

COM S 339 Assignment 1

Alec Meyer

February 9, 2021

Refactoring Architecture

The idea of satisfactory architecture emerging from refactoring sprouts from the use of AGILE development practices. AGILE is one of the most common software development practices and is used widely throughout the tech industry. I am going to discuss the downfalls to producing software through constant refactoring and how it can fail a projects needs.

Satisfactory architecture emerging from refactoring is a large debate in the sofwtare architecture community. The idea is that a project can be developed without a traditional architectre through refactoring and editing code. This claim is what AGILE development depends on. This ideology allows engineers to disregard the future of the projects architecture on the basis of refactoring any issues in the future.

The architecture developed from refactoring is loose. "Failure [architecture did not emerge] is also more prevalent in cases of large pre-existing

products which either had no discernable architecture to begin with or had their architecture erode away over years of maintenance and haphazard addition of new features by persons who didn't understand the pre-existing architecture or ignored it for one reason or another"

One way to look at AGILE strategies is that it is a development pattern whos organization and structure is based on its lack of structure. This makes it unlikely for an architecture to be made through refactoring. Refactoring doesn't allow a structure since it is a strategy based on the in-the-moment development. Thapparambil stated "no agile methods discuss Architecture in any length".

35% of participants indicate factors under projects. Change/size/type/
Maturity of AK/System Age/type of ASR/Criticality

Since there is not a set architecture in a refactoring based development it is necessary that new changes are communicated. "They [the team] were used to work heads down in their cubicles for months without speaking to anybody. After that time they simply deposit hundreds of pages of useless diagrams and felt good about it. [...] the sense of responsibility that comes with Agile is not there