SE 339
Assignment 1 - Emergence of Architecture

Stamati Morellas

Due: 9/16/19

Part I:

In general, this article addresses the debate of whether or not small refactoring in agile development practices establish a strong and effective software architecture. Some of the categories of factors that were found to impact the emergence of architecture through refactoring are project, organization, practice, and team-based contexts. Some of these include design principles, team size and skill, culture, organizational structure, rate of change, the size of a project, etc. The study concluded that different people have different views on the question that was trying to be answered, however with the study's findings on the twenty contextual factors that influence software architecture in agile development show that if used accordingly, one will be able to make informed decisions about how they want to run their agile development processes with the hopes of architecture emerging from small refactoring as a result.

Part II:

(a) Satisfactory architecture emerges from refactoring by looking at potential poor architecture solutions in the code and making a better architecture.
(b) In cases of failed large projects where the architecture did not emerge from refactoring.
(c) Emergence of architecture through refactoring is rendered unlikely through the identification of key non-functional requirements at later stages of development and through a high rate of requirement changes

Part III:

(a) There are many factors that can impact the emergence of architecture. For example, system age plays an important role because more recently designed software systems are more likely to be amenable and to gain structural and behavioral integrity through refactoring than older ones.
(b) Maturity helps emerging architecture through refactoring because a more mature team of developers will have the necessary experience to foresee future architectural patterns as they emerge by having relative vision on potential evolutionary paths of the architecture they started with.

(c) Bad communication could lead to the failure of addressing important information to the rest of the team that could greatly impact the emergence of architecture through refactoring.
(d) Management support is important to the emergence of architecture through refactoring because architecture transformation becomes a priority and the team will have enough time to look at the system and change its architecture.

Part IV:

I believe this article emphasized the significance of always keeping software architecture in the back of your mind when practicing agile development. Although there is a lot of various factors to consider, the fundamental principle of making small periodic changes to develop an architecture is very beneficial in the long-run. By doing this early on, teams will be able to not only save time, but they will also decrease the risk of developing an unusable software as an end result.