Developer Vs. PM's perspective

written by Manoj Khanna | April 5, 2003

Estimation & Invention

We simply can't estimate by the minute. Some teams estimate by days and is more often wrong then right. There are simply too many facts not controllable by a project manager to give an estimate by minute a meaning.

We often say: We INVENT things. Every program deals with a new thing and it's nearly impossible to estimate the needed time for inventions.

As far as I can see PM and the team should use something to track down the time actually needed and compare it with their estimates. This will give them a better idea on how and where their estimates were wrong. Over time their estimates will improve by doing this.

Bug Fixing

Estimates for fixing a bug are one of the hardest things to do. This is because very often you need some idea where to start checking things. Even if you got this idea, it is often not clear what exactly needs to be changed. I find myself often in a situation, where some modules simply don't work together very well. It would be easy to change one of the components to fit the needs, but then a 3rd module would break, so as one possible option I have to accept a hack in the 2nd module. Or I can rearrange the whole thing of 3 modules to work together perfectly (preferred way!) These decisions can only be made when the clock for fixing the bug has long started ticking.

KNOW where to SAVE

Some people think that software engineering is an expensive work. And often the only thing that counts is the time used to get a program running. The quicker, the better. Regardless of the remaining errors, the maintainability of the code or its documentation.

I'd say don't accept this. This is short time thinking. If you need two days more to get it done correctly, do it now. The "saved" two days WILL come back later (maybe a year or two years later), but they WILL come back, but then it will be a week.

Coders Haven…where??

One of the best methods the coders follow is "Think first, code later". The theory behind this is that the time spent in design pays back by lowering coding time. As it is widely known that:

"How do you know what to code, if you have no idea of the problem?". And Moreover structured programming is the only way to get larger projects done.

There is a saying:

"Plan to throw one away! You will anyway".

Fact is: If you analyzed the problem and thought about different solutions, about their pitfalls, drawbacks and advantages you will end with a design that fits your needs. But sometimes it happens that it turns out that your theoretical concepts don't work that good in practice. Even if it is hard: Throw it away!

I will quote here an example of my previous company where I was reworking on the code that one of our former employees had left. He was one of the guys, who fixed a bug by adding a special layer on top of his code to handle a (what he thinks

is a) special case. Needles to say, that this additional code has bugs by itself, so a next layer is needed to take this into account. When I rewrote his code I manage to get rid of 2/3 of his code, making the whole thing smaller and thus more manageable. The whole process took

4 weeks of my time and which I could have finished in one and a half weeks.

Conclusion

Well, if someone plans to make things work without even knowing how, it’s his choice but the last thing to take into consideration is that his program is going to work. Needless to say that the most famous scientists just say: “If you want to make something, build the model first”. Software engineering is just building models for OO and similar programming techniques. If you don’t intend to program using certain methodologies and style you should consider taking track of all the variables through the code. PM is not a coder, but someone who must know what and when to include in the project. After making a good project representing using the best terminologies (estimation and planning) then rest is just a piece of cake.

© Manoj Khanna 2003 – 2012.