July 2018

Retrospectives in Agile methodologies, the base of improvements.

Paper written by Juanjo Escribano
Epic Engineer

Taking some time for analyzing ourselves, our status and also our internal processes, is a key factor for improvement either in software development or in our own lives. It is said that even the longest trip starts with a small first step, but knowing where we are is certainly important to go in the right direction. **You need to know where you are in order to know how to get to your destination**. But as we are all humans we all make mistakes. All the time. Everyday. So, even if you know where you are and the direction you must take to get to your destination, you can miss that step.

This can occur because of a lot of different factors: **maybe you failed calculating the route; maybe you did not fail, but you found something unexpected blocking your way**; maybe a meteorite crashed just in front of you and the road is now a mixture of stone and fire (yes, it happens... sometimes...).

Here is the importance of retrospective. **You should stop from time to time (how about every 100 steps on your trip?) and do a sincere analysis**: What did I do well those last 100 steps? What did I do wrong? Am I nearer or further from my destination? Is there any way to do the trip easier and faster? If I did fail when calculating the route, what can I change not to fail again? Retrospective, in a word.

We can't avoid going into the topic of retrospectives when talking about agile methodologies. Moreover, **I think these are the most important part of every agile methodology**. Good sprint planning, backlog grooming and daily meetings are certainly needed, but without retrospective sessions, these techniques would only be a good organization framework with a dangerous lack of personal and professional improvement, so we would always be the same, and we all know that this would not have a happy ending. So, **when talking about agile software development, these retrospectives must be done at the end of each sprint, and also at the end of the project**. Personally, I think that it is a good idea to take some time (how about 30 minutes?) at the end of the week to do your own retrospective of the last 7 days: Did you reach your goals for the week? If something went wrong, what could you do to avoid it next time? If there was a success, what factors helped to achieve it? Could these factors be applied to other circumstances and other projects?

All these questions must be answered in a retrospective session, but sometimes it is difficult to structure the ideas. When talking about this, **there are some techniques to help team members to structure their thoughts so the ideas flow in an easy way in the session. One of these techniques is called the 4L technique**. Let's see how it works:

A) **Draw 4 columns on your wallboard: Liked, Learned, Lacked and Longed for**. Then, ask the team to write in cards things that they Liked, Learned, Lacked and Longed For in the sprint (or in the whole project). Give them 15 minutes to write it and put the cards in the right column.

B) **Read all the cards and check if some explanation is needed**. It is not the moment for discussing, just a short explanation of the card if needed. Most of the cards should be self-explanatory. **This part should not take more than 5-10 minutes. Then ask the team to assign kudos to the cards in each column**. For example, each team member can assign 3 kudos in each column (drawing a tick on the card you want to assign kudos to is enough). This process should not take more than 5 minutes.

C) **Reorder the cards in each column so the cards with most kudos will be at the top of each column**. At this time, the team should have a good idea of what is going well and what is going wrong in the sprint/project.

D) **Go through the cards and discuss about them if necessary (if you do not have enough time, you can go through the most voted cards in each column)**. At this point, the columns Liked and Learned won't be a problem. Keep "Liked" as things that the team should continue doing. **When going over "Lacked" column, it is important to be positive: the idea is not to do a catharsis or just complain about something, but to get a real solution, or at least a partial solution to the problem.** Something similar happens with the "Longer For" column. Can the team do something to get these so longed for things? It is important to go through this part with a positive approach, looking for solutions, not for blaming. Be careful with "infinite discussion loops" that don't go anywhere. Try not to extend this part more than 1 hour (it depends on several factors, of course).

So, the retrospective session could be done in 90 minutes, and the output should be a list of things that the team should start doing, a list of things that the team should modify and a list of things that the team should stop doing.WWW

Juan José Escribano

Epic Engineer

✓ Software Architect

✓ Full Stack Engineer

Biography

Juanjo Escribano has been a Software Engineer at Epic Labs since 2017. He studied Computer Engineering at UPM and is currently studing degrees in Mathematics and Spanish Language and Literature at UNED. He loves learning such different areas because “a good engineer must be a good person first, and being a good person is impossible if we don’t understand who we are and why are we here, so the more I study the more I understand the world I live... or maybe not...”. He has been working as a software developer for 18 years althought as he says, he is not an expert at any technology, but he knows a little of a lot of differents technologies. He always carries a 10 sided dice in the pocket because “you never know when you will have to take an important decision for your life or fight with a goblin”.

epiclabs.io

epic labs

Calle Salvatierra, 4
28034 Madrid
+34 666 069 988
info@epiclabs.io