
Why Johnny Can't Test and Can Test-Driven Development Help: A People-Centered Analysis of Testing

Orit Hazzan

Department of Education in Technology and Science
Technion

Email: oritha@techunix.technion.ac.il

-
- Who likes testing?
 - Who hates testing?

 - Why?
-

Outline

- Why people don't like testing?
- May Test Driven Development (TDD) help?
If yes - how?
- TDD and Risk Management

■ Thanks: Uri Leron and Yael Dubinsky

Outline

- Why people don't like testing?
 - Cognitive
 - Social
 - Affective
 - Managerial
 - Think about TDD...
 - May TDD help? If yes- how?
 - TDD and risk management
-

Why people don't like testing?

- **Observation 1:** In traditional development environments, testing appears as one of the last stages and is usually done under pressure.
-

Software Engineering/Hans Van Vliet



- P. 386-397: ... the testing activity often does not get the attention it deserves. **By the time the software has been written, we are often pressed for time, which does not encourage thorough testing.**
- p. 397: Postponing test activities for too long is one of the most severe mistakes often made in software development projects. This postponement makes testing a rather costly affair.

Yet, this book presents the traditional life cycle of software development.

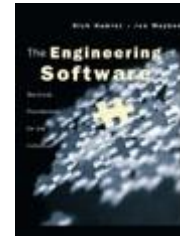
Why people don't test

- **Observation 1:** In traditional development environments, testing appears as one of the last stages and is usually done under pressure.
 - Time pressure is a naive explanation: it is based on an external/technical factor
 - The truth should be solicited in people's behavior, emotions and cognition
-

Some Observations

- **Observation 2:** Testing may give a negative feedback (and who likes it?)
 - The game ends with a failure
-

Hamlet and Maybee



Testing a program ought to be more fun than designing and coding it, certainly more fun than negotiating with its users about the requirements. For the first time in development, you get to see it work! There's no denying that watching all that information processing horsepower in action is interesting, even exciting. After so much work, running the program should be a reward.

Maybe the reason testing is not always thought of as fun is that there's a flip side: the program may not work. In the earlier parts of development, things can go wrong, but failures are not as absolute and graphic as they are in testing. A developer can even (unconsciously or on purpose) sweep problems under the rug during requirements, specification, design, and coding—but when those problems show up as failed tests, it's no longer possible to kid yourself.

Or maybe the fun of testing diminishes if it isn't your own code being tried. But testing is still detective work that would be a serious challenge for Sherlock Holmes. The program and its specification and design documentation contain the clues to the crime (the lurking crashes), and the tester must find them as a triumph of deduction. Then again, not everyone likes detective work, and many people have no talent for it. Dr. Watson had none.

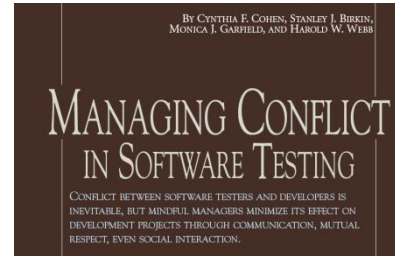
Some Observations

- **Observation 3:** **Testing** in traditional environments is **done by someone else**
 - The responsibility is transferred
 - Especially, if it gives a negative feedback!
-

Some Observations

- **Observation 4:** Testing in traditional software development environments is carried out at the end of the production line, and, inspired by tradition working class jobs, gets **low status**.
 - **Tension between different groups**
-

CACM, 47(1): Managing Conflict In Software Testing, *Cohen et al.*



- Though most organizations recognize the need for high-quality testers and their specialized skill set, **testers still struggle to win the respect they deserve.** One manager told us, "If you had a diagram with God at the top, the engineers [developers] would put themselves above that." Many testers feel they struggle to maintain their place relative to that of developers.
- P. 80: **The lack of status and support makes the tester's job more difficult and time consuming, as the struggle for recognition becomes part of the job itself.**

Additional Observations

- **Managerial difficulties:**
 - Testing slows down development
 - It's hard to manage testing (when)
 - **Cognitive difficulties:**
 - It's hard to know what to test
 - It's hard to know how much testing should be done
-

Summary of Part A

- Why people don't like testing?
- Testing is hard
 - Time pressure
 - Testing gives negative feedback
 - The responsibility is transferred
 - Low status of testing
 - Conflicts *between* the individual and the team
 - More...
- Plausible explanation: Software intangibility

Outline

- Why people don't like testing?
 - May Test Driven Development (TDD)
help? If yes - how?
 - TDD and risk management
-

How TDD copes with the challenge

- **Time pressure** - TDD inspires a **controlled process**:
 - it is not postponed to the end of the process
 - it is done all the time
 - it is done in small steps
 - **Testing gives a negative feedback** - in TDD the rules of the game are reversed
 - TDD ends with a success
-

Practitioners' reflection

- Code Unit Test First <http://c2.com/cgi/wiki?CodeUnitTestFirst>

Why don't people like testing? Well, the traditional way of testing is tough to take. You write what seems to be perfectly sensible code, then you write a test and the test tells you that you failed. No one wants to hear that.

Let's turn it around. Write the test first; run it. Of course it fails.. You haven't written the code under test yet. Start writing code.. keep testing. Soon, the test will tell you that you've succeeded! MichaelFeathers

How TDD copes with the challenge

- **Time pressure** - TDD is done all the time; it is not postponed to the end of the process
 - **Testing gives a negative feedback** - TDD ends with a success
 - **The responsibility is transferred** - TDD is done by the developer who writes the code
 - **Low status of testing** - All developers are testers
-

Additional Observations

- **Managerial difficulties:**
 - Testing slows down development
 - It's hard to manage testing (when)
 - TDD fosters development processes
 - TDD turns development (and testing) to be a controlled process
 - Specific rules
 - **Automatic (not manually) process**
-

Practitioner's reflection

- A key aspect of this process: don't try to implement two things at a time, don't try to fix two things at a time. Just do one. When you get this right, development turns into a very pleasant cycle of testing, seeing a simple thing to fix, fixing it, testing, getting positive feedback all the way. Guaranteed flow. And you go so fast! Try it, you'll like it.

RonJeffries

Additional Observations

- **Cognitive difficulties:** TDD improves understanding of what you develop
 - It's hard to You know what to test (what)
 - It's hard to You know how much testing should be done
-

Outline

- Why people don't like testing?
 - May Test Driven Development (TDD) help? If yes - how?
 - TDD and risk management
-

Risk management and TDD

- מקור: ניהול סיכונים בניהול פרויקטים
 - ניהול פרויקטים לעולם יהיה בסביבת אי וודאות.
 - **סיכון: כל גורם העלול למנוע מהפרויקט להגיע לסימומו המוצלח ולעמוד במטרותיו.**
 - ע"י ניהול סיכונים אנו מנסים לצמצם כל סיכון אפשרי שנוכל להעריך שיקרה ושעלול למנוע את סיום הפרויקט בהצלחה.
-

Risk management and TDD

TDD ■

□ במקום "כיבוי שריפות"

□ במקום **דחוף וחשוב**

חשוב ולא דחוף

□ מערב את כל צוות הפרויקט בניהול סיכונים

Conclusion

- TDD may help in coping with
 - Cognitive
 - Affective
 - Social
 - Managerialchallenges of testing.
-

Practitioner's reflection

- <http://xp.c2.com/UnitTest.html>

It's a wonderful ... experience

I don't write code any other way anymore.

My code has less problems, I have more confidence and management has more confidence. - sg

Questions?

Orit Hazzan

Email: oritha@techunix.technion.ac.il