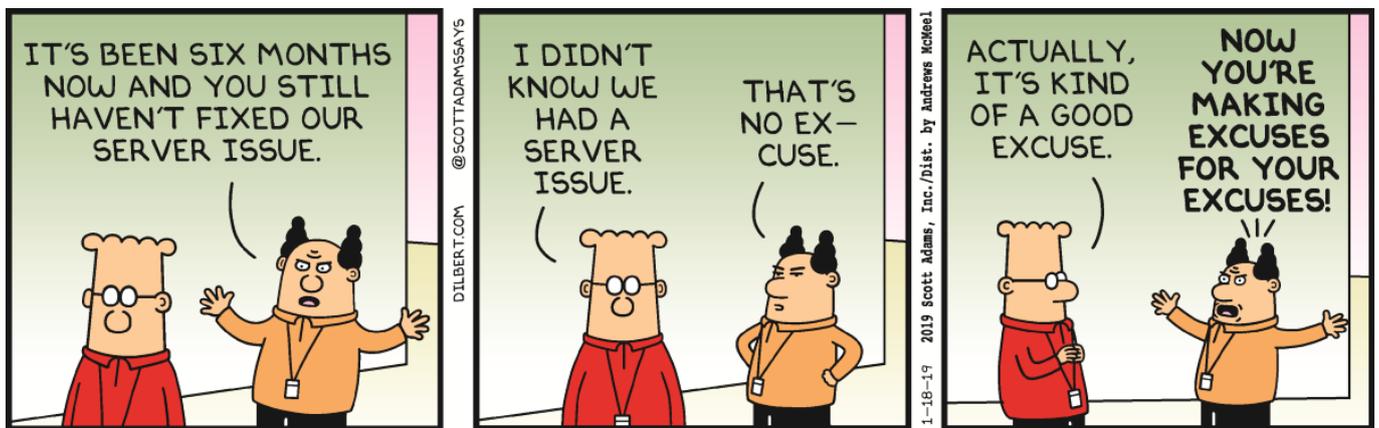


Dilbert still struggles working Agile

In my previous blog featuring Dilbert in 2016 - [Dilbert saves the Agile day](#) - we elaborated on a few of the challenges he encountered. Now 2,5 years later, new ones have arrived. That is why I have made a short selection of observations I had in the area of Agile team management. Let's continue our journey and see how DevOps and Agile are working out for our friends...

'One Dilbert still says more than a thousand words.'

1. Not my job



Two of the pillars in the Agile universe are autonomy and ownership of teams. With great power comes great responsibility. So, when you are part of a DevOps team and you have the responsibility for a production environment, you should know its health. Good teams will take the necessary steps to get the information needed and act upon it. However, this philosophy is not only applicable for teams. It is also true for Scrum Masters, Product Owners, Team Manager or any other role. For example:

- Scrum Masters should know when the team is feeling down or is experiencing difficulties in its delivery process.
- Product Owners should know, which features are being used and how the accumulated technical debt will impact future development.
- Team Managers should know the top three impediments their teams are facing and where they need help.

The fact that sometimes you simply don't know, is not bad in itself. As long as you perceive this as an indication to get informed. "I did not know" is not an excuse, it is your job to know.

2. Who actually need good engineers?



Although the percentages are perhaps steep, bottom line is that I have seen this a few times in practice. It requires strong leadership and vision on skill and people development to keep your employees in shape. This responsibility works both ways. In my opinion, employees should be equally accountable for their personal development. As part of a good feedback cycle from peers and managers, people should develop a fair insight in their own capabilities. Companies and departments are in trouble when management attention is lacking or is too weak. In that case you are at risk of employing workers with deteriorating skill sets that have less and less value. Developing a continuous learning culture is vital in software companies who are rapidly moving from Scrum to DevOps and beyond. I would like to see teams, Scrum Masters and managers team up for this challenge!

3. That's just not Agile



At times you find yourself being part of a discussion that doesn't seem to make sense while the people surrounding you seem to think otherwise. A topic I find particularly intriguing is the popular phrase "That is not Agile". It is like you can silence all questions or discussions with that one line. I am a strong believer in the Agile way of working. However, at times you need to understand that change - especially cultural change - does not happen overnight but requires small steps. It is not a terrible thing if some stuff does not feel a 100% Agile, yet. For example, it can be very fruitful to have that extra check or coordination in place when lots of teams are involved in releasing a new software version. I favor the idea of experimenting and enlarging

team's autonomy. Especially when team basics are in place and embedded in the right organizational conditions (that last part is not meant as rigid as it may sound). Remember, being Agile is not a goal in itself, it remains a means to an end.

4. Please don't use that language here



“My manager has absolutely no idea what I am doing all day” it is not the first time that I have heard that sentence. It is part of a recurring discussion I have with Scrum Masters, engineers and my fellow managers. How much knowledge, insight and hands-on experience does a manager need in order to lead a group of well-trained engineers? I don't dare to think that there is only one truth, as always it probably depends. Without speaking too much to the choir, I can only elaborate on my own experiences as a manager and Scrum Master. It has helped me a great deal that I have a technical background. On the other side I have seen some talented colleagues without any technical background, working outstanding without any problems. Technical background or not, teams have their own responsibility in transparent communication; they have to be able to explain what they are doing and why it is important. And not only in bits, bites and *flurm* but also in a comprehensive manner so Product Owners, customers, users, business colleagues and managers are well informed.

5. Agile strategy in place, check



As all companies seem to be going Agile it is normal that employees need information on the actual changes in their company. Whether you are introducing Scrum or trying to move to DevOps, people will want to know what the impact is going to be. What does it mean for my role, my team or my career path? The people - or team - in charge of the change should be able to explain the new expectations towards the teams and individuals. What are the responsibilities, tasks and accompanying authorizations? Even more interesting, how is the balancing act envisioned of team freedom & mandates on one hand and organizational governance on the other. The expressed desire to make teams more autonomous often contrasts with an underlying basic lack of trust. As a result, in practice teams are not being allowed to act autonomously. It all comes down to breaking free from previously existing structures and organizational boundaries. A major shift has to be made in the managerial aspects of leading teams and delegating decision making to teams.

Many companies are well under way in their Agile journey. At times it proves to be difficult and challenging. Thank God Dilbert is here to make us laugh along the way...

Copyright:

[1] DILBERT © 2019 Scott Adams. Used By permission of ANDREWS MCMEEL SYNDICATION. All rights reserved.

[2] DILBERT © 2018 Scott Adams. Used By permission of ANDREWS MCMEEL SYNDICATION. All rights reserved.