Re: Let's put this to rest

Source: http://coding.derkeiler.com/Archive/General/comp.object/2006−06/msg00274.html

- *From*: AndyW <foo_@xxxxxxxxxxxxxxx>
- *Date*: Fri, 16 Jun 2006 15:19:36 +1200

On Thu, 15 Jun 2006 10:19:02 +0100, "S Perryman" <a@xxxxx> wrote:

> "AndyW" <foo_@xxxxxxxxxxxxxxx> wrote in message
> news:pt6292pkq8tgd04tajqng44679vlqvrcp9@xxxxxxxxxx
>
> > On Thu, 15 Jun 2006 08:53:07 +0100, "S Perryman" <a@xxxxx> wrote:
> >
> > > "AndyW" <foo_@xxxxxxxxxxxxxxx> wrote in message
> > > news:s39192p6h62sr9vopidhhep9lc3tlos779@xxxxxxxxxx
> > >
> > > My question to this. How did you initially
> > > know that a stack was
> > > required.
> >
> > My answer to this : irrelevant to the discussion.
> >
> > 1. The "requirement" is to produce the "OOA" for a stack.
> > 2. Not to argue the toss about where the requirement may
> > have come from.
>
> I would suggest to you that the term analysis generally means to break
> something down into its parts. A requirement is simply a
> characteristc that must be implemented in the final system for it to
> be considered acceptable.

> Requirements Analysis is the process of discovering requirements. Usually by the breaking down of an existing system to determine its characteristics.
>
> Using that definition − you cannot already have an existing requirement for a stack − you havint performed the requirements analysis that determins that is a required characteristic. So it is not irrelevant to the discussion if you are using the terminology incorrectly.
>
> Obviously my words about 1 and 2 have been ignored. :−(

The reason I ignored it, is that you had in my mind made an incorrect assumption and to build a response to an argument based on an incorrect assumption would have been a bad thing to do. It would have just re−inforced what I thought was your incorrect view.

If you say that a requirement already exists for a stack, then by definition there is probably no need to do any formal analysis, because as I said above, analysis is the breaking down of a whole into parts [to see how they work] − but you already know how it works, because you just said there is a stack. Likely now you would just go and design some solution.

So it was a self defeating argument

However....

Iin the other part of the thread I suggested that requirements analysis was different from OOA (or normal analysis). In this case one performs an elicitation of requirements and builds a model of the problem space (as it exists).

So, one might have a piece of software, the requirement that was elicited from the customer is that the piece of software needs to be changed to implement a stack. One does not know 100% how the software works. So one builds an Analysis model of how the software works by breaking the original piece of software down into respective chunks, looking at the code, then building a model. One now has a requirement + a model of the existing system and can now procede on working out a desired solution.

In MDA there is the issue of platform independence, so both the analysis model and the solution model would likely be platform independant and contain a generic stack. When the solution is created for the target platform, one maps the generic stack to a specific stack.

Think of it this way, the original model may be created by a system architect, the solution may be done by a platform architect.

So

These two entirely different approaches are WHY it is so important to give 'a toss' of when the requirement came from :)

Now, in both solutions different models of stacks may be used. In MDA you have both the generic stack (think of a design pattern for a stack) and the implemenation solution (for which the model maps to). In the non−MDA approach, you simply have the implementation solution of the stack.

Thats how I understand things.

.