Test Driven Development: Is It Worth It?

Feel free to do your own web searches for points and counterpoints to using TDD or other testing methods. There are many resources for you to research and use to help shape your own opinion. The biggest challenge is finding articles and papers based on data over gut feeling. We've collected a few documents here that we feel give a good supporting case for TDD. We've also provided some compelling counterpoints or critiques of certain details.

Ultimately, of course, it's up to you to decide on your testing approach. In general, more testing of just about any sort leads to better code, more peace of mind, lower overall costs, and other often surprising benefits. The techniques we illustrate are flexible enough to allow you to do what you feel is right. If you're on this journey with us, we only ask that you consider going "all in" on the examples and homework so you absorb everything before deciding what bits to keep and which to throw out. As Donald Knuth said, "Premature optimization is the root of all evil (or at least most of it) in programming."

Why TDD is Worthwhile
- "An Initial Investigation of Test Driven Development in Industry"
- "Realizing quality improvement through test driven development: results and experiences of four industrial teams"
- "Empirical study of Software Quality Evaluation in Agile Methodology Using Traditional Metrics"
- "Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?"
- "Empirical Software Engineering"

Counterpoints
- "Does Test-Driven Development Really Improve Software Design Quality?"
- "TDD is dead. Long live testing."
- "TDD leads to an architectural meltdown around iteration three"
- "Why Most Unit Testing is a Waste" [pdf]