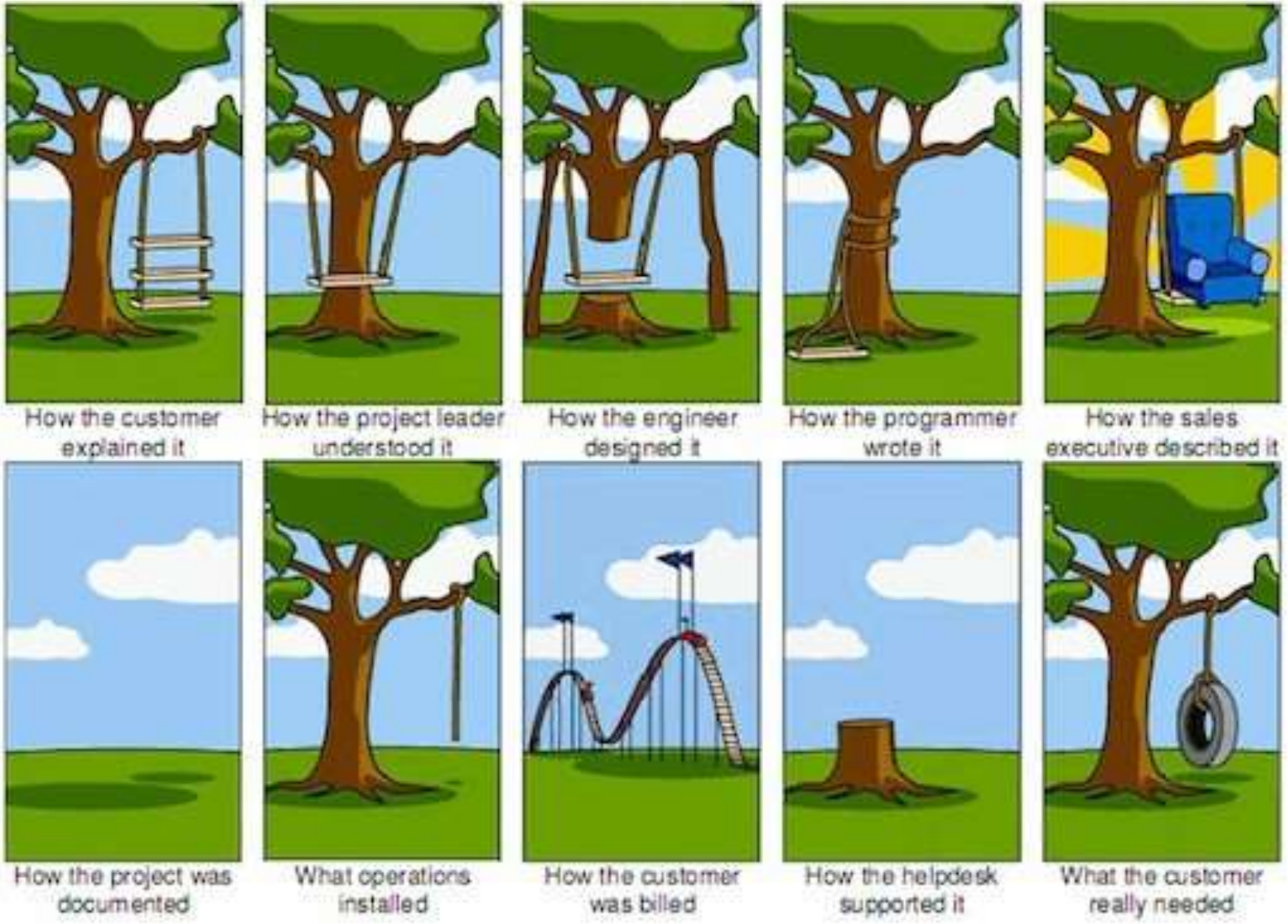

How the customer explained it
How the project leader understood it
How the engineer designed it
How the programmer wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

11.1 C
Life cycle of a system

Learn objectives:

✓11.2.1.1 Explain the stages of the life cycle used in solving problems

✓11.2.1.2 Analyse the advantages and disadvantages of **cyclical**, **waterfall** and **spiral** models

Criteria for success:

1. **Know the stages of the SDLC**
2. **Distinguish models**
3. **Explain the choice of the model**

The systems life cycle is the process of stages which occur during the development of a new ICT system

