

**A tester learned how to use data in a powerful way, you won't believe what happened next...**

Linda.Hoff@qlik.com  
[www.linkedin.com/in/hofflinda/](http://www.linkedin.com/in/hofflinda/)



**Abstract:**

As testers, I bet you all felt the feeling of hopelessness when no one thinks your opinion is worth listening to. If that opinion is backed up with data, it is suddenly not debatable and if important enough, it will get prioritized. Data can be one of your most important tools but there are also situations where data can mislead us and trick us into making wrong decisions.

If you want to get a good result it is not enough using data, you need to learn how to understand data and how to communicate data as information.

This session will provide you with a glimpse of what data literacy is and how you can improve your skills how to turn data into the most powerful tool you ever had.

**Key takeaways:**

- Basic understanding what Data Literacy is
- Hands on tips and tricks for how to work with data

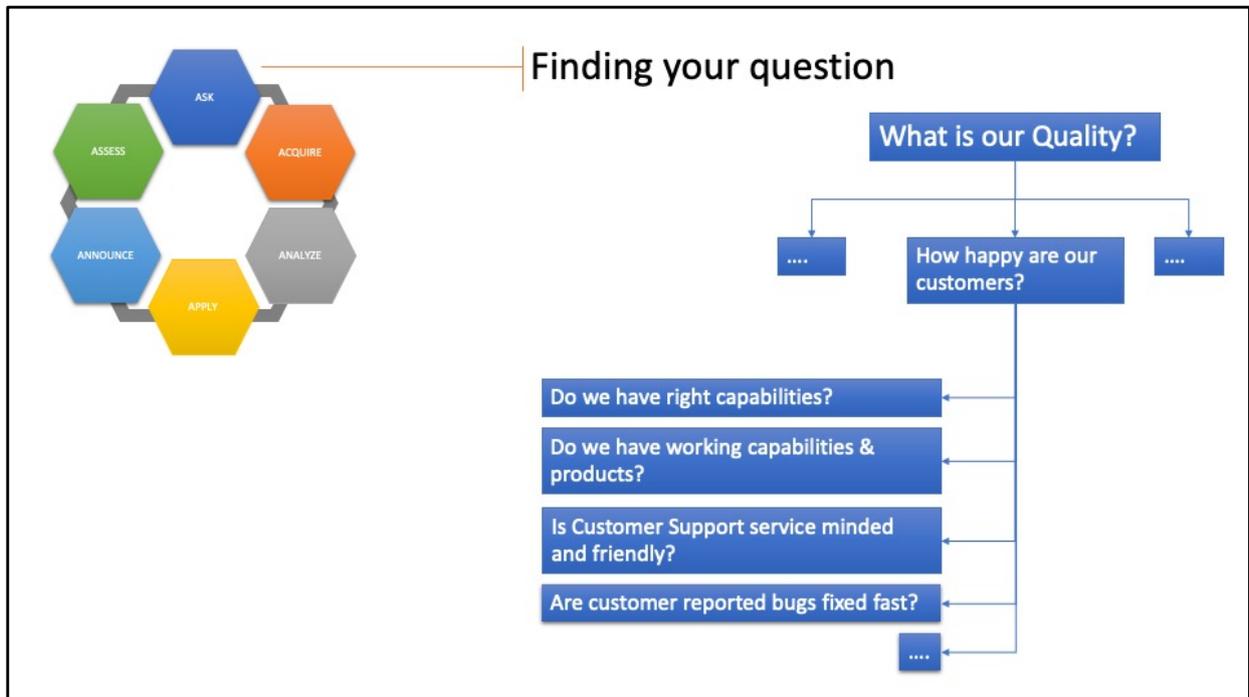
**Data literacy** is the ability to  
**read, understand, create,** and  
communicate **data as information**

Source: Wikipedia, 2021

Data can bring tremendous value to organizations. The best way of leverage your data is to develop your data literacy skills. That will give you a clearer picture, deeper understanding leading up to better decisions and predictions.

