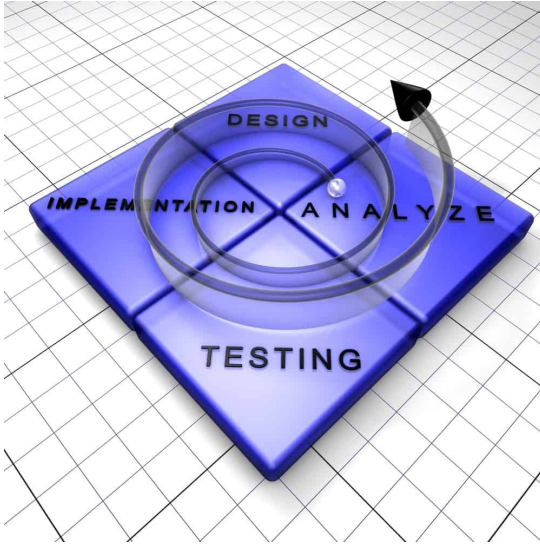


In the last two months I was testing software for [Enrise](#) via Polteq as a part of their development team The Impediments. Testing for them has been nice and instructive. The people at Enrise did not stop asking questions. Most of the questions I answered immediately, but some took a while to find the right answer. The question that lingered the most was: “How do we sell software testing to our customers?” I have to say that I’m very glad that Enrise asks this question! It shows me that Enrise sees the value and the importance of testing. The high quality software that Enrise delivers, can only be delivered when there is time set aside for software testing.



So how do you sell software testing?

- Sell a project (testing is part of the project).
- Only provide guarantee for production incidents when the customer pays for testing.
- Make the customer aware why you value testing.

Basically, you do not sell testing, you sell quality!

Software development is practiced in teams. Testing is a set of tasks that need to be executed in the teams, so there is implicit time for testing. You need to see that testing is essential to quality software and the time needed to test has to be allocated. The software may appear more expensive to your customer, but this will pay off! In the long-term, you will see less disturbances of production when the software is professionally tested. Investing in quality up front is worth it.

You’re probably thinking: “How will testing add to the quality?” Well, in many ways... I’ll give you three:

1. If you plan to test, you will need testable requirements. Ask questions on user stories

until you have clear what your customer actually wants to have. Developers usually make more assumptions and think about the solution, where testers think about value for the customer. More questions in the beginning will result in an easier development process and a smaller chance on defects.

2. A tester tests software instead of checking if the software works. A customer will use the software to see if it works, where the tester tries to find the edge cases and out of the box situations. A negative age is an unlikely situation, but a typo in a birthdate is quite plausible... The tester thinks of these situations, so when these typos arise in production the situation has already been covered.
3. Testing is about mitigating risks. This implies that the risks need to be identified. Testing will create a better view on risks. So even risks that are not mitigated, can be communicated to the customer. Then it's up to the customer to decide what to do: Invest more to mitigate these risks, or live with the risks?

In short: if you want quality products, you really need to test!

Jeroen Mengerink