Quick & Dirty (&Right)

Ted Neward
Neward & Associates
http://www.newardassociates.com | ted@tedneward.com

Quick & Dirty

Everybody agrees... it's bad!

Quotes

Everybody agrees... it's bad!

"If 10 years from now, when you are doing something quick and dirty, you suddenly visualize that I am looking over your shoulders and say to yourself: 'Djikstra would not have liked this', well that would be enough immortality for me." --Edsger Djikstra

Quotes

Everybody agrees... it's bad!

"... doing things the quick and dirty way sets us up with a technical debt, which is similar to a financial debt. Like a financial debt, the technical debt incurs interest payments, which come in the form of the extra effort that we have to do in future development because of the quick and dirty design choice. We can choose to continue paying the interest, or we can pay down the principal by refactoring the quick and dirty design into the better design. Although it costs to pay down the principal, we gain by reduced interest payments in the future." --Martin Fowler

Quotes

Everybody agrees... it's bad!

"THROWAWAY CODE is quick-and-dirty code that was intended to be used only once and then discarded. However, such code often takes on a life of its own, despite casual structure and poor or non-existent documentation. It works, so why fix it? When a related problem arises, the quickest way to address it might be to expediently modify this working code, rather than design a proper, general program from the ground up. Over time, a simple throwaway program begets a BIG BALL OF MUD." -- Brian Foote, Jospeh Yoder

Architecture

What defines the difference between these two buildings?

- goals
- scope
- complexity
- materials
- process
- … ?

Quotes

Everybody agrees... it's bad!

... so there's no further point to this keynote, right?

Quotes

Everybody agrees... it's bad!

... so there's no further point to this keynote, right?

... except we don't always seem to agree that it's bad

Quotes

Google "Quick and Dirty"

- "Quick and Dirty Guide to ..."

  (adjective) introduction; first-steps; tutorial
- "A Quick and Dirty On ..."

  (noun) one- or two-pager on a given subject
- Urban Dictionary: [[CENSORED]]

Quick & Dirty (& Right)

Time to re-examine

Re-examination

Everybody is OK with "quick"...

... but what does "dirty" mean?

Re-examination

"Dirty" == "Technical debt"

"... doing things the quick and dirty way sets us up with a technical debt, which is similar to a financial debt. Like a financial debt, the technical debt incurs interest payments, which come in the form of the extra effort that we have to do in future development because of the quick and dirty design choice. We can choose to continue paying the interest, or we can pay down the principal by refactoring the quick and dirty design into the better design. Although it costs to pay down the principal, we gain by reduced interest payments in the future." --Martin Fowler

Re-examination

Technical debt (according to Fowler)

https://martinfowler.com/bliki/TechnicalDebtQuadrant.html

- "people have made a considered decision to adopt a design strategy that isn't sustainable in the longer term, but yields a short term benefit, such as making a release. The point is that the debt yields value sooner, but needs to be paid off as soon as possible."
- "The debt metaphor reminds us about the choices we can make with design flaws. ... So the useful distinction isn't between debt or non-debt, but between prudent and reckless debt."
- "... there's also a difference between deliberate and inadvertent debt."

Re-examination

Four Quadrants to Technical debt

- Deliberate/Reckless

  "We don't have time for design"
- Inadvertent/Reckless

  "What's layering?"
- Deliberate/Prudent

  "We must ship now and deal with the consequences"
- Inadvertent/Prudent

  "Now we know how we should have done it"

Re-examination

Conclusions?

- "the useful distinction isn't between debt or non-debt, but between prudent and reckless debt"
- is the debt being taken a prudent one?
- do you have a plan for paying it off?
- be careful of leaving the debt in place for too long

when temporary soutions are allowed to linger long past their sell-by date, you have a problem

Re-examination

Reminder:

"THROWAWAY CODE is quick-and-dirty code that was intended to be used only once and then discarded. However, such code often takes on a life of its own, despite casual structure and poor or non-existent documentation. It works, so why fix it? When a related problem arises, the quickest way to address it might be to expediently modify this working code, rather than design a proper, general program from the ground up. Over time, a simple throwaway program begets a BIG BALL OF MUD." -- Brian Foote, Jospeh Yoder

Sidebar: Not a problem...

... but desirable?

Desirable?

- DZone Article: "Is This the Age of Throw-Away Software Systems?"

  https://dzone.com/articles/is-this-the-age-of-throw-away-software-systems

  – Just Make it Work: Usually Quick, Dirty and Cheap
    - **time to market is all that matters**
    - **"How fast can I get it? Does it work? How much does it cost?"**

  – Systems are no longer targeted to be long-lived
    - **companies don't try to extend the lifetime of older systems; they just rewrite it**
    - **thus, the lifecyle of the built system is likely to be 3-5 years**

  – "Agile" forces us to implement incomplete and unapproved requirements
    - **create a working hypothesis from what you know**
    - **supplement it with best-guesses to get to an initial solution**
    - **if the result is too far away from correct, throw it away**

Desirable?

INC. Article: The Importance of Quick and Dirty

https://www.inc.com/magazine/201305/jason-fried/the-importance-of-quick-and-dirty.html

- "When we work on something, we pay careful attention to the details and take the time to polish every little thing until it's just right. As a result, 37signals is known for a focus on quality, and we take that reputation very seriously. It turns out that that's a problem."

Desirable?

