



1 Person Can't Wear 6 Hats
in a successful software project

**Considering hiring one person to fill your software needs?
Think again!**

Executive Summary

A software development project requires a wide variety of skillsets. Just as you can't hire a plumber to design and build your house, you can't hire a programmer to define, design, manage, code, and deliver your software system.

An innovative software developer who thrives on figuring out new technologies may not have the strong people skills necessary to be a good project manager.

A requirements analyst who excels at working with subject matter experts and translating business needs into software specifications may not have the technical skills necessary to write code.

A thorough tester who presses every button and follows every possible path through the system may not have the communication skills vital to an effective trainer.

One person can't wear all the hats required for a successful software project.



Steps in a successful software project

Here are the steps in a successful software development project.

1. Define the requirements in writing and with pictures that show you what system pages will look like.
 2. Architect the system - similar to the way a construction architect develops blueprints for a building.
 3. Develop test plans. This is not required if you will test against the requirements document.
 4. Develop the software – also called programming or writing code.
 5. Unit and end to end testing. Iterative testing is completed as each screen and function* is coded. End to end testing occurs when the system is complete.
 6. Install software on the hardware/server/network on which end users will test it.
 7. Train people who will be testing and who will use the system in production.
 8. Testing and feedback from the people who will use the system.
 9. Final coding that incorporates test results and feedback.
 10. Complete system documentation (if desired).
 11. Install software for production use (up to this point, the software would have been installed on hardware/server/network used for testing).
 12. Go Live with the new system.
 13. Provide system support and ongoing enhancements.
- *Function: work the system needs to perform that is not shown on a screen or report – often called the “backend processes.”*

The size of the project determines the time it takes for each step; for example, smaller projects may be defined in 8 hours or less, while it may take more than 100 hours to define requirements for larger projects. For big projects, consider an iterative approach that breaks the work into manageable, well-defined sub-projects.

Software Project Roles

1. Project Manager

To ensure that a large software project is completed on schedule and meets your requirements, you need a good project manager who will:

- Develop a project schedule.
- Ensure deadlines are met.
- Communicate with the development team and all stakeholders regarding project status.
- Identify and overcome obstacles to project success.
- Handle change requests.

2. Requirements Analyst

The person who develops your requirements will:

- Listen to your explanation and quickly understand how your business works.
- Understand how software can support your unique business process.
- Communicate with you – verbally and in writing – about your business needs and your software.
- Translate your business needs into screens and functions that can be modeled to show you how your system will work.
- Document requirements in a format that programmers can use to code your system.

If you are developing a new system and want to add only one full-time employee, hire an experienced maintenance programmer and contract with a software development firm for the other positions on the team. You pay for each role only when you need it, and you avoid adding the overhead of full-time employees whose job ends when the software project is complete.

3. System Architect

The system architect understands your business needs, sees the big picture, and creates a technical blueprint for your system. The architect enjoys challenge and likes figuring out new technologies and creating patterns others will follow.

4. Software Developer

Software developers use the system architect's methodology and plan to write code that brings to life the screens and functions defined in the requirements. Developers tend to fall into two groups:

- *New development specialists*: these programmers enjoy working with the system architect to figure out new methodologies and ways to use the tools to solve business problems. They like challenges and prefer exploring new technologies; they are innovators who thrive on solving complex problems and do not enjoy repetitive tasks.
- *Maintenance programming specialists*: these programmers excel in using proven methodologies and technologies to quickly and accurately complete the functions and screens required for a new system. These programmers enjoy the challenge of getting a system working reliably and predictably. They are also good with creating system reports.

Generally programmers who enjoy new development don't like the repetitive nature of maintenance programming, and developers who enjoy maintenance programming are often uncomfortable with the unpredictability and challenge of figuring out new technologies and solutions.

What's in a name?

What's the correct name for someone who writes software?

- Programmer
- Developer
- Analyst
- Systems Analyst
- Systems Engineer
- Software Engineer
- Coder

As if the list isn't long enough already, you'll also see these titles used in conjunction with senior, junior, or lead – for example, a lead developer or junior programmer.

Don't be distracted by the title; focus on the role and the work to be done.

5. Software Tester

Testers are willing to press every button, enter information into every field on a page, and clearly document the steps followed and results. They use the requirements and/or test plans to test each screen and function, document any deviations from the specifications, and re-test after the developer makes changes. The person who writes the code should always test it before declaring it complete; however, final testing should be completed by someone other than the developer, who may not follow the same steps as a "real" user of the system.

6. Trainer

If you develop a custom software application that automates your business processes, it should be easy to train people to use the new system. The person who conducts the training should understand your system, have great communication skills, and be comfortable answering the many questions that come up during training classes.

What about big projects?

For larger projects, it's impossible for one person to fill all the roles required. Many software developers are not skilled project managers or requirements analyst, in part because programmers often prefer talking with computers to communicating with people. Project managers and requirements analysts may not have the expertise required to be a successful programmer. Someone who is a great tester may not have the verbal skills necessary to be a good trainer, and programmers who are great with figuring out new things may find report writing boring.

