Planning for Performance and Reliability

This topic has always been a bit of a groaner. Everyone knows the importance of planning (who hasn’t heard the adage “failing to plan is planning to fail”? but no one really wants to do it. Let’s face it, planning is simply not as fun as writing code. It’s very easy to come up with reasons why the planning can wait.

The problem with this approach is that by the time you start thinking that planning might have actually been quite a good idea, you’re already miles out to sea and heading off into even deeper water. So deep, in fact, that a sign proclaiming, “Here be monsters!” would not only not go amiss but might even be considered mandatory. Now you’re thinking that starting off with a plan might have been a swell idea. But if you’re not entirely sure where it all went wrong, it probably won’t be all that obvious what you can do to fix it.

Fear not! Help is at hand! As the famous Chinese saying goes, “if you want to know what lies ahead, ask someone coming back.” In short, the best way to avoid ending up waist-deep in a swamp is to speak to more experienced developers and ask them for help and guidance.

This chapter aims to provide a map that you can use to help steer yourself past the worst bits. Some of the things covered here will seem obvious, others hopefully less so. There’s no magic sauce here; all of this stuff is fairly straightforward. (But everything is straightforward once you know about it!)

We’ll start off with a very basic look at project management and how to plan your way through any project. It’s a lightweight process with a focus on giving you some structure to follow. You are most welcome to take what works for you and ignore anything that doesn’t. Feel free to enhance it in any way that suits your needs. What we provide here is really just a starting point; you should change whatever you need to change to make it as useful as possible to you.

The last section is admittedly a bit of a stretch for a chapter on planning in a book on performance. However, the number of sites we’ve seen taken down or severally damaged due to poor backups is really quite disturbing. So we’ll cover some of the basics of making good backups, the different types, and how to go about it.
As promised, we’re going to start off with good old planning and management. You might be strongly considering skipping this section. We strongly suggest that you at least skim through this section! Even if you only pick up one new idea that rings true, it will pay dividends from now on!

Because project management can be boring and because few people really want to read huge tomes on the subject (certainly not us, anyway), we’re going to introduce a very powerful yet refreshingly simple system. As luck would have it, the system has an easy to remember mnemonic of yoU MAke DInner In TiME which breaks down to

1. Understanding
2. Making Decisions
3. Design and Implementation
4. Installation
5. Testing
6. Maintenance
7. Evaluation

We cover each of these points in the following sections. Although it looks pretty full on, it’s really easy to use; quite honestly we’ve never found anything better! Although these ideas are mostly applied to development projects, they can also be easily applied to developing your system for performance. In fact, you can apply these techniques to pretty much any project successfully. The idea is that with this roadmap you’ll stray from the path less than if you just winged it (which is what most of us tend to do). So even a passing familiarity with this stuff will likely be quite helpful in the long run.

Geeks (or computer professionals, if we want to be politically correct—but we don’t) are often a little trigger-happy when it comes to problem solving. There is nothing like a good problem to stretch those mental muscles, especially as we spend most of the day explaining where the Start menu is and why when you left-click you really do need to click the left mouse button and not the one on the right. Perhaps this is why we tend to jump into a problem with both feet—to recover from the mental starvation we have been suffering. Whatever the reason, this approach has a nasty habit of either backfiring immediately or helping us to paint ourselves into a corner.

Now, before anyone else says it, we know that sometimes those really simple problems really are that simple. Sometimes when we say “It will just take a minute,” it actually does. More often that not, though, what was described as a really easy problem turns out to be a rat’s nest of tangled requirements, dependencies, and confusion. And let’s not forget deadlines, which are usually as realistic as founding a unicorn ranch staffed by fairies. The problem is that most of the time we find out about the hidden hell under the carpet once there’s no possibility of turning back.

Fortunately, there is a solution. Before jumping into the pool of project happiness, have a look around first. Speak to people and try to understand more about the situation. If you find out during your conversation about the piranha-infested swimming pool, you will be very grateful you didn’t decide to test the waters using yourself as bait.