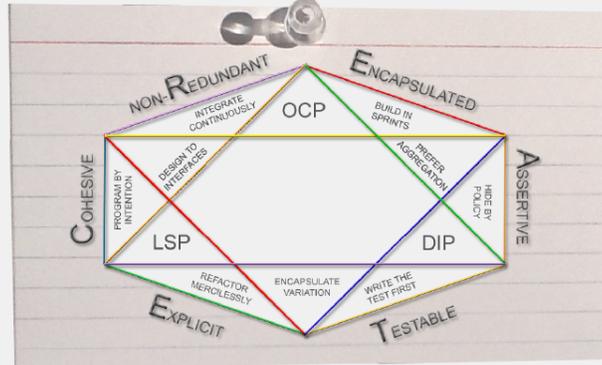


### Five XP Practices for Agile Development



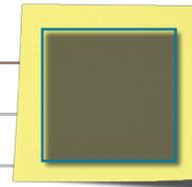
David Bernstein – To Be Agile (<http://ToBeAgile.com>)

<http://ToBeAgile.com>  
[info@ToBeAgile.com](mailto:info@ToBeAgile.com)

© Copyright 2012-2018 To Be Agile

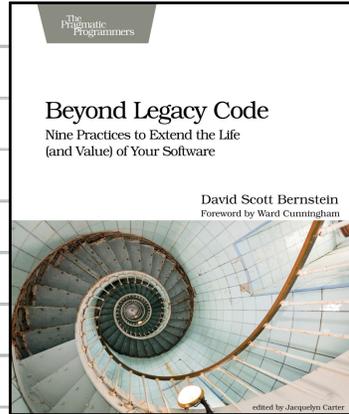
DB20180610

## David Scott Bernstein



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

## Book – Beyond Legacy Code



<http://BeyondLegacyCode.com>

- Nine practices to design and build healthy code, plus some tips on dealing with legacy code.
- Discusses the value and reasoning behind the technical practices, so both managers and the team can get on the same page as to their value.
- It's not a "How To" book, it's a "Why To" book.

## Some Questions for You



- How many of your teams are doing Scrum?
  - More than a year? Two? Five? Seven?
- How many of your teams are doing XP?
  - More than a year? Two? Five? Seven?
- Which technical practices?
  - CI, TDD, refactoring, pairing?



*"This book shows readers how to use SCRUM, an Agile software development process, to quickly and seamlessly implement XP in their shop-while still producing actual software. Using SCRUM and the Agile process can virtually eliminate all downtime during an XP implementation."*

**"Agile Software Development with Scrum"**

-- Ken Schwaber and Mike Beedle

## 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 radiate knowledge throughout an organization.
5. **Create CLEAN Code:** Share 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.
9. **Refactor Legacy Code:** Incorporate learning and pay off technical debt.

