**Task 7.6-2:**

*What is the difference between a function and a usage pattern?*

Usage patterns define *how* a system is supposed to be carried out, while functions on the other hand describe *what* is supposed to happen. When functions were first conceived, they worked by taking some sort of data as input, and then transformed it to give some other data as output. Now-a-days however, functions have grown more complex and do shit like help people keep track of information such as in a nuclear m’fucking reactor.

**Task 7.6-3:**

*What are the 4 function types?*

**Updates:** Update functions are activated through flags from the problem domain (Shit that needs to be tracked), and are responsible for updating the overall system state. Or some parts of it.

**Signaling:** Functions which respond to state changes from update functions. These are used to either inform applicants of the application domain or make direct changes to the problem domain.

**Reading:** These functions react in response to an actor’s request for information. They don’t really do much other than fetching the info and display it.

**Calculating:** Basically a reading function, except it also takes input to transform into an output. Classic function types, basically.

Please note that a function does not necessarily need to consign to any of these types, nor does it need to fit into just one.

**Task 7.6-5:**

*Which sources can be used to identify functions?*

Use patterns, classes, and events.

**Task 7.6-7:**

*How do you evaluate the functional criteria?*

For complex functions, this can either be done by detailing the function as a formula (u = f(x)), an algorithm, or by splitting it into smaller functions.