Data-informed decision making model



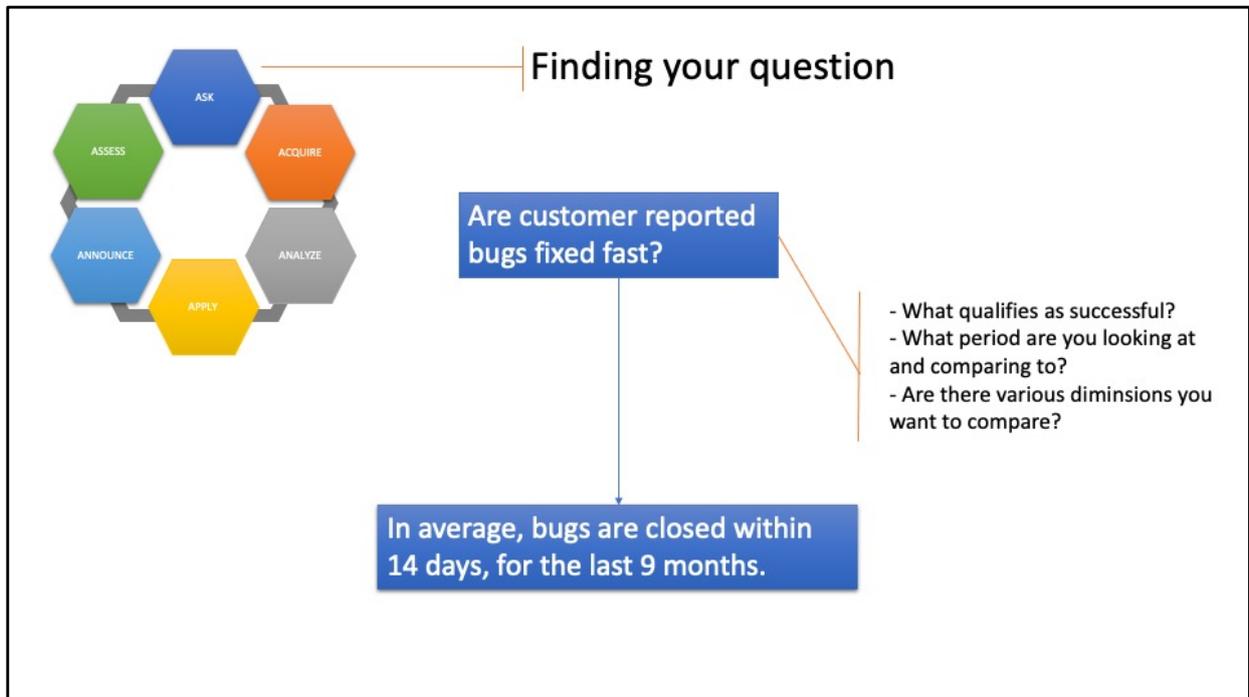


Find your business questions.

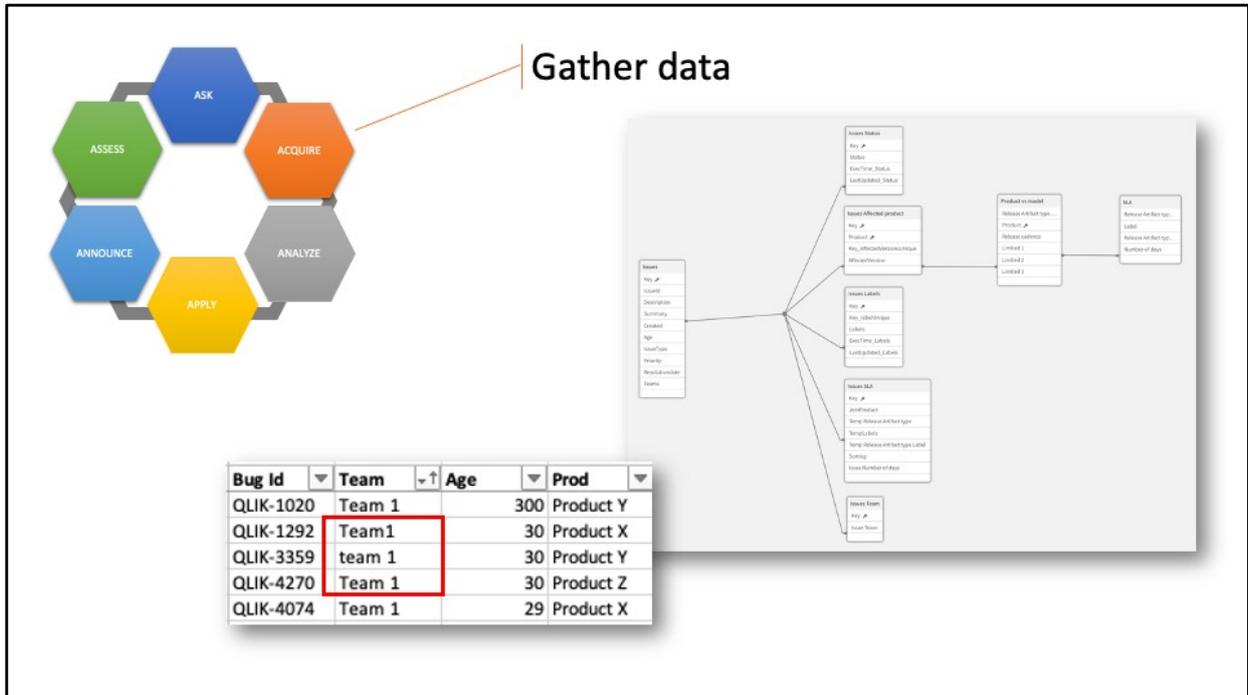
In order to be able to answer what our quality is, we need to specify what leads up to quality.

Usually you will end up with several levels of questions. Think outside the box and try to include everything. For example, can external competition affect your quality? Probably, since users are comparing with other similar products and perceive the quality less good if another product is much better.

