**Daniel Kifle**

**ID#: 108589**

**Mindsets Between Application Developers And Framework Developers**

Apart from writing codes, framework development is completely different from application development as we need to view problems in completely different perspective, think about solution with boarder range and most importantly provide support for different application development at same time making it simple to use.

Code reusability, granularity and simplicity comes with a good design of frameworks. However, to achieve all the advantages, designing a well-functioning framework is crucial. This raises a question into what considerations should be taken in designing frameworks rather than simply developing an application.

During my working days as developer, all I need to think about is what need to be done. Then I think how it can be done and then try to find more better solution and finally implement it. But as framework developer, I needed to think more than just a better solution. I also needed to think how framework will be used and then again how it can be reused. As application developer, I never expect by code to be reused but as framework developer, it was one of my top priority. The knowledge of design pattern becomes handy while developing framework as I always focused in designing solution first rather than developing it.

One of the main challenges in designing a framework is predicting of all the possible common features that applications come up with and trying to make the framework support those functionalities. Each type of application would have its own major requirements on the framework and the desired behavior for one may be impairment to the others. From the very beginning of designing a framework, a critical thinking and analysis is required so as the framework should be able to accommodate one or more of the application types that need to be built over. Inversion of Control (IoC) is one of the main features in frameworks that follows the Hollywood Principle (Don’t call us, We will call you). This architecture enables application processing steps to be customized by event handler objects that are invoked via the framework’s reactive dispatching mechanism. The framework’s dispatcher reacts by invoking hook methods on pre-registered handle objects, which perform application-specific processing on events. Another difference lies then in this case where inversion of control allows the framework (rather than each application) to determine which set of application-specific methods to invoke in response to external events.

Developing an application, I need to think about User experience which can be solved by implementing proper design, features, following trends and working on feedbacks. But as framework developer we have to think how developer will use our framework and design whole architecture based on it.

As developer, I only need be to concern if there are any bugs and other errors in particular scenario or not. But as the framework developer, I had to be more alert for any error that can occur in framework as any error may break application code that uses it. Also, the test case need to cover much more scenarios than that of application development.

In application, all component that are added are features and we are concerned on how to add whereas in frameworks components are the toolset and we have to be concerned on what to add so that it will be beneficial addition to the framework.

With many good and valid ways to approach and solve the given problem, I found framework development more subjective and difficult to pin down best idea. On the other hand, with straight forward and few choices to solve given problem software development is much more objective.

As application developer, first I used to research about appropriate pre-designed and tools that boost my effort in developing the required software. So, I already have template to kickstart project so I only need to think about the specific business logics. But while developing framework, I didn’t have any pre-built tool, I have to start everything from start.

As a framework needs to be developed to ensure that many programs can be developed in a consistent, efficient and accurate manner by developers with different skills, we thought in generic way to solve problem. I found framework development to be much more advanced than application development