

TESTING ANARCHY

To manage or to lead software testing | By Stefaan Luckermans



Stefaan Luckermans

-The Thaster -



Anarchy

a state of disorder due to absence or nonrecognition of authority



Introduction

What came first? Who decides what? Why do we do this? When and where will it happen?

All reasonable questions, but if there is no authority or the authority is not recognised, these questions can become issues, because nobody can or dares to answer them.

The reason why I called this article 'Testing anarchy' lies in the definition of the word 'anarchy':

'a state of disorder due to absence or nonrecognition of authority'

In this definition we recognise the following indicators:

- A state of disorder
- Absence of authority
- Non-recognition of authority



This means in fact “Something is wrong because nobody acts in the right way, everyone acts in a different way, nobody is leading, everyone thinks they know best, etc.”

Anarchy can paralyse a company or make it dysfunctional. In any case, any form of anarchy creates burden and causes unbalance, 2 things we don't want in a good functioning company. In this article we will focus on a couple of situations in which this 'attitude' can cause a problem and we'll see what the right way is to react on these situations:

- ✦ When testing becomes a goal
- ✦ When improvement becomes a goal
- ✦ When a test lead manages
- ✦ When hierarchy rules

When testing becomes a goal

Unless your company is a 'testing company', a company aimed to deliver testing services, software testing can never be the goal of a company.

expressions I encountered a lot in my career are:

- That error shouldn't occur. Why wasn't it tested?
- We tested this, but we don't know why it now shows a bug.

This really becomes a problem if these bugs are reported by end-users and/or end-clients. If a certain number of defects are only discovered once a product is in a production environment, it can influence the shareholders of a company, who than can pressure the management to take measures.

Often these measures result in hasty conclusions in which the testing department becomes the scapegoat (even if they are the ones to be blamed, it isn't the way to go). Often the measures taken will make software testing part of the management goals.

By making the software testing department the centre of focus in resolving the problems encountered in production, has the opposite effect of what is desired.

The message that management is sending out to the rest of the company is dual:

- Hey people, the problems we encounter in production should have been detected (and resolved) by the testing guys and girls.
- By the way, blame them for future problems ...

You could say, this is not anarchy, because although it causes disorder, it doesn't prove a lack of authority or non-recognition of authority. But, only at first sight. Management is saying "it's the testing department" and everyone else believe it, meaning that management is doubting the "testing authority" of its testing department.

Now it is easy to say that the management is giving the wrong signal, but the problem remains as does the question: "How do we handle this in the right way".

Rule #1: Anything that causes testing anarchy will cause disorder and mistrust and therefore has to be avoided.

Rule #2: Quality is the result of a carefully constructed cultural environment. It has to be the fabric of the organisation, not part of the fabric (Phil Crosby).

Rule #2 says something about the problem encountered. It is a quality problem. People are causing/reporting the problems, the product is faulty, the processes in the company didn't trap the defects going to production, the project didn't meet its criteria, or the criteria in place were not sufficient. Perhaps the product was released because the budget was under pressure.

It's this what should be the basis of the company goals, not "testing failed and must be better".

So when stating goals for a company, department, project, or individual always do this in terms of the 5 Ps that determine quality:

1. People
2. Project
3. Product
4. Process
5. Price

When improvement becomes a goal

So, testing shouldn't be a goal, but we saw that something has to be done and that in fact we have to do something about the quality

in our company. We need to improve quality. That should be a goal in our company.

What is there to disagree? It's a good goal, but let us see how the management in our company can tackle this in the wrong way, than to see how to do it in the right way.

What often happens is that management will look inside the company and figure out that since the problem lies within the company, nobody within the company can be trusted with the improvement at hand.

They doubt the authority that is within the company to handle the disorder.

Rule#1 states that any decision that causes anarchy should be avoided, so that should already be a red light.

But, if the improvement is handled wisely, at a certain point people from within the company will be involved, no?