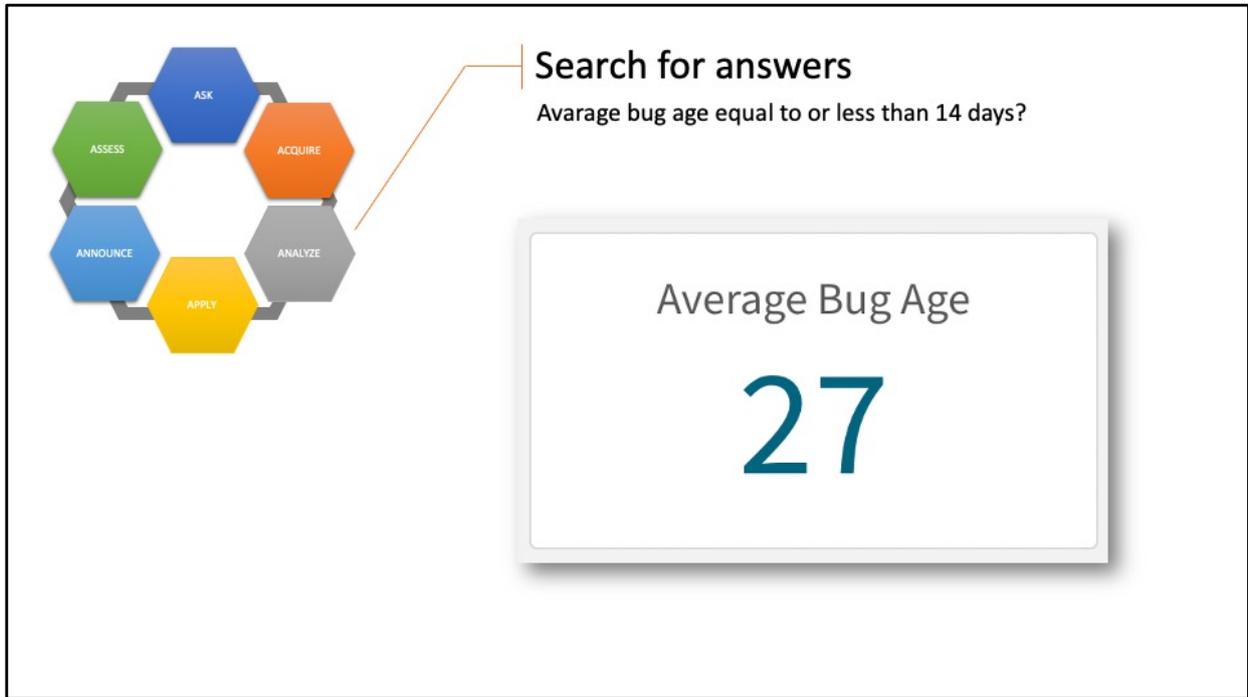
Make sure that all questions are really important for your company and related to overall business goals.



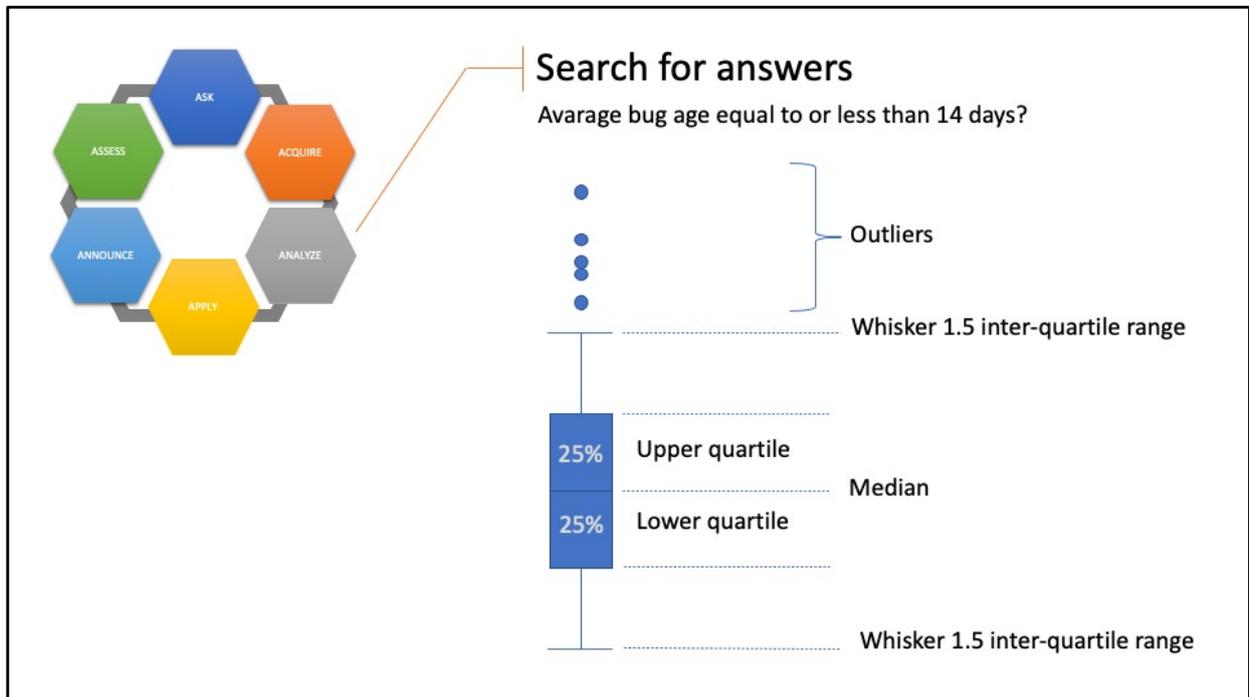
Transform your business questions into analytical question.  
How do we work proactively with fixing bugs fast? We could monitor bug age.



