

Adaptive Vs Predictive

Scenario 1

A large organization wants to centralize their IT hardware distribution. Currently, team admins order computers one at a time from a list of approved models. The director of IT has asked you to implement a new system for ordering in bulk and distributing internally. Luckily, you have several years worth of historical data on available suppliers, hardware requirements, and usage within the organization.

Possible Methodology can be followed: **Predictive - WaterFall**

The requirements(Ordering and Distributing) are stable and well-understood, and the likelihood of change is low.

So I can follow waterfall methodology as follows

Analyze - Understands the Requirement and ask necessary questions, understands the budget for ordering and analyzing the timeline

Design - This phase includes planning - like which company is best to place the order and who would provide the best delivery and service. How many resources are needed to distribute internally.

Build - This phase is the actual execution phase, ordering and distributing

Test - Making sure all the business use cases are met and working as expected.

Deliver - Satisfying the exit criteria- meaning meeting all requirements and delivering the product without any issues.

Scenario 2

A restaurant chain wants a new way of interacting with their customers via a mobile device app. Marketing has already announced publicly that the new app will be available in six months, but some stakeholders are still arguing internally about which features are the highest priority.

Possible Methodology can be followed: **Adaptive - Agile**

Since the requirements are unsure and chances of change is most likely, we would follow Adaptive approach and I would like to follow Agile Methodology since the agile approach is optimized for change

Within sprints, the team will first prioritize to ensure that they are doing the work that will deliver the highest value to the customer which is creating a mobile device app for interacting with their customers.

The team will then do all of the work necessary to analyze, design, build, test, and deliver a working component of the product.

The agile life cycles account for potential instability in requirements or conditions and allow for regular change and reprioritization from sprint to sprint- meaning after the first or a few release of product to the customers, the team will gather all the feedback and make alterations to the mobile device app in accordance to the customer suggestions, so the change is expected and new patch will be release on the upcoming sprints.

The iterative and incremental nature of agile development helps the team harness change, which they know is inevitable. By keeping the sprints short while still delivering a working product increment each sprint, priorities can change without efforts being wasted.