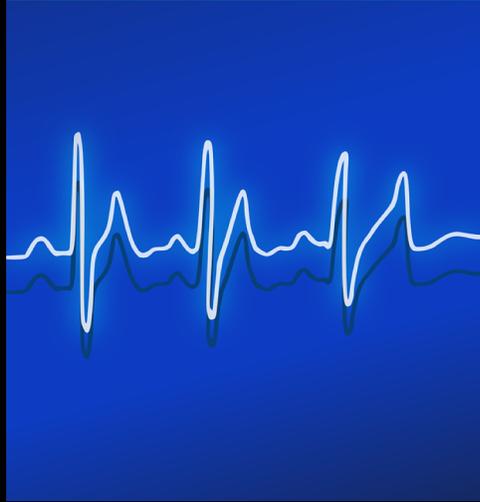
to be agile

5

### Practice 3



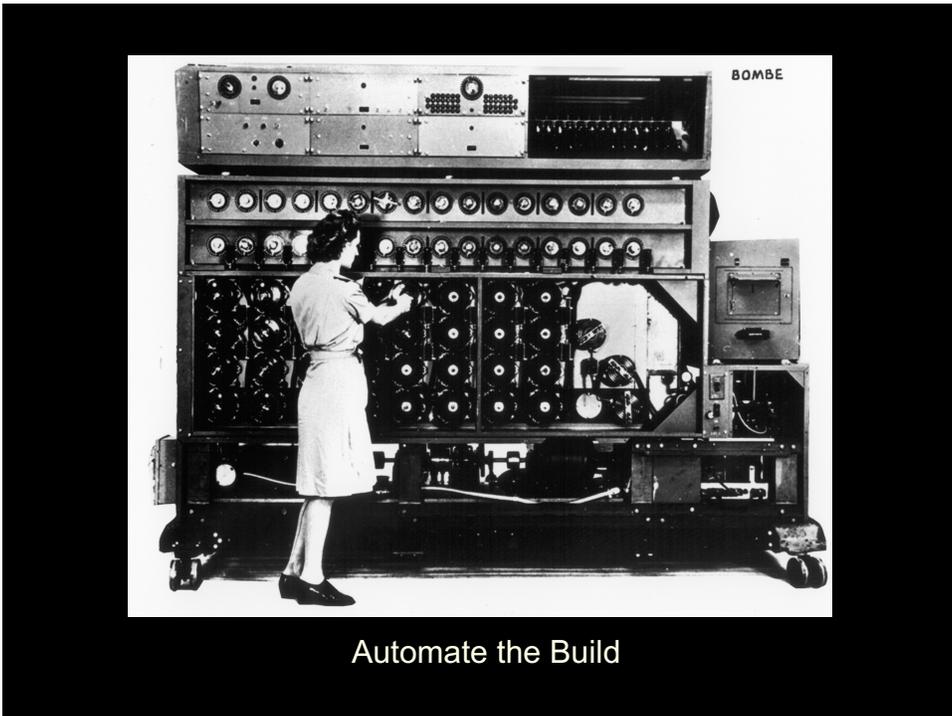
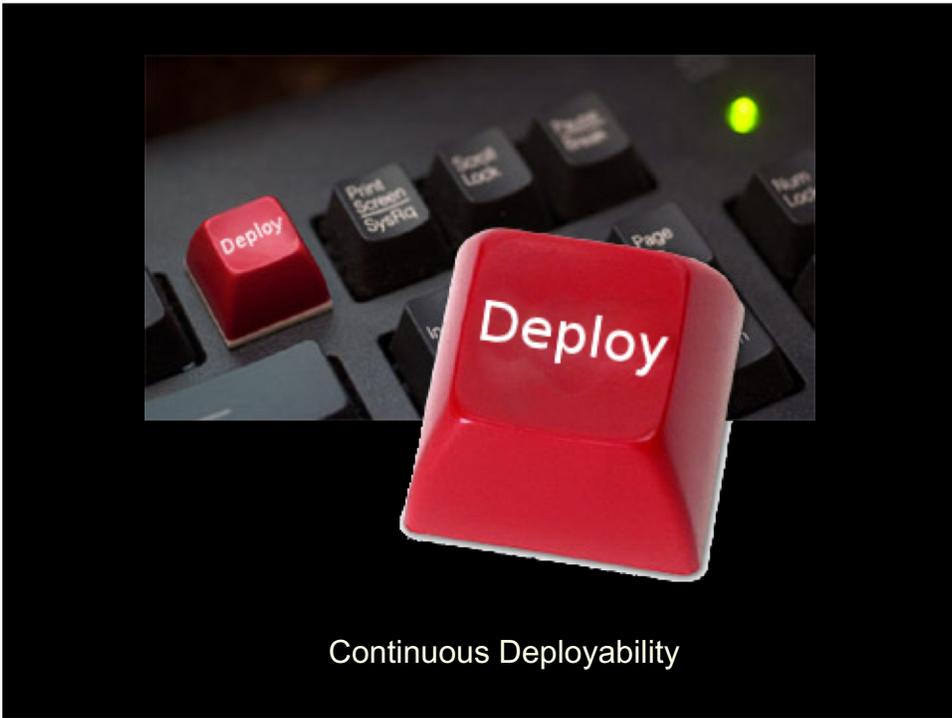
Integrate Continuously