- Reliable source
- Tursted source
- All relevant data
- Correct/suitable format
- Create a good data model
- Clean and verify the data



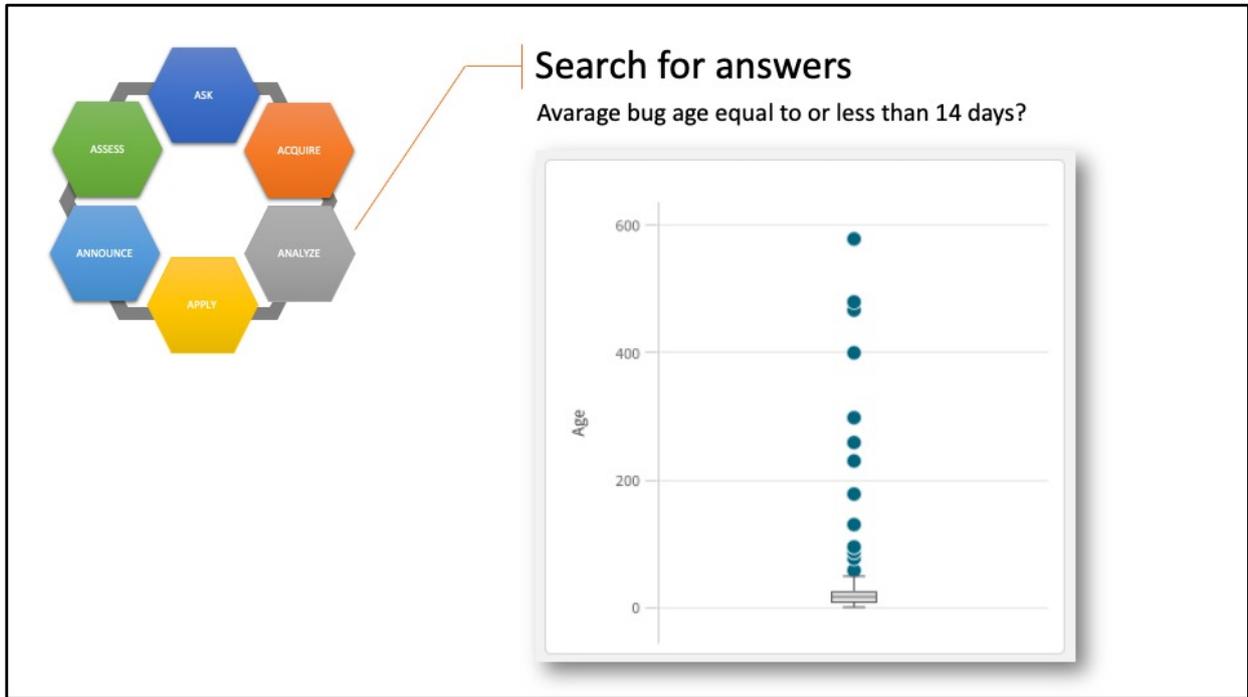
A single KPI for average could hide a lot of interesting insights.  
Do we have a few, very old bugs, that increase the average alot?



**Box and whisker plot** tell you more about your data.

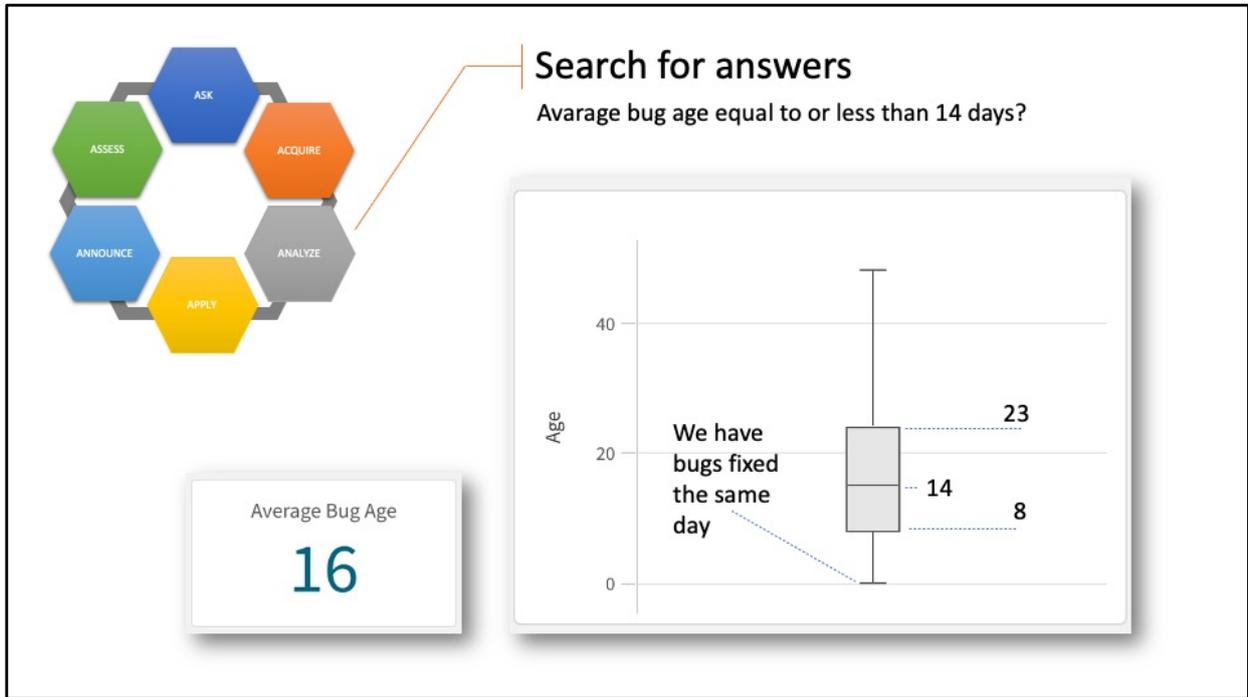
**Outliers:** Data points that are far from most of the data points. Probably “special cases” or “corner cases”.

