14. A cég, ahol dolgozik, prototípus alapú agilis módszertannal dolgozik. Minden hét végén elkészül a szoftver egy újabb verziója. Magyarázza el az ügyfélnek, hogy mi a módszer lényege, és mi ebben az ő feladata!

– Minden héten le kell tesztelnie a szoftvert.

– Folyamatosan kapcsolatban kell lenniük, ami sok idejét igénybe veheti.

– A fejlesztés végén az ügyfél pontosan olyan programot kap, amit szeretne.

A tételhez szótár használható.
Kulcsszavak, fogalmak:
– Szoftverfejlesztés lépései
– Kommunikáció fontossága

Source:
http://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_szoftverteszteles/ch02s03.html
https://en.wikipedia.org/wiki/Agile_software_development

- Can you explain what my role is in agile software development?
- This development style is based on the prototype model. That means that before the final handover, several prototypes will be delivered to find out any misunderstandings as soon as possible. And you can see what you can expect from the system. Because your expectations can change over time, ~~and~~ we can refine the prototypes. You can give us feedback about the prototype, and we will change them. In this approach, the delivered system is also a prototype.

The modern methodologies are mostly prototype-based. Iterative methods usually link a prototype to every milestone.

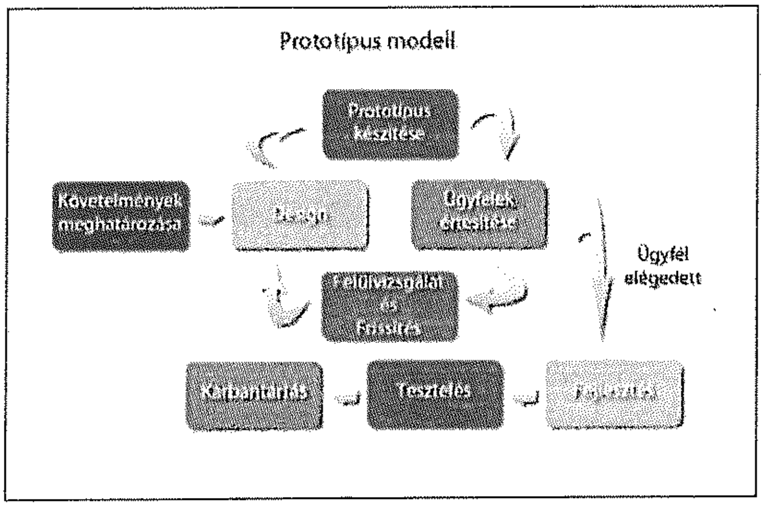
- How often do you deliver these new prototypes?
- We create a new prototype once a week.
- What are the steps of the development process?
- **The development of the initial prototype usually consists of the following steps:**

Step 1 **Define the basic requirements** such as input and output data. Usually, performance or security requirements are not addressed.

Step 2: **Develop an initial prototype**, we only develop user interfaces with a suitable case tool. The functions behind it, except for the opening of new windows.

Step 3: **Presentation:** This is a kind of user acceptance test. End-users examine the prototype and indicate what they think differently or what they would do.

Step 4. **Clarification of requirements:** Using your feedback, we develop the prototype further. With every step we are going to get closer and closer to your ideal programme.



Step 5 The cycle starts again.

- What does this mean for me?
- You have to test the newest prototype every week and give us feedback. This can be very time-consuming for you, but the program will be suited for your needs.
- So then why agile development so much better than just giving you specifications at the start?

Agile software development has the following **common features:**

- **Easier documentation.**
  The project is split into small parts and you only have to deal with a small piece of the product, which is produced in cycles of one to four weeks (also called frames or time boxes), and these cycles repeat. This makes it easier to manage it.
- **Increasing flexibility, decreasing risk.**
  Iterations are used. An iteration is like a traditional life cycle: it includes design, requirement analysis, encoding, and testing. Flexibility increases and risk is reduced as there is a takeover test at the end of the iteration, after which the customer can change his or her requirements.
- **Easier communication, improved collaboration.**
  Communication is easier because you only have to test a small part of the whole project every week. This makes it easier to communicate your needs to the programmers.
- **Inviting the customer to the development.**
  A designated person from the developers is constantly available for you to answer any questions that may arise on the fly as soon as possible.

  So your role is to stay in contact with us, and help us make the best program possible for your needs. ☺

Agile development methods are also called **adaptive**, is an important feature that **participants will try to adapt to the project as much as possible**. That is why it is important for developers to learn constantly

Agile software development values:

- ***Individuals and Interactions*** *over processes and tools*

Self-organization and motivation are important, as are interactions like co-location and pair programming. It is better to have *a good team of developers who communicate and collaborate well*, rather than a team of experts each operating in isolation. Communication is a fundamental concept.

- ***Working Software*** *over comprehensive documentation*

Working software is more useful and welcome than just presenting light, rather than heavy documents that take a lot of effort and quickly become outdated.

- ***Customer Collaboration*** *over contract negotiation*

Requirements cannot be fully collected at the beginning of the software development cycle, so *it is better to directly involve the paying customer and their end-users*, or a proxy for them so that detailed requirements can be progressively elaborated and adapted based on feedback.

- ***Responding to Change*** *over following a plan*

Agile software development methods are focused on quick responses to change and continuous development.