When developing software you need to focus on various things. Writing code is not your only task though it is still vital. Basically, you should answer the following questions: what you're going to do, how, using which methods, and why. To take it all into account and effectively organize the workflow, you need to choose a suitable software development model.

**Software development models**

To navigate through this complicated process you will require a more complex thing than a simple to-do list. Thus, you will definitely be in need of a development model, which describes in detail not only your steps in the right order but also the methods you use and the whole structure of a future product. There exist a lot of such models. In general, they are called SDLC, i.e. **Software development life cycle**. Below, we will discuss the most popular models, look into their essence, and compare their core features.

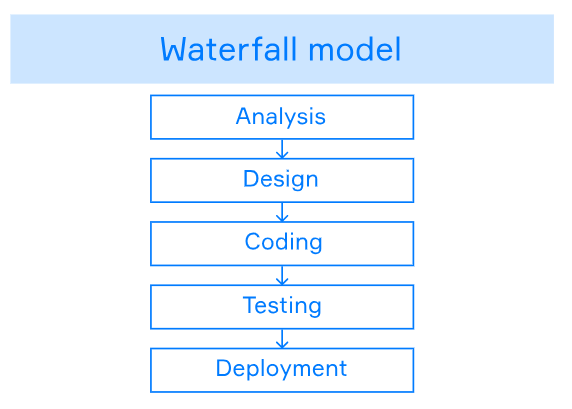
All SDLC models can be divided into several groups depending on the workflow organization, type of relationships between the teammates, and their communication with the customers.

The workflow may be linear or iterative, i.e. you either work on your product step by step or you can get back to the previous stages and change them. In terms of relationships and communication, you can either take the task from the customer and then simply show what you've got or you can consult with the customer at every stage.

Now let's observe the specific models and consider their pros, cons, and application.

**Waterfall**

The process in this model moves in a linear cascade mode through all the development stages: analysis, design, coding, testing, deployment. The waterfall model means there's no way back, so you cannot make any changes. Moreover, you cannot start the next step till the previous one is completed.



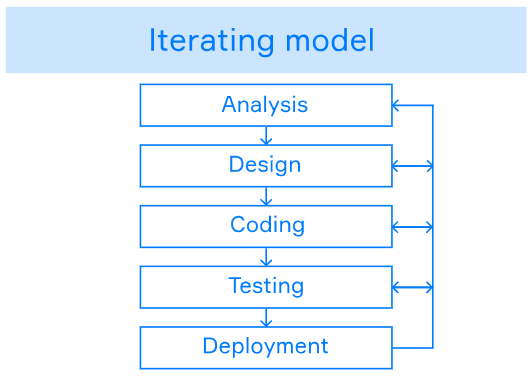
The final result will be seen and checked only at the very end of the development. It means that testing is often rushed, and errors may be costly to fix.

However, this model has some advantages. The waterfall allows to predict project costs and to introduce precise deadlines. This model suits for small projects with short deadlines and specific tasks and for those in which the tasks are to be completed no matter how much money it will take.

The waterfall model assumes that the client receives the finished product as soon as all functionality is ready, and not when all the individual features are done. This can be a problem if the client doesn't like something and wants to make some changes. This problem is solved by the iterative model, where each feature can be demonstrated right after it is finished.

**Iterative model**

In reality, errors always arise during development, and it's better to correct them as soon as possible before everything crashes. Iterative model allows to check problems and thus fix the mistakes in time. Besides, you may need to make some changes if your customer decides to amend the task. Also, the external environment and market conditions may change, and the original version of the product might be no longer relevant and useful.



Among the advantages here we can single out a higher probability of getting what the customer wants. And in general, such a development is very flexible, since changes can be made at any stage. However, it is unknown how long the editing will take and how much money will have to be allocated for it. These can be considered the disadvantages of this model.

This SDLC model typically entails some customer involvement because it's possible that small changes will be required during the development process. But human interaction is also a complex process. This means that it must be described in detail, the rules for this communication must be established in order to increase productivity. For this, agile group models were invented.

**Agile group**



So far, we have noted that it is important to take errors into account, consider the costs, and act according to an algorithm. But we forgot the most important thing: the people who do all this. Agile was invented to fix this. Agile is a set of principles and rules that facilitate the development process and take into account the human factor. All principles are written in the Agile manifesto. We will analyze them in more detail in a separate topic. For now, it is important to know that the developers are more important than the development, and if they cannot agree on what they do and how they do it, there will be no good result. It is also important to follow the wishes and mood of the customer. After all, if they don't like it, then again, the result won't be satisfactory.

