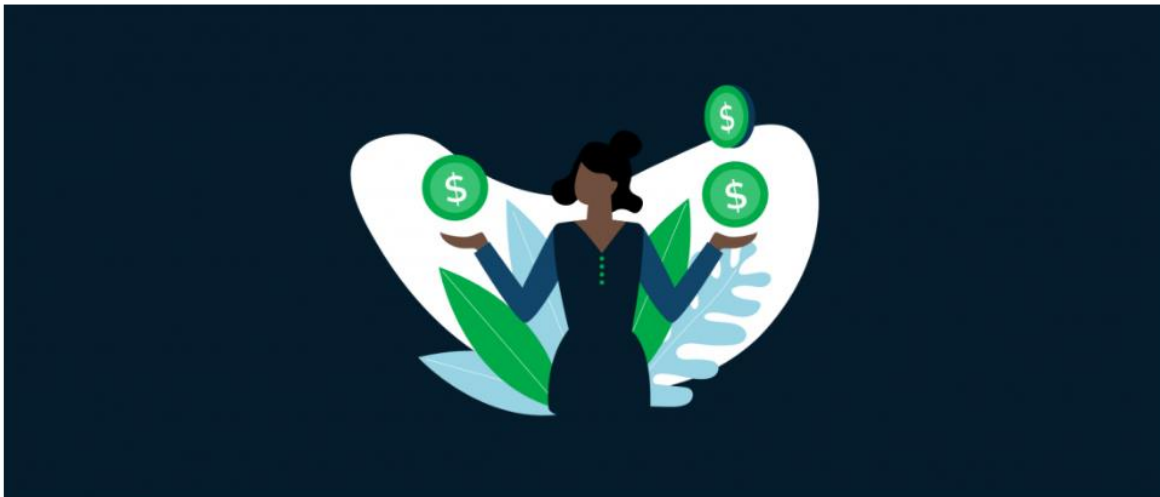


# Ghostwritten blog post for field service management software company Dispatch



## 7 Hidden Costs of Building Your Own Enterprise Software

Published March 20, 2020



Thinking about building your own enterprise-level field service management software? It might be tempting—you can customize it to your unique needs, and you may make it with a smaller price tag as well.

But there are a lot of hidden costs waiting to derail your best-laid plans. If you want to keep from being blindsided, check out our list of hidden costs to building (instead of buying) your enterprise software.

## 7 Hidden Costs of Building **Your Own Enterprise-Level Software**

Here are the costs you'll incur when building an enterprise field service management solution. Many of these are time-related costs, but you know the old saying: time is money.

1. **Before-You-Begin Costs** like the time you sink into creating a budget.
2. **Planning Costs** of building out sprints.
3. **Project Management Costs** to make sure everything stays on track.
4. **Project Development Costs** of the QA process and bug fixes.
5. **Training Costs** for everyone from your tech team to your end-users.
6. **Troubleshooting and Support Costs** to deal with new bugs and user error.
7. **Maintenance Costs** of creating patches and new versions of the software.
8. **BONUS: Opportunity Cost**—the time you spend on all this instead of your core business goals.

Now let's get into the nitty-gritty on the time-and-money outlay that comes with creating your own field service software.

## 1. Before-You-Begin **Costs**

Before you even start the actual software development, you will have time costs associated with your project. One of the first (and, arguably, the most important) is nailing down a general budget.

This also tends to be one of the more difficult costs to determine, according to consulting firm McKinsey. Their research found that [the average large-scale IT project runs a whopping 45%](#) over budget—and 7% over time. Some of the reasons they cited for these overruns were unclear objectives, an unprepared team, and unrealistic schedules.

You can mitigate some of these issues by hiring a business analyst or product manager, which is the second hidden cost of your software development journey.

If you don't already have this person on your team, it'll take time to hire someone competent to fill this role. The phrase "time is money" became a cliché for a reason—all of the time you invest in hiring a project manager is time taken away from your other business projects and goals. On top of that is the time for the onboarding and training your new hire will need.

