1. **What are the reasons of a successful and unsuccessful software project?**

Ans:-

Reasons behind Successful software project:-

 Consider End Users and Implement User Testing.

 2.) Have Clear Objectives and Specifications.

 3.) Hire Experienced Developers.

 4) Have a Plan.

 5.) Practice Effective Project Management.

 6.) Have Clear Communication.

 7.) Focus on Your Design.

Reasons behind Unsuccessful software project:-

* Lack of user participation
* Changing requirements
* Unrealistic or unarticulated project goals
* Inaccurate estimates of needed resources
* Badly defined system requirements
* Poor reporting of the project’s status
* Lack of resources
* Unmanaged risks
* Poor communication among customers, developers, and users
* Use of immature technology
* Inability to handle the project’s complexity
* Sloppy development practices
* Poor Project Management
* Stakeholder politics
* Lack of Stakeholder involvement
* Commercial pressures

1. **What types of problems may arise if a software project is developed on ad hoc basis?**

The meaning of word Ad-hoc is something which is not in order or not organised or unstructured. In the similar note the Ad-hoc testing is nothing but a type of [black box testing or **behavioural** testing](http://tryqa.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/).

## Characteristics

1. Ad-hoc testing is done after the completion of the formal testing on the application or product.
2. This testing is performed with the aim to break the application without following any process.
3. The testers executing the ad-hoc testing should have thorough knowledge on the product.
4. The bugs found during ad-hoc testing exposes the loopholes of the testing process followed.
5. Ad-hoc testing can be executed only once until and unless a defect is found which requires [retesting](http://tryqa.com/what-is-retesting/).

Disadvantage of ad hoc testing:-

1. Since ad-hoc testing is done without any planning and in unstructured way so recreation of bugs sometime becomes a big trouble.
2. The test scenarios executed during the ad-hoc testing are not documented so the tester has to keep all the scenarios in their mind which he/she might not be able to recollect in future.
3. Ad-hoc testing is very much dependent on the skilled tester who has thorough knowledge of the product it cannot be done by any new joiner of the team.

**3.Provide three examples of software projects that would be amenable to the waterfall model. Be specific.**

Ans:-

The waterfall model is amenable to the projects that focus on the attributes such as the data structures, software architecture, and procedural detail and interface characterization of objects.

1. **Provide three examples of software projects that would be amenable to the prototyping model. Be specific.**

Ans:-

Software applications that are relatively easy to prototype almost always involve human-‐machine interaction and/or heavy computer graphics.

Other applications that are sometimes amenable to prototyping are certain classes of mathematical algorithms, subset of command driven systems and other applications where results can be easily examined without real-‐time interaction.

Applications that are difficult to prototype include control and process control functions, many classes of real-‐time applications and embedded software.

1. **What process adaptations are required if the prototype will evolve into a delivery system or product?**

Ans:- If a prototype is evolved into a delivery system or product, it begins with communication. The software engineer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory.

The prototype serves as a mechanism for identifying software requirements. If a working prototype is built, the developer attempts to make use of existing program fragments or applies tools (e.g., report generators, window managers, etc.) that enable working programs to be generated quickly.

1. **Provide three examples of software projects that would be amenable to the incremental model. Be specific.**

Ans:- Each linear sequence produces deliverable "increments" of the software for example, word-‐processing software developed using the incremental paradigm might deliver basic file management, editing and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in the third increment, and advanced page layout capability in the fourth increment.

The process flow for any increment may incorporate the prototyping paradigm. Incremental development is particularly useful when staffing is unavailable for a complete implementation by the business deadline that has been established for the project.

1. **As you move outward along the spiral process flow, what can you say about the software that is being developed or maintained?**

Ans:- As work moves outward on the spiral, the product moves toward a more complete state and the level of abstraction at which work is performed is reduced (i.e., implementation specific work accelerates as we move further from the origin)