The Agile Invasion

Adrian Kosmaczewski

2021-02-26

Something very big happened during the last 20 years: The Agile Invasion™®©.

A set of white men signed with their names a website called the Agile Manifesto, adding a picture of them in front of a whiteboard in 2002. Then lots of people, including me, added their "signature" on the same website, due to a severe case of Distributed Fear Of Missing Out™®© (DFOMO.)

One of them, Ward Cunningham, created both the first wiki, and then Extreme Programming in 1996. Another one of the signers, Kent Beck invented Test-Driven Development. Alistair Cockburn invented Crystal. Martin Fowler invented Refactoring and took Ward's idea, mixed it with a blog and turned into a bliki. Andrew Hunt brought pragmatism to the industry. Ken Schwaber and Jeff Sutherland invented Scrum.

Not bad for a collective resumé.

Thus a whole new industry was born, and since that moment all developer teams hold a sacrosanct daily standup meeting, even if they still work in noisy open spaces and still struggle to get their software to compile, let alone to run properly.

The Agile Manifesto is an interesting case of a document that has been thoroughly misinterpreted by the whole industry. That must have been, most probably, the reason of its staggering success: let us analyse it a bit.

> Individuals and interactions over processes and tools
> Working software over comprehensive documentation
> Customer collaboration over contract negotiation
> Responding to change over following a plan

If you look at the source code on the website of the manifesto, you will see the following:

```
<font size="+3">Individuals and interactions </font><font size="+2">over processes and tools
<font size="+3">Working software </font><font size="+2">over comprehensive documentation<br>
<font size="+3">Customer collaboration </font><font size="+2">over contract negotiation<br><
<font size="+3">Responding to change </font><font size="+2">over following a plan<br></font>
```

Try not to look at the `<font>` tags for too long, but just enough to see the relative sizes of the text; the `+2` and `+3` and where they are located. "That is, while there is value in the items on the right, we value the items on the left more," they said. And here is a scoop: based on the font size, we can argue that they value the items on the left 50% more than those on the right.

The problem was, what the average manager understood of this manifesto was completely different:

- There are no tools. You should bring your own device, by the way, because BYOD or forced home office during a pandemic.
- We do not write any more documentation ever. It is actually forbidden. Just provide software that does not crash on start. The rest nobody cares.
- We do not negotiate anything else anymore. It is the way of the CEO or the highway.
- There is no plan to follow. Shut up and code. Pizza, foosball and hackathon! Now you do not have to go home on the weekend anymore.

Oh, and please, be here at 9 AM, we have our standup meeting.

At some point I even became a certified Scrum Master but my certification expired and that link is a 404 now.

Of course, not everything is bad in Agile. There are at least three things I consider to be the most important to ever have come out of the Agile trend.

First, unit tests. The generalization of unit testing in pretty much every programming language on Earth is, in the humble opinion of the author of these words, the most important breakthrough in the history of software engineering. Unit testing brought many side effects to our industry, like new testable architectures, the identity of "legacy code" with untested code, the coupling between refactoring and testing, all of these side effects have made software better.

Second, the concept of "technical debt." Finally MBAs had a metric that could be used in a conversation with developers. Problem is, nobody can actually measure it objectively. But that is just a detail. There is, for the first time in the industry, a common concept that both developers and managers can actually use in the same phrase with the same semantics, and this is actually a big thing. The big task now is to actually provide actual meaning to it. But that is another problem.

Third, books like Bertrand Meyer's "Agile! The Good, The Hype And The Ugly" or Steve McConnell's "More Effective Agile" which are the definitive references in the field.