**Team Project - Journal**

**Simplicity**

* 17th September 2016

In the of practice of Extreme Programming , Simplicity plays a very important role.

Simplicity ‘simply’ means that “concentrate on today’s work”. It follows an approach in which - in first phase, a team focuses on building a Simple solution probably because of factors such as, requirements might change or there are two many unknowns at first.

More functionalities can be added later on when need arises.

Implementing Simplicity in the project can also improve communication, as simple design and coding will definitely be more convenient to communicate among the development team.

Additionally, as at the start of the project, no one is sure that which requirements will be essential and which will be extraneous, so using simplicity we can follow “You aren’t gonna need it”(YAGNI) approach. Which state that programmers should not add more functionality to the project until it’s deemed necessary. Because requirements might become irrelevant and if so all the time and money spent on the implementing that will go to waste.

As we are at start of our project, with a team of five people who don’t know much about each other’s technical skills and depth and breadth about the subjects, i believe that keeping a project simple at the start can lead to greater good and efficient solution. As we are finalizing our project, there are many unknowns, we haven’t decided yet that how we can expand our initial solution. So it will be better that we start with simple solution of our project, which will give us not only confidence but also the facility of expanding the projects by incorporating additional features later on like adding an Object Oriented approach, design patterns et cetera.

In our first meeting, we have all decided to follow approach of Simplicity by designing and coding in a simple approach, covering requirements in phase by phase manner.

These are key values of Simplicity which i learned from variety of sources and i along with the team surely try to incorporate Simplicity Value into our programming so that we can follow best practices of Extreme programming.

- 22th September 2016

On the 20th September, we had a project meeting to discuss the project requirements, scope, design and work flow. We all members presented our ideas regarding the Finite State Auto meta – Treasure Hunt game implementation. We all agreed upon the idea of developing a Simple Treasure Hunt scenario with few island each having two paths to different island, a user controlled Ship which travels to specific island based on the user’s decision.

We will developed and deploy the initial-simple solution first, and after iteratively will continue to add more functionalities to it. I think there are two benefits of following this methodology. Firstly, it will provide simple and easy to understand design and code. All team members can easily understand and review each other’s code, as the design is simple. Secondly, it will be easier to implement new requirements or change requirements as only a portion of the code will need to be altered. Instead of removing a total module and having to rewrite it completely.

We have also discussed, extended scope of the project by making the game more interactive, but we will progress iteratively and will add new requirements as necessary, so that we can maintain simplicity.

In this week’s meeting we decided to implement a simple working model of the game, with two islands each having routes using which a user can travel from one island to another.

I created a sample project in the greenfoot accordingly. I have implemented the basic logic on which we can initiate a trip when user clicks on a particular ship. Provided that the ship is on that island i.e user is currently on that island. Whole team is currently working on that sample project, to do necessary changes and create classes with attributes and behaviors in it.

I believe, the simplicity can be very helpful in the big projects as you can easily generate a prototype to work on using simplicity and can use is to implement more advance features later on.

Following next week, we will be focusing on extending the sample and implementing the whole solution as a prototype of the project.