The Heartbeat of a Project



Three Kinds of Done





Integrate Early and Often



Take the First Step

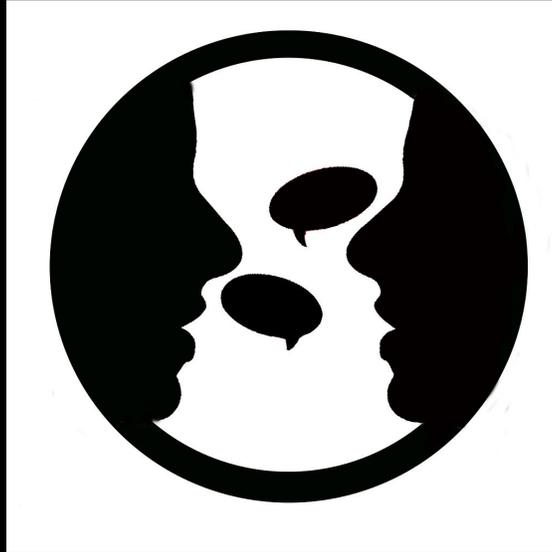
## Practice 4



Collaborate



Extreme Programming



Communicate and Collaborate



Pair Program



Buddy Program



Spike, Swarm, and Mob





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



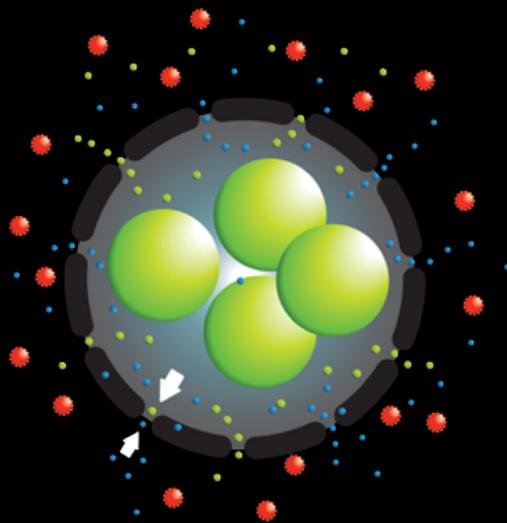
Create CLEAN Code



Quality Code is Cohesive



Quality Code is Loosely Coupled



Quality Code is Encapsulated



Quality Code is Assertive



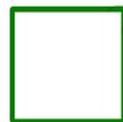
Quality Code is Non-Redundant



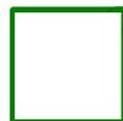
Code Qualities Guide Us



**EXCELLENT**



**GOOD**



**AVERAGE**

Increase Quality Today...

