***SOFTWARE DESIGN DOCUMENT FUNCTIONS***

Ask a fellow programmer what the purpose of writing a design document is, and the most likely answer you are going to get is "to let the programmers know what code to write". This is true, but there are several other reasons for writing this document, and they are by no means less important.

* **Communicating your design to others:**
  + **To implementers.**  
    Goes without saying. The programmers do have to know what code to write.

» In order to help them write efficiently, the design should be clear, complete and well defined.

* + **To fellow designers.**  
    Many designs, especially the ones describing larger systems, are given to fellow software designers to read and comment. They will ask questions, find bugs and mistakes and suggest improvements. This interaction is very important from the company's point of view - it will produce a much better design and save a lot of time and money in the long run.

» In order to optimize this interaction, you should specify the decisions you made in building your design and explain the reasons behind each one of them.

* + **To successors.**  
    It is most likely that at some point in time you will move on to other projects and let someone else take over and be in charge of your former design. This person will extend it, or modify it, as the requirements change and the system evolves. Your original design document will then be an invaluable resource of your design philosophy, intent and assumptions. Your successor will really need this insight in order to be able to expand the design without breaking it.

» Thoroughly explain your design philosophy and intentions.

* + **To managers.**  
    You probably have to report to someone. That someone needs to see the results of your work, either to evaluate your progress, or to gain a better understanding of the design and the resources it will require when implemented, or even to get an [INDICATION[http://cdncache-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.bitformation.com/art/writing_design_docs.html)](http://www.bitformation.com/art/writing_design_docs.html) of how good you are at what you do.

» Form carries just as much weight as content. The document should be carefully arranged and edited so that it would be aesthetic to look at. Additionally, the reader should be able to derive time estimates from the content.

* + **To guests.**  
    **New employees** usually go through a period of training during which they are given various internal documents to read in order to familiarize themselves with the company products. Your software design document is almost surely to be one of these documents. **Investors** sometimes come to inspect the company. One of the things they will want to see is the software design document.   
    During the initial stages of the project there is no finished product to show potential customers. Instead, it may be necessary to convince them that the future product is a viable solution to their problem. One of the means your company might use to do so is your design document.

» The document should be presentable. It should give the various readers a general view of the purposes and context of the project, taking into account the probability that not all readers will know in advance what the project is all about.

* **Processing your ideas and converting them into a solid design**  
  I find this to be a central goal in writing the document. When thoughts are put into writing, it is much harder to overlook problematic issues. This is particularly true if you do your best to anticipate every possible question and objection your readers may have.
* **Allowing for increased accuracy in time estimates**  
  At the beginning of a project, one of the first steps is to estimate the amount of work involved. This is a very rough estimate, and it is based on previous experiences with projects of a similar size. As such, it is bound to be off by a factor of two or more. As the work progresses, there is need to come up with more accurate estimates, and the best tool for this is a detailed design. Once you have the design document and you iron out all the problems and details you could think of without actually writing the code, you can give a fairly accurate estimate for every module and function in the design. Adding these up and allowing some margin for unforeseen problems will give you a rather accurate measure of the work involved.

» The document should reflect the size and complexity of every part of the software. It should also mention factors that may affect the duration of the coding phase, such as the use of new technologies, languages, tools or environments, or the requirement for a special skill set, etc.

* **Allowing for content adjustments**  
  In the course of virtually any software project, a moment arrives when it becomes clear that the required feature set cannot be developed within the given time frame. Normally, under these circumstances, the less important features are identified and removed in order to meet the (possibly extended) time frame. In some companies, this may be done without referring to the design document while in others the design document will be a major reference for this adjustment.

» The design document must elaborate on the dependencies among the various coding phases, and should attach a concrete cost to each of the project requirements.

.