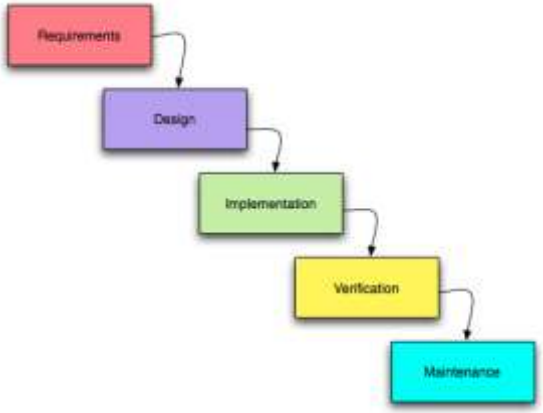
https://www.youtube.com/watch?v=4xCldpbDZ10

Stages of SDLC

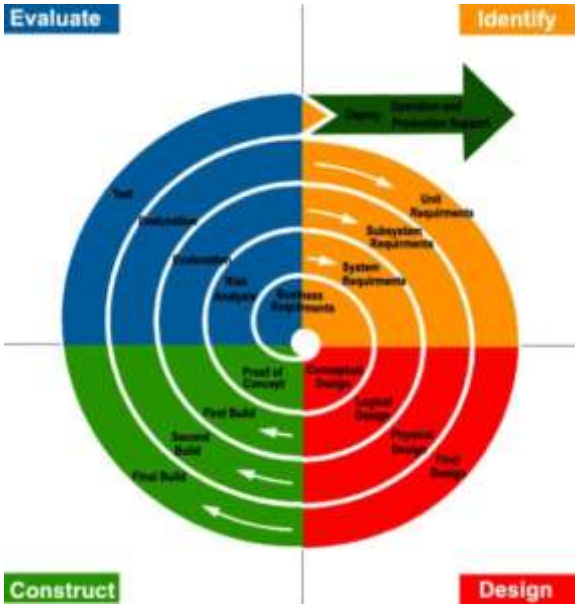
- **Analysis**. This is the first step when the team decides what the software needs to do. The main point is to think about what the user will want from the program. At this stage, it might be a good idea to ask other people what they want from the software. Who is going to use it? What information do they need to input? What information or data does it need to output? Consider existing similar systems or determine how the current system works.
- **Design** - the teamwork out the details of the program by breaking it down into smaller chunks. This includes thinking about the visual appearance and the programming behind the software. The team will use pseudocode and diagrams to work out how the program should go.
- **Implementation** - the program code is written. Good pseudocode allows the implementation stage to be relatively easy. The code is normally written in a high-level language.
- **Testing** - this involves testing the program under various conditions to make sure it is going to work. You need to think about what devices it could be used on and what might cause the program to crash.
- **Evaluation** is the final stage of the SDLC, which discusses the level of achievement of the goals. The key evaluation criterion is the use of the system by the intended users.

Group work Ambassadors

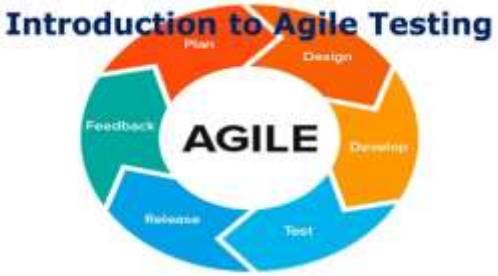
System life cycle models:



Waterfall model



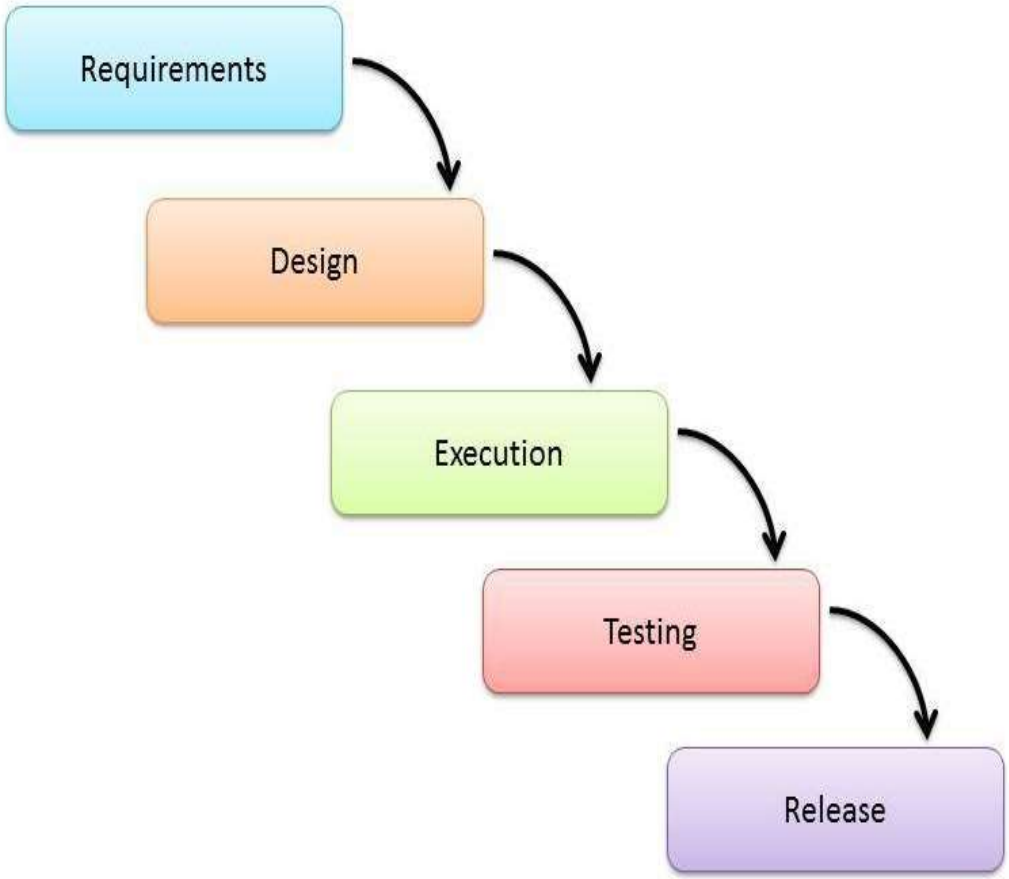
Spiral model



Agile model

Study the proposed material.
Select from it a description of the model, its characteristics, advantages and disadvantages.
Tell another group about your model.

Waterfall Model



One of the oldest, involves the sequential passage of stages, each of which must be completed completely before the beginning of the next. In the Waterfall model it is easy to manage the project. Due to its rigidity, the development is quick, the cost and time are predetermined. But this is a double-edged sword. Cascade model will give an excellent result only in projects with clearly defined requirements and methods of their implementation.

Waterfall Model

- as one stage is completed, the developers move on to the next step
- developers can’t go back to a previous step
- project outcome and an extensive plan must be set in the beginning and then followed carefully

When should you use Waterfall methodology

 When there is a clear picture of what the final product should be.

 When clients won’t have the ability to change the scope of the project once it has begun.

 When definition, not speed, is key to success.

