cycle



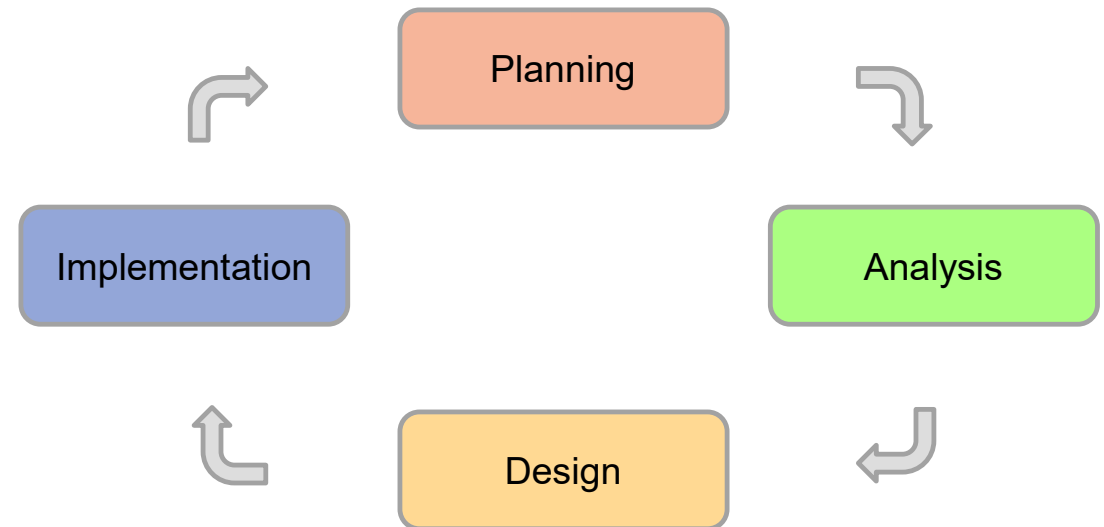
# ***SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)***

- SDLC is a software development life cycle consisting of several stages



# TAHAPAN SDLC

1. **Planning: Mengapa software harus dikembangkan?**
2. **System request**
3. **Feasibility Analysis (project size estimation)**
4. **Analysis: Who are the users and workflow of the software?** Requirement gathering
  - Business process modeling



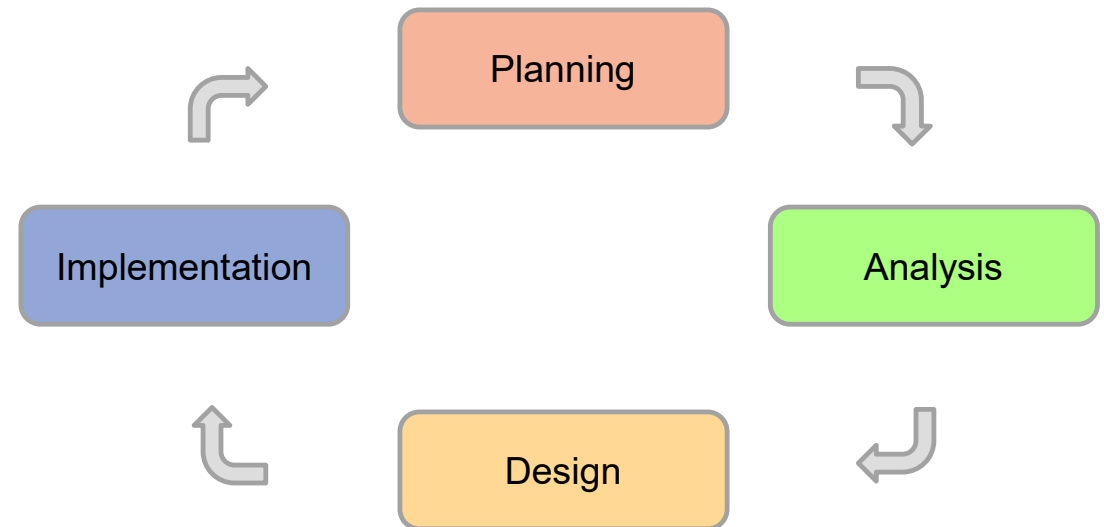
# TAHAPAN SDLC

## 3. **Design:** What is the composition of the software?

- ❑ Program design
- ❑ User interface design
- ❑ Data design

## 4. **Implementation:** Construction and submission of software

- ❑ System construction
- ❑ Testing
- ❑ Documentation
- ❑ Installation

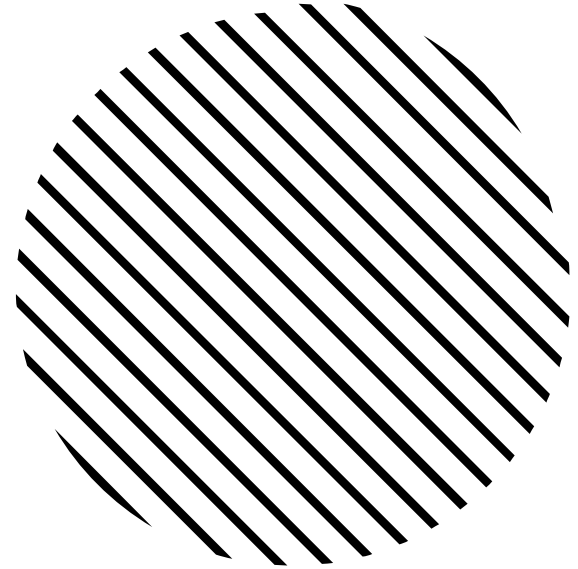




**02**



## **Software Development Model**



# **MODEL PENGEMBANGAN PERANGKAT LUNAK**

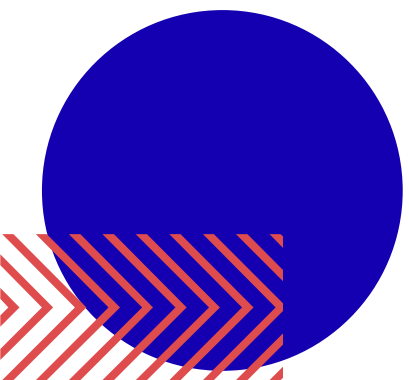
The process model for software engineering is selected based on the nature of the application and project, the methods and tools to be used.

A popular development model consists of:

- Model Waterfall
- Model Prototyping
- Model RAD
- Model Spiral

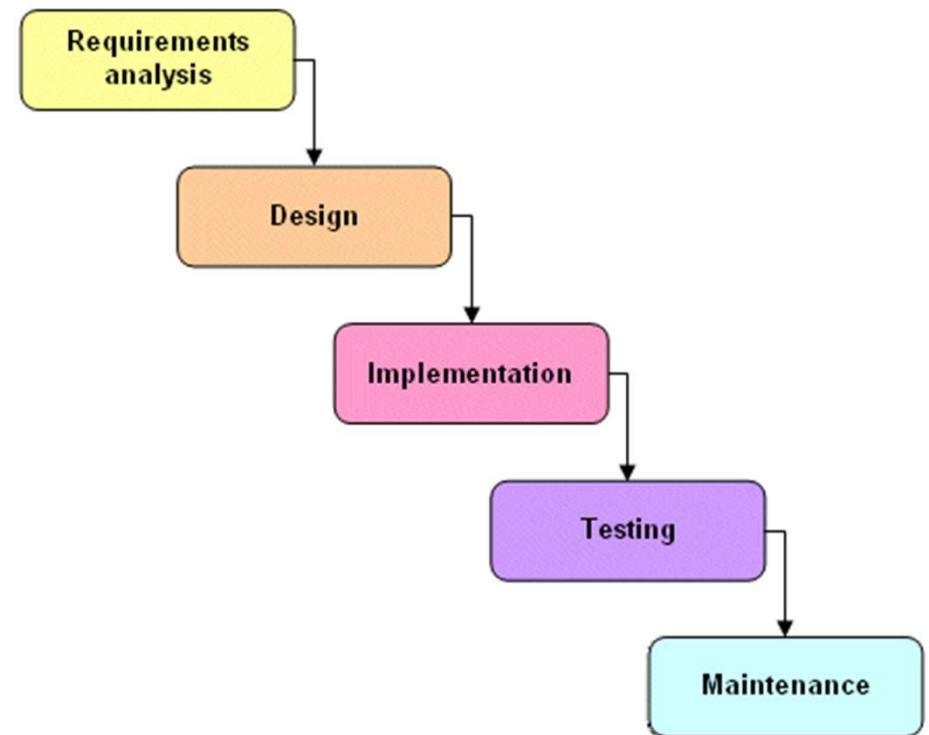


**WATERFALL**



## MODEL ***WATERFALL***

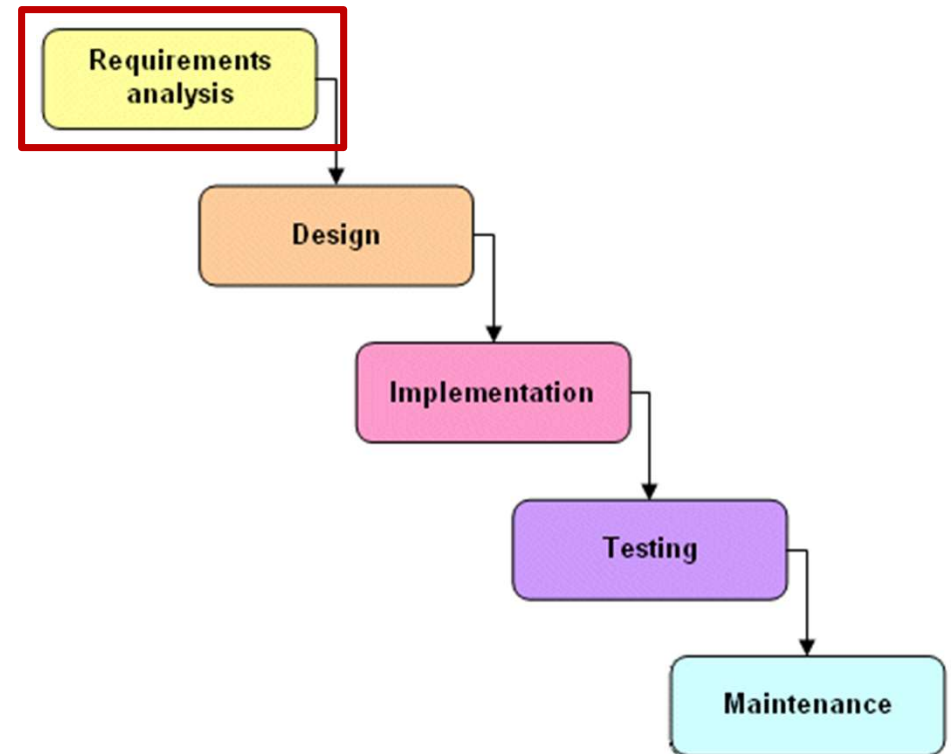
The software development process is sequential, where the work on each phase in Waterfall must be completed first before proceeding to the next phase.



