Problem Set:

Prototyping plays a major role in DSDM. One of its main purposes is to replace documentation describing the final product with a working version of the product. That is, DSDM emphasizes the creation of software artifacts rather than documents in order to capture requirements and design intentions.

- Describe at least two advantages of using prototypes rather than documents for the self-parking car software we discussed in earlier assignments.
- 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?
- 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.

Solution:

1. **Describe at least two advantages of using prototypes rather than documents for the self-parking car software we discussed in earlier assignments.**

Prototypes are frequently employed in the design of physical products. They may, however, be constructed in a similar manner for software design and development. Many software engineers think that prototyping is a superior technique to seek the end design. For example, designing autonomous lane shifting software for an automobile manufacturer is a difficult task. However, creating prototypes based on the demands of the client may make software development more productive and efficient. The prototype model's core principle is to produce something to understand the customer's requirements. Based on current requirements, this prototype can be created. Customers may "experience" the system by using this prototype model. It can aid the client's comprehension of the intended product. Below are a few advantages of using prototypes over tradition documents:

- Customers have an active role in the development process.
- End users have a better grasp of the system that is being built.
- Bugs might appear early in the process.
- User feedback is useful in improving the final product.
- Functions that are difficult or ambiguous can be discovered early in the 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?**

Prototypes are extremely beneficial for both the developer and the client in terms of improved understanding. Creating a prototype model, on the other hand, might be hazardous. The following are some downsides or hazards associated with prototyping:

- Prototyping is a time-consuming and costly procedure. The creation of prototypes is sometimes hampered by a limited budget and time frame.
- The building system's technique can be shifted from planning to executing and finally repairing.

- The intricacy of the prototypes might sometimes alter the initial scopes of the system.
- One of the biggest hazards of prototyping is insufficient analysis. Developers may begin prototyping without first assessing the entire project.

When the system requires many contacts with end users, the prototypes model is particularly successful. Online systems and web interfaces, for example.

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.**

Identifying vacant parking spaces and parking correctly necessitates several customer contacts. This feature should be available through the software. Prototyping is the greatest technique to include end-users in this sort of program. The user may really feel the system and offer input on what they truly desire in the result. It is probable that our first prototype will not have all functionality or needs. However, it is critical to construct one since we may subsequently adjust the functionality based on the needs of the customer. Producing documentation in addition to prototype takes even more time and money. Due to time constraints, most projects have an extremely tight timeline. Furthermore, documentation has a relatively minor function in this sort of application for end-users. The ratio can be 80-20%, with 80 percent relying on prototype and 20 percent relying on documentation. I would work hard to produce a better prototype that could mirror the customer's specifications. Furthermore, end-user feedback would give encouragement to achieve more along the process.