Waterfall model

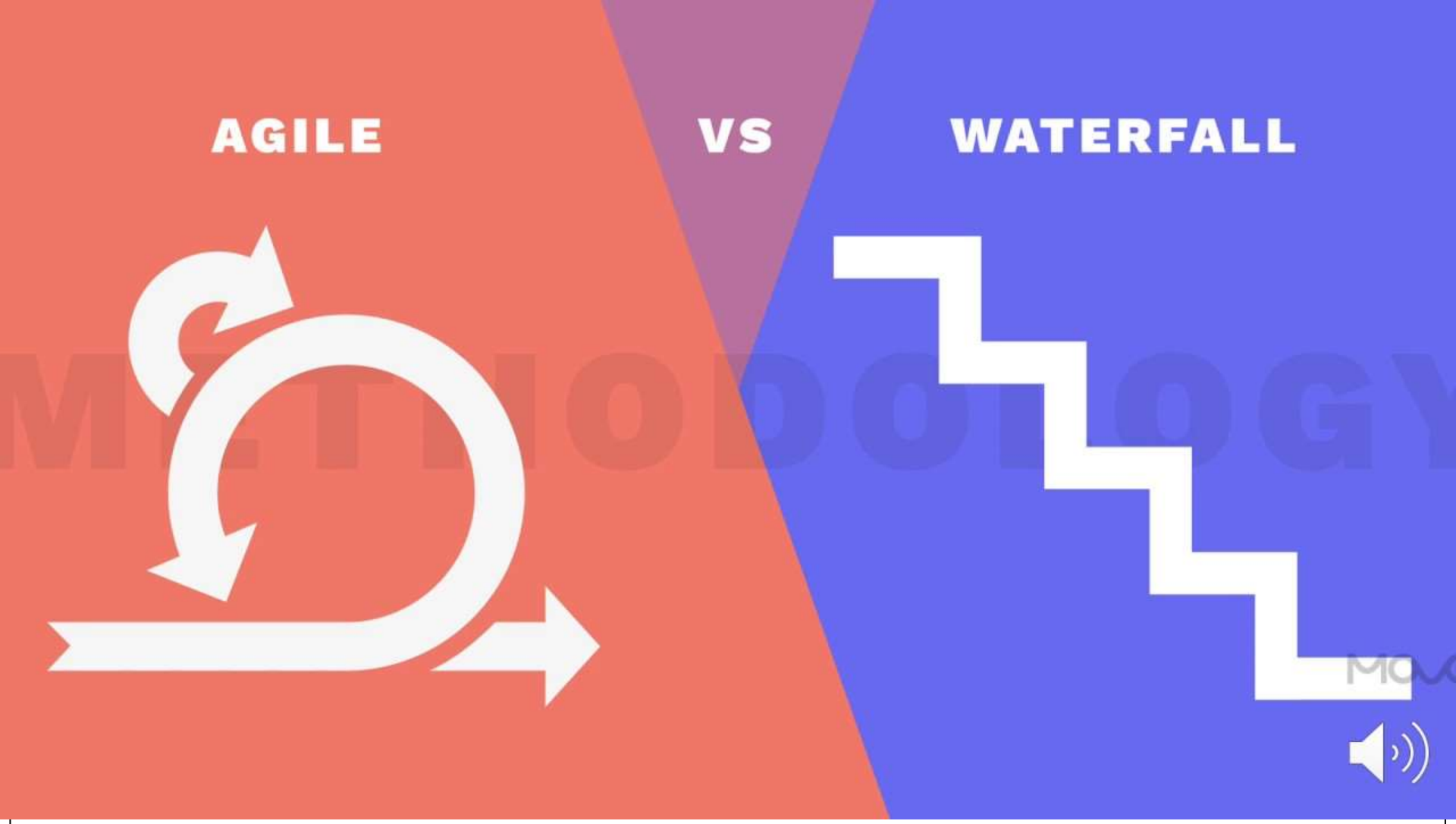
1. With the waterfall methodology, the client knows what to expect. They'll have an idea of the size, cost, and timeline for the project.

2. Once a step has been completed, developers can't go back to a previous stage and make changes.
3. Waterfall methodology relies heavily on initial requirements. However, if these requirements are faulty in any manner, the project is doomed.

4. If a requirement error is found, or a change needs to be made, the project has to start from the beginning with all new code.
5. The whole product is only tested at the end. If bugs are written early, but discovered late, their existence may have affected how other code was written.
6. The plan doesn't take into account a client's evolvingneeds. If the client realizes that they need more thanthey initially thought, and demand change, the projectwill come in late and impact budget.

Advantages and Disadvantages of Waterfall Model

Advantages	Disadvantages
• Easy to explain to the users. • Total cost known • Structures approach. • Stages and activities are well defined. • Each phase has specific deliverables. • Solid understanding of what is required • Delivery date known • Good control over project	• Very difficult to go back to any stage after it finished. • Costly and required more time, in addition to the detailed plan. • A little flexibility • A lot of pressure • Works bad for long-term projects

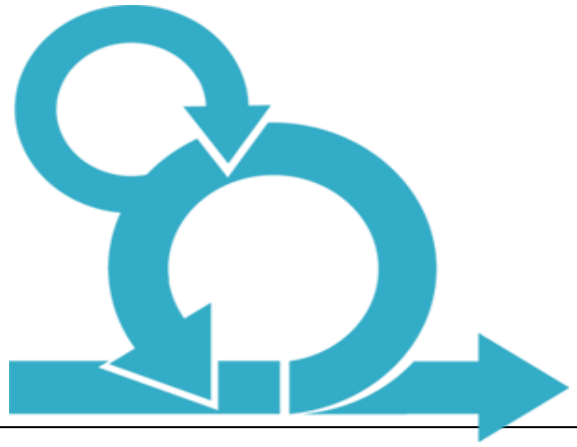


Developers start to work on small modules. The work on these modules is done in weekly or monthly sprints, and at the end of each sprint, project priorities are evaluated and tests are run. These sprints allow for bugs to be discovered, and customer feedback to be incorporated into the design before the next sprint is run.

