In this class we discussed user interface design, its position in the development lifecycle model, and ways in which we can structure our projects to best take advantage of our knowledge of human factors. The traditional point in time where user interface design occurs is indicated below.

What has occurred over time, however, is a realization of the importance of understanding usability. Sadly, the first approaches were implemented in a way that did not change our model of development, as shown below. The main problem with this is that if you test after the construction phase you are left with a tricky decision. Imagine that the system functions (it basically works), but that it is difficult to use. Should you spend the resources to fix the usability concerns? You might be forced to say no if the cost is prohibitive. In other words, you cannot increase usability through testing, in the same way that you cannot increase quality. You can only verify whether something is usable - or not.
There may be different ways of thinking of usability, but I am interested in pursuing measurement of usability in terms of:

- Effectiveness
- Efficiency
- Satisfaction

We did a quick exercise to see the diversity of design ideas that a small group can generate. Then, we discussed a lifecycle for user interface design:

Because you all have taken the HCI course, you probably know what each of these elements represents. The important thing is that the techniques all hang together in a cohesive manner. By the time you get to usability testing it is only verifying things, and hopefully all of the major discoveries occur earlier in the life of the project.

One of the frustrations we face with systems is that we sometimes find them poorly designed. We discussed the frustration from the developer’s viewpoint, where they probably did not set out with the objective of developing an unusable system.

A powerful technique we need to employ is that of generalization. Through the development of generalized models we recognize that we are designing for tasks, and not specific people. Sure, specific people do those tasks, but they may not always do them. We can generalize by job titles, locations, or even along task categories.

Although interface designs represent an important element of the overall design, they are insufficient in that they only point to external views. You will need to design internally how your system will operate. The interface designs can help guide this, and can be helpful in planning.

We discussed the tradeoffs of low fidelity and high fidelity prototypes. In class we had debates about this subject, and also on how innovative our designs should be.