

TDD

Test Driven Development

XP Values...

1. Communication
2. Simplicity
3. Feedback
4. Courage
5. Respect

Also: Safety, Security, Quality of Life,
Predictability

XP Principles...

Flow, Opportunity, Redundancy,
Failure, Quality, Baby Steps, Accepted
Responsibility, Humanity, Economics,
Mutual Benefit, Self Similarity,
Improvement, Diversity, Reflection

XP Primary Practices...

Weekly Cycle, Quarterly Cycle, Slack,
Ten Minute Build, Continuous
Integration, Sit Together, **Test Driven
Development**, Incremental Design,
Whole Team, Informative Workspace,
Energized Work, **Pair Programming**,
Stories

TDD IS DESIGN!!!

- Devs write **automated** tests as they go
- Tests are written in **advance** of code
- Design **a little** at a time

Upside

VS

Downside

- Simpler/Less code
- Maintainable code
- Cleaner API
- Safety net
- Documentation
- Drives pairing

- Hard to learn
- Slower at first
- Can still TDD the wrong thing

Automated tests

- Automated
- Test a single behavior (not method)
- Intention revealing (tell a story)
- Repeatable
- Have no side effects
- Are production code

Unit tests

vs

Integration tests

- Fast
- Small
- Run in-memory

- Slower
- Can cross boundaries
- Can depend on external resources

Useful tools

- Package manager (NuGet)
- Test framework (NUnit)
- Test runner (TestDriven.NET/Resharper)
- Mocking fx (Rhino Mocks)

Useful patterns

- Red-Green-Refactor
- AAA Syntax
- Builder Pattern

8 Useful guidelines

- Get end to end feedback asap
- Make test green asap then refactor duplication
- All tests pass all the time - no broken windows
- Try to never debug
- If its not important for the test, it is very important not to show it
- It's ok to temporarily suspend design judgment to discover more about the domain (DANGER!)
- State vs Interaction tests
- Mocking more than 3 things is probably a design smell

Contact me...



kevint@drivensoftware.com

[@kevintrethwey](https://twitter.com/kevintrethwey)

www.drivensoftware.com