**Interquartile range (IQR)** : is the distance between the upper and lower quartiles.

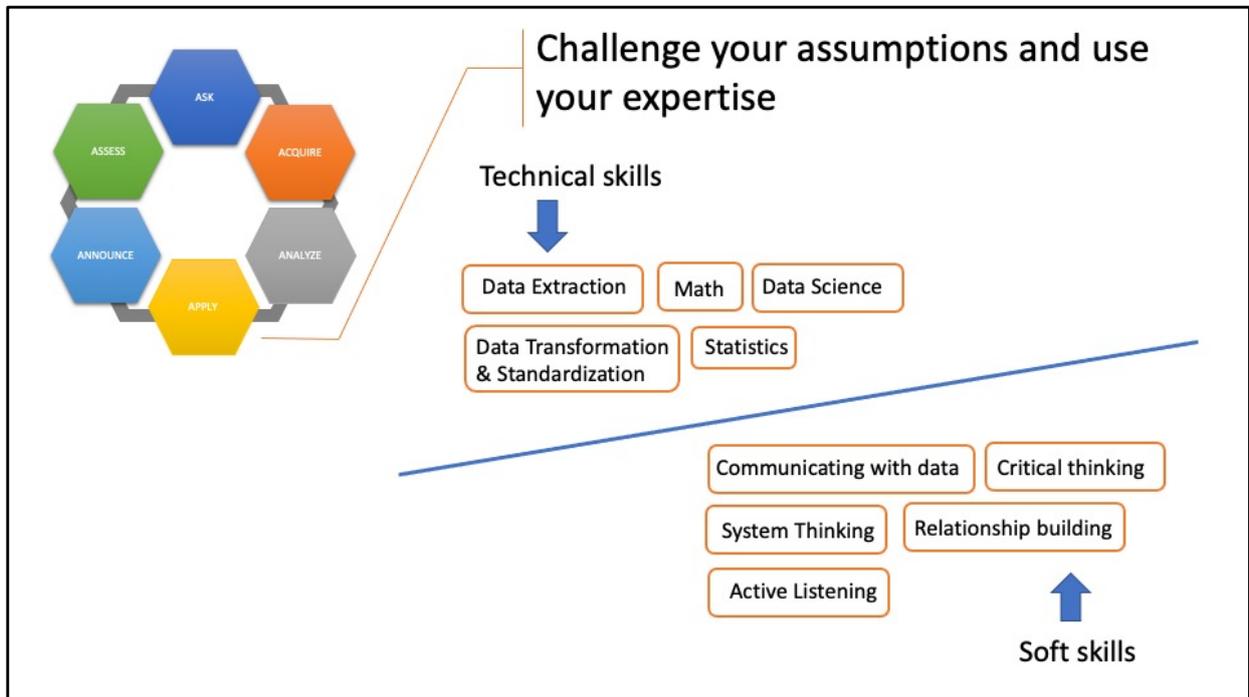


There is in total 370 bugs, 16 outliers (4%).

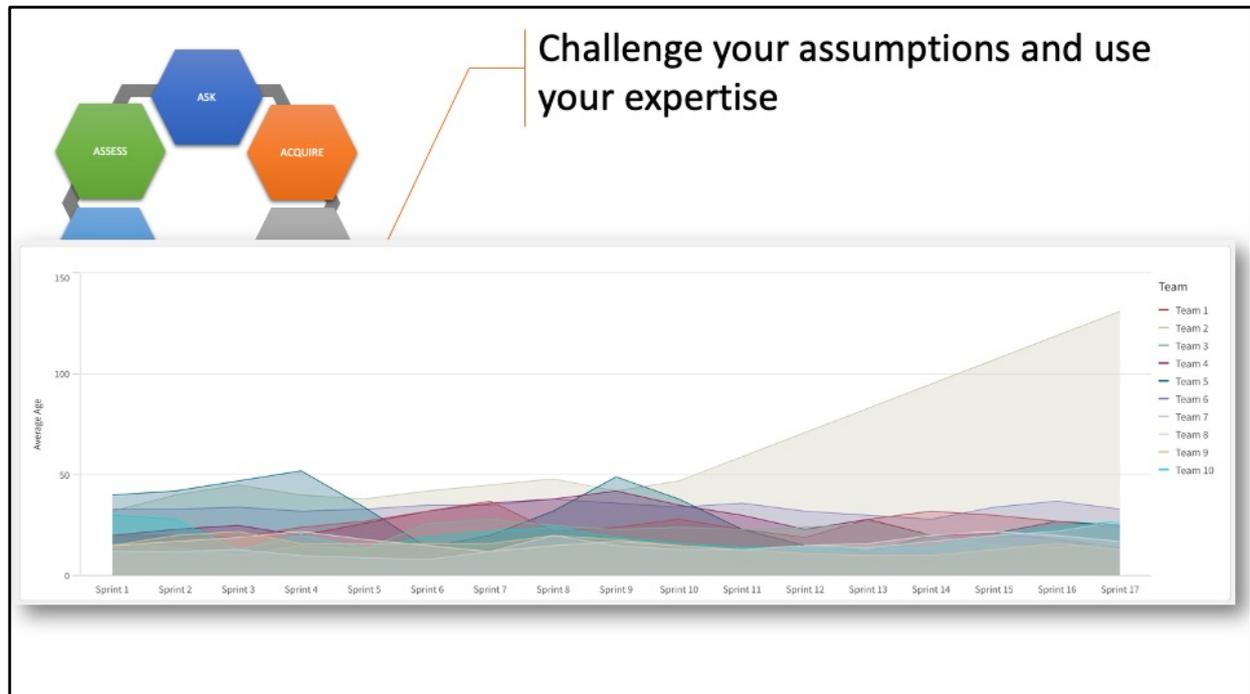
Let's explore what it looks like without outliers and take a separate **action** on outliers to see if they can be closed.



