

CUSTOMER JOBS

The intention behind developing the software from the view point of each user is as explained below.

- **Solution Architect as a Customer**

The main task on this class of customer is to design the overall architecture for the system being developed. The inherent problem faced by a solution architect is usually the large number of requirements and the need to analyze each of them. From the functional aspect, our solution would reduce time spent on sub-system design thereby creating more time for detail system design and integration plan. Additionally, the application reduces the work pressure thus enabling him to focus on.

- **Developer as a Customer**

As a developer, the major task is to analyze and implement user stories while having to deal with problems like little experience in coding standards, little/no knowledge in Design Best Practices, and time constraint for analyzing requirements. Our solution greatly reduces time invested in deciding the best way to implement the solution to the given requirement resulting in on-time implementation. As an added bonus, the developer also learns modern design pattern and gains experience in design and best practices.

- **Entrepreneurs as Customers**

Often, entrepreneurs have little/no experience about best practices. This software helps to provide an overview of modern design process thereby resulting in the creation of quality software architecture for a given idea. A direct consequence of this is a decrease in the initial investment which is of high importance to entrepreneurs.

- **Project Managers / Team Leads as Customers**

These customers want to decrease anomalies and increase maintainability in system. But, they face problems like blotted code/design and unmaintainable software due to lack/improper use of design patterns. Our solution indicates anti-patterns and enables early/short reengineering phase. The customer also gains design experience, learns modern design pattern and practices.

GAINS

- A well designed software results in scalability and lesser investment of time/ money.
- Eases the difficulty of design and development when an existing pattern can be readily applied.
- A quality design eventually results in lower costs of development as it avoids further rework and refactoring.
- A good software ensures the following key features: Good design, Quality, Performance, Cost effectiveness etc., and serves as an ideal for designers and developers for self-actualization.
- A software application that adheres to all features of being a 'good' software for a given set of functionalities, may catapult the organization/individual owning the software to financial success and fame.

PAINS

- Too many requirements exist before the development stage commences. Each requirement needs sufficient time to be addressed individually.
- Insufficient experience/no experience in matters of effective software design, coding standards and systematic development.
- Enormous amount of time required to design a software without a definite pattern: Time is money.
- Expensive to develop software without a predefined pattern as this may result in unnecessary rework and refactoring.
- A badly designed software is bound to be non-scalable and would result in re-development from scratch thus proving to be unreasonably expensive.

PRODUCTS AND SERVICES

- The application provides services to users from multiple viewpoints. From a buyer's perspective, the application supports the user in deciding which pattern is most applicable for the current scenario. In addition to the suggested pattern, the application also provides a sample blueprint based on which the solution can be implemented which is helpful from the viewpoint of a co-creator. Additionally, the application also helps in the detection of anti-patterns which helps in the refactoring of existing solutions from the viewpoint of a transferor.
- The most important outcome from the application is from the viewpoint of buyers and transferors. It acts as a decision maker to decide on the right pattern. It acts as a pain reliever by identifying the deviations from standard practices. Additionally, the blueprint of the patterns acts as a reference tool to aid solution implementation. The deliverables are crucial in order to ensure adherence to standard SDLC.

GAIN CREATORS

- The application enables efficient time management and reduces overall development effort
- It also helps in the design a high quality software that adhere to standard SDLC.
- It enables greater maintainability by reducing risks involved during refactoring and rework of design
- There is reduced overhead between solution architect and developers. Both use the application that provides high quality solutions.

PAIN RELIEVERS

- The application produces significant savings in the form of time, money and effort.
- It also reduces work pressure on the users and helps them gain design experience
- It reduces the need to come up with patterns manually and hence eliminates some risk
- The users can focus more on innovation and solving domain specific problems rather than having to spend time on design related decisions.

USE CASES**Case 1**

Brief Description	Application can be used as a support tool in order to design high level architecture
Actors	Solution Architects
Preconditions	Requirement Document from the Requirement Elicitation phase
Basic Flow of Events	The user answers a set of questions. The answers are internally mapped against a set of business rules in order to suggest the best possible design pattern
Post Conditions	The output will be the best possible design pattern for the given use case. Additionally, a sample diagram of the particular pattern will also be shown which will serve as a point of reference

Case 2

Brief Description	Application can be used as a support tool so that the right design pattern is used during development process
Actors	Developers
Preconditions	User Story or a task which has been assigned to the developer
Basic Flow of Events	The user answers a set of questions based on the user story/task which has been assigned
Post Conditions	The output will be the best possible design pattern for the given use case. Additionally, a sample diagram of the particular pattern will also be shown which will serve as a point of reference for the development

Case 3

Brief Description	Application can be used as a support tool to ensure a stable design
Actors	Entrepreneurs with relatively less or zero experience
Preconditions	Requirement Document from the Requirement Elicitation phase. In case of absence of a Requirement Document, a concrete idea which can act as a substitute for the document
Basic Flow of Events	The user answers a set of questions based on which Requirement Document or the idea
Post Conditions	The output will be the best possible design pattern for the given use case. Additionally, a sample diagram of the particular pattern will also be shown which will serve as a point of reference for the development

Case 4

Brief Description	Application can be used as a support tool to detect anti-patterns in existing solutions
Actors	Project Manager / Team Leads / Developers
Preconditions	Anomalies in the behavior of an existing system. Also, difficulties in development or maintenance which could also indicate presence of anti-patterns
Basic Flow of Events	The user answers a set of questions related to the anomalies/difficulties being experienced
Post Conditions	The output indicates possible anti-pattern being present. Additionally, corrective measures will be suggested