Agile Model (Cyclical Model)

Agile came about as a “solution” to the disadvantages of the waterfall methodology. Instead of a sequential design process, the Agile methodology follows an incremental approach.

In the "flexible" development methodology, after each iteration, the customer can observe the result and understand whether it satisfies it or not. This is one of the advantages of a flexible model. Its drawbacks include the fact that, because of the lack of specific wording of the results, it is difficult to estimate the effort and cost required for development. Extreme programming is one of the most well-known applications of a flexible model in practice.



When should you use Agile methodology

- When rapid production is more important than the quality of the product.
- When clients will be able to chance the scope of the project.
- When there isn’t a clear picture of what the final product should look like.
- When you have skilled developers who are adaptable and able to think independently.
- When the product is intended for an industry with rapidly changing standards.

TASK 2 Agile model

+	-
+customer satisfaction	-not suitable for handling complex dependencies
+saves time	-
+The testing at the end of each sprint ensures that the bugs	-
+suitable for changing requirements	

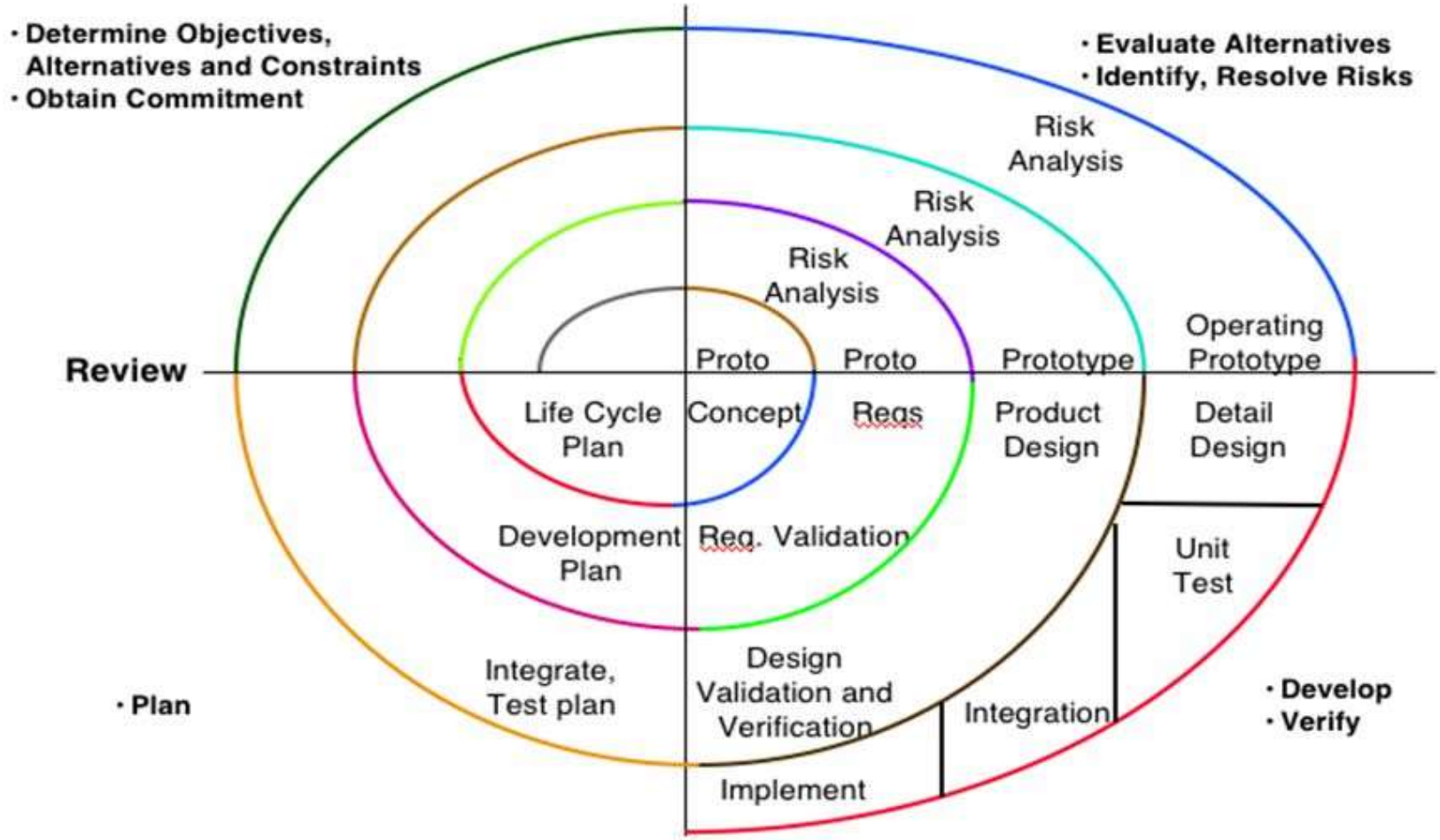
Agile Model

- Customer Satisfaction.
- Saves Time.
- Testing.
- Suitable for fixed or changing requirements.
- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.

3. "Spiral Model" (spiral model)

- "Spiral model" = incremental + waterfall model, but with an emphasis on risk analysis. It works well for solving mission-critical business tasks, when failure is incompatible with the company's activities, in the conditions of issuing new product lines, if necessary, scientific research and practical testing.