So, Agile means working in close collaboration both within the team and with clients. At the end of each iteration, stakeholders review development progress and re-prioritize tasks for future iteration to improve return on investment and align with user needs and business goals. This model is typically used in large projects where user needs are constantly changing.

**Conclusion**

So, we've briefly discussed why we need development models, what they are, their advantages and disadvantages. Now we also understand which models are useful in which specific cases. Further on we will deepen this knowledge with specific examples.

**SDLC models groups**

All SDLC models can be divided into several groups. What factors does this division depend on? Choose several correct answers.

The correct answers are:

* ✅ **interactions between teammates**
* ✅ **communication with customers**
* ✅ **workflow organization**

Select all the development stages that should be considered when working on a project.

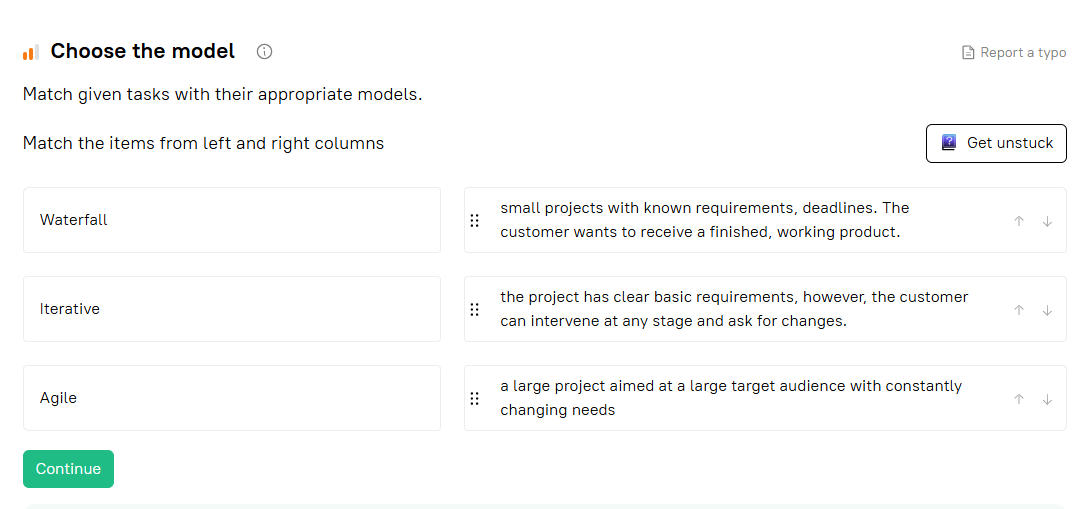
analysis

design

testing

deployment

coding



Where are all the principles and rules of Agile model written?

Agile Manifesto

What two types of workflow have we discussed in this topic?

Hint

Waterfall is linear model.

Linear

Iterative

**What is SDLC?**

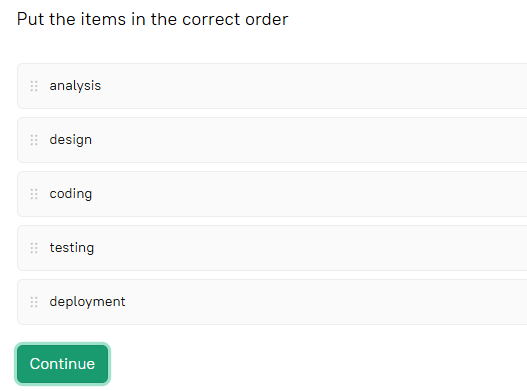
What does SDLC stand for?

Select one option from the list

software development life cycle

**Waterfall actions order**

What is the order of actions we perform when using the Waterfall model?



**Waterfall advantages**

What are the two main advantages of the waterfall model? It allows us to

predict project costs

introduce precise deadlines

**Agile factor**

How is Agile fundamentally different from waterfall and iterative models? What additional factor does it take into account? Choose one right answer.

human factor

**Which model to choose?**

A group of developers received an order to make a website. Their client turned out to be very moody. They immediately realized that it would take a very long time to do the task and they would have to constantly make changes during the development.

Which model should the developers choose from those discussed in the topic?

Fill in the gaps with the relevant elements

The developers should choose the iterative model.