**Component architecture**

Classes, stable relationships, related components, logical level and structure for descriptions.

**Process architecture**

Objects, dynamic relationships, coordination of processes, physical level and structure for execution.

**The three principles of architectural design**

*Define and prioritize criteria*

In short, you need some criteria to anchor the minimal quality of specific aspects within the system. These criteria can be things like speed, security, usability, etc. Since no company has the funds to perfect all criteria, they will need to be prioritized based on various factors.

*Build a bridge between the criteria and the technical platform*

The criteria set for a given architectural design should be grounded in the environment which the architecture is designed for. Furthermore, one should generally aim to ensure the architecture allows for effective use of the technical platform. No, I am not certain what this actually entails, because it’s written very vaguely in the book. Also, try to base your design on factual data as much as possible. Speculation has a high likelihood of being wrong, so basing decisions on speculations just mean you will have to rework half the project later on.

*Test the design as soon as possible*

You want to create as barebones of a program you can as soon as possible, and then test it. The prototype should give a general sense of what the program is supposed to do when it is completed. Testing on this prototype will reveal core problems with the design, but since it’s a prototype it can be easily rewritten to fix those faults from the get go, rather than having to develop the entire thing twice.

**Task 9.2:**

*Why is it important to define design criteria for every project?*

Criteria are set to ensure that the software follows good design, which is to say that it has no inherent weaknesses, and is easy to use (Usability, flexibility, understandable). Refer to page 174 in the book for a list of general criteria. Also helps provide oversight as to which parts of the software should be focused more on than others.