Excluding outliers gives us an average of 16 instead of 26



Technical skills like knowing how a Box and whisker plot works and when to use it is important but it's not enough. You need to add some soft skills as well.

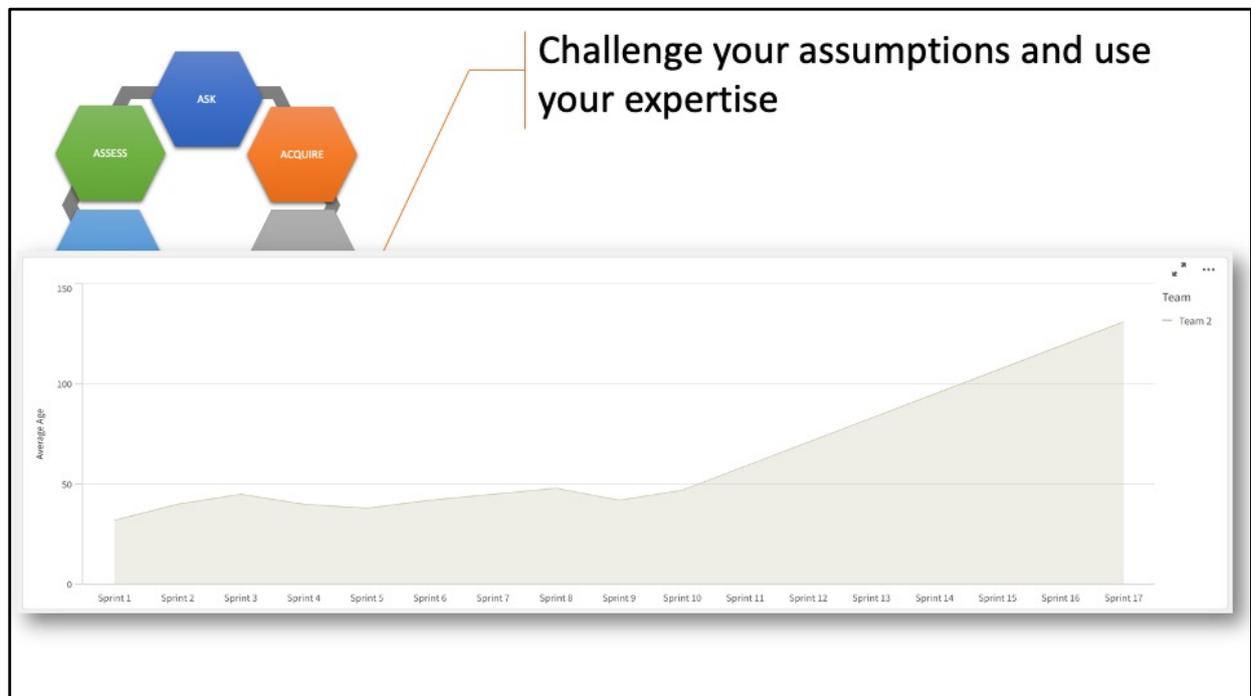


Our assumption is that all teams are same and work the same and can be compared. Let's challenge that assumption.

The graph shows average age sprint over sprint (2 weeks).

Team 2 have a suspicious curve! It turns out that that team doesn't exist anymore.

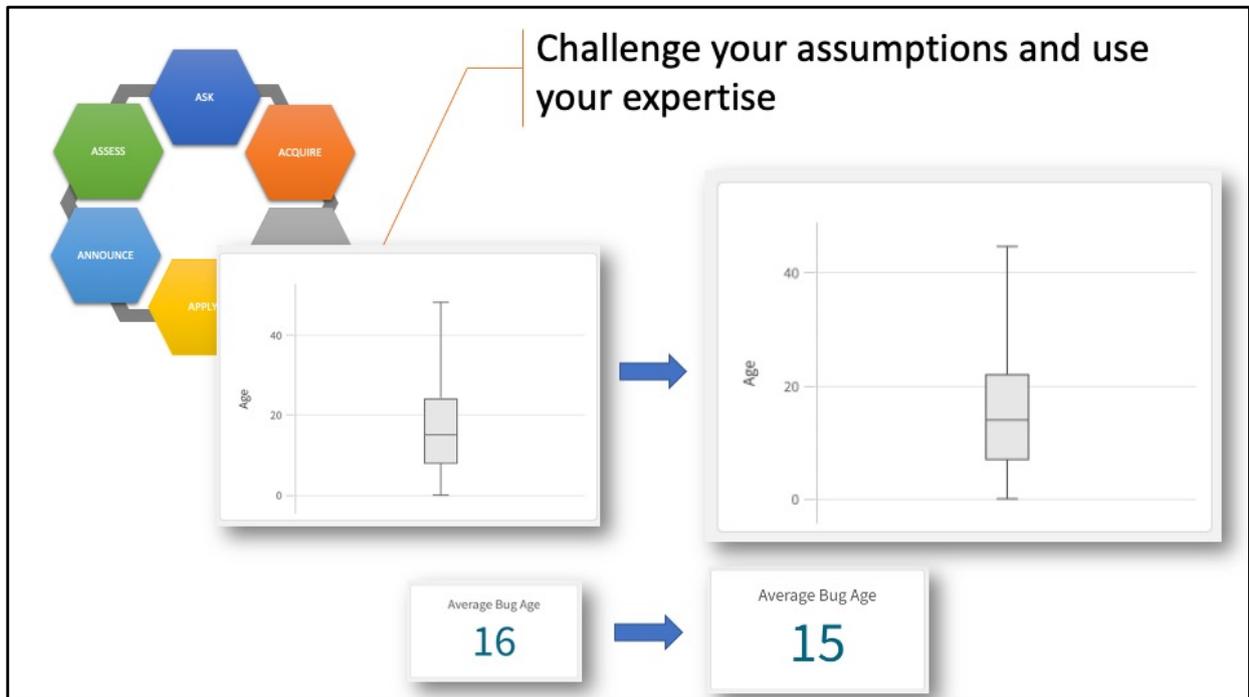
The bugs are abandoned and the functionality doesn't exist anymore. They should not be included in this analysis.