# TAHAPAN MODEL *WATERFALL* (1)

## 1. *Requirement Analysis*

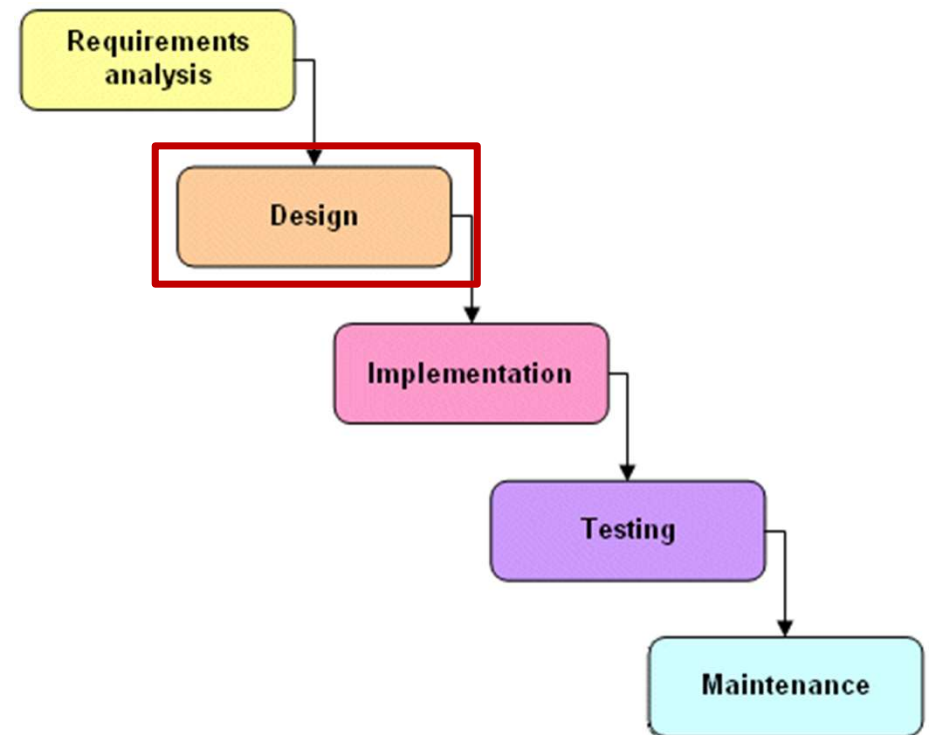
All software needs must be obtained in this phase. Data collection can be done by means of research, interviews, or literature studies. This stage will produce user requirement documents or can be said to be data related to user wishes in making the system.



## TAHAPAN MODEL *WATERFALL* (2)

### 2. *Design*

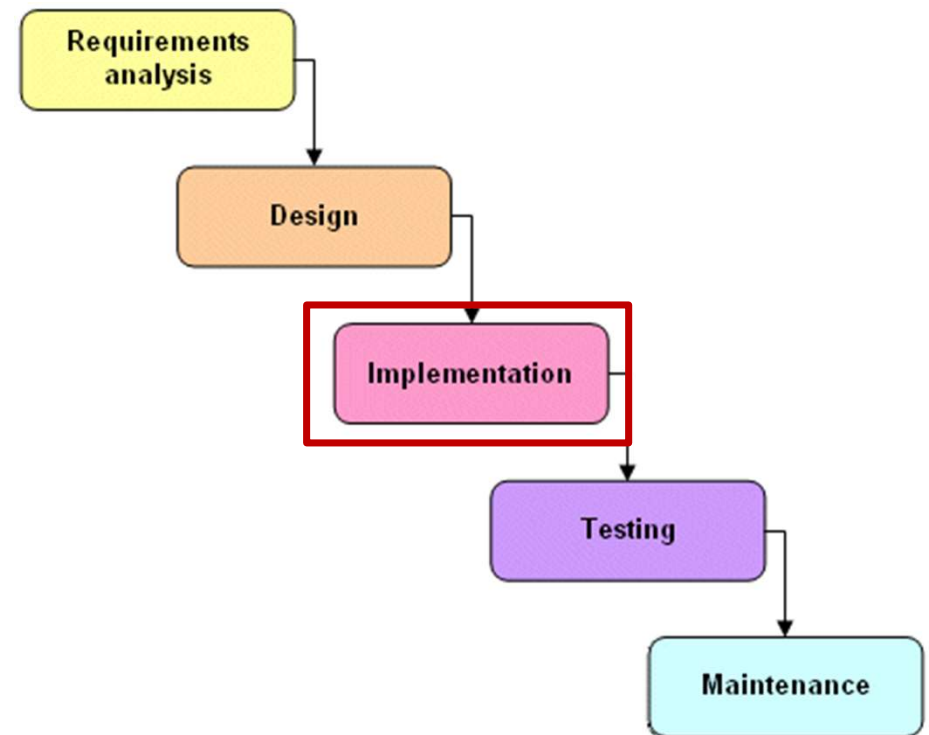
The design process will translate the requirements into an estimated software design before coding. This process focuses on data structures, software architecture, interface representations, and procedural details (algorithms). This stage will produce a document called software requirements.



## TAHAPAN MODEL *WATERFALL* (3)

### 3. *Implementation*

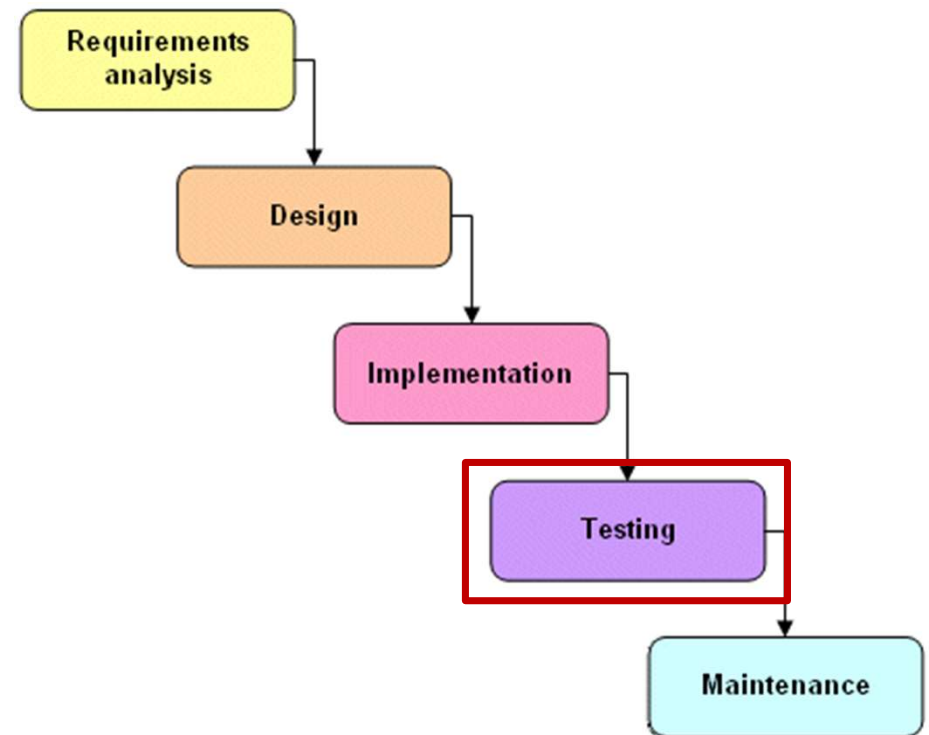
In this stage, programming is carried out. The creation of software is broken down into small modules that will later be combined in the next stage. In addition, in this stage, an examination of the module made is also carried out, whether it has fulfilled the desired function.



## TAHAPAN MODEL WATERFALL (4)

### 4. *Testing*

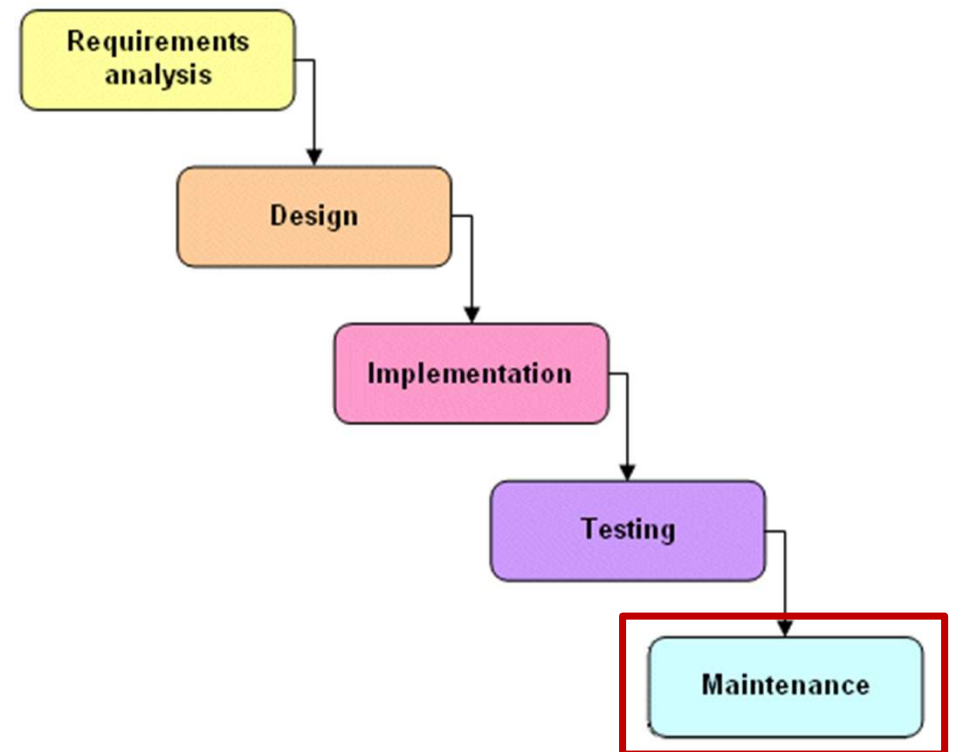
At this stage, the modules that have been created are combined and tested to find out whether the software created is in accordance with its design and whether there are still errors.



