Top 5 Most Common Missed Issues in Software Estimation

eBooks by INTERTECH

Table of Contents

Infrastructure Costs

Software development can require very little infrastructure. Hire a few developers, give them each a computer and a list of features to build.

Voilà!

You have a software project. As the industry has matured we realized that while this can work, it rarely produces an acceptable solution. Successful projects require infrastructure to reduce risk and increase efficiency of the developers.

Setting up infrastructure is an expensive upfront cost, but is paid back many fold over the cost of the project. Setting up infrastructure for a software development project often involves time and materials from other groups within an organization.

Provisioning virtual machines usually requires involvement of the IT group, setting up a new GIT repository might require a DevOps team, and setting up a test environment could require the QA group. Each of these groups have separate priorities, and might not be able to perform the required setup immediately. This can cause delays and increased cost before a project can even start.

A few infrastructure costs which are frequently missed:

- Setup / Configuration of Source Code Repository
- Setup / Configuration of CI Server
- Artifact Management server
- Acquisition of Test Servers
- VMs or physical servers for each
- Permissions to each of these

Staff Training and Turnover

Each development project requires a specific set of technical skills, and this set of skills is frequently different from the last project you completed. Keeping your staff's technical skills current can be difficult and/or costly.

Utilizing the newest libraries, frameworks, and tools can increase developer performance, reduce defects and keep employees engaged. However training for this newer technology, or slowing the project down to allow developers to learn the technology themselves comes at the cost of time and money.

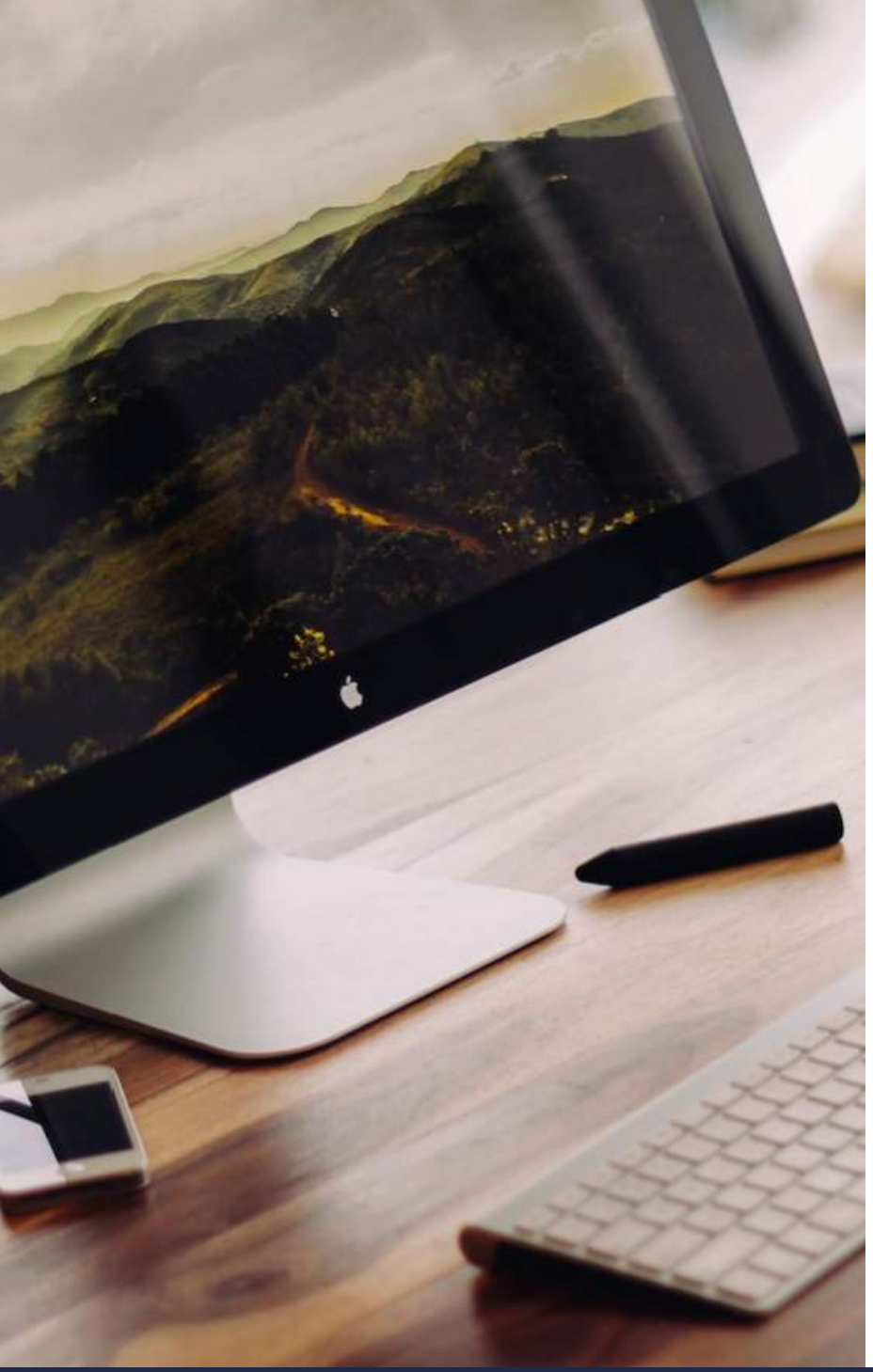
On top of the constant need for new skills, the software development industry has the highest rate of turnover of any industry. This frequently causes unexpected delays and cost overruns for new projects. It is rare for a medium to long term project to end with all of the same staff it started with. The staff hired to replace those who left, will not only take time to get up to speed, but will also take time and effort from other members of the team to get them up to speed.

