Re: Freezing requirements in embedded product development

Source: http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2006−12/msg01305.html

- *From*: "Paul E. Bennett" <peb@xxxxxxxxxxxxxxxxxx>

- *Date*: Thu, 28 Dec 2006 09:53:20 +0000

ssubbarayan wrote:

> Dear All,
> This question is slightly OT.Because this is going to deal with
> managing embedded product requirements then into involving in nittie
> grits.
>
> I am currently working on a consumer electronics embedded product.On
> every now and then when we are about to deliver, requirements keep
> changing.I have informed my top management and they say when customers
> need we need to give thats what we are paid for.While theres a
> obligation to deliver to what customers want,how to manage these adhoc
> requirements in a efficient manner?
>
> When should we take a call to freeze the requirements?At what stage you
> can say "Boss this point of time we cannot take any more
> requirements"?Most of the time we fail to say straight no because of
> fear of loosing the customer.Whats the best way of putting across to
> the customer to freeze the requirement?
>
> Can any one having good experience in managing requirements for
> embedded product share their experience ?
>
> Sorry if this question looks naive and amature.I am in learning stage
> of my project.Incase this is not the right group please forgive me and
> direct me to right group to address such issues.

There is no such thing as a dumb question, only dumb answers. I have already seen that Geoff, Tim and Vivekanandan have provided some quite reasoned answers pointing out it really is a question of how you manage your boss.
As well as following up on the Jack Ganssle links at <http://www.ganssle.com> and <http://www.embedded.com> you should also read a few other books on project management. One book that is well worth a read, "Software failure: management failure − amazing stories and cautionary tales" by Stephen Flowers [John Wiley & Sons ISBN0−471−95113−7],

because it reports on a number of projects that failed miserably and explores the reasons for those failures.

Vivekanandan's suggestion about ensuring you properly capture the customer requirements is a really good point. As I deal with the more industrial user I tend to have to learn a lot about my clients processes (I have worked mostly in Energy [Nuclear and Oil] and Transport [mostly Rail]. I have also worked for companies that supply to the food industry [a most fickle clientel]).

To capture the requirements fully I have even generated presentations to give to the customers which describes the approach we are planning to take in providing their solution and ensuring that they fully understand that approach (using lots of diagrams and simple demonstrations) then asking them to sign up to the technical specification. At this point the specification is frozen and the customer is made aware that any changes from then on will cost. Naturally, the earlier they request any changes the less expensive it will be but you should go through the presentation stage again and get new signatures on the updated specifications. It makes the Technical Specification contractual. Beware that the cost of change works both ways in this situation. Any change you make is your own cost and may involve penalties to be paid to your client.

The other aspect that you will need to your project management is to have a very clear and resilient change management process. You should know what the status of every single component of the system is at any given time during its development, whether or not it meets its specifications, how it performed during unit testing, has it met with client approval (if client inspection of testing was a requirement) which issue is being incorporated into the final product. The change management system should be monitoring changes not only to the software but the hardware, specification documents, data−sheets, certificates of conformity from suppliers, and the contractual documentation and correspondence between you and the client.

None of the above should prevent you from being fairly agile in your development process. In fact, if you use a more component oriented approach in your development you may find it easier to be very agile in development and thus minimise the cost of making changes.

−−
****
Paul E. Bennett .....<email://peb@xxxxxxxxxxxxxxxxx>
Forth based HIDECS Consultancy .....<http://www.amleth.demon.co.uk/>
Mob: +44 (0)7811−639972
Tel: +44 (0)1235−811095
Going Forth Safely ..... EBA. www.electric−boat−association.org.uk..
****
.