## TAHAPAN MODEL WATERFALL(5)

### 5. *Maintenance*

The finished software is run and maintained. Maintenance is included in fixing errors not found in the previous step.



# ADVANTAGES & DISADVANTAGES OF WATERFALL MODELS

## Advantage

- The sequence of the working process becomes more regular.
- The schedule of each process can be determined exactly.
- Clients can plan and prepare all data needs and processes that will be needed.

## Lack

- It is rigid, making it difficult to make changes in the middle of the process.
- Requires a complete list of needs at the beginning.
- The processing time is long because the input of the next stage is the output of the previous stage.



## **WATERFALL MODEL USE SITUATIONS**

The Waterfall method is suitable for systems that have a stable product definition and there is no change during development for any reason such as external changes, changes in goals, changes in budget or changes in technology, and all requirements are well understood at the beginning of the work.

# PM ProjectManager

	ASSIGNEE	STATUS ▾	%	PRIORITY	DUE	⌵ ⌵
		To Do	28%	—	18 Apr	
		To Do	41%	↑	20 Apr	
tion		To Do	5%	🔥	01 Sep	
umentation		To Do	5%	—	30 Apr	
e		To Do	0%	—	09 Sep	
arketing		To Do	0%	—	12 Sep	
ackaging ◇		To Do	0%	—	12 Sep	
in Sourcing		To Do	25%	▾	22 Mar	

## Design

PROGRESS

**28% complete** ☒

4 Subtasks

Engineering Drawings

**100%** ☒

3D Rendering

**0%** ☒

3D Printed Prototype

**0%** ☒

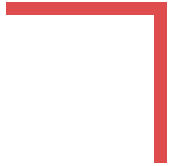
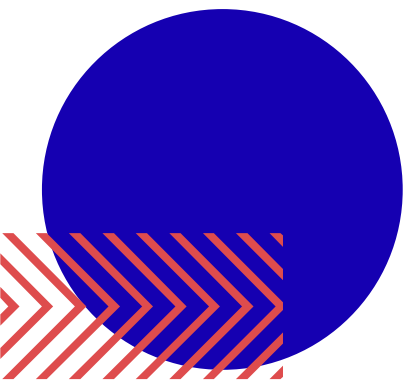
Stakeholder Feedback ◇

+ Add Subtask

DESCRIPTION ▾

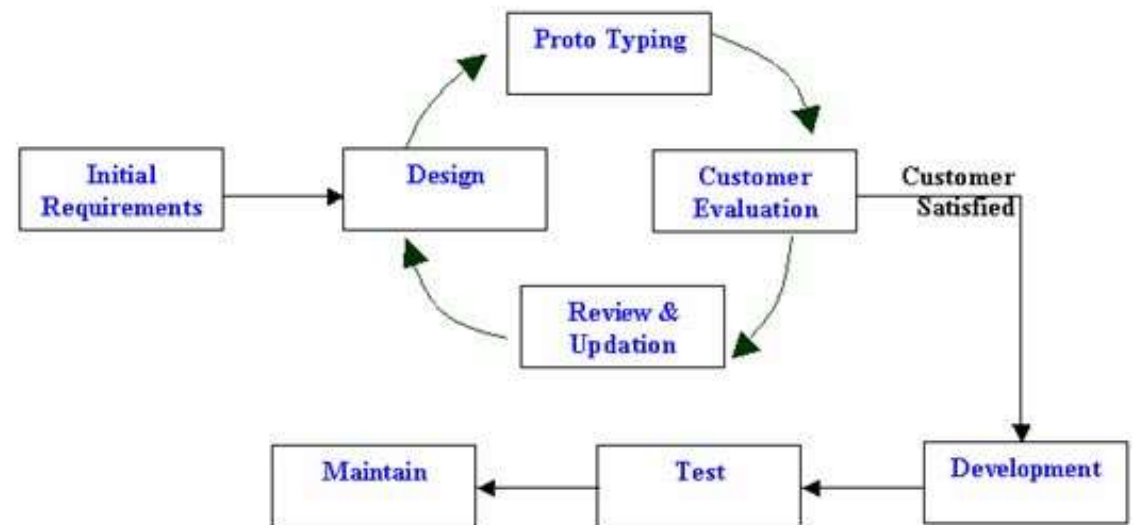
Add a description

# MODEL PROTOTYPING



## MODEL *PROTOTYPING*

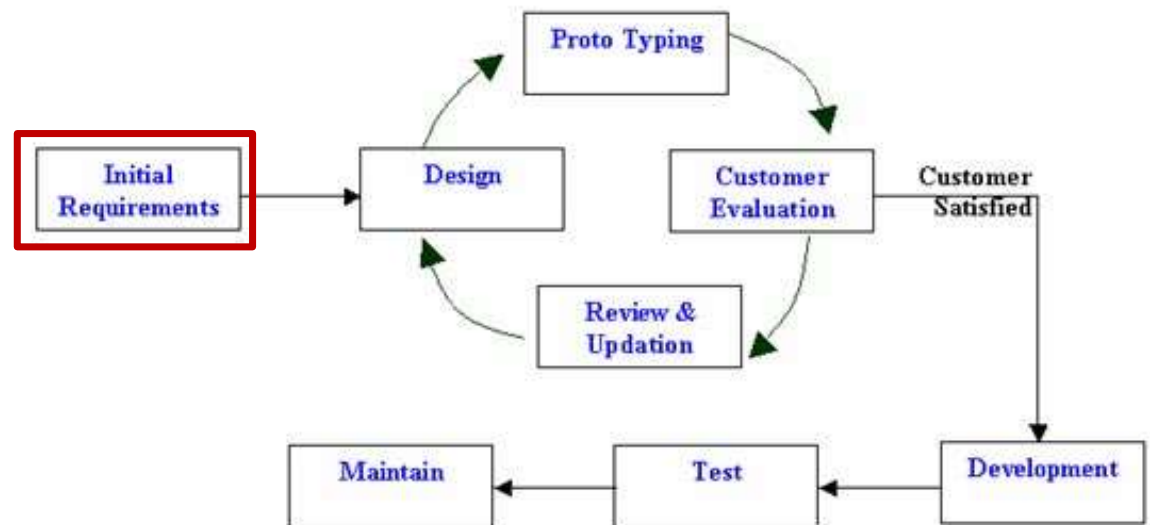
An approach that directly demonstrates how a piece of software will work in its environment before the actual construction phase is undertaken.



# STAGES OF MODEL PROTOTYPING (I)

## 1. Initial Requirements

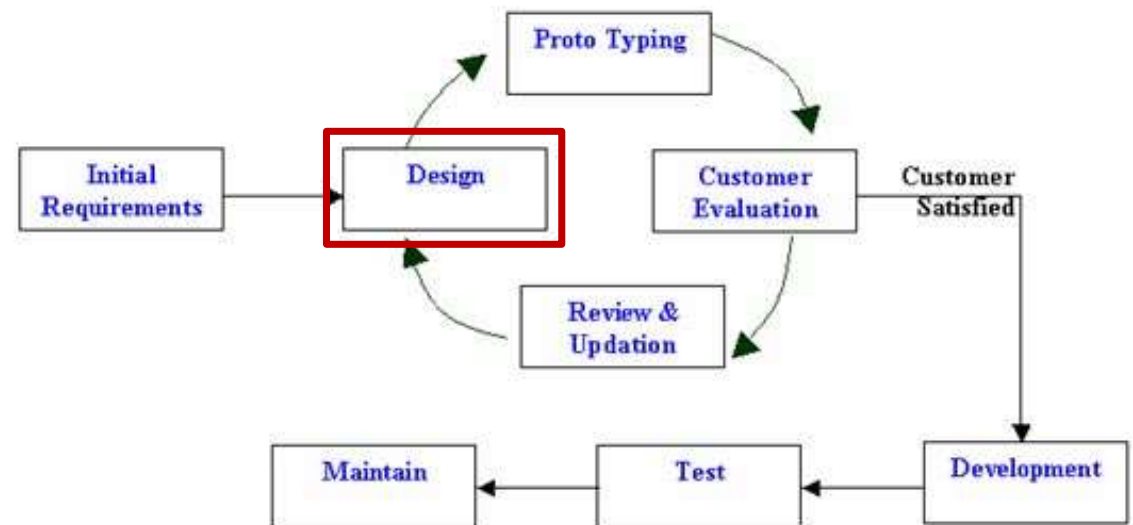
The client and developer jointly define the format of the entire software, identify all requirements, and outline the system to be created.