To mitigate this impact on your project plan, plan on scheduling training early in the project, and encourage cross training in not only different parts of the project, but also in different technologies.

Your biggest asset in this area is having employees and/or consultants who are not only skilled in the technology, but also skilled in training and teaching other developers.

Project dependencies

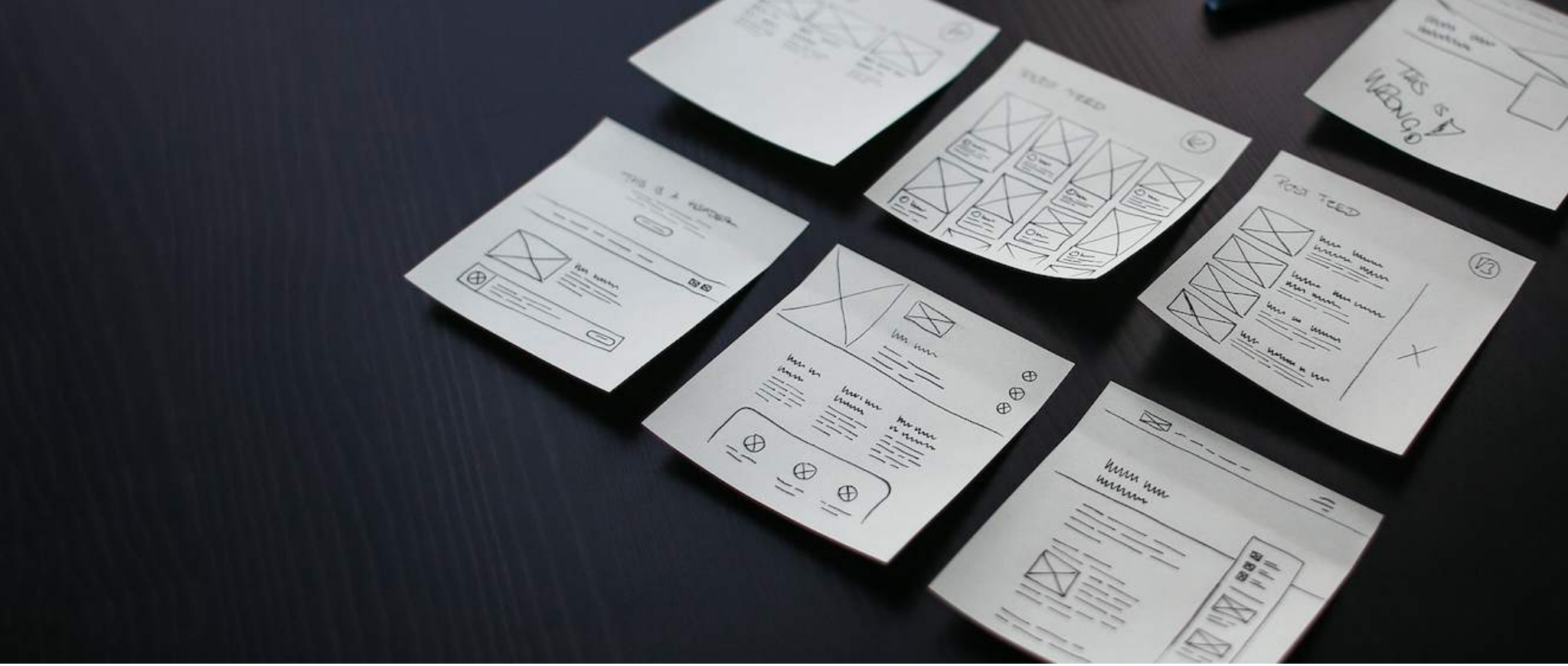
Very few projects are built in isolation. Most projects are dependent upon one or more group's delivery to complete and sometimes even start the project. In an industry where less than one third of projects are delivered on time and on budget, depending on another project can be a serious risk. Scheduling and budgeting for a project which depends upon the success of several others can be very difficult. It is important to not only look at the success rate of a specific team, but of the organization in general as this team may have dependencies they have been waiting for.



