further problems, but postponing this kind of discovery is a risky strategy. What happens if that Priority 3 bug you've been ignoring has been camouflage a Priority 4? How long will it take to fix? Should we fix bug 12345 or bug 12346? These are all questions that could be avoided (including the often lengthy conversations that follow) if teams fixed bugs in time as they found them. Without doubt, every project will have its share of bugs that need special attention, but few projects require this level of attention to be the norm. Don't waste time with unnecessary discussions; fix bugs as soon as you find them.

7. UNFIXED BUGS HINDER SHORT-Notice RELEASES

Once in a while an event occurs that forces a team to release all or part of their software when they least expect it. As a result, an emergency patch is needed to fix the bug in the production environment, or on an unexpected visit from the project customers requires the latest release to be deployed on a client laptop. These events can take at the best of times, often make for an interesting patch, but when these events happen, bugs that have long been left unattended are suddenly re-test the bugs to consider. The result is often a delay rather than a release, because the team is no longer able to fix bugs as soon as you find them.

8. UNFIXED BUGS LEAD TO INACCURATE ESTIMATES

New bugs arrive all the time. Some require more seconds to investigate; others take hours to diagnose. Some take minutes; others take hours, and some take days. Major bugs can be automatically re-tested, others require manual verification. When determining project milestones, it's important to ask how many unfixed bugs a project has to achieve its estimates. Some people believe this is less important than the complexity of the bug. Regardless, every bug has a duration, and its duration affects how much work is left. Projects, and therefore estimates, can be inaccurate with even one unfixed bug. Don't let your releases be delayed by unfixed bugs; fix bugs as soon as you find them.

9. FIXING FAMILIAR CODE IS EASIER THAN UNFAMILIAR CODE

The human brain is capable of amazing feats of memory, but relaxing information indefinitely is not one of them. Consistently re-learning everything is often required to fix bugs that are unfamiliar. Don't delay your releases by leaving bugs un固定ed; fix bugs as soon as you find them.

10. REASONS WHY YOU FIX BUGS AS SOON AS YOU FIND THEM

1. UNFIXED BUGS CAME FROM OTHER BUGS

One bug often causes others, so every bug needs to be fixed as soon as possible. Don't delay finding important problems; fix bugs as soon as you find them.

2. UNFIXED BUGS SUGGEST QUALITY ISN'T IMPORTANT

We're all professionals at heart, but it's surprising how quickly a team can find themselves in a downward spiral. A developer working on software that already contains bugs, and most software does, will likely add more code of the same nature. Similarly, a tester who has seen tons of reported bugs winds up being unrealistic about reporting many more. Of course, it's not just developers and testers that are affected. Over time, every member of the team will start to ask themselves, "Who's left?" Why pay for a high-quality product when a substantial one is the missing piece? Don't starve your team with quality expectations; fix bugs as soon as you find them.

3. DISCUSSING UNFIXED BUGS IS A WASTE OF TIME

Regardless of whether they're part of project planning or a dedicated triage meeting or just gathered around a desk, discussing unfixed bugs is a waste of time. There really is no one opinion that needs weighing. "Open this bug need to be fixed?" Everything is subjective, which says it's just noise. Should we fix bug 12345 or bug 12346? All of these are questions that could be avoided (including the often lengthy conversations that follow) if teams fixed bugs in time as they found them. Without doubt, every project will have its share of bugs that need special attention, but few projects require this level of attention to be the norm. Don't waste time with unnecessary discussions; fix bugs as soon as you find them.

4. UNFIXED BUGS LEAD TO DILAPIDATED EFFORT

The greater the number of unfixed bugs, the harder it is to identify whether a bug has been reported. Imagine a scenario where there is a bug. When a "new" bug is discovered, it's easy to identify whether that bug has been reported by someone else. Now imagine trying to perform the same task when there are 50 unfixed bugs to sift through. It's either going to take a disproportionately long amount of time (which could be better spent looking for other bugs) or the bugs are transferred to the back of the shelf. Cut costs and to a greater extent. Don't put your project at risk, fix bugs as soon as you find them.

5. UNFIXED BUGS LEAD TO UNRELIABLE METRICS

Different teams analyze bugs in different ways. Some casually monitor how many are left to be fixed, whilst others track everything from their density to the project. Regardless of the complexity, every bug-based project relies on accurate analysis of bugs. As the number of unfixed bugs increases, it becomes increasingly difficult for management to make accurate decisions. Even if the information was correct at the time, the longer a bug has been unfixed, the greater the chance that information will change. The resulting implications may be severe for the team. Don't fall into the trap of decisions based on unreliable data. Fix bugs as soon as you find them.

6. UNFIXED BUGS DISTRACT THE ENTIRE TEAM

When somebody encounters an unfixed bug a number of distracting questions are planted in their mind. Take a developer who is about to make an enhancement when they notice a bug. Should they fix the bug first, or should they focus on the new feature? Fixing the bug might require several hours, but who can you trust to do that? Similarly, a tester who has stumbled across a bug that this bug needs to be fixed? They probably have a need to test another. Should they postpone this kind of discovery is a risky strategy. What happens if that Priority 3 bug you've been ignoring has been camouflage a Priority 4? How long will it take to fix? Should we fix bug 12345 or bug 12346? These are all questions that could be avoided (including the often lengthy conversations that follow) if teams fixed bugs in time as they found them. Without doubt, every project will have its share of bugs that need special attention, but few projects require this level of attention to be the norm. Don't waste time with unnecessary discussions; fix bugs as soon as you find them.

7. UNFIXED BUGS LEAD TO INACCURATE ESTIMATES

New bugs arrive all the time. Some require more seconds to investigate; others take hours to diagnose. Some take minutes; others take hours, and some take days. Major bugs can be automatically re-tested, others require manual verification. When determining project milestones, it's important to ask how many unfixed bugs a project has to achieve its estimates. Some people believe this is less important than the complexity of the bug. Regardless, every bug has a duration, and its duration affects how much work is left. Projects, and therefore estimates, can be inaccurate with even one unfixed bug. Don't let your releases be delayed by unfixed bugs; fix bugs as soon as you find them.

8. UNFIXED BUGS LEAD TO DUPLICATE EFFORT

There's not a team that doesn't have bugs. Some casually monitor how many are left to be fixed, whilst others track everything from their density to the project. Regardless of the complexity, every bug-based project relies on accurate analysis of bugs. As the number of unfixed bugs increases, it becomes increasingly difficult for management to make accurate decisions. Even if the information was correct at the time, the longer a bug has been unfixed, the greater the chance that information will change. The resulting implications may be severe for the team. Don't fall into the trap of decisions based on unreliable data. Fix bugs as soon as you find them.

9. FIXING FAMILIAR CODE IS EASIER THAN UNFAMILIAR CODE

The human brain is capable of amazing feats of memory, but relaxing information indefinitely is not one of them. Consistently re-learning everything is often required to fix bugs that are unfamiliar. Don't delay your releases by leaving bugs un固定ed; fix bugs as soon as you find them.

10. FIXING A BUG TODAY COSTS LESS THAN TOMORROW

For all the reasons listed in points 1 to 9, fixing a bug today costs less than fixing the same bug tomorrow. If a bug is left to fester in the software you are delivering, configuring or maintaining it can become aggerate other bugs, demotivate the team by suggesting quality isn't important, become the basis of painful conversations, cause duplicate effort, lead to incorrect project decisions, distract the project team. Watermark statements, invalidate estimates and lead to unnecessary frustration. And the longer you keep a bug exposed, the more likely these things become to occur and to a greater extent. Don't put your project at risk, fix bugs as soon as you find them.