The software development process that we will use in senior design involves a submitted weekly requirements/analysis/design/implementation/testing document entitle Weekly Deliverables. A copy of a great example of Weekly Deliverables from a team that was in senior design a few years ago is attached to this announcement. The example is from the middle of the second semester, so it contains lots of detail that won’t be in a starting project’s early Weekly Deliverables.

Please have a look at the example as soon as you can. When you do, you’ll see that the sections of the document have the following titles:

1. Intro

2. Technical Plan 3.Team Members 4.Delegation of Tasks

5.Components 6.Dependency Graph

7. Completion Schedule

8. GUI Prototype Changes ­ 4/14/2015 9GUI Prototype

Home Page Event Detail: Event Overview: Settings:

iOS Mockups 10. DB Design

11. Test Plan Version

If you think about it, some of the sections involve functional design, i.e., design of what the software being developed should do, and how it should do it AND some of the sections involve implementation design, i.e., technical details about how the functional design will be implemented.

Since functional design has to be done before implementation design can be done, it would make more sense for the sections about the former to preced sections about the latter; so, starting this semester all Weekly Deliverables’ sections will be ordered as follows:

1. Intro

2. Team Members

3. Delegation of Tasks

4. Completion Schedule

5. GUI Prototypes

Home Page Event Detail: Event Overview: Settings:

6. GUI Prototype Changes 4/14/2015

7. iOS Mockups

8.Technical Plan

9. Components

10 . Dependency Graph

11. DB Design

12. Test Plan Version

Note that our development process is not Waterfall; rather, any section can be updated, and even radically changed, as required because: (i) the project client has realized that changes to the functional design must be made in order for the software to satisfy his/her needs; or (ii) the team – or the team and, in the case of a tech savvy client, the team and/or the project client – realizes that the technical design will not work.

So…At the very start of each project, the only required sections are sections 1., 2., 3., and possibly 4. Since software must be designed before it can be implemented, most teams will then go on to developing the software’s GUI prototype; that is, because most, BUT NOT ALL, projects will require GUIs. (In the case of projects that involve mostly algorithmic design, the GUI is either far less significant, or, in a few cases, nonexistent.)

As far as adding/updating other sections is concerned: (i) most, BUT NOT ALL, projects that require significant GUIs should start adding GUI design to their Weekly Deliverables relatively soon, and should have a realistic version of the GUI prototype finished by ½ to ¾ of the first semester; the rest of the sections should be started earlier, and finished by the end of the first semester. Unless it’s unrealistic, for some reason, for your project, implementation and testing should be started in the first semester; (ii) for projects that involve mostly/entirely algorithmic development and implementation, GIVEN THAT SUCH PROJECTS ARE VERY DIFFERENT FROM ONE ANOTHER, algorithmic development should be started, and its documentation should be started, ASAP; implementation should, in most BUT NOT ALL, cases be started by the end of the first semester.

Because, as noted above, projects can be very different from one another, there can be no specific due date for any of a specific project’s aspects to be started. But, updating the Completion Schedule on a weekly basis, and using it to assign weekly tasks, is the best way to maximize the probability of a project’s success.