For a big system, we recommend following the same process with one change – break the project into smaller deliverables.

- Define the business process (graphically) at the beginning of the requirements phase.
- Identify the vision for the overall system, but break the project into logical phases.
- Create requirements and code the project in phases. (This means many of the steps above become iterative.)

The benefits to breaking a project into phases are:

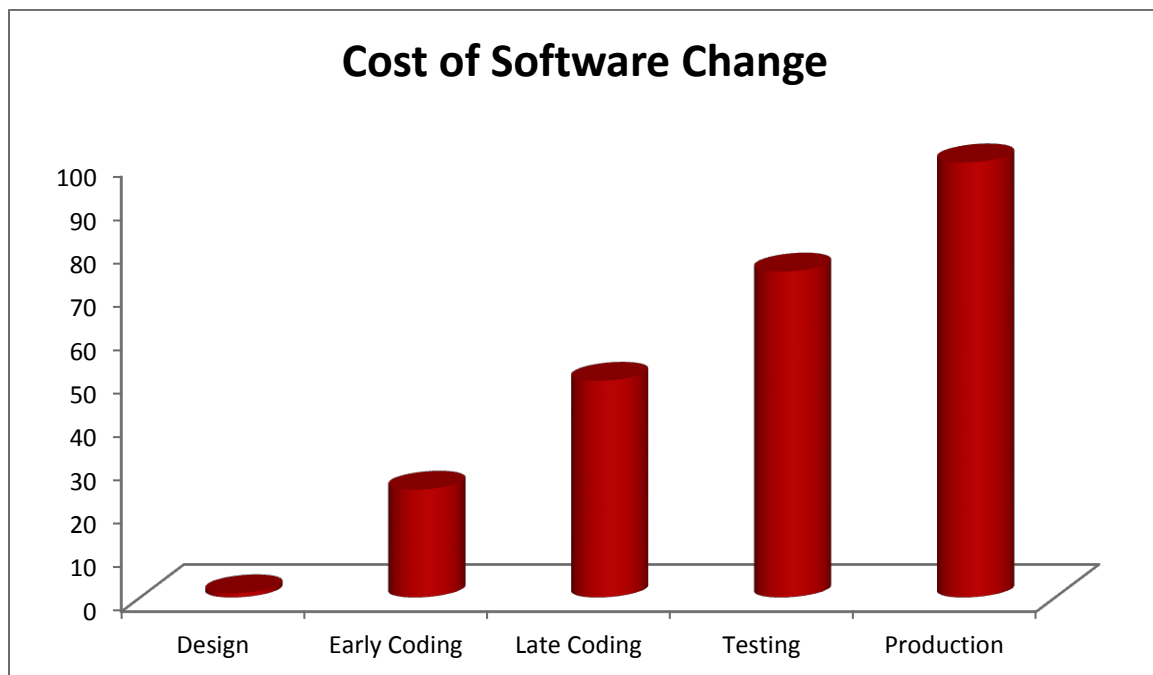
- Begin using parts of the working software very quickly.
- Incorporate feedback from each phase into future work. (*The chart below shows why this is very cost-effective.*)
- Create manageable, well-defined deliverables, so you know whether your project is proceeding as planned.

Identify changes early in the project

As illustrated in the chart below, the cost of a change discovered late in a software project may be 90-100X more than one found early in the process.

Why are late changes so expensive?

- **Late changes often require retrofitting existing code and/or modifying the database.** The database is the underlying foundation of your software, and modifying it may be like trying to change the foundation of a completed skyscraper.
- **Late changes necessitate end to end re-testing of the system,** since the changes may have ripple effects throughout the application.



If you want to hire your own programmer . . .

If you choose to hire a full time software developer to work on the new system and continue to maintain it, we recommend that you look for an experienced maintenance programmer who enjoys working with a familiar system, writing reports, and enhancing existing applications.

Contract with a proven software development firm to avoid the cost of hiring permanent employees to fill the rest of the positions for your software project team. You enjoy the skills of a project manager, requirements analyst, system architect, tester, and trainer and pay only for the time you need them to work on your project; when their part of the project is done, your cost ends.

Adding a software firm's programmers to your team:

- Gives you access to new development specialists only when you need them and avoids the cost of finding, hiring, and paying benefits to expensive full-time employees.
- Shortens the timeline for completing your software by dividing the work among a number of programmers.
- Provides your developer with resources to brainstorm solutions and collaborate on solving technical problems.

Cost-effective, on-time projects that meet your needs

When you have the right people working together in each of the six roles in your software project, you significantly increase the probability that it will meet your business needs and be completed on time and in budget.





About DragonPoint

DragonPoint has 25 years of experience designing, developing, and enhancing custom software for businesses.

Call DragonPoint today at 877-542-0657 to talk about your software needs and get expert assistance writing a specification that communicates your business needs!



This document is for information purposes only. DragonPoint makes no warranties, express, implied or statutory as to the information in this document. The information in this paper may be reused and re-transmitted provided DragonPoint, Inc. is cited as the original owners.