## STAGES OF MODEL PROTOTYPING (2)

### 2. *Design*

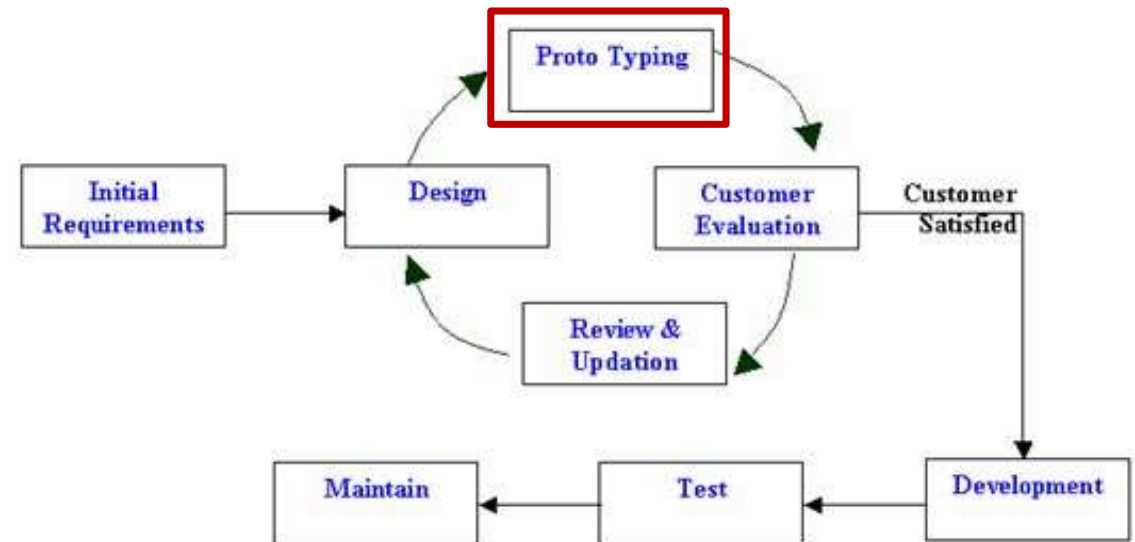
At this stage is carried out the translation of the requirements or data that has been analyzed into a form that is easy to understand by the user.



## TAHAPAN MODEL *PROTOTYPING*(3)

### 3. *Prototyping*

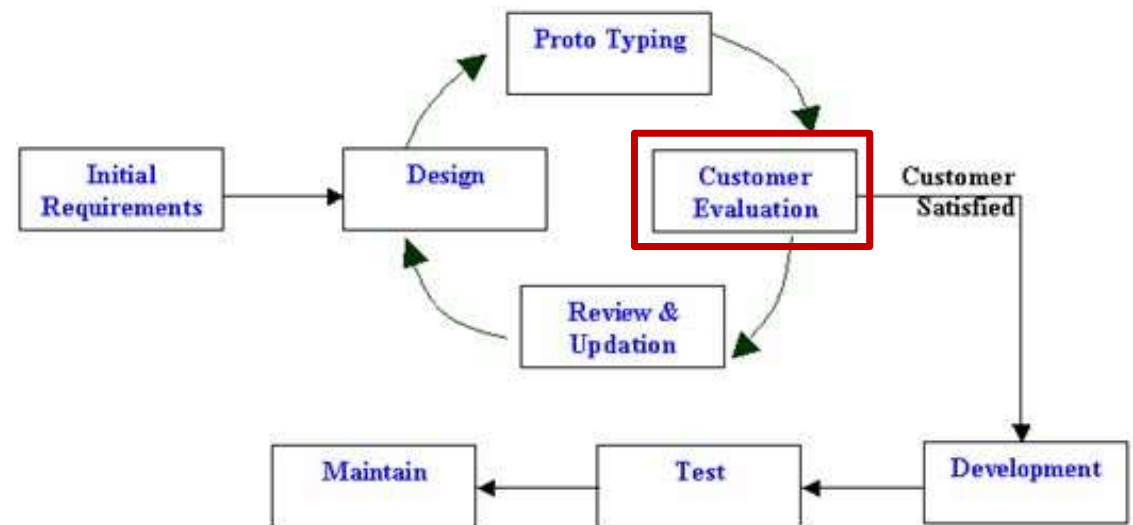
Build prototyping by creating temporary designs that focus on presenting to the client, for example by creating inputs and output formats.



## STAGES OF MODEL PROTOTYPING (4)

### 4. Customer Evaluation

This evaluation is carried out by the client, whether the prototyping that has been built is in accordance with the client's wishes. If it is appropriate, then the process proceeds to the next stage. But if not, prototyping is revised by repeating the previous steps.

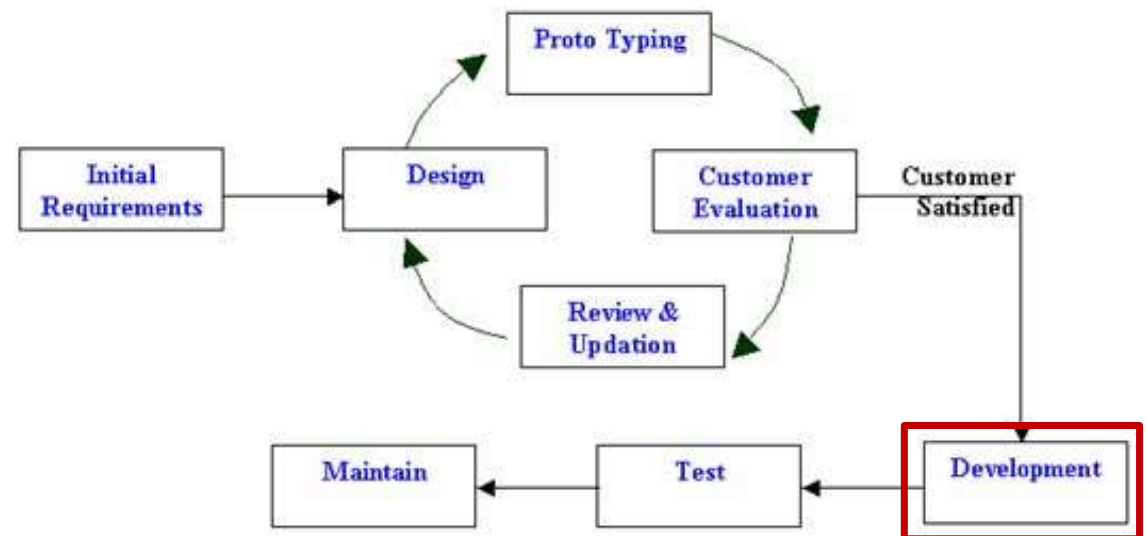




## STAGES OF MODEL PROTOTYPING (5)

### 5. Development

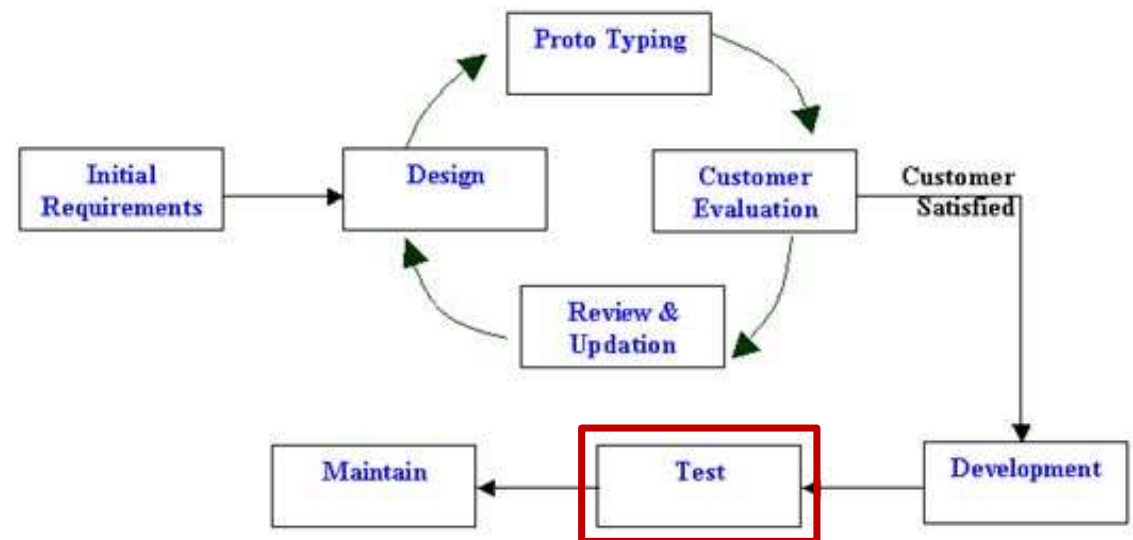
At this stage, the agreed prototyping is translated into the appropriate programming language.



## STAGES OF MODEL PROTOTYPING (6)

### 6. Test

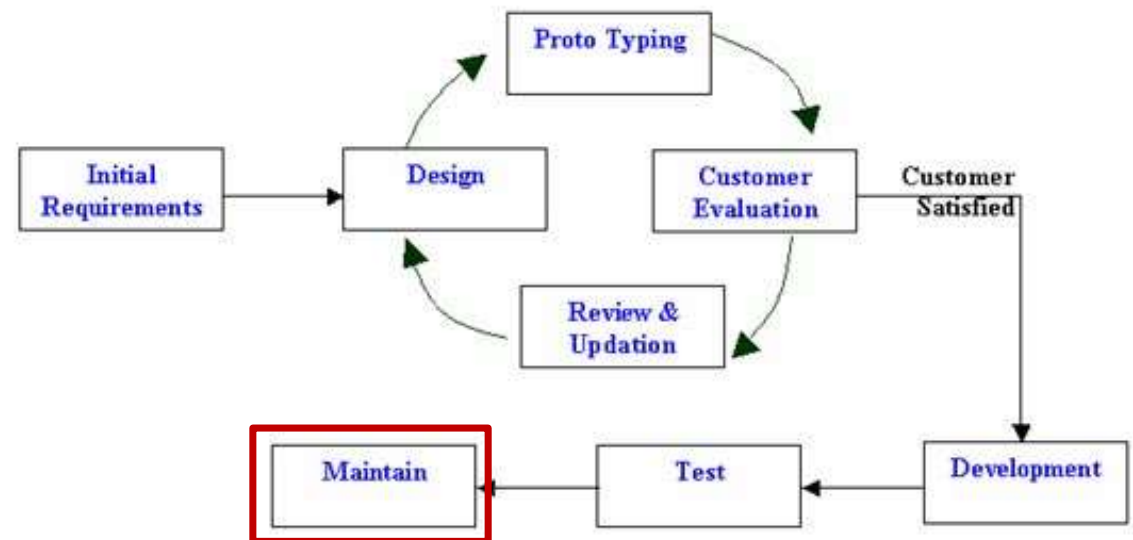
After the system has become a ready-to-use software, then the testing process is carried out.