INC. Article: The Importance of Quick and Dirty

https://www.inc.com/magazine/201305/jason-fried/the-importance-of-quick-and-dirty.html

- "We recently began exploring an idea for a new software product. ... For six weeks, we sketched out a bunch of ideas for the product until we finally felt we had a great idea. ... All of us were excited and working hard, but a week later, we had almost nothing to show for our effort. Nearly two months had passed since we had set out to test our idea, and we still had no idea if it would work."
- "What happened was we forgot we were just building a quick-and-dirty demo for ourselves."

Desirable?

The Lean Startup world embraces the MVP

- Minimum Viable Product
- the smallest set of features possible that still allows a customer to purchase a thing
- used to get you to customer #1
- might not even be software!
- prove the market exists; validate or invalidate the idea

Desirable?

INC. Article: The Importance of Quick and Dirty

https://www.inc.com/magazine/201305/jason-fried/the-importance-of-quick-and-dirty.html

- "... obsessing about quality too early in the creative process prevents a lot of good ideas from taking shape. As businesses grow, all sorts of things that once were done on the fly--including creating new products--have a way of becoming bureaucratized. As a result, the wrong sets of pressures are brought to bear. Doubts, deadlines, resource planning...all of this stuff is essential. But only later on. Fretting about such matters at the outset only gets in the way."

Desirable?

- INC. Article: The Importance of Quick and Dirty

  https://www.inc.com/magazine/201305/jason-fried/the-importance-of-quick-and-dirty.html

  – "... I see [startups] overthinking simple problems, adding too much structure too early, and trying to get formal too soon. Start-ups should embrace their scrappiness, not rush to toss it aside. The ability to run with scissors is a blessing, not a curse."

Desirable?

Anybody remember a time when prototypes were prototypes?

- set a hypothesis
- build something
- prove or disprove the hypothesis
- throw away the prototype and get on to the real work!

Iterations

Let's talk about that again

Iterations

Why does agile suggest iterations?

"Responding to change over following a plan."

"Working software is the primary measure of progress."

"Simplicity--the art of maximizing the amount of work not done--is essential."

Agile Manifesto

Iterations

Why does agile suggest iterations?

"An iterative process is one that makes progress through successive refinement. A development team takes a first cut at a system, knowing it is incomplete or weak in some (perhaps many) areas. The team then iteratively refines those areas until the product is satisfactory. With each iteration, the software is improved through the addition of greater detail."

https://www.mountaingoatsoftware.com/blog/agile-needs-to-be-both-iterative-and-incremental

Iterations

It isn't just agile that suggests iterations, by the way

- introducing John Boyd, and his military strategy tool, OODA

Iterations

OODA: Observe, Orient, Decide, Act

- decision-making occurs in this 4-part loop
- Observe: get information from the context
- Orient: determine what that information means and what options are
- Decide: choose a course of action
- Act: enact said action

Iterations

OODA: Observe, Orient, Decide, Act

- "the fundamental, unavoidable and all-pervasive presence of uncertainty is the starting point."
- "we must be able to form mental concepts of observed reality... and... change those concepts as reality itself appears to change."
- applied to scientists as much as to warfighters

see his diagram at http://tinyurl.com/sciloop

Iterations

OODA: Observe, Orient, Decide, Act

- this is just like agile!
- except...
    - **Boyd's stated aim was to "get inside" the opponent's OODA Loop**
    - **"In order to win, we should operate at a faster tempo or rhythm than our adversaries—or, better yet, get inside [the] adversary's Observation-Orientation-Decision-Action time cycle or loop."**

Iterations

In other words, it's not the loop itself we want to focus on...

Iterations

In other words, it's not the loop itself we want to focus on...

... but the tempo and speed in which the loop executes

Iterations

In other words, it's not the loop itself we want to focus on...

... but the tempo and speed in which the loop executes

... which means, we want feedback! Quickly!

Iterations

In other words, it's not the loop itself we want to focus on...

... but the tempo and speed in which the loop executes

... which means, we want feedback! Quickly!

Which, of course, we can't get without something up and running

Desirable?

- Recall: The Importance of Quick and Dirty

  https://www.inc.com/magazine/201305/jason-fried/the-importance-of-quick-and-dirty.html

  – "... I see [startups] overthinking simple problems, adding too much structure too early, and trying to get formal too soon. Start-ups should embrace their scrappiness, not rush to toss it aside. The ability to run with scissors is a blessing, not a curse."

Iterations

What if the first iteration is good enough?

- Congratulations!
- You have delivered the value the customer wanted
- and it's time for you to STOP!

Wrapping Up

What does it all mean?

Wrapping Up

The Human Side

- Q-and-D is often thought of poorly by developers
  - **"It's a hack!"**
  - **"It's so ugly!"**
  - **"I'm a *craftsman*, for God's sake!"**
- Don't let your ego turn you away from useful tools

Wrapping Up

When do we abandon Q-and-D?

- has the problem lasted longer than we thought?
- does the problem have more angles to it than we thought?
- are we considering keeping this system for more than 5 years?

Wrapping Up

Final thoughts

- Nobody suggests Q-and-D is the only or best way to build things

  but it is a tool in your toolbox, use it as such
- Understand the goals of the project as a whole

  don't create a masterpiece when a simple brace will do
- Get feedback as quickly as possible

  in order to refine and improve as quickly as possible

Wrapping Up

Now go.

Be Quick.

Get a little Dirty.

And feel Right doing it.