Often we see that management doesn't choose for a people-approach. They tend to choose for a framework, because that is what a company is, no? A balanced network of entities all working to achieve a common goal, just like a framework.

Rule #3: A framework is an essential supporting structure of a building, vehicle, or object (Google).

Important are 'balanced' and 'supporting' and these words imply that an improvement is NOT a goal. One could say even that these words imply that a framework is part of the strategy to reach the company's goals. Therefore ...

Rule #4: 'Improving' is never a company goal. It is always part of a strategy to achieve a company's goals.

For this reason the company management decides to manage the company goals which are related to quality with the support of a framework.

When a test lead manages

So management looks at their goals, matches them with a framework and ...

they do what they are supposed to do ... manage. They define a number of KPIs that match their goals and expect people to follow the guidelines in the framework and report the KPIs on a regular basis.

This way they cover rules #3, #4 and #2. But what about rule #1 'avoid anarchy'?

One of the most striking quotes I ever heard in my career is one made by a project manager that had to report his KPIs. In fact he had forgotten to make his report and because I reminded him it was the last day to send in his report he started making it.

5 minutes later he finished, stating: "It doesn't matter what I put as number, because I always have enough reasons on why to give this number. Everyone has his own way to interpret the KPI".

What he really said was: "there is no authority on these KPIs".

That's what happens if management manages ... but how can they do better?

Rule # 5: There are times when a leader must move out ahead of the flock, go off in a new direction, confident that he is leading his people the right way.

(Nelson Mandela)

A leader sets out the lines, the guides, the rules, the constraints and does this with the confidence that is expected from a leader. A leader knows that he does this to lead his company in the right direction.

A management that leads will make sure that its goals and KPIs cannot be misused nor misinterpreted. The author of the previous quote also gives the word 'leading' a new dimension, because he not only lead by example, but in the process inspired billions of people all over the world.

Rule #6: An average teacher tells. A good teacher explains. A super teacher demonstrates. A great teacher inspires (William Arthur Ward).

A management that leads, will teach its people the values of the company. It will inspire its people to greatness by giving them the tools and support to achieve the company's goals.

When hierarchy rules

Leading by example will also make people responsible and accountable for their actions. It recognises their 'authority'.

This means that all 6 rules are followed and yet reality teaches us that there still is anarchy, because people are hesitant to try out new things and therefore will resist any form of change. They will

do anything to stop the change. Question is whether they do this because they don't recognise the authority of their management, or their own authority.

Rule #7: Men are not prisoners of fate, but only prisoners of their own minds (Franklin D. Roosevelt).

During my career I came in situations where people were forced to change, where people were not asked for their opinion, where people were afraid to change their ways just because they were pressured and stressed out.

In such state it is very difficult to embrace change.

Rule #8: People love as self-recognition what they hate as an accusation (Elias Canetti).

The line between leading and ruling is very thin. In case the management tends to lean in the direction of ruling, people will often see this as an accusation that they are doing it wrong. Even if the intents of the management are genuine, it can demotivate people to embrace change.

Better is to enable people to see the benefits of change in their own way and time, because than they will see it as an opportunity to do better.

If I tell you that you made an error in your code, you probably will go in defence mode. On the other hand, if a coder has to start his day by reading the comment on his code, it becomes impersonal. If a compiler tells them that there 5 errors in their code, they will accept this without any problem. Making sure that developers can find their defects in an impersonal way will motivate them to correct their code.

This is only one example, but in any situation it is possible to let people keep their self-respect.

This way you eliminate any form of anarchy.

Conclusion

1. Rule #1: Avoid anarchy
2. Rule #2: Make goals related to quality
3. Rule #3: Use a framework only as support
4. Rule #4: Improvement on itself can never be a goal
5. Rule #5: Lead instead of manage
6. Rule #6: Inspire people by teaching
7. Rule #7: Recognise the reasons for resisting change
8. Rule #8: Give people the chance for self-recognition.