# STAGES OF MODEL PROTOTYPING (7)

## 7. *Maintain*

Software that has been tested and received by customers is ready to use, in addition to maintenance that includes fixing errors not found in the previous step.



# **ADVANTAGES & DISADVANTAGES OF MODELS *PROTOTYPING***

## **Advantage**

- Increase user engagement.
- Reduce time and cost.
- Errors that occur in prototyping can be detected early.
- Implementation becomes easier because the user knows what to expect.

## **Lack**

- The analysis and design process is too short.
- The cost of prototyping is quite high.
- Usually less flexible in dealing with change.

## **MODEL PROTOTYPING USE SITUATIONS**

The prototyping method is suitable for projects that require a short time and users know how the process of creating projects to how to implement the project because between developers and users good communication is established.

# PROTOTYPE

*How to build representations of potential solutions:*

1

Keep the user in mind.



2

Plan the scope of testing.



3

Take action and fail fast.



4

Choose the right tools and fidelity.

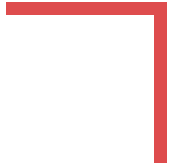
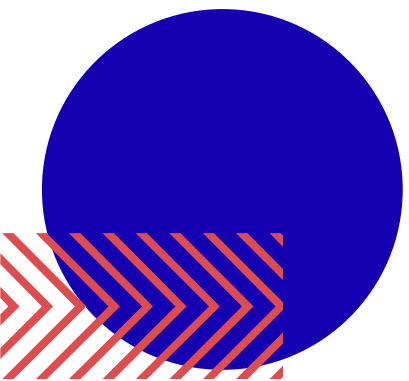


5

Record questions for testing.

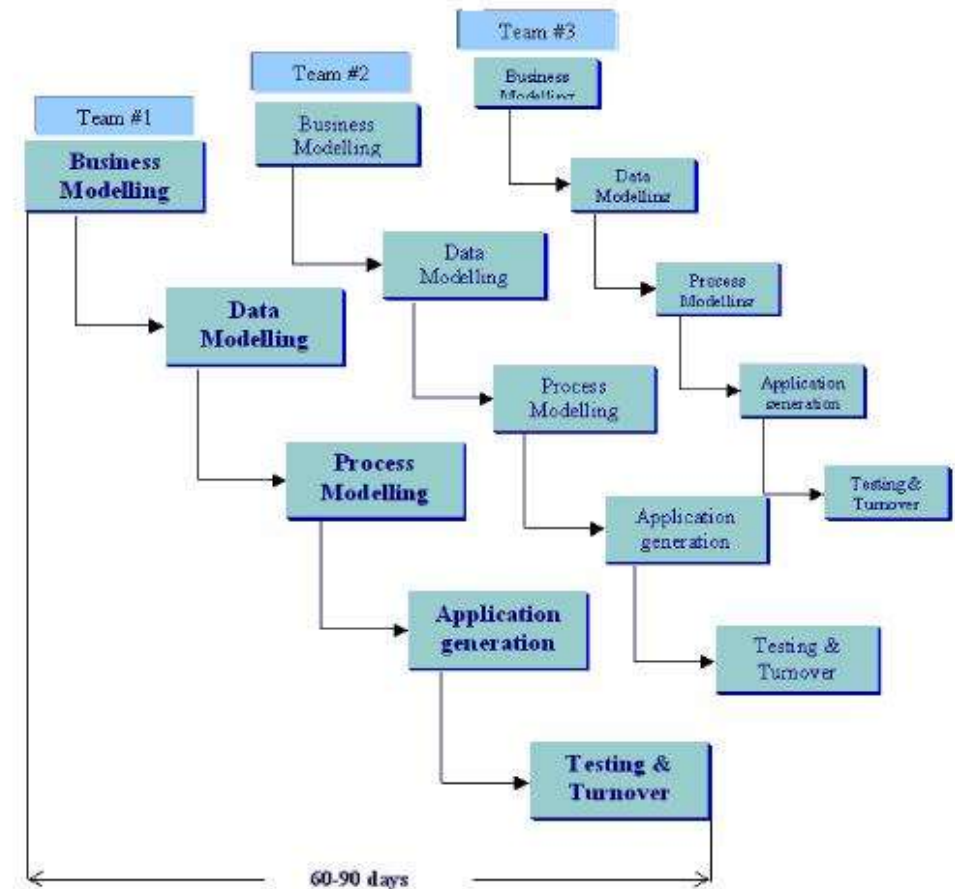


**MODEL RAD**



# MODEL *RAD*

*Rapid Application Development (RAD) is a linear sequential software development process model that emphasizes very short development cycles. RAD focuses on gathering customer needs through workshops or discussions*

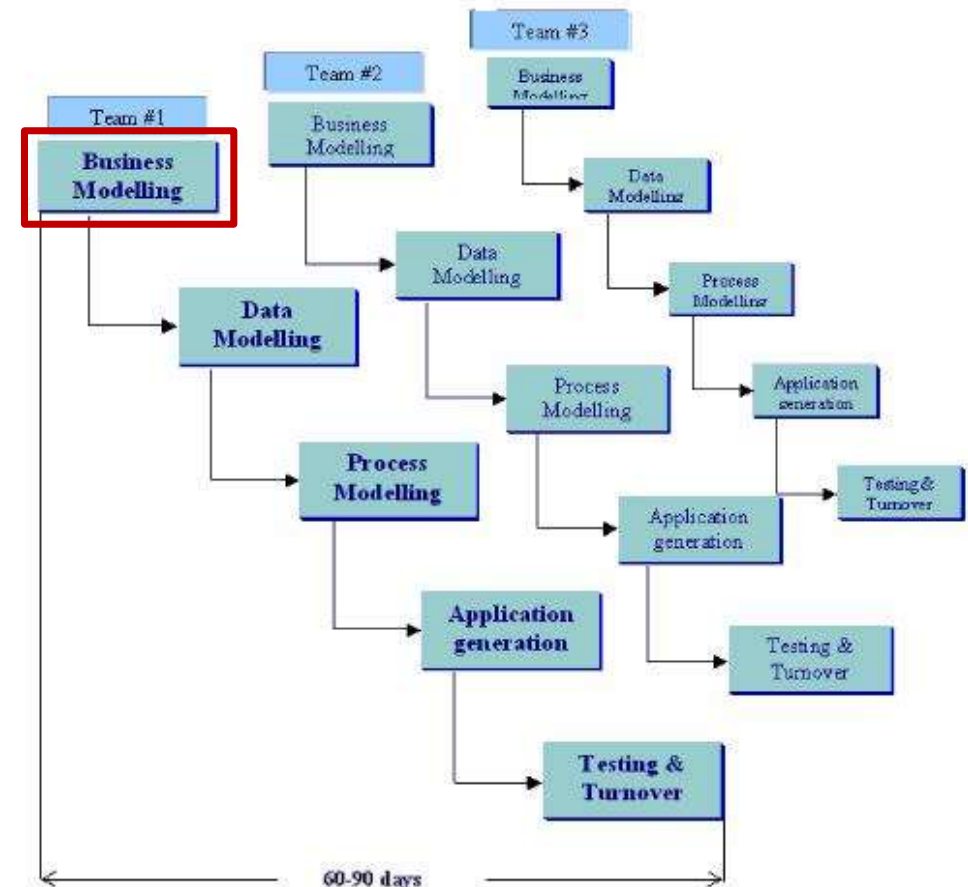




# RAD MODEL STAGES (I)

## 1. Business Modelling

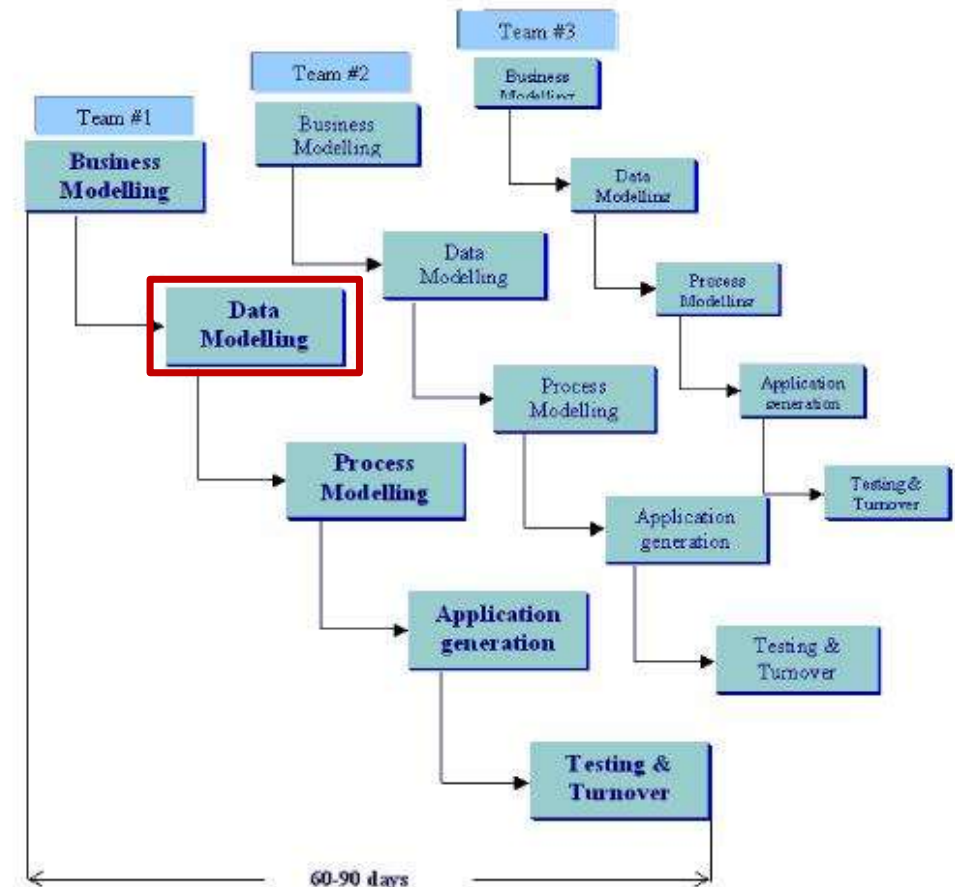
Business analysis is conducted to find important information related to the product being developed such as how the information can be obtained, how and when the information is processed, and what factors drive the successful flow of information.



