HOW MUCH BETTER SHOULD YOU LEAVE THINGS?

Overview: When finding a previously unnoticed bug you may feel compelled to fix it before moving on. But there can be unintended consequences to doing so. Before making any changes, speak with stakeholders and determine if a fix is necessary immediately and the consequences of such an action. And be sure to document what you've done!

We have all been there: you're deep into code and find a bug that has not been noticed before, or some code that should be cleaned up. You are right there so why not fix it? Sometimes the answer is yes, it makes sense to fix it then, following the old backpacker's mantra of leaving things better than you found them. But other times the answer is no. How do you decide?

Here are some questions to ask. To proceed, the answer to each must be favorable.

• Will this prevent you from meeting the requirements of your Sprint commitments? Will this push you over the allotted time?

• Do you have a high level of confidence in your understanding of the code and that your change will not cause any side effects? Is it better for you to do this now rather than having someone else to do it later as a bug fix?

• Should time be spent now to fix this? Speak with the Product Owner to set priority, there could be other things which are more strategically important.

• Are you the one to fix this? Should this be put on the team and allotted to someone else? It could be that they have better skills in this area or the reverse; this could be a suitable place for someone to expand their skills, improving the team.

• Can the fix be easily tested, particularly using automation? Does it require any complex configuration in order to reproduce?

• If this relates to UX, can you make this change now without the time for feedback on your solution?

Discuss and Document

Sometimes a second or even third opinion helps. You can provide a demo to the important stakeholders to gauge their opinion and how they prefer this problem be fixed. Stakeholders could include the Product Owner, quality assurance (QA) and user experience (UX) teams, or even your fellow development team members. If the bug was introduced recently and it is possible to talk to the person who wrote the code, they may have a unique perspective on how to fix it.

If you do decide to fix something "while you are in there" make sure that this is carefully documented. Do not try to cut corners to keep your time commitment by forgoing UX, comments, documentation, and testing. Failure to do so will end up causing more problems in the future. Do not forget that there is a code review.

Preparing for a Future Fix

So, what do you do with things you cannot fix in your sprint? Create the reports in your issue tracking tool and document them thoroughly. Document why you thought it was a problem. If you have a suggested fix, lay out why you feel it will fix the issue and point out any questions you may have about your approach. Also, keep in mind that fixing legacy bugs may have unintended consequences. This could mean adding more bugs than you are actually intending to fix or that users have created a process that accommodates the current behavior.

On that note, if you feel that it should be fixed sooner rather than later, before the release deadline, then it might be a good idea to speak to the stakeholders involved such as the Product Owner and UX team in order to relay that a fix for this bug is required. This then allows the Product Owner to plan ahead of time and re-shuffle any upcoming sprint commitments, if needed.

Establishing Guidelines

There is a balance to leaving things better while still meeting your commitments. This will not happen naturally; it will need your attention and the cooperation of your team. If you feel that the process is not well-nurtured amongst your team, then reach out to other developers you work with and see how they feel about this topic. By not having team guidelines related to this slippery slope, you may be increasing the list of tech debt items or slowing down your sprint velocity. Consider what guidelines you personally follow and let those spark a discussion with your teammates.