

# What essential design artifacts do you produce? [closed]

Asked 16 years, 3 months ago   Modified 1 year, 7 months ago

Viewed 11k times



2



**Closed.** This question is [opinion-based](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 6 years ago.

[Improve this question](#)

In the course of your software development lifecycle, what essential design artifacts do you produce? What makes them essential to your practice?

The project I'm currently on has been in production for more than 8 years. This web application has been actively enhanced and maintained over that time. While we have [CMMI](#)-based policies and processes in place, with portions of our practice being well defined, the design phase has been largely overlooked. What are the best practices?

[Share](#)[Improve this question](#)[Follow](#)

edited May 17, 2023 at 21:07

**Peter Mortensen**

31.6k ● 22 ● 109 ● 133

asked Sep 4, 2008 at 22:39



user290

## 6 Answers

Sorted by:

Highest score (default)



6



Having worked on a lot of [waterfall](#) projects in the past and a lot of adhoc and agile projects more recently, there's a number of design artifacts I like to create although I can't state enough that it really depends on the details of the project (methodology, team structure, timescale, tools, etc.).

For a generic, server-based 'enterprise application' I'd want the bare minimum to be something along these lines:

- A detailed functional design document (aka specification). Generally something along the lines of Joel s' [WhatsTimeIsIt example specification](#), although probably with some [UML](#) use case diagrams.

- A software technical design document. It is not necessarily detailed for 100% system coverage, but detailed in all the key areas and containing all the design decisions. Being a bit of an UML freak, it'd be nice to see lots of pictures along the lines of package diagrams, component diagrams, key feature class diagrams, and probably some sequence diagrams thrown in for good measure.
- An infrastructure design document. Probably with UML deployment diagram for the conceptual design and perhaps a network diagram for something more physical.

When I say document, any of the above might be broken down into multiple documents, or perhaps stored on a wiki/some other tool.

As for their usefulness, my philosophy has always been that a development team should always be able to hand over an application to a support team without having to hand over their phone numbers. If the design artifacts don't clearly indicate what the application does, how it does it, and where it does it then you know the support teams are going to give the app the same care and attention they would a rabid dog.

I should mention I'm not vindicating the practice of handing software over from a development team to a support team once it's *finished*, which raises all manner of interesting issues; I'm just saying it should be possible if the management so desired.

Share Improve this answer

Follow

edited May 17, 2023 at 21:15



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Sep 4, 2008 at 23:28



Ubiguchi

3,086 ● 1 ● 25 ● 24

---

Would your answer be the same for an Agile delivery method (e.g SCRUM), with JIT design? Anything you would add? Anything you would remove? – [mathewbutler](#) Aug 4, 2011 at 12:14

---



6

Working code...and whiteboard drawings.

:P



Share Improve this answer

Follow

answered Sep 4, 2008 at 22:41



Jason Bunting

58.9k ● 15 ● 104 ● 94





1



Designs change so much during development and afterwards that most of my carefully crafted documents rot away in source control and become almost more of a hindrance than a help, once code is in production. I see design documents as necessary to good communication and to clarify your thinking while you develop something, but after that it takes a herculean effort to keep them properly maintained.

I do take pictures of whiteboards and save the JPEGs to source control. Those are some of my best design docs!

Share Improve this answer

answered Sep 4, 2008 at 23:14

Follow



[Eric Z Beard](#)

38.4k ● 27 ● 101 ● 147



1



In our model (which is fairly specific to business process applications) the design artefacts include:

- a domain data model, with comments on each entity and attribute
- a properties file listing all the modify and create triggers on each entity, calculated attributes, validators and other business logic
- a set of screen definitions (view model)

However do these really count as design artefacts? Our framework is such that these definitions are used to

generate the actual code of the system, so maybe they go beyond design.

But the fact that they serve double duty is powerful because they are, by definition, up to date and synchronised with the code at all times.

Share Improve this answer

answered Sep 4, 2008 at 23:26

Follow



[Leigh Caldwell](#)

11k ● 4 ● 26 ● 31



1



This is not a design document, per se, but our **unit tests** serve the dual purpose of "describing" how the code they test is supposed to function. The nice part about this is that they **never get out of date**, since our unit tests must pass for our build to succeed.



Share Improve this answer

answered Sep 5, 2008 at 1:54



Follow



[Matt Dillard](#)

14.9k ● 7 ● 52 ● 61



0



I don't think anything can take the place of a good old-fashioned design specification for the following reasons:

- It serves as a means of communicating how you will build an application to others.
- It lets you get ideas out of your head, so you don't worry about tracking a million things at the same time.



- If you have to pause a project and return to it later you're not starting your thought process over again.

I like to see various bits of info in a design specification:

- General explanation of your approach to the challenge at hand
- How will you monitor your application?
- What are the security concerns and how are they addressed?
- Flowcharts / sequence diagrams
- Open issues
- Known limitations

Unit tests, while a fantastic and arguably critical item to include in your application development, don't cover all of these topics.

Share Improve this answer

Follow

edited May 17, 2023 at 21:17



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Jun 26, 2009 at 0:01



Ian Suttle

3,402 ● 2 ● 25 ● 28

---