## RAD MODEL STAGES (2)

### 2. Data Modelling

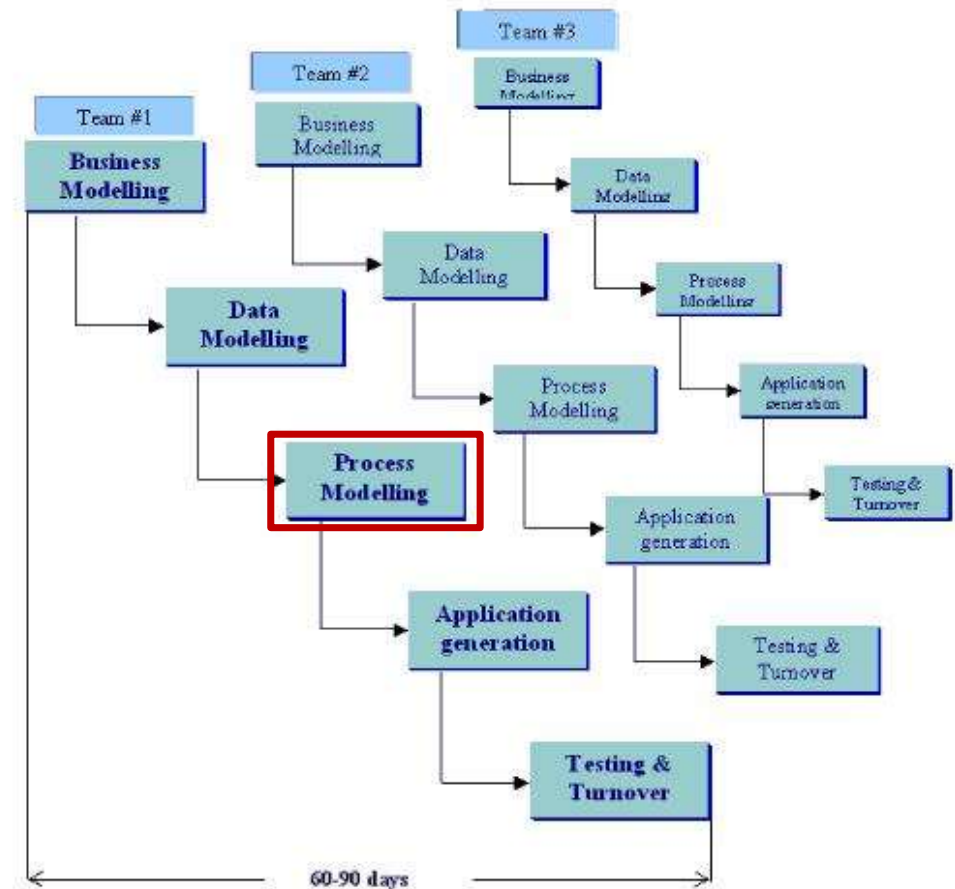
The information collected in the Business Modelling phase is reviewed and analyzed to form a collection of data objects that are essential to support the business.



## RAD MODEL STAGES (3)

### 3. *Process Modelling*

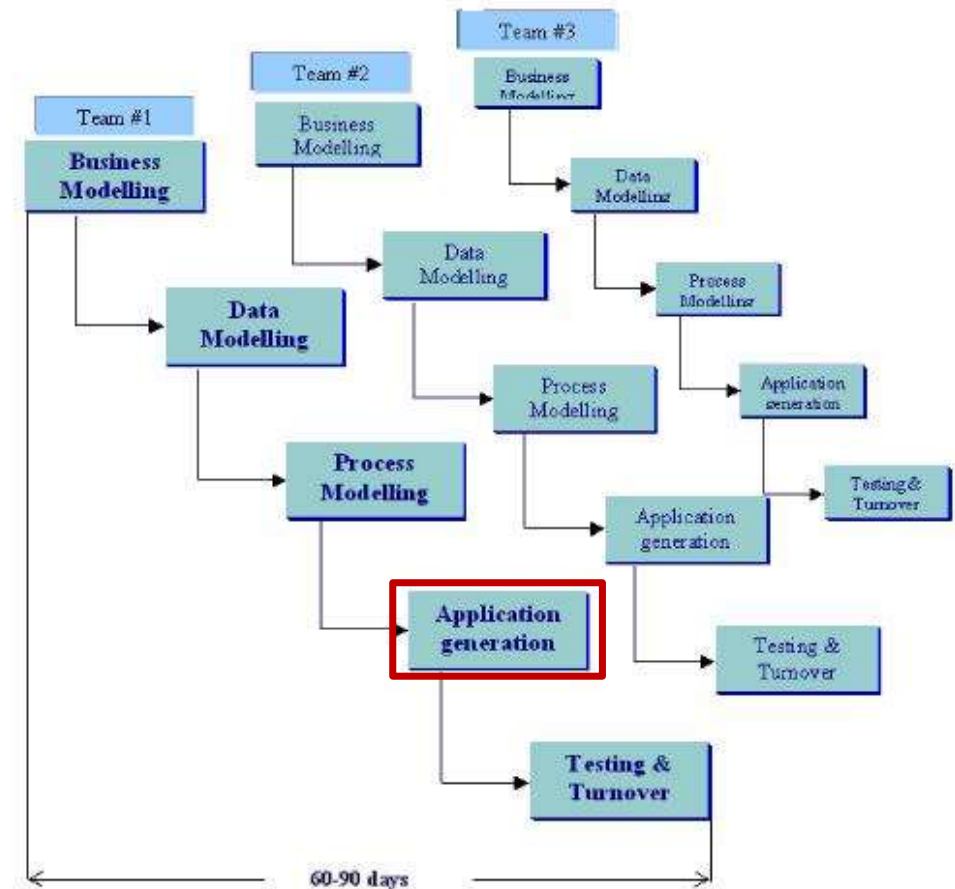
The set of data objects defined in the Data Modeling phase is transformed to establish the flow of business information required to achieve specific business objectives as per the business model. There is information about the process for adding, deleting, or modifying data objects.



## RAD MODEL STAGES (4)

### 4. *Application Generation*

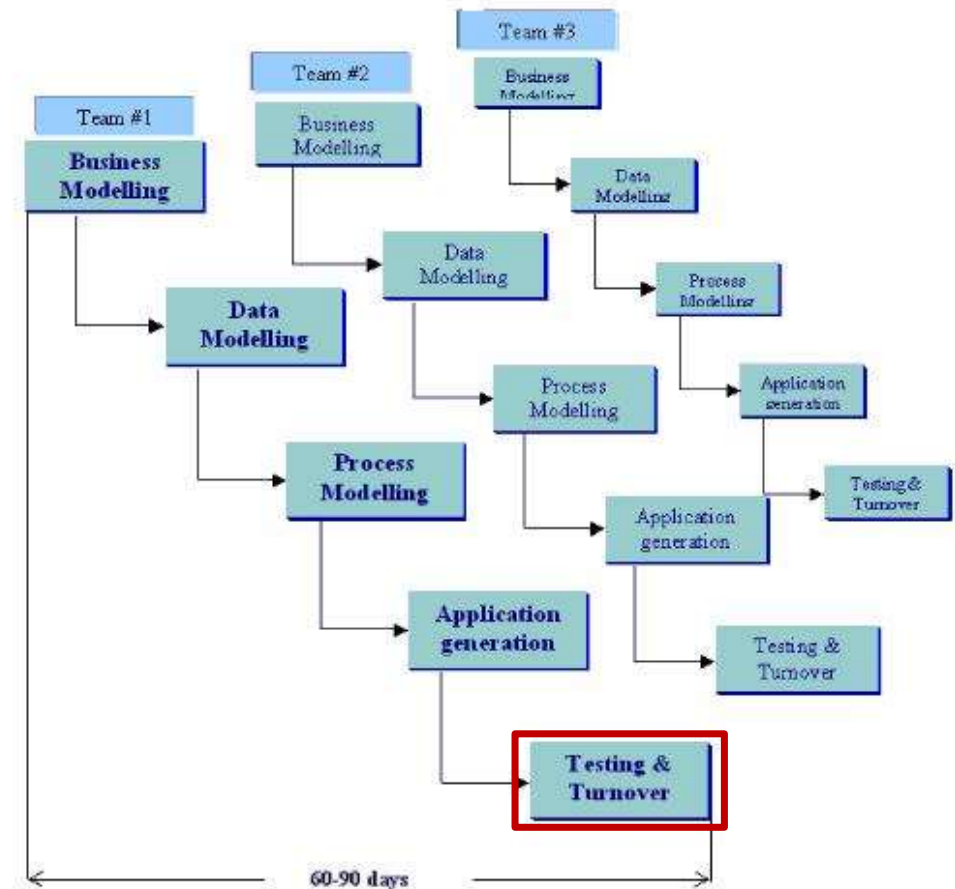
Systems are built by coding to transform processes and data models into prototypes.



## RAD MODEL STAGES (5)

### 5. Testing & Turnover

Because the RAD process emphasizes reuse, many program components have been tested. This reduces the overall testing time. However, new components need to be tested and all interfaces need to be fully tested.



