The Bermuda Triangle of Test Automation

There is a great chance you have heard once in your lifetime about The Bermuda Triangle. Some people believe it is a conspiracy theory while others insist that it is real. In short: The Bermuda Triangle is supposed to be situated in a section of the North Atlantic Ocean and is seen as the reason for the disappearance of dozens planes and ships. I won't discuss if it is real or not, but the clue for me is: "there are certain disastrous situations that look like they happened out of nothing". Which makes me think about other situations where people say: "How did this happen?". One of them is: how has the Test Pyramid of Mike Cohn become an action plan for some people?

The test pyramid principle

Mike Cohn's concept describes that Test Automation should contain many more low-level unit tests than high level end-to-end tests running through a GUI. It also mentions that when you see an ice-cream cone instead of a pyramid, this is a sign that the implementation of test automation is not done the right way.

I totally understand the idea behind this concept and I think it is a useful tool to gain an overall view of the different types of automated tests and emphasize their role in the total picture. What concerns me is the fact that when I hear people talking about the test pyramid it is like they talk about an action plan. I'm sorry to say this, but in my opinion it is not.

I believe The Test Pyramid is a way to make an abstract concept more tangible. When used as an action plan for implementing you will probably end up with a checklist which you translate to tasks that must be performed. Assigning the unit test to the developers and the service and UI level tests to the testers. Then the time will come that you will finally see the pyramid and you can celebrate the achievement. What you probably don't realise is that this will have led to automatic behaviour.

Why do I believe that automatic behaviour should be avoided?

I believe if you want to make optimal use of the test pyramid you should never forget that it is a model. A model to use as a guide to start asking questions like:

- How can each type of automated tests solely contribute in debugging and maintaining the product?
- How should different types of test automation work together to improve and maintain product quality?
- How to set up automated tests which are making detecting defects faster and less

intensive?

In order to gain optimal benefit of the test pyramid it is important that you keep remembering that it is just a tool to assist in improving your test quality. Otherwise you will fool yourself with statistics showing you what you want to see: a pyramid which "indicates" that everything is fine. But think about it: what is the point of having a bunch of automated tests which are perfectly in ratio if you keep writing code that you don't need? And yes, at this point the test pyramid will indeed become your Bermuda Triangle.

Tuba Aydemir