M8.C1: Assignment 8: Prototyping

Sahil Mahendra Mody

CWID: 20007262

1) **Describe at least two advantages of using prototypes rather than documents for the self-parking car software we discussed in earlier assignments.**
   Prototyping is a relatively quick and inexpensive process that offers significant benefits for application development.
   1. Helps Gain Buy-In and Validate Ideas:
   If you're still in the process of trying to obtain budget for a self-driving car software, a great way to help your stakeholders visualize the software and show value early on is to start with a low-investment, low-risk prototype phase. You can put your prototype in front of real users to validate your ideas and designs to ensure the software fits a need and is feasible to build before jumping into development.
   2.Reduced time and costs:
   Prototyping improves the quality of customer specifications and requirements with the self-driving car software. Customers can anticipate higher costs, needed changes, potential project hurdles, and, most importantly, potential end result disasters with prototyping. Prototyping effectively can ensure software quality and cost savings for future years.
   3.Improved and increased user involvement:
   Most customers want to feel as though they are a part of the technical aspects of their project. Prototyping necessitates user participation and allows them to see and interact with a working model of their software. Customers can provide immediate feedback, request software changes, and change model specifications using prototypes. Most importantly, prototyping aids in the elimination of misunderstandings and miscommunications during the development process.

2) Describe at least two risks of using prototypes rather than documents for the self-parking car software we discussed in earlier assignments. What could possibly go wrong?

- **Insufficient analysis:** A focus on a limited prototype can distract developers from properly analyzing the complete project. The potential end result: A potential overlooking of better solutions, incomplete specifications or the converting limited prototypes into poorly engineered and developed final software that is hard to maintain.
- **User confusion:** Customers mistaking a prototype for the finished product is the worst-case scenario for any prototype. Customers who see a rough prototype may not realize it simply needs to be finished or polished. Customers may also mistake the prototype for accurately modeling the

performance of the final software. Customers may also develop a preference for prototype features that will not be included in the final software.

- **Developer misunderstanding of user objectives:** To ensure the success of this project, developers and customers must be on the same page and share the same project objectives. Customers who demand that all proposed prototype features be included in the final product can cause team and mission conflicts.
- **Excessive Development Time:** Prototypes needs to be developed quickly. If a developer spends too much time developing a complex prototype, this self-driving car software can run into roadblocks and run over both time and cost budgets.

3) Your consulting customers at DriverlessCars have asked you to build a prototype of a software module that would allow cars to change lanes at highway speeds simply by turning on the turn signal. Will you deliver a prototype or will you convince the customers that extensive documentation along with the prototype is a better plan? Justify your decision by describing the advantages, disadvantages, and costs of delivering just a prototype versus a prototype plus extensive analysis and documentation.

- Prototyping is a relatively quick and inexpensive process that offers significant benefits for application development.
- Changing lanes by simply turning on the turn signal appears to be a significant step in addition to a difficult task It is, at the very least, a very detailed task where developers would have to consider all possible edge cases as well as scenarios. In my opinion, it is best to use a prototype in conjunction with extensive documentation, owing to the module's importance more.
- Difference between prototype and document after seeing a working model is that a prototype would help the customer understand and clarify the requirements. However, documentation would assist developers in clearly stating and understanding the requirements so that there is no room for error.
- Prototyping and Documentation would be expensive, but the benefits like being cost and time effective far outweigh the drawbacks, which are simply increased costs and time.
- Advantage of both the methods: The customer can be certain that the functionality they have described is a viable feature and, if necessary, can come up with additional requirements to make the feature in good shape.
- Disadvantage of both the methods: The developers time and attention would have to be divided between prototype and documentation which would increase the amount of effort versus time, as well as cost.