

Technical Debt

Elizabeth Naramore
Dutch PHP Conference
June 11, 2010

Technical Debt:

What is it?

Technical Debt:

Why should we care?

"Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite...

The danger occurs when the debt is not repaid. Every minute spent on non-quite-right code counts as interest on that debt.

--Ward Cunningham, 1992

"I think that there were plenty of cases where people would rush software out the door and learn things but never put that learning back into the program, and that by analogy, was borrowing money thinking that you never had to pay it back.... eventually all your income goes to interest and your purchasing power goes to zero."

What is technical debt?

Code or practices
that will hinder your progress
in the future.

Why should we care?

Too much technical debt
can cripple your team
or organization.

You have technical debt if you have:

lack of documentation

untested code

suppressed errors

unshared knowledge between teams or people

confusing code, inconsistencies, "workarounds"

local changes you've not committed

Or, if you have:

code that doesn't follow coding standards
non-existent or improperly used version control
unstable deployment process
duplicate code blocks
individual code ownership
3rd party software that needs updated or patched

Guess what?

***We all* have it.**

Don't panic.



**Some technical debt
is ok.**

It might make sense if:

It lets you hit a deadline

It lets you test an experimental feature

The code is rarely touched

You are coding based on reasonable expectations

It allows you to extend startup money

Your code is at the end of the life-cycle

This is also known as
prudent
technical debt.

**So when is
technical debt
not ok?**

All the other times.

This is also known as
reckless
technical debt.

**Sometimes we know
we are taking shortcuts,
but we do it anyway.**

This is also known as
deliberate
technical debt.

**Sometimes we don't
know any better, or
the debt is not our fault.**

This debt could come from:

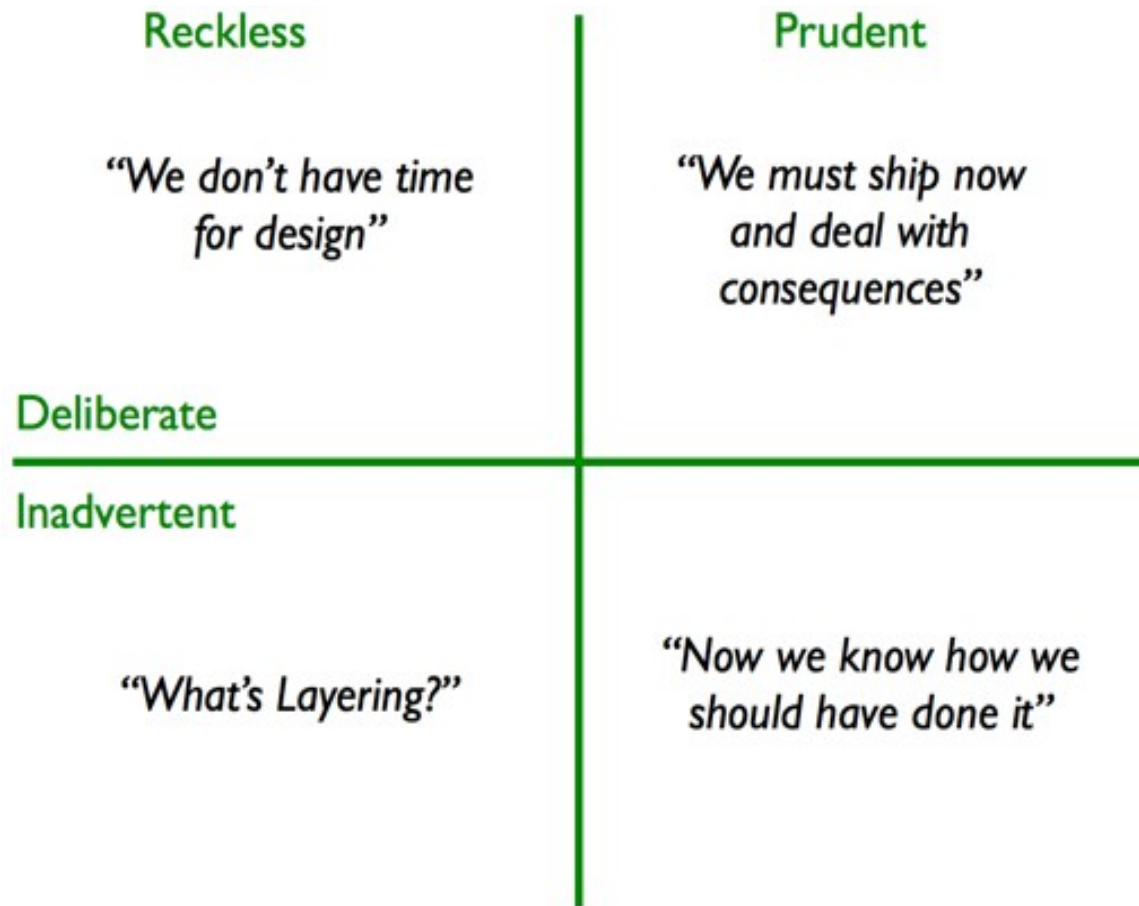
inexperienced team members
changing project requirements
post-release retrospectives
security patches or updates from 3rd parties