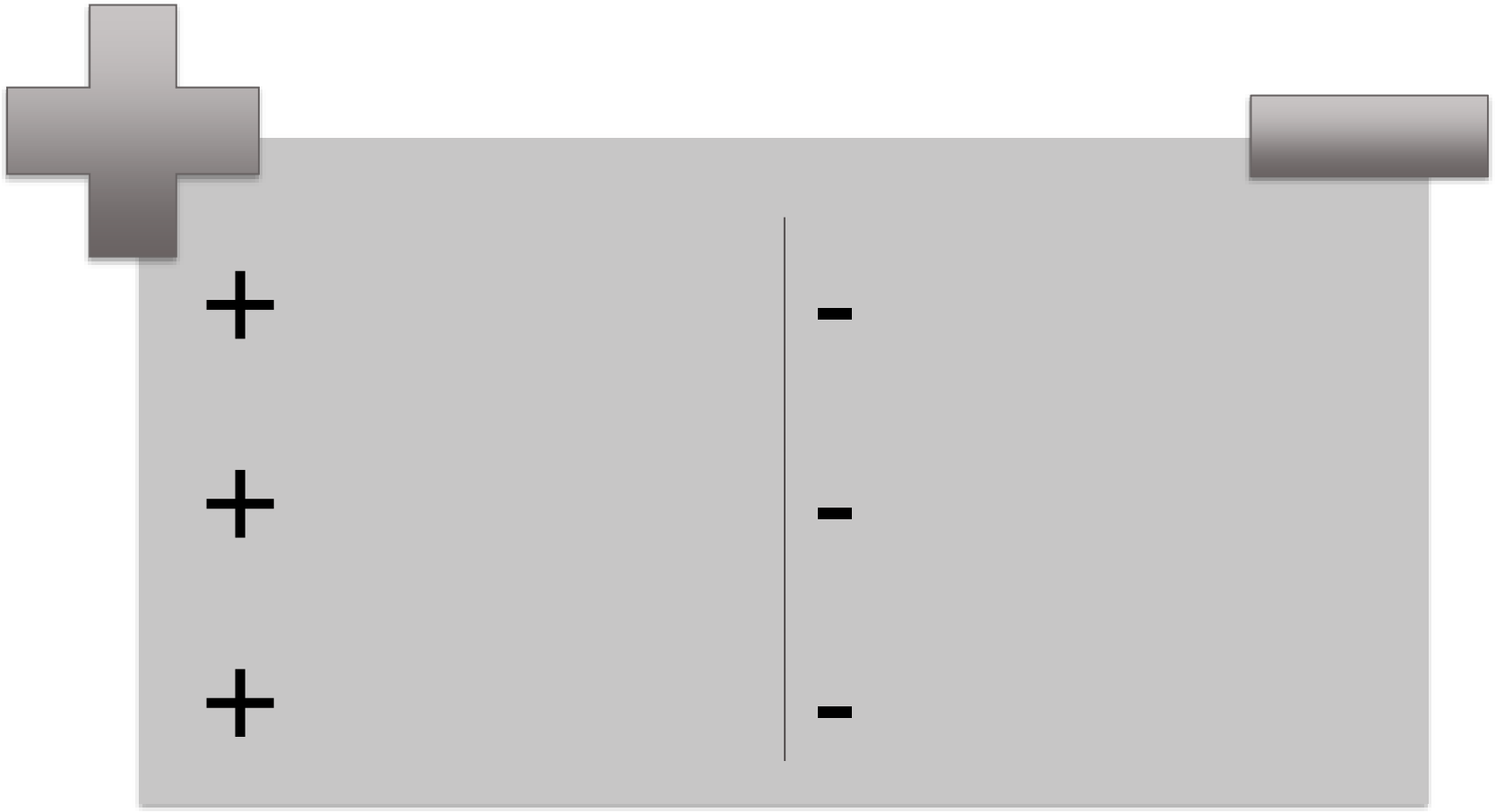
The Spiral Model



When should you use Spiral methodology

- When costs and risk evaluation is important
- For medium to high-risk projects
- Long-term project commitment unwise because of potential changes to economic priorities
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)

TASK 3 Spiral model



Advantages and Disadvantages Spiral Model (SDM)

Advantages	Disadvantages
● Estimates (i.e. budget, schedule, etc.) become more realistic as work progressed because important issues are discovered earlier. ● Early involvement of developers. ● Manages risks and develops the system into phases.	● High cost and time to reach the final product. ● Needs special skills to evaluate the risks and assumptions. ● Highly customized limiting re-usability

Spiral model

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.
- Software is produced early in the software life cycle

Individual work

Fill the table

SDLC model name	Description	Advantages (at least 2)	Disadvantages (at least 2)
Waterfall model			
Spiral model			
Agile model			

Successful criteria

- explain the differences and advantages of development models;

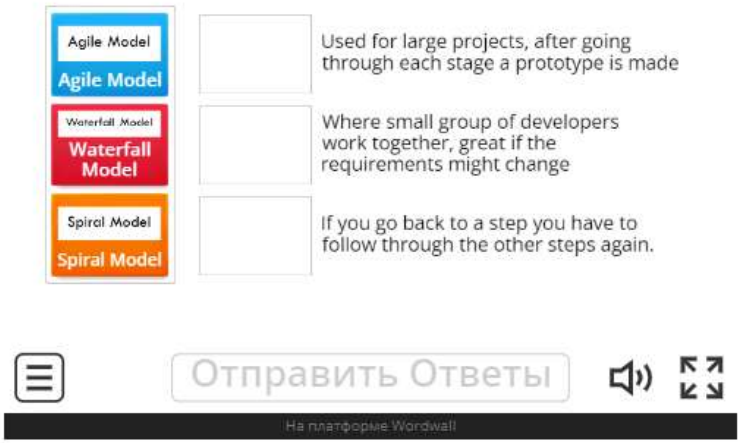
11.2.1.2 analyse the advantages and disadvantages of cyclical, waterfall and spiral models

SLC model	Advantages	Disadvantages
Waterfall model	• The deliverables can be shown to clients to inform them of the progress on the project. • A sense of discipline is maintained throughout due to the deadlines for each stage. • Requirements must be considered before work is begun.	