M1.B7: Assignment: Social Media 1

Name: Sahil Mahendra Mody

CWID: 20007262

1) **URL:** https://dzone.com/articles/reverse-pull-requests
2) **Description:**

Title: Reverse Pull Requests

In today's world, more teams are implementing microservices architectures and shifting to Continuous Deployment and Trunk-Based Development. With pair programming, no feature branches, and such continuous change, code reviews would appear extremely difficult, with limited tooling support. In addition to code reviews, GitHub's Pull Request review features played an important role in the process using an approach we called '**Reverse Pull-Requests.**' The product by one of the teams was a set of APIs used by other apps in the business and the team was made up of developers with a wide range of experience. The team had made a decision to maximize flexibility by using a single Git repository per microservice, overcoming the challenges with repository automation. It was working very smoothly and even delivering business faster than any other company like the team was able to change the course of development on request immediately. But something was missing which was wider visibility of changes being applied to code. This resulted in becoming inconsistent projects in terms of design patterns, code conventions and code quality but how could we improve this thing? Code reviews was an option but the challenge was how to do code reviews continuously and we knew we can't work from a feature branch because it would destroy the pace, we were able to work at. So, the solution of code-reviews failed and they came up with another approach called reverse pull-requests. The working of reverse-pull requests has been explained. As much as we find reverse pull requests useful there were limitations such as hard to separate the changes made by one pair from another while working on the same codebase. As the team implemented the reverse pull request solutions, it boosted the code quality, knowledge transfer and consistency. It also improved overall design in a way that was non-blocking. For these reasons, the team decided to retain the reverse pull request approach. This can be concluded by saying that the team found out that even after having limited applicability in the right environment, the reverse pull requests can be a better solution than code reviews as it can boost the code quality, knowledge consistency and can improve the overall design as it can make the code led in a state that was easy to understand and change in future.

3) **Recommendation:**

Yes, I would like to recommend this article as this article states the importance of reverse pull requests and how can it be useful in projects. It also showed how it overcame the limitations of code reviews, the working model of reverse pull requests, its advantages and limitations and even with the limitations why it is better than code reviews.