Just what is 'Best Practice' exactly?

Posted At : 31 January 2012 09:14 | Posted By : Shaun McCran
Related Categories: Best practices

I've been an advocate of 'best practice' for a while now, in any random psychometric test I'll always come out rated as a perfectionist which may go some way as to explaining why I am constantly striving to make whatever process I am involved in either more efficient or generally just a smoother transaction.

But some recent musing has left me wondering just what 'best practice' is? When I sat down and really thought about what the term means to me I came up with some surprising answers.

When I really started thinking around the terminology and meaning of 'Best practice' I soon realised there were several obviously different facets to the term for me.

Technical Best practice

Technical best practice is the most easily defined facet. This is an adherence of what is considered the best technical solution to meet the project requirements. The confusion often comes here where different parties have different views on exactly what the best course of action is.

For me the key factors in Technical Best practice are reliability, simplicity and flexibility. Will the technology work in the same way, every time? Also can it sustain whatever uptime is required, and a good level of performance? As an example don't write a Rich front end RSS aggregator to be deployed on a site with a very narrow internet connection. This also branches into wider project decisions such as 'is there a wide pool of resource available to support the tech choice'?

The most common conflict I have seen in this arena is where your developers have discovered a new and interesting way of developing an application that may involve non standard technology, or cutting edge code, that no-one else knows or can support. Sure it might be fun for the developers to build, but how easy is it to extend or support later?

Moral Best practice

This is more ambiguous facet of the term, and one I have come to appreciate more and more. I have spent a lot of time building a career in software Architecture and I'd like to think I know a fair bit about it by now (14 years no less!). Based on all this experience I have a pretty good gut feeling about projects. You know when you are working on something and there is a general feeling of uneasiness about it? Well you should listen to that feeling. It means that something isn't going the way it should be.

For me this feeling comes from the project heading in a direction that I suspect may not be the most successful. Sure we are all paid to work on projects, but don't you enjoy them more when you really believe that it the outcome will be a success? There is nothing worse than having a hand in something that everyone suspect will fail as soon as it is released.

The conflict for me here is that often this feeling stems from client requirements that conflict with Technical intelligence. It is often a careful balancing act managing clients' demands whilst trying to align them with what you consider to be the best possible project outcome.

In Conclusion

Looking at these two definitions it is easy to see that the first is easy to document. Without any real trouble you can (and should) document any best practices in technology as and when you implement them. It's quite black and white really.

The second definition is a little harder to pin down, and really is a bit more of a minefield.