

Object-oriented programming

[Design](#)



We know from the world of engineering that a manufacturer will develop a prototype model before mass producing the final product. So why not apply this concept to the development of software product? This is what the pioneers of prototyping have set out to achieve by attempting to demonstrate how a system or a component of a computer-based information system will function in its environment. Users find such demonstrations very helpful in visualizing what proposed systems will do for them.

They often find it difficult to accurately assess what they are getting from a system by reading large requirements specifications. Prototyping can result in a set of requirements that are better understood by the user and is more likely to be complete and accurate. Its advantages are that it is dynamic, flexible and provides a collaborative methodology that both aid the end user of the product and the development team. Some benefits of developing a prototype: Misunderstandings between software developers and users may be identified as the system functions are demonstrated. Missing user services may be detected. Difficult to use or confusing user services may be identified and refined. ; Software development staff may find incomplete and/or inconsistent requirements as the prototype is developed. ; A working, albeit limited, system is available quickly to demonstrate the feasibility and usefulness of the application to management. ; The prototype serves as a basis for writing the specification for a production quality system. Nice and Housemaster stated other uses: ; It can be used for training users before the production-quality system has been delivered. It can be used during system testing. The same tests can be applied to the prototype and the final system and results compared. Boone specified four stages in the prototype

development: Establish prototype objectives. Select functions for prototype inclusion and make decisions on what non-functional functions must be prototyped. ; Develop the prototype. ; Evaluate the prototype system, for rapid software development. Some of these management problems are: Planning, costing and estimating a prototyping project is difficult for project Procedures for configuration management are unsuitable for managers. controlling rapid change inherent in prototyping. ; Management pressures to reach swift conclusions may result in inappropriate requirements. Types of prototyping models Exploratory programming This method is based on producing an initial basic model that is shown to the customer for any comments, then refining the model until it reaches an acceptable system. This model may have an advantage for developers who have previously developed a similar system from which the initial model can be produced. Figure 1 shows the layout for the exploratory programming cycle.

Prototype Model Use the Refined Model Refine Model Check to see if the Model meets the Customer NO Requirements YES Final System Figure 1 Exploratory programming The system uses an iterative process to refine the initial model. The success of this approach lies in how rapidly the developer can process these iterations in order to Exploratory programming tends to result in a system that is not well defined. There will be a lot of 'change' going on within the software that can lead to errors being produced. This often results in problems with maintenance that can be costly and time consuming especially for large-scale systems.

Verification is also a problem as it checks the project against its original specification which is not clearly defined. Students will often start with this <https://assignbuster.com/object-oriented-programming-essay-samples/>

approach without realizing it. Given an assignment a student may start with a basic model and refine it by adding additional components until they have satisfied the requirements. More often than not this is achieved by accident and a tested analysis and design has not been adopted. Throwing away prototyping This method starts with an initial model, a prototype, which is checked with the customer to ascertain its correctness then developers start to build the final product from the beginning.

The prototype is thrown away. It is used as an aid to understanding and does not form part of the final product. Because it is only an experimental system all non-functional requirements can be ignored. The initial model or prototype may also have some of the functional requirements missing. The use of a front end graphical user interface can be ignored and systems that handle error recovery do not need to be included at this stage. Result of prototype Clear statement of requirements Figure 2 shows a typical layout for a throwaway prototype system.

The prototype is developed from an outline specification and presented to the customer to ensure they are satisfied with its functional objectives. The diagram starts with the prototype that is evaluated to ascertain a full system specification. Once this has been achieved the prototype is thrown away and work starts on the full version. Outline User Analyses Design Main System Specification Implementation Test Figure 2 Throw-away prototyping There are several problems with this method: 1) Often many important functions are left out to aid the speed of the development prototype. 2) We have seen with the traditional lifestyle that the negotiated statement of requirements can form the basis of a contract between the customer and developer. This <https://assignbuster.com/object-oriented-programming-essay-samples/>

model is incomplete at the prototype stage and therefore cannot be an adequate basis for such a contract. (3) It does not consider non-functional requirements. (4) The user has a restricted use of the prototype model which can lead to a false impression as to how the system will function. Having stated these problems this method does have its advantages, for example, incept of a prototype gives the customer an early view of the product.

As systems grow even more complex there becomes a problem in developing a full specification from scratch. It is expected that the concepts of prototyping will grow even though they introduce problems of their own. Reverse engineering This is not strictly a prototype method, but often comes under that heading in many texts. It starts with some completed code (or a complete system) that may be used as a base for a current project which is developed to meet the needs of the new

The main objective of this model is to use lines of coded that are already written and may be useful for the new application. The code is then reconstructed to suit the user requirements of the new system. This activity contains the essence of code reuse that is becoming more popular in current programming paradigms, especially in the field of object-oriented designs.