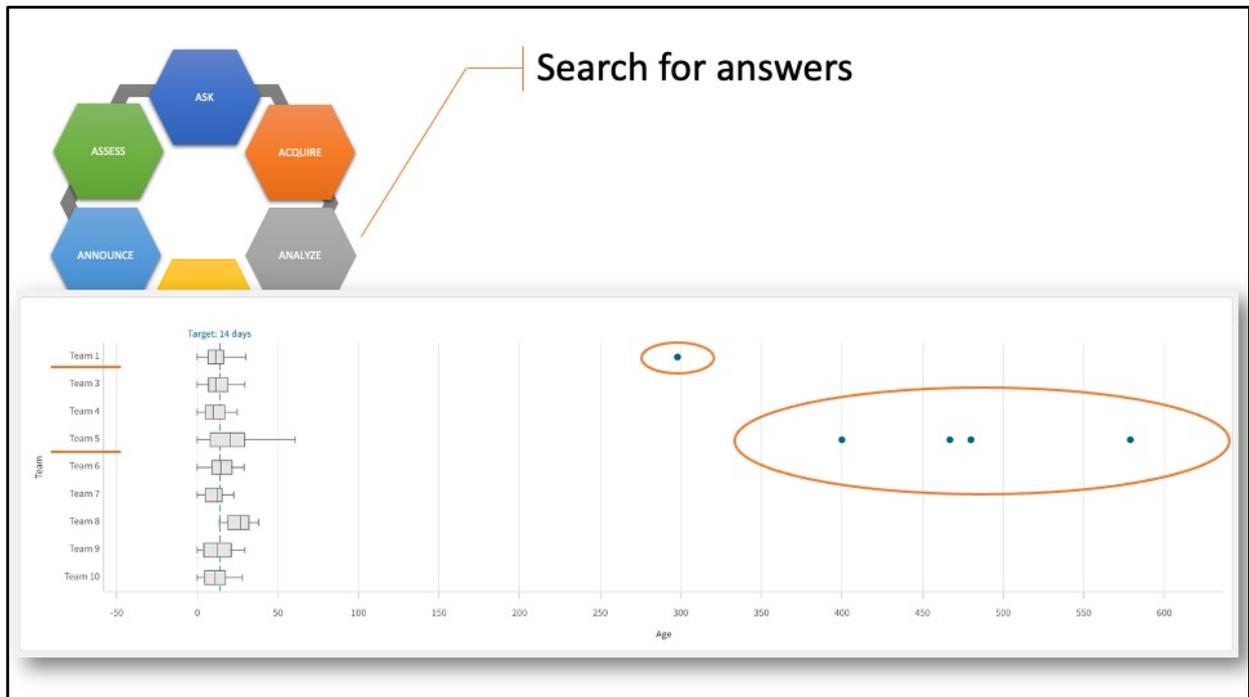
Team 2 have a suspicious curve! It turns out that that team doesn't exist anymore. The bugs are abandoned and the functionality doesn't exist anymore. They should not be included in this analysis.



Going back to average and box plots with and without outliers, after excluding Team 2.  
 Average bug age is > 14. What can we do to lower average?