While you're thinking about hiring, you'll also have to add developers to your team. This can be a pricey proposition, especially if you need multiple new employees to build your enterprise software – not to mention the salaries associated with each hire.

## 2. Planning **Costs**

Your next big chunk of costs involves all of your planning activities before the actual development begins. You'll need time to decide on the most important business objectives (what you want your new software to actually do), the technical requirements to get there, and a blueprint for what that solution will look like.

Once you've got your goals and a blueprint, your developers need to [build out sprints for the project](#). Each sprint requires a planning meeting between you and your developers to decide what work will be done in the time period defined by the sprint. The number of sprints and the length of each depends on what you need your software to do for you, but even for small projects the time needed to plan them adds up.

### 3. Project Management **Costs**

Now that you've got the project and sprints defined, a product manager on board, and developers to actually build your enterprise software, you can just let them get to work, right?

Not exactly. Project management can be a time-consuming process on its own. You'll have to manage your developers to ensure they get work done on schedule, make sure all development teams are communicating and collaborating effectively, analyze past sprints and plan new ones, and put project monitoring processes in place.

### 4. Project Development **Costs**

Once your software is nearing its finished stage, a new set of time-intensive costs pop up. First, you'll have to put your software through a quality assurance process. Your team should be doing regular QA tests throughout the development process to avoid last-minute disasters, but you'll still have to sink some time into testing the final (or near-final) version before releasing it.

If your QA testing reveals bugs, you'll have added time costs for fixing them. Finally, before actually handing your software over to end-users, you'll have to perform an audit to make sure it meets security and compliance requirements.

### 5. Training **Costs**

You can't just hand your software to your users and let them figure it out on their own. Everyone will need training, from your end-users to your tech support team. That means developing training materials (either written documentation or e-learning courses).

When your materials are finished, you'll have to provide time for your users to learn the new system—which means staff costs that aren't going to the regular operation of your business. Depending on the complexity of your new software (and how many hours your employees need to learn it), those costs can be steep. Not quite parallel to this situation but still enlightening, *Training Magazine's* 2017 Training Industry Report found

that small businesses [spend between \\$1,100 and \\$1,886](#) to train new employees for 36 to 43 hours. That translates to \$30 to \$43 per hour.

## 6. Troubleshooting and Support **Costs**

When you build your own software, you're also on the hook for any ongoing tech support required to deal with new bugs and troubleshooting. Just about any software solution has its own unique set of bugs (and, yes, user errors) that appear regularly.

Beyond the regular troubleshooting time, your IT team will also have to collaborate with your developers to fix these recurring issues in the next iteration of the software (which starts the sprint process all over). They also will have to consider operational costs – monitoring, care and feeding of databases and storage systems, etc.

## 7. Maintenance **Costs**

So, you've got your shiny new enterprise software built, your end-users and support staff have been trained, and your teams are collaborating to keep up with bugs. All done, right?

No again. Maintenance tasks—like patches and upgrades—can add up, too. Your developer team will have to spend time creating these patches and new versions of the software, while your IT staff spends time implementing and troubleshooting them. Additionally, new security vulnerabilities will need to be mitigated as it pertains to both patches but also the software packages used.

## 8. Bonus: Opportunity **Cost**

All of the costs related to building your own enterprise software can add up fast, especially if your solution needs to be complex and integrate with a lot of existing systems.

But one cost a lot of businesses may not think about is the hit to the core business. While you're involved with planning and setting up this new software—which can take months—you're not as focused on your overall business goals and opportunities.

If you've got a great business analyst and project manager, the cost here may be minimal. But if you're really involved with the software production and launch, you may miss other potential areas of growth, which can set your business back.

## The Case for Buying Instead of **Building Enterprise Software**

The cost of a purchased software solution may look enormous at first. But when you think about all of the staff time and management/IT costs that go into building your own software, a purchased solution looks a lot more attractive.

With purchased enterprise software, you don't have the headaches involved with meeting development deadlines, creating your own training materials, or monitoring software health and creating patches and updates. With those issues in mind, a purchased software solution becomes more budget-friendly.