Subject: FMC
Posted by fudadmin on Thu, 08 Dec 2005 02:07:05 GMT
View Forum Message <> Reply to Message

http://www.f-m-c.org/quick-intro/

Subject: Re: FMC
Posted by mirek on Thu, 08 Dec 2005 13:12:53 GMT
View Forum Message <> Reply to Message

Actually, I always thought that those funny diagrams are the great way how to fetch more money from your customer, but do not really help to develop applications. But I can be wrong, I often am

Subject: Re: FMC
Posted by fudadmin on Thu, 08 Dec 2005 13:36:17 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 08 December 2005 08:12Actually, I always thought that those funny diagrams are the great way how to fetch more money from your customer, but do not really help to develop applications. But I can be wrong, I often am

I'm not a fan of UML etc. For me at the moment Doxygen is most useful. But you hit the point about making money ... I've just posted it for general knowledge...

Subject: Re: FMC
Posted by exolon on Fri, 27 Oct 2006 02:27:02 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 08 December 2005 13:12Actually, I always thought that those funny diagrams are the great way how to fetch more money from your customer, but do not really help to develop applications.
Yeah. It's only useful in design for rough sketches of ideas, say, which the different parts of the app will handle different tasks and how they interact with each other.

The real use, I think, is to document existing code so that others can come along and look at it without having to read and understand all of the code first. I know when I started working on quite big Java legacy projects, the auto-generated UML diagrams I made helped me understand what was going on better.

People who expect a rigid design to be implemented before diving into the code are probably going to be misguided... you're rarely going to know everything before you start working on it

So I'd think of UML as a documentation tool rather than a design tool/necessity. For me, design should be as limited in depth as requirements and user stories from XP planning.

Subject: Re: FMC
Posted by asif on Thu, 07 Aug 2008 20:01:12 GMT
View Forum Message <> Reply to Message

Hi,

I would like to respectfully add that UML is not just for diagramming. It's a complete design tool for object-oriented analysis, modeling and design. Just think about designing a greater-than-300-tables database without an entity-relationship diagram! Sameway, you are sure to hit design bugs if you try to write large object-oriented apps without expressing your design using UML. I am not talking about just using UML. The UML tool you use should allow complete forward and reverse engineering of code - that is, your diagrams should change your source and your source should change the diagrams. The higher you go design/architecture wise, the more you realize the importance of UML.

--
Asif

Subject: Re: FMC
Posted by mirek on Thu, 07 Aug 2008 20:17:35 GMT
View Forum Message <> Reply to Message

asif wrote on Thu, 07 August 2008 16:01Hi,

I would like to respectfully add that UML is not just for diagramming. It's a complete design tool for object-oriented analysis, modeling and design. Just think about designing a greater-than-300-tables database without an entity-relationship diagram!

Well, my infamous agenda project has over 600 tables And I never felt even the need to ER type diagrams.

I guess that once you are accustomed dealing with memory data structures, relation tables are not much different.

Quote:
 Sameway, you are sure to hit design bugs if you try to write large object-oriented apps without expressing your design using UML.

I would argue you are going to hit design bugs with UML too...

Also, IME, UML implies sort of waterfall model, which, in my mind, is flawed from consumer's perspective. I mean, it is in fact very good for programming company, but does not lead to solving the problem. I am strong believer in the iterative approach. The application does not need to be "design bugs free" as long as you have playground area wide enough to fix them.

Quote:
I am not talking about just using UML. The UML tool you use should allow complete forward and reverse engineering of code - that is, your diagrams should change your source and your source should change the diagrams. The higher you go design/architecture wise, the more you realize the importance of UML.

Quite possible.

It is also true that "regular engineers" likely need some approach like that. Me, solving some problem, I am thinking about it 24 hours a day (even when sleeping). In that case, all these schema pictures just look funny

Mirek