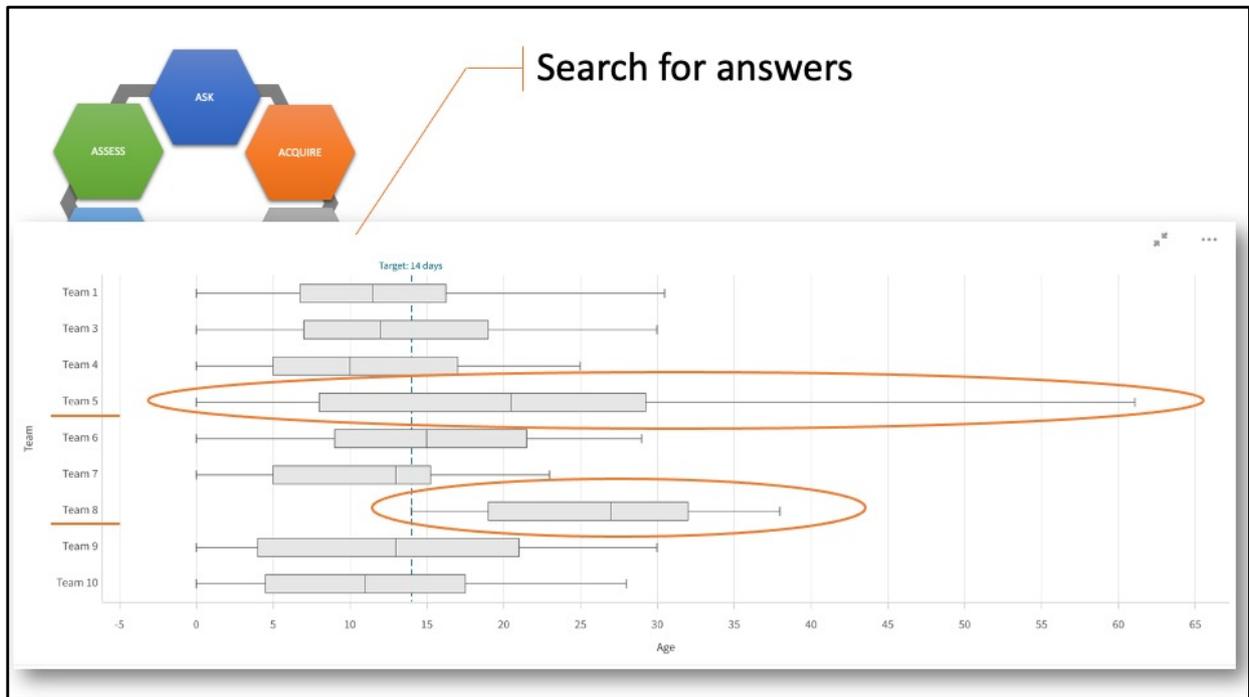


Going back to average and box plots with and without outliers, after excluding Team 2.  
 Average bug age is > 14. What can we do to lower average?



Let's do an analysis from Teams perspective. We could do it from product, or product area, or program language, bug quality or....

2 Teams have all outliers. **Action:** ask them to close them or come back with context. If they cannot be fixed, practice active listening for why and use your new knowledge in order to do better analyses next time.



By excluding outliers we can get a better picture of teams' "normal" situation. 2 Teams have an interesting distribution. We could continue to look deeper into the analysis of what is driving that but for the presentation we will stop here with identified **action** for teams to continue the analysis and do a retrospect of how they are working with bug fixes.



**Who is your audience:** Managers with limited data literacy skills. They need clear actions so it's easy to act. But they also need to understand why this should be prioritized.

**What is your message:**

Action 1: Team 1 + Team 5 to close their outliers

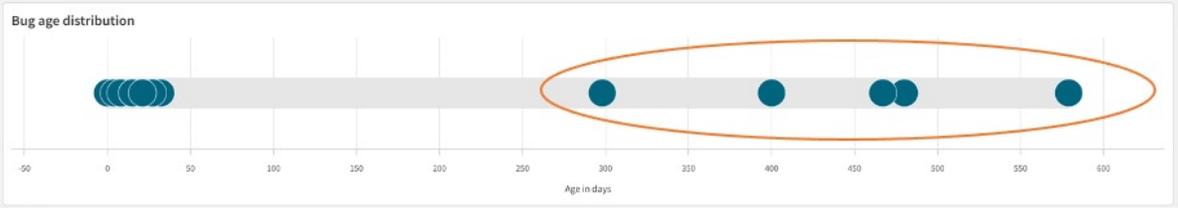
Action 2: Team 5 + Team 8 to retrospect their WoW/do their own analysis

**Craft your story:** What elements to use. How to visualize.

**Tell your story:** Story telling skills. For example, start with why.

# Decide & communicate

| Bug Id    | Q | Team   | Q | Age | Q |
|-----------|---|--------|---|-----|---|
| QLIK-1007 |   | Team 5 |   | 400 |   |
| QLIK-1015 |   | Team 5 |   | 579 |   |
| QLIK-1020 |   | Team 1 |   | 298 |   |
| QLIK-1044 |   | Team 5 |   | 480 |   |
| QLIK-1097 |   | Team 5 |   | 467 |   |



My first choice is to not use box plot. Too few people understand it. A simple distribution plot shows the outliers good enough for everyone to understand that they are a few “special cases”. I use a circle to highlight. I also choose to add a table with exactly what bugs and what teams. This will make it easy for managers to act.

Did the actions result in improvements?



- Action 1
- Action 2
- Action 3
- Action 4

Continuous monitoring and follow up analyses.

Remember, you cannot expect any difference if no one take actions. So make sure you follow up on decided actions to make sure they are completed.



What did we do?

1. We figured out a “real” question and how to measure it.
2. We acquired the correct data
3. We did a thorough analysis to find a good answer to our question
4. We challenged ourselves
5. We planned how to present and create our story
6. We announced clear actions in an inspiring way

But it is a loop/circle....

1. Ask – Trend – compare week over week: Are we moving in the right direction?
2. Ask – Bug fix vs feature developed: Should we focus less on feature development?
3. Ask – Bug fixes per team member: Should we hire more people?
4. Acquire – Number of days in different context, like Cloud or OnPrem
5. Analyze – Priority & severity
6. Analyze – Separate time for fix and release
7. Analyze – Any difference if we already knew about the bug when customer reported it?

8. Analyze – Any relationship with customers, where in the product, type of issue (performance, UI etc)
9. Apply – Why 14 days? Am I being anchored?
10. Apply – I assume all bugs are logged correctly and on time, is this true?
11. Apply – I assume teams are following ways of working and on time, is that true?
12. Announce
13. Assess

**A tester learned how to use data in a powerful way,  
you wont believe what happened next...**



**Thank you for listening!**  
linda.hoff@qlik.com