Five XP Practices for Agile Development

Welcome.
Thank you!

http://BeyondLegacyCode.com

David Scott Bernstein

- Software developer since 1980
- Trained 8,000 developers since 1990
- Published author since 2015
- Website: http://ToBeAgile.com
Nine Essential Practices

1. **Say What, Why, and for Whom before How**: With a Product Owner defining the next most important features to build, the need for upfront requirements goes away.

2. **Build in Small Batches**: Building incrementally increases feedback, helps simplify the construction of complex systems, and reduces risks.

3. **Integrate Continuously**: Sets up the infrastructure for incremental development.

4. **Collaborate**: Spiking, pairing, and swarming as a team to solve problems and increase knowledge dissemination.

5. **Create CLEAN Code**: Shares standards and practices for building software with code qualities that support testability.

6. **Write the Test First**: Drops the cost of building and maintaining software dramatically.

7. **Specify Behaviors with Tests**: Uses tests to define and document behaviors.

8. **Implement the Design Last**: Paying technical debt can pay back dividends in the short term as well as the long term.


---

**Practice 3**

Integrate Continuously
The Heartbeat of a Project

Three Kinds of Done
Continuous Deployability

Automate the Build
Integrate Early and Often

Take the First Step
Practice 4

Collaborate

Extreme Programming
Communicate and Collaborate

Pair Program
Time Box Unknowns

Code Reviews, Retrospectives
Amplify Learning

Be Mentoring and Mentored
Practice 5

We can infer good development principles and practices through five key code qualities:

- Cohesive
- Loosely Coupled
- Encapsulated
- Assertive
- Nonredundant

Create CLEAN Code

Quality Code is Cohesive
Quality Code is Assertive

Quality Code is Non-Redundant
Code Qualities Guide Us

Increase Quality Today…
Practice 6

Write the Test First

The Things We Call Tests
Quality Assurance

Testing

- ✔️
- ✔️
- ✗

Write Good Tests
Positive Feedback (A Virtuous circle)

TDD Gives Rapid Feedback

TDD Supports Refactoring
Write Testable Code

TDD Can Fail
Introducing TDD to a Team

“Warning: Promiscuous pairing can lead to becoming test infected.”

Become Test Infected
Practice 9

Refactor Legacy Code

Investment or Debt?
Become a “Deadbeat”

Time for Change

When Code Needs to Change
Refactoring Techniques

Refactor to Accommodate Change
Refactor to the Open-Closed

Refactor to Support Changeability
Thank You!

- We have just scratched the surface, to learn more:
  - Read my blog: [http://ToBeAgile.com/blog](http://ToBeAgile.com/blog)
  - Sign up for my newsletter: [http://ToBeAgile.com/signup](http://ToBeAgile.com/signup)
  - Follow me on Twitter (@ToBeAgile)
  - Attend my one of my Certified Scrum Developer trainings
    - See [http://ToBeAgile.com/training](http://ToBeAgile.com/training) for my public class schedule
    - Or contact me to arrange a private class for your organization
  - Visit [http://ToBeAgile.com](http://ToBeAgile.com) for more information