

There Is No Perfect Design

Many developers, especially those fresh out of college (though older developers are just as prone), fall into the trap of believing in an absolute concept of “the perfect design”. I hate to break such youthful idealism, but there’s just no such thing.

Design is always a series of trade-offs in an arduous struggle to implement the best solution given a set of competing constraints. And there are always constraints.

Not too long ago, I had an interesting discussion with a young developer who was unhappy with the design of a project he was working on. This project had a very aggressive schedule, and he complained about the poor design of the system.

“So why do you think it is poorly designed, the system appears to have met the requirements, especially given the short time constraint”, I asked him. He explained how he would have preferred a system that abstracted the data access via some form of Object Relational Mapping, rather than simply pulling data from the table and slapping that data on a page via data binding. He also would have liked to clean up the object model. It wasn’t in his mind, “good design”.

I pointed out that it also wouldn’t have been good design to spend time choosing and getting up to speed with an ORM tool, only to deliver the software late (which was not an option). Sure, the code would have been well factored, but we had a hard deadline, and missing it would have been a huge burden on the company.

I suggested to him that **constraints are necessary** for a software project. I told him,

If a project doesn’t have a time constraint, it will never get finished.

That lit a lightbulb for this developer.

That explains why I never finish my personal projects.

Absolutely! With no time constraint, this developer would spend more time after more time attempting to hit that elusive goal of the “perfect design”. But that goal will never be reached **because perfect design is asymptotic**. You can get infinitely close, but you can never reach it.

In the end, I told the developer that he’ll have the opportunity to refactor the code into a better design in the second phase of the project, as the time constraint is no longer so aggressive. I also suggested he skim [Small Things Considered: Why There Is No Perfect Design](#) by Henry Petroski. The book makes its main point in the first chapter, that design is about compromise and managing trade-offs to meet constraints. The rest of the book is a tour of various design decisions in history that illustrate this central theme.

Author: Phil Haack

Copied from: <http://haacked.com/archive/2005/05/31/ThereIsNoPerfectDesign.aspx>

Article downloaded from page eioba.com