



TCR: A More Divisive TDD

Where TDD enforces testing, TCR enforces TDD.

Kent Beck has had a substantial influence on my career and workflow. Pair programming, TDD and XP have been essential tools for me a long time developer.

I'm catching up on Mr. Beck's latest process innovation TCR. [test && commit | | revert. As part of Limbo on the Cheap, we... | by Kent Beck | Medium](#) will best describe the process in his own words.

What I want to comment on is how weirdly divisive these tools have been on both sides. I expect TCR to be even more radical to some.

(Note: TCR reminds me a bit of Chad Folwer's POV on microservices and branches: microservices are not precious and **should** be rewritten and unmerged branches **should** be deleted at 5pm: [From Homogeneous Monolith to Heterogeneous Microservices ...YouTube · GOTO ConferencesJul 13, 2015](#))

I've had the chance to work with effective and talented developers on both sides: those who see no value in testing or pairing, and those who are dogmatically attached to them.

I've never fully understood either of these POV. I tend to think of TDD and pairing as tools in a tool box. I think I'll put TCR in the same toolbox.

These are tools to apply at the correct times, when the costs (there are costs) are worth the benefit. It takes experience to know when to use them.

The other large piece to consider is how well setup your codebase is setup for these practices and how comfortable developers are with these tools. Pairing in a team with a high number of projects per developer has a larger cost than pairing in a team where devs aren't trading time on 'their' projects for 'your' project. Similarly if your first test requires setting up a test harness for a test on non testable code, will require significantly more effort than if testing is already part of the workflow. The scale of projects and audience and criticality also weigh in heavily in the 'is a test worth it?' equation.

Development processes are subtle and nuanced, and it's easy to get opinionated and geared about the "right way," in either camp. Nearly, 40 years later, No Silver Bullet by Fred Brooks (<http://worrydream.com> > [refsPDFNo Silver Bullet – Essence and Accident in Software Engineering](#)) is as relevant as ever.

TL;DR: I'm looking forward to playing with TCR but I wouldn't demand it of anyone. Team alignment is the most important factor (<https://medium.com> > [@jessitronSymmatheicist \(n\). A quick definition, without the... | by Jessica Kerr | Medium](#))

a labor of  by jsk  © 2022-2025

Tags [Archive](#) [RSS feed](#)

Made with [Montaigne](#) and by [anton](#) 