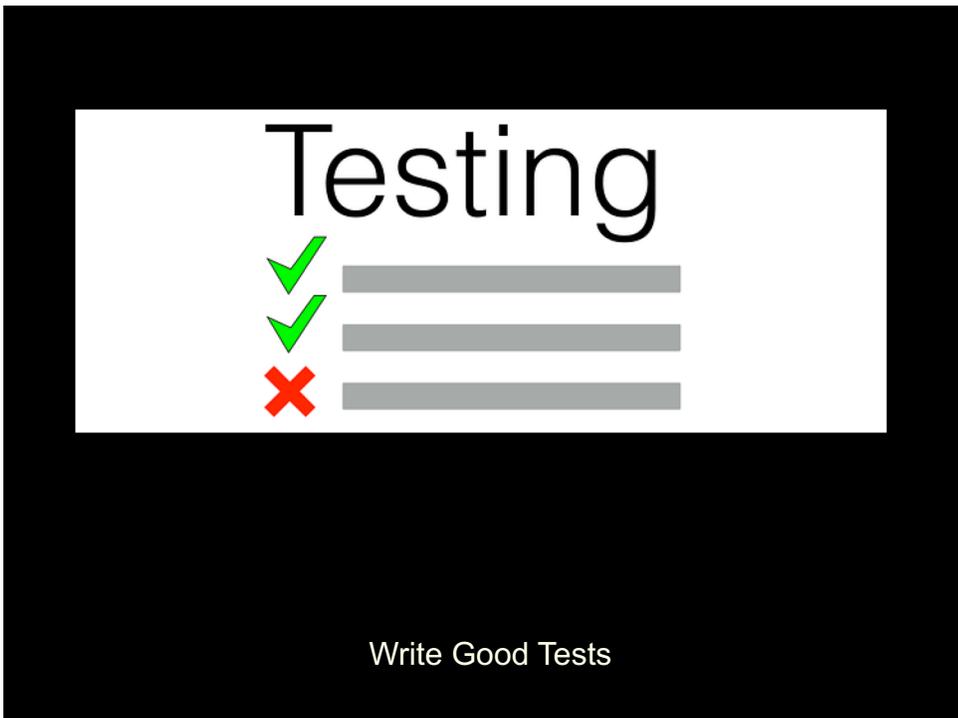
Practice 6



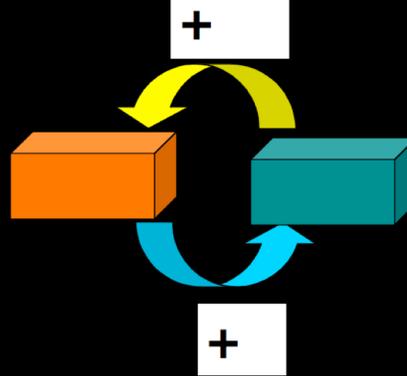
Write the Test First



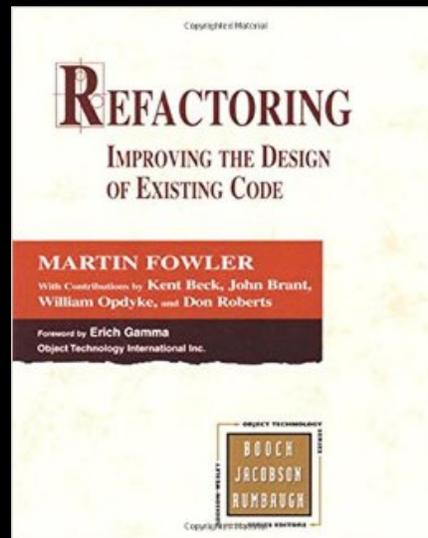
The Things We Call Tests



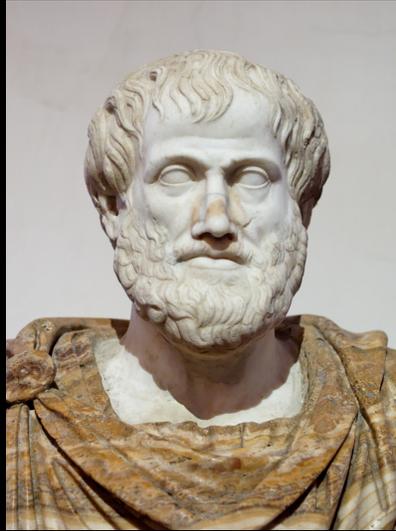
# Positive Feedback (A Virtuous circle)



