The value of an iterative and incremental Approach

One of the greatest benefits of using Scrum is that it prescribes an iterative and incremental approach to software development. This is by far the most effective and efficient approach for creating software in today's world and in the next sections, we'll explain exactly why that is the case. Let's begin by remembering how we used to develop software…

The waterfall legacy

When I first started programming, we used to build our systems in distinct, single stages: first analysis, then design, then coding, and so on. Each stage would cover everything we would need to consider in order to deliver the whole system, down to the finest detail. Once the stage was complete, we would move on to the next stage of the development lifecycle and never re-visit the previous, completed stage. This is now known as the **waterfall approach** because each stage was like a distinct level of a waterfall, one following the other in sequential, non-repeatable fashion, as illustrated in the following figure:

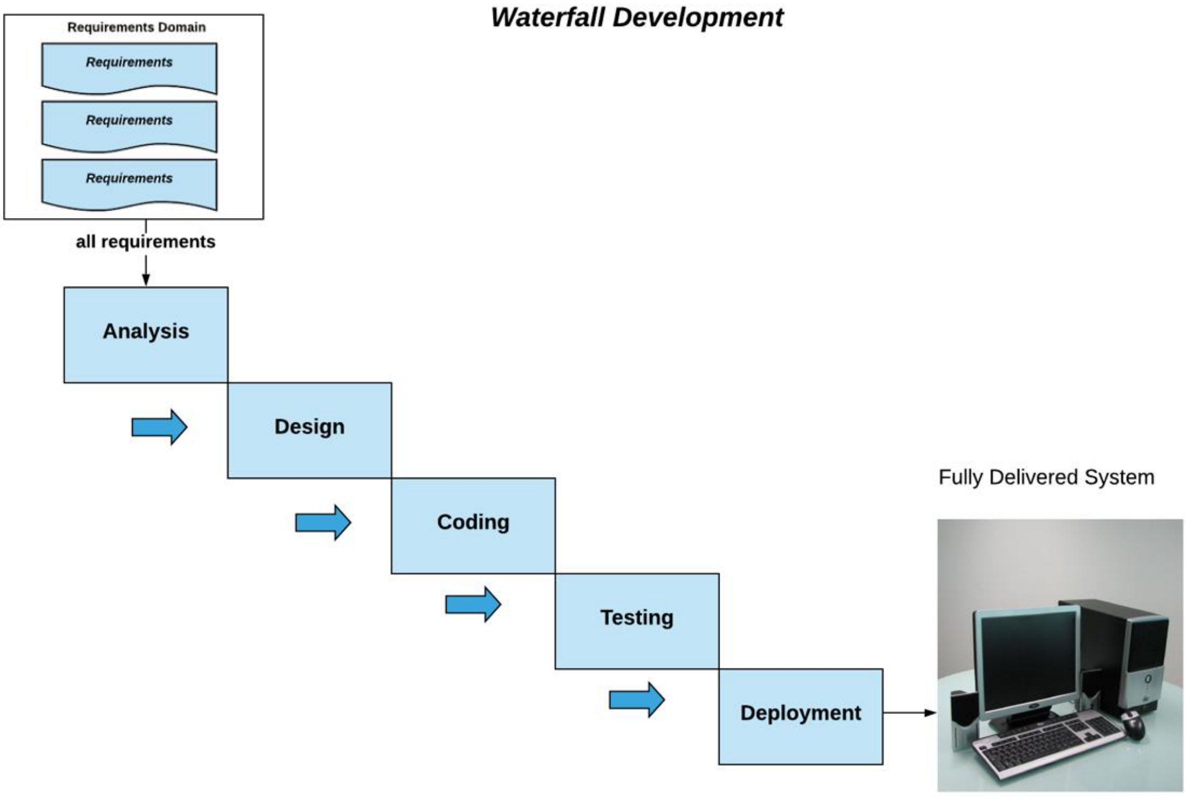

Figure 1.2 – Waterfall development

As we soon came to discover, there were some serious drawbacks to this approach:

• First, it took a long time to actually deliver software to our users. Since we had to consider every possible requirement, and design and document every possible functionality before we could start coding, it would take months or often years to progress from system inception to system deployment. By that time, a competitor would have beaten us to the punch by delivering their system first or the business need for our system would have simply passed, overtaken by circumstances and changes in the market.

- Secondly, since we were moving sequentially from stage to stage, any design flaws or false assumptions that were discovered after deployment could not be fixed without a major re-haul of our system. This took a lot of time and effort.

- Finally, if requirements were changed by the customer once we were past the design stage, we would have to start pretty much from scratch again.

In short, the waterfall approach was inflexible, risky, and time-consuming. It worked well for projects with rigid, unchanging requirements that were not affected by market conditions and weren't time-critical. However, as software applications started to become more prevalent in our lives and the market expanded and diversified, such projects started becoming rarer. Gone were the days in which consumers were happy to sit and wait for the next version of their spreadsheet application to come out from one of the two companies that produced them, or to wait for their email provider to fix a bug in their email system due to there being no real alternative.

Today, customers have plenty of choices and they value the speedy delivery of working software over dependence on monopolistic software providers. For software providers nowadays, time-to-market is an essential factor in their strategy and waterfall development is just too risky and rigid to follow. Luckily, the people who came up with Agile methodologies saw this at an early stage and almost all of the Agile methodologies that were developed, especially Scrum, follow what is known as an **iterative and incremental development approach**. Let's find out what that means.