# ADVANTAGES & DISADVANTAGES OF RAD MODELS

## Kelebihan

- The duration of development time may be reduced due to the reuse of existing components
- Encourage customer feedback
- Can accommodate changing needs

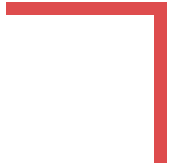
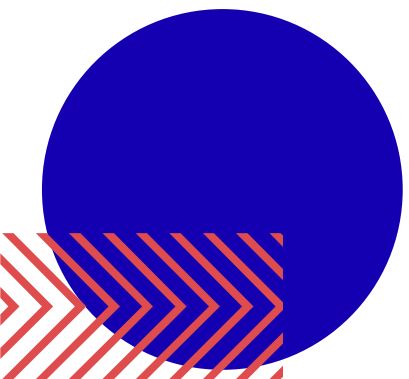
## Kekurangan

- There is a very strong reliance on team members to identify needs (business models)
- Only modularable systems can be built using RAD
- Requires user engagement throughout the software lifecycle

## **RAD MODEL USE SITUATIONS**

The RAD model is suitable for use if there is a need to create a system that can be modulated in a fairly short time of 2-3 months. In addition, designers are available to do modeling and there is a fairly high cost budget.

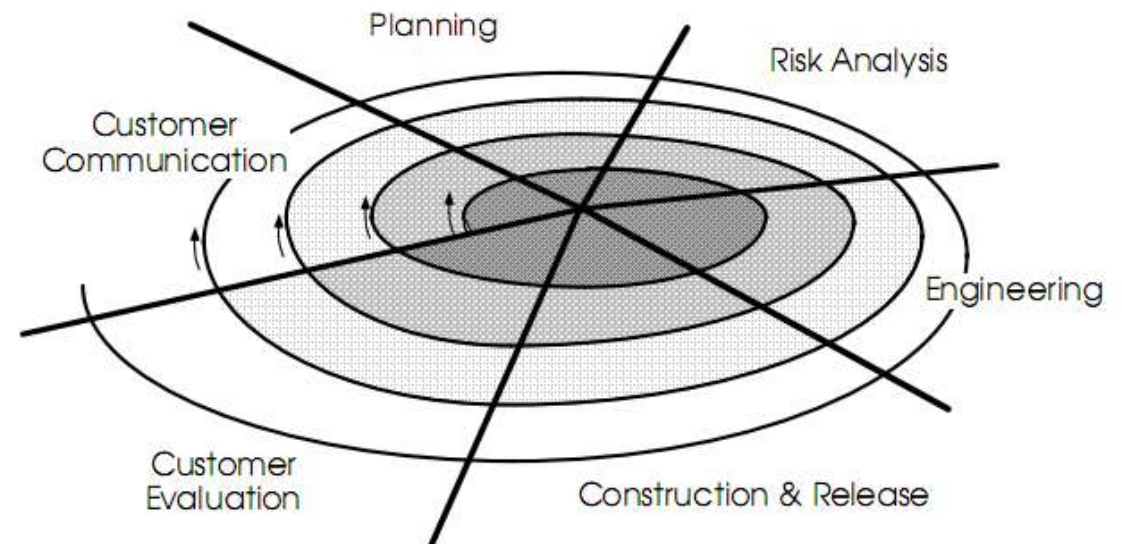
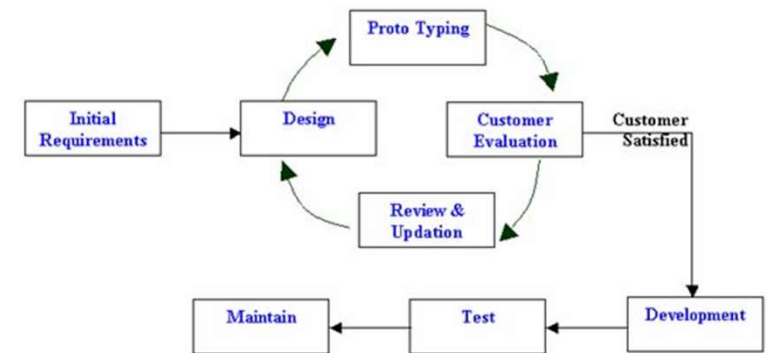
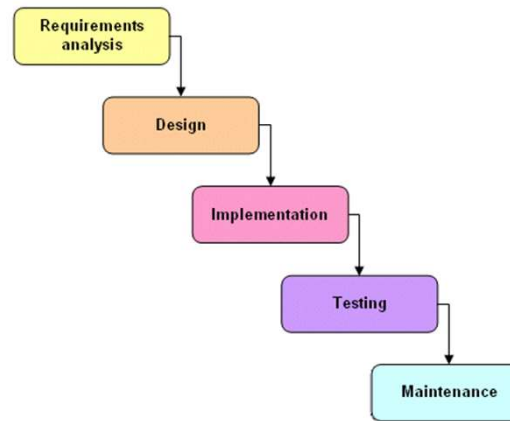
# MODEL SPIRAL





## MODEL *SPIRAL*

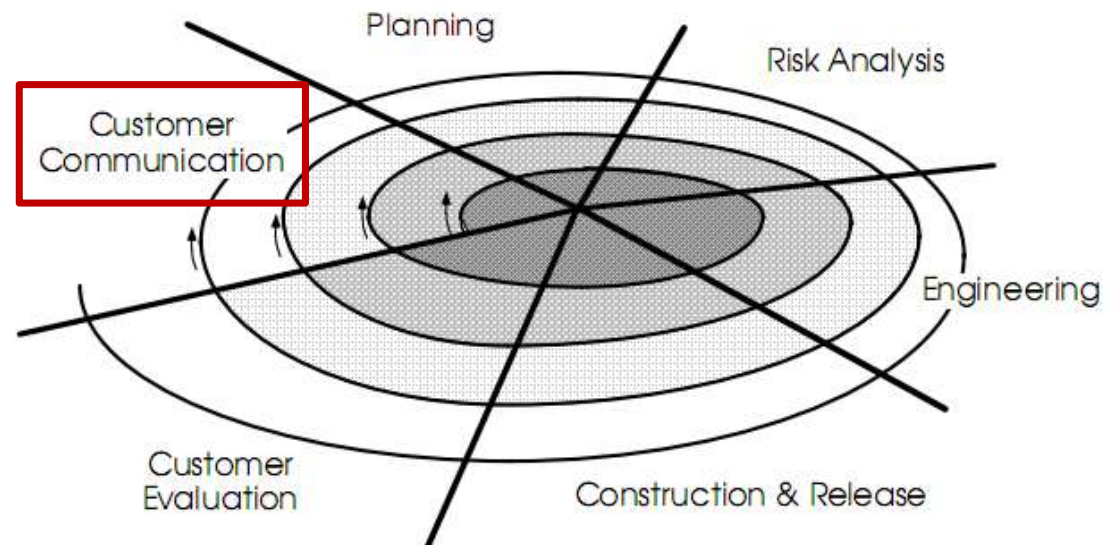
A form of evolution that uses the natural iteration method owned by the prototyping model and combined with systematic aspects developed with the Waterfall model.



# SPIRAL MODEL STAGES (I)

## 1. *Customer Communication*

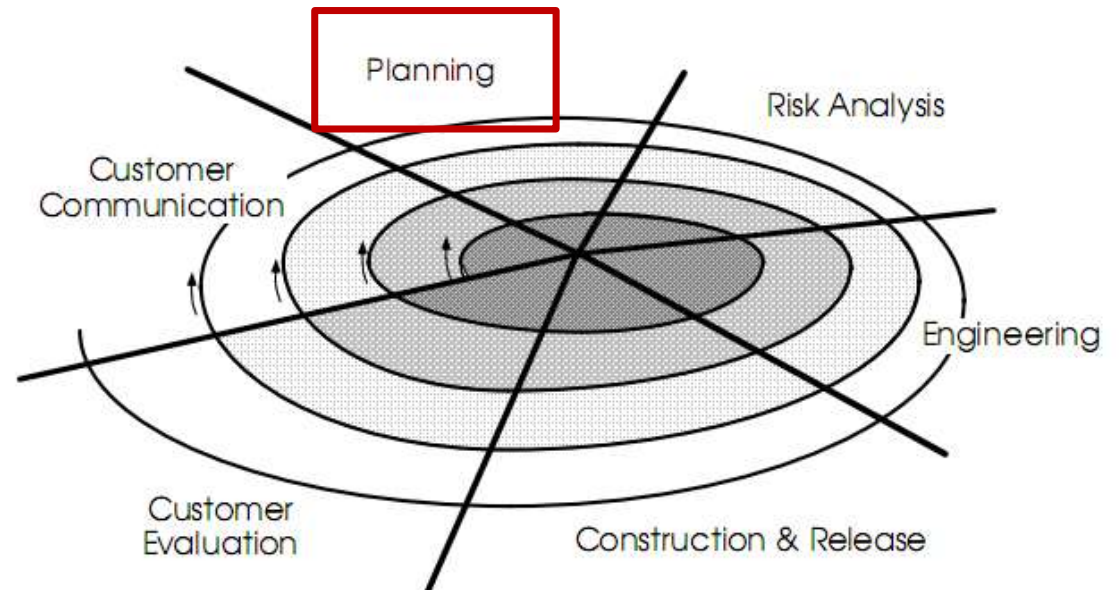
Activities needed to build effective communication between developers and users or clients, especially regarding the needs of clients.