These dependencies may not even be for internal projects. Sometimes a project depends upon integration with a third-party. This might be accessing a public API to collect weather forecasts, or integrating with a credit card company to process payments. If all of these integration points worked as advertised, life would be easy. More often then not the process of integration will discover defects and quirks. If the third-party fixes these defects and quirks, your project will be delayed. If the third-party cannot or will not make any changes, your project may take significantly longer, or be canceled.

Depending on other projects is rarely avoidable, the best things you can do to reduce this risk is to add as many communication channels between your team and the leadership of the project which you depend on.

Make sure you are monitoring not only the other project's timeline and development rate, but also their bug counts and trends. Some projects are "done", but have so many bugs as to need an immediate new version to make it useful to you.



Following an Agile methodology makes these lines of communication and status monitoring much easier. If possible attend the demo of your project dependencies, this will give you a good understanding of not only where the project is “supposed” to be, but how well it is actually working.

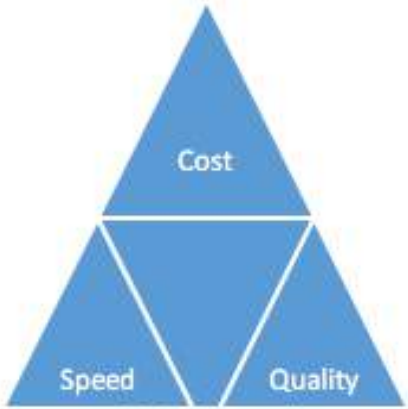

Don’t plan for Deployment
/ Maintenance

A project doesn't stop when the final line of code is written, and the final test passes. This is just the first step in the life cycle of a software product. Most developers don't work in an environment where they are concerned about how the software is deployed. What supported network configurations are supported? Are upgrades installed by professional services, or by the customer? Who monitors the software to make sure it is running properly, and do they have sufficient information to know that it is?

These are just some of the questions which start to get asked towards the end of many projects. Since these were not asked at the beginning, many of the answers mean extra development time. If the software is not maintainable in production, it is unlikely that it will be successful.


Lack of communication on
Speed vs. Cost vs. Quality
(SCQ) mix

Each project has unique market requirements for how much it can cost, how quickly it must be done, and what level of quality it must reach. Without clear communication of where on the SCQ triangle, developers, project owners and management spend significant amounts of time discussing which is more important for each feature.



SCQ Triangle

An internal development tool might work being quickly created by one developer (fast and cheap) and any development beyond that is gold plating. Whereas a medical device needs to be primarily of high quality, otherwise failure to meet regulatory requirements, or future lawsuits could be extraordinarily expensive. Each developer needs to know where on the triangle the project sits, so they can make quick decisions about how much and what type of work needs to occur on a daily basis

Determining where on this triangle the project needs to be will require the product owner, project management, sales & marketing, and the system designer / architect.

This is usually not a quick decision, and the location may drift over time as the need for the project comes into better focus. The earlier in the project lifecycle this can be clearly communicated, the easier many decisions will be.

Thinking about starting a new software development project? Check out Intertech's Software Development Project Feasibility Study page.

We will find the most cost effective approach for your project, map out requirements, and provide recommendations for how you should tackle this new software development project.

651.288.7001