TDD Gives Rapid Feedback



TDD Supports Refactoring



Write Testable Code



TDD Can Fail

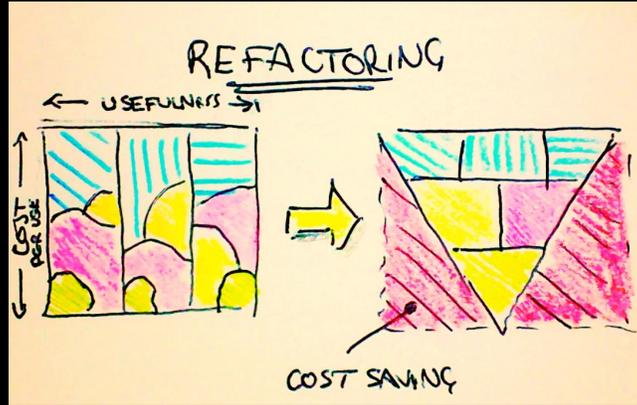


Introducing TDD to a Team

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

Become Test Infected

### Practice 9



Refactor Legacy Code



Investment or Debt?

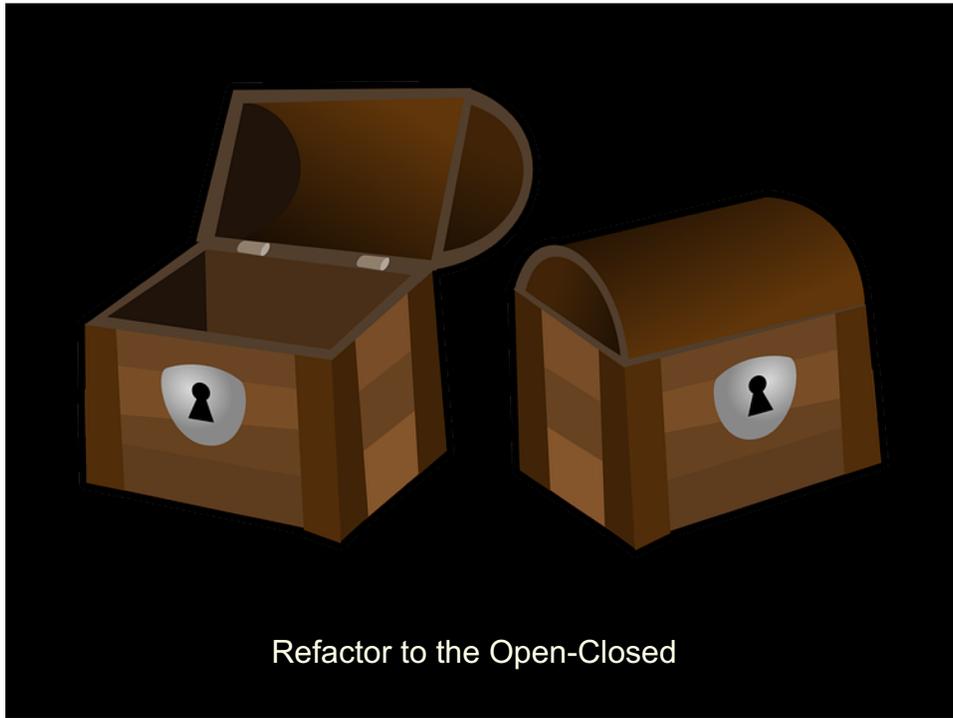




Refactoring Techniques



Refactor to Accommodate Change





## Thank You!



*Please fill out your feedback forms!*

- We have just scratched the surface, to learn more:
  - Read my blog: <http://ToBeAgile.com/blog>
  - Sign up for my newsletter: <http://ToBeAgile.com/signup>
  - Follow me on Twitter (@ToBeAgile)
  - Read my book:
    - *Beyond Legacy Code: Nine Practices to Extend the Life (and Value) of Your Software* (available from <http://BeyondLegacyCode.com>)
  - Attend my one of my Certified Scrum Developer trainings
    - See <http://ToBeAgile.com/training> for my public class schedule
    - Or contact me to arrange a private class for your organization
  - Visit <http://ToBeAgile.com> for more information

to be agile

50