• Takes a long time to deliver a project using this approach. • Loss of flexibility - you can't traverse the stages as you see fit, there is a rigid order to follow. • Cannot change the project down the line even when new innovations come to market. • Any mistakes/overlooked issues will mean you need to completely restart the project. • Requirements are almost impossible to grasp before beginning to develop.
Spiral model	• High amount of risk analysis • Good for large and mission-critical projects. • Software is produced early in the software life cycle. • Users can be closely tied to all lifecycle steps • Early and frequent feedback from users • Cumulative costs assessed frequently • Focuses attention on early error elimination.	• Can be a costly model to use. • Project's success is highly dependent on the risk analysis phase. • Doesn't work well for smaller projects. • Time spent for evaluating risks too large for small or low-risk projects • The model is complex • The design does not have to be perfect • Risk assessment expertise is required • Spiral may continue indefinitely
Agile model	• The agile approach believes you can't know the requirements at the start of the project. • The project is very adaptable, since the plan can change.	• Paired programming is not liked by many. • Customers who don't understand this approach will not like the lack of documentation. • People work in sprints, meaning code is usually spaghetti code.

Activity

- https://www.bzfar.org/publ/sdlc/types_of_sdlc/types_of_system_development_life_cycle/44-1-0-54

Ex. 1



Ex. 2