

On GameSpot: [Watch the Academy Awards of gaming](#)

- BNET Business Network:
- [BNET](#) |
- [TechRepublic](#) |
- [ZDNet](#)

Design specifications: How much detail is enough?

by [Jeff Davis](#) | Mar 13, 2002 8:00:00 AM

Takeaway: The first step in delivering a great application is creating a solid design specification document. Find out why you should demand a design document and avoid coding on the fly.

When your team members develop an application, how do you know they are coding the functionality correctly? Without a complete, unambiguous design specification document, you could be setting yourself up for costly rewrites.

“A great functional design specification is like a pyramid,” according to Peter Schickler, president of [Button Systems, Inc.](#), a Vermont-based software development company. “At the top is a broad-brush overview that describes the wide spectrum of the system and its primary components. At each level below that, you have an overview of each of the primary components and as much detail as your developers require.”

Schickler and his team of developers prefer to work with design documents that include sample screen shots, detailed descriptions of each function, database layouts, and report layouts. In cases where a customer submits a design document without sufficient detail, Schickler’s standard operating policy is to “respond in writing with a breathtaking number of questions.”

The art of writing a design spec

“Writing a design specification is more of an art, something to be learned, more than something that can be taught,” said Todd Tincher, call center manager for [Appriss](#), which provides intelligent communication solutions for vertical markets. Tincher was the project manager for an 18-month, multimillion-dollar call center development project. The job came in under budget and on time, but it wasn’t easy.

“When I first started out as a developer and project leader, I tried to find a cookie-cutter explanation of how to write a good design spec,” said Tincher, “but there wasn’t one. So I wrote a basic outline, then filled it in based on the personalities of the developers and the needs of the customer.

“I’ve struggled to come up with one document that fits the needs of both the developers and the customer,” Tincher said. “I don’t always show my customer the whole design document, because they won’t understand it all.”

And then there were two

Tincher sometimes goes so far as to create two design documents: a broad-based description for the customers and a detailed document for the developers.

“Some developers want as many details as possible,” Tincher said. “But the artistic developers want a broad-based design spec. They just want to know what functions the application needs to perform, and they like to fill in the details themselves.

“I start off the document with a high-level description of the functional design: Basically, here’s what the application is going to do. When the customer signs off on that general description, I create a second file. In that document, I go into more details, if necessary, for the developers, and ask for their buy-in.

“The extra details aren’t always necessary if the developer, or at least the project lead, has been involved in the process since the initial discussions,” Tincher said. “Getting the developer involved early on gives him [or her] a feel for the customer’s needs and scope of the project from the beginning.

“It also helps avoid situations where the developer comes back to the project lead and says, ‘You should have asked me before you told the customer we could do this!’”

Advice for new project managers

Tincher offers this advice for new project managers as well as developers who aim to move into a project-management role: “Don’t get bogged down in the details.

“If you’re tackling your first big project, you naturally want to succeed. You put together a great design specification to provide the road map for the process. But you don’t want to wind up doing the work instead of managing the work. Meet with your developers regularly. Let them know you’re following up on their issues. Keep the design specification document updated. But don’t jump in and try to do everything yourself.”

Customer sign-offs are a must

When Tincher writes a design document, he reserves the last page for the customer's signature.

"The customer has to sign that page indicating they agree to everything in the document, and there are two reasons for that. One is to cover your [bases], which is always important. Two, if you sign something, you're going to read it. I don't know how many times I've handed a design document to a customer who says, 'Oh yeah, that looks good.' Then you deliver the application and the customer says, 'Where'd that come from?' You say, 'It's right here in the document,' and the customer says, 'Well, I never read that thing!'"

Schickler takes the sign-off one step further. "I make my customers initial the bottom of every page of the design document," Schickler said. "That protects my billable time. When the client asks for a change or a new feature or functionality, if the program doesn't work according to the design spec, I fix it for free. If the client wants something that isn't in the design spec, it's billable."

Details required for QA testing

Anna L. Marra, a quality assurance analyst for Appriss, thinks a design spec should get down to the nitty-gritty. "A great design specification document should be extremely detailed," Marra said, "down to the level of what each function does and what result is expected after each action."

Marra said the quality assurance team should meet with the developers and review the first draft of the design document, line by line.

A detailed design spec can make life easier for everyone involved in quality assurance. For starters, "it's easier to write a test plan from a good design spec," Marra said. That, in turn, makes the testing easier, particularly if the person doing the testing isn't the one who wrote the test plan.

Copyright © 2008 CNET Networks, Inc. All Rights Reserved. [Privacy Policy](#) | [Terms of Use](#)