Topic: Test Driven Development

Test-driven development (TDD) is a method of implementing software programming that interlaces unit testing, programming, and refactoring on source code (Heusser, 2017). TDD is an Agile software development model first introduced as part of the Extreme Programming methodology. TDD argues that the most effective way to code is to first create unit tests, then write just enough code to satisfy them. The developer will then go back and refactor to make the software more efficient and clearer, without changing the behavior. TDD evangelist argue that it requires developers and testers to accurately anticipate how the application and its features will be used in the real world (Heusser, 2017). This results in higher quality work in a shorter time span. Critics argue that the skill required to successfully implement TDD cannot be consistently maintained, while decrying it as impossible to use whatsoever for legacy systems.

TDD works best in well-trained teams working on new development. It can struggle when team members lack the required skills to create unit tests, or clean code. It is also nearly impossible to implement on a legacy system not already using TDD. Despite those flaws, TDD has made its way into many teams, which all argue that the benefits far outway the potential issues.

Reference:

Heusser, M. (2017, May 25). What is Test-Driven Development (TDD)?: Definition from TechTarget. Software Quality. <https://www.techtarget.com/searchsoftwarequality/definition/test-driven-> development

Response 1:

Anitha,

TDD methodologies are deeply connected with Agile and Extreme Programming paradigms. Organizations interested in an Agile software development approach should consider adopting TDD as a methodology specifically suited for that process.

Response 2:

Christopher,

I think it is common for most organizations to pick and choose aspects of a theoretical process. I see that all the time in “agile” companies, which have no interest in sprints or tracking points. Hybrid approaches are very common and I am not surprised you have worked with hybrid TDD approaches.