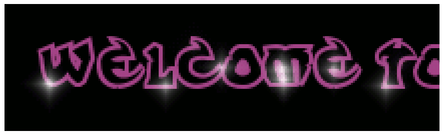

SOFTWARE ENGINEERING

What is the order of diagram drawing in a design?

Ask Question

7

5

I'm new with OOP and UML and I have some confusion here. I'd like to know where to start, I mean, somebody comes to you and ask you to do something (involves software design of course), once you have determined what has to be done, what is the order in which you have to start the software architecture? I mean, is it first the use case diagram or the class diagram? or should I draw some diagrams in parallel?

But basically, where should I start? and what UML diagram goes first? Thanks for helping!!

design object-oriented uml

edited Jun 11 '12 at 16:16
umlcat
2,012 9 15

asked Jun 11 '12 at 4:07
Manuel Malagon
36 1 2

P.S. I added "uml" and "design" tags so other people may look into your question. Besides, many people still use non U.M.L. design & techniques. – umlcat Jun 11 '12 at 16:21

A related thread: stackoverflow.com/questions/33229243/... – smwikipedia Oct 20 '15 at 6:17

3 Answers

9

There is no fixed order. It is all based on what you think is most useful to you/your team.

Initially, there should be some collection of requirements. If you start with SW architecture once someone tells you to do a software for X, then you already missed the most important step there.

During the requirements engineering phase, not all UML diagrams make sense yet. But some do make a lot of sense, in particular use-case diagrams. I have often seen use-case diagrams as the basis for discussions from which the requirements are then determined.

Similarly, state machines are already popular on the requirements level as well.

Once you have a set of requirements and actually start thinking about your architecture and later the design, more and more diagrams will start to make sense. Obviously, the class diagram comes in handy on lower levels of detail, but others, like sequence diagrams, may also make sense if you have complex sequences in your architecture.

Are you developing a distributed system, or something where components have to communicate a lot with each other? Try out the communication diagram.

Unless your process somehow requires you to use a specific UML diagram, there is no one forcing you to do so. Always remember that you do not model your software in UML for its own sake. If you don't feel like it'll help you, then simply don't do it.

Obviously, it takes some experience to decide this. I suggest you at least invest time to try out state machine, class, activity, and sequence diagrams, as those are the most popular. If you have enough time, take a look at the others as well.

answered Jun 11 '12 at 6:31

Frank
13.1k 2 34 64

*

4

Although I would agree that there is no hard and fast rule, I have come up with a rough outline of steps after reading Martin Fowler's book on UML.

1) Analyse the actors first (a uml diagram on the actors and how they interact with the system)

2) Use-Case Diagrams

3) Analyzing external factors that would impact a system (not really a diagram just including for information sake)

4) Analyzing risks in the systems and estimating their impact (again not really a diagram)

5) Sequence Diagrams (with data-flow in them)

6) Class Diagrams

From the book: "with a big system...It is easier ...to arrive at the list of actors first, and then ...the use cases for each actor."~Martin Fowler - UML Distilled

Recommend that book as a nice introduction to diagrams.

One more advice from the book: **its better to have a few diagrams that you keep updated rather than having lots of diagrams that are least updated**

answered Jun 11 '12 at 16:48

Imran Omar Bukhsh
1,619 14 25

*

Good suggestion, but first, one should define the objectives of the system. Also, Use-Case, Sequence and Class Diagrams could be built in parallel. Somewhere along the line the architecture of the solution should be documented before detailed design. – NoChance Jun 11 '12 at 17:38

@Emaad Kareem - yes ofcourse there are lots of other steps that I have not mentioned (e.g. feasibility study, requirements collection etc.) but I wanted to keep my answer to diagrams only as much as possible as the question is about diagrams only and not about the whole software life-cycle – Imran Omar Bukhsh Jun 11 '12 at 18:16

0

There is not a rigid rule, but, some guidelines.

Most developers / analysts will start, depending on the scenario.

A lot of analysts get started with the "use cases" diagrams, and eventually, add other diagrams, usually architectural diagrams, like packages or components.

And, then, with "Classes", "Activity", "Sequence" diagrams.

Cheers.

answered Jun 11 '12 at 16:20

umlcat
2,012 9 15

*