This is also known as
inadvertent
technical debt.

4 ways to classify technical debt:

**Prudent vs. Reckless
Deliberate vs. Inadvertent**

Martin Fowler's Quadrant



**By minimizing
deliberate debt,
you make it easier
to handle
inadvertent debt.**

**So now that we know
what it is,
what do we do with it?**

**We tackle our debt
with**

D.E.B.T.

(see what I did there?)

Discover
Estimate
Break Down
Task & Track

Discover

Where is our technical debt?

Ask the developers on your team

Use the checklist mentioned earlier

Use software like PHPUnderControl or Sonar

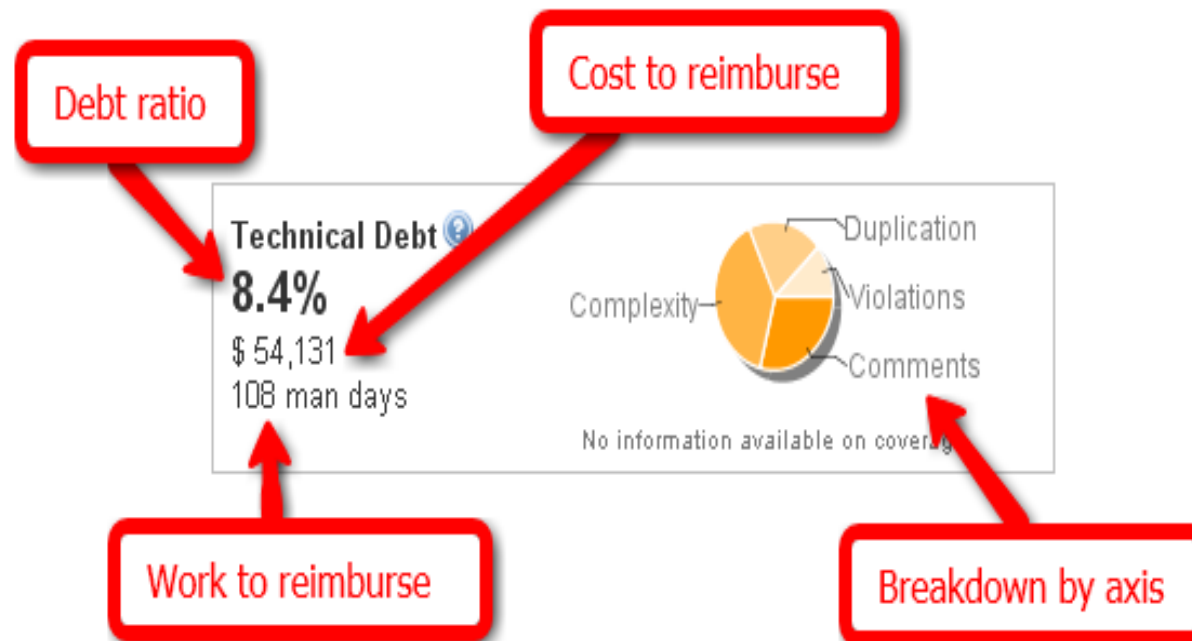
Estimate

The Sonar Formula

Debt(in man days)= cost_to_fix_duplications +
cost_to_fix_violations +
cost_to_comment_public_API +
cost_to_fix_uncovered_complexity +
cost_