## SPIRAL MODEL STAGES (2)

### 2. *Planning*

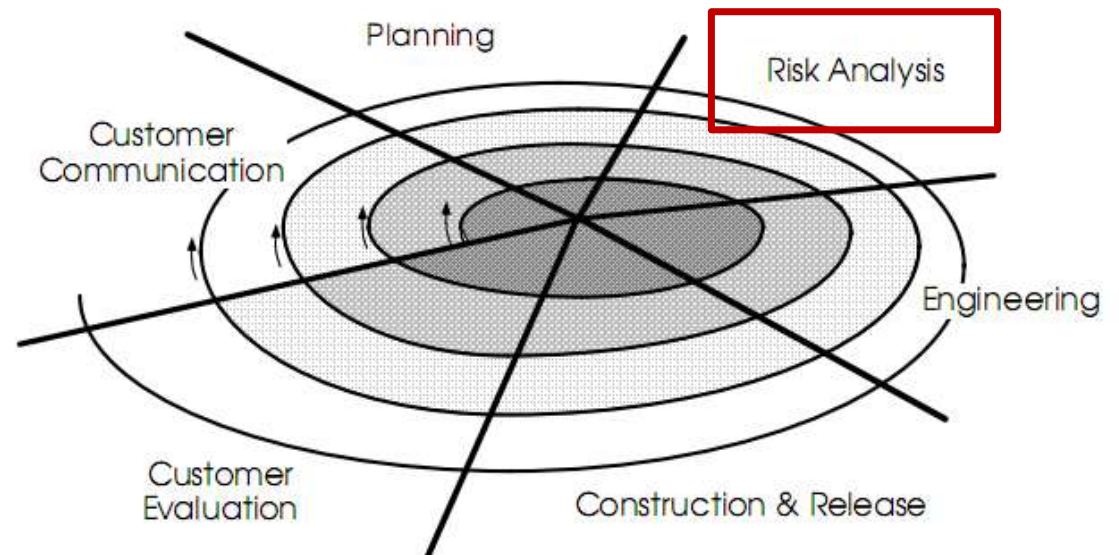
This planning activity is needed to determine resources, estimated processing time, and other information needed for software development.



## SPIRAL MODEL STAGES (3)

### 3. *Risk Analysis*

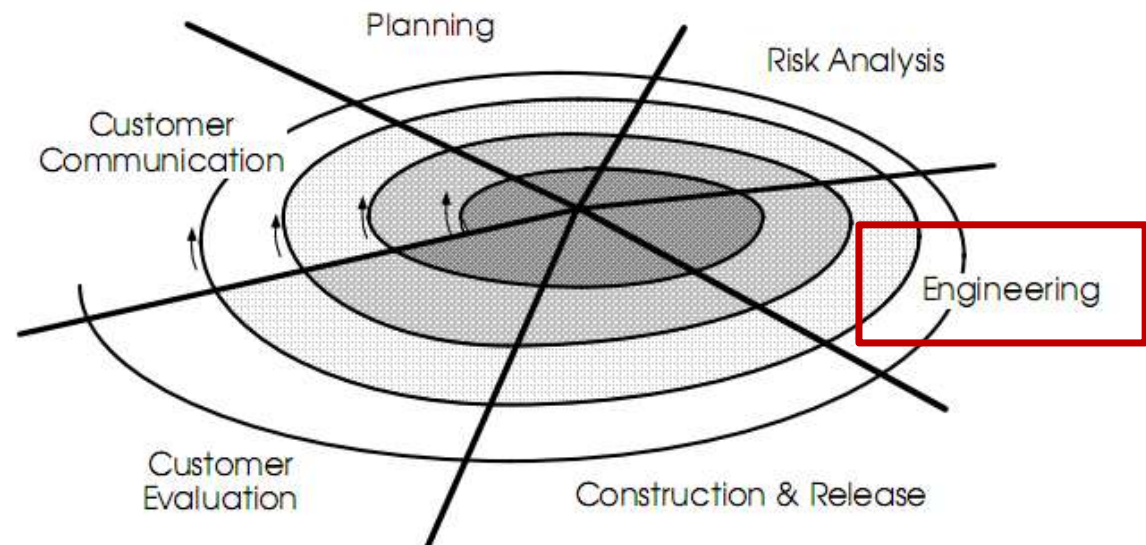
This risk analysis activity is carried out to analyze both technical and managerial risks.



## SPIRAL MODEL STAGES (4)

### 4. *Engineering*

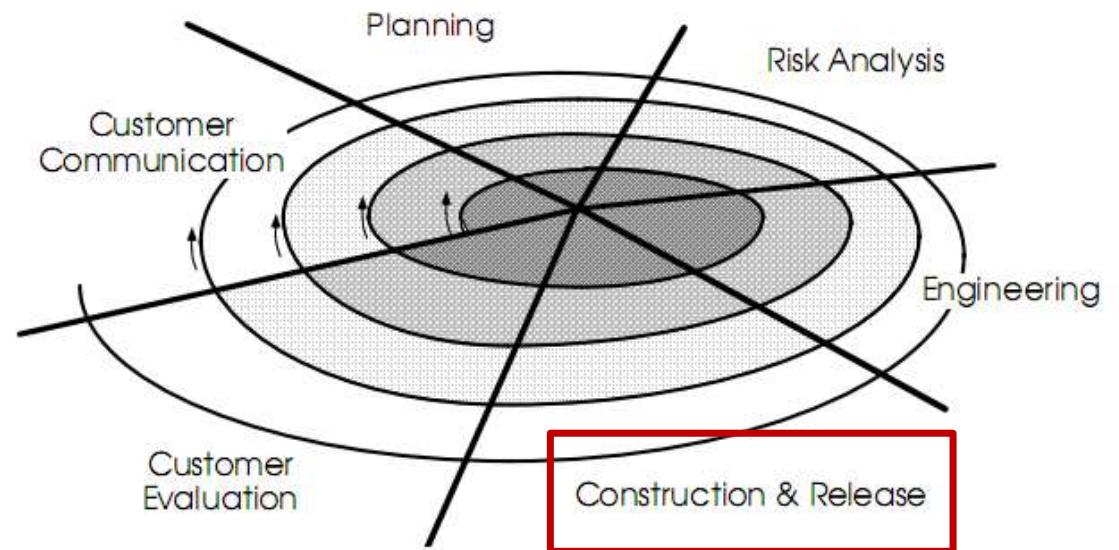
The activity required to build one or more technical representations of the application, namely in the form of designing diagrams needed for the coding process.



## SPIRAL MODEL STAGES (5)

### 5. *Construction & Release*

Activities required for software development, testing, installation and provision of user support such as training on software use and documentation such as software usage manuals.

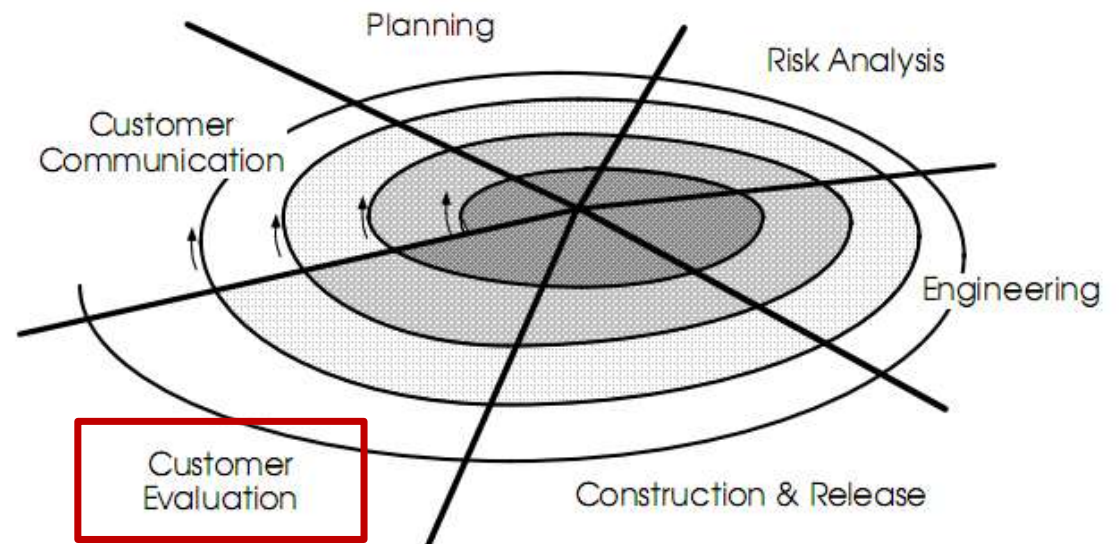




# SPIRAL MODEL STAGES (6)

## 6. *Customer Evaluation*

Activities required to obtain feedback from users or clients based on their evaluation during software representation in the engineering stage as well as in implementation during software installation in the construction & release stage.



# ADVANTAGES & DISADVANTAGES OF SPIRAL MODELS

## Advantage

- *Users and developers can understand well the software built.*
- *Estimates become more realistic as the project progresses as problems are discovered as soon as possible.*
- *Software engineers can work faster.*

## Lack

- It takes a long time and costs a lot.
- Requires a good long-term plan.
- Has risks that must be reconsidered by clients and developers.

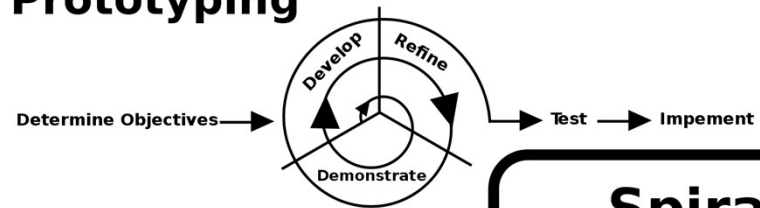


## **SPIRAL MODEL USE SITUATIONS**

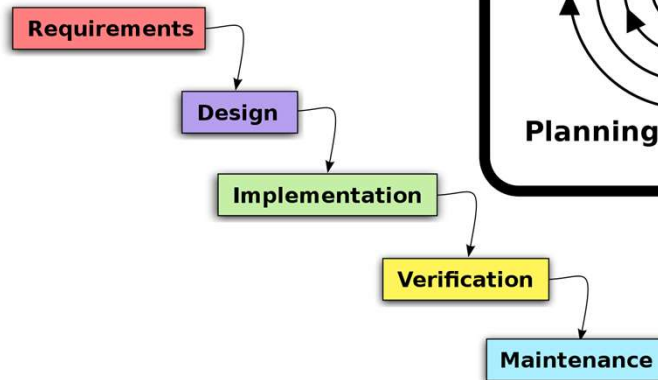
The spiral model is suitable for developing large-scale software systems because it has a risk analysis process that can greatly minimize possible risks and with time and cost targets that are not too binding.

The spiral model allows developers to use prototypes at each stage to reduce risk.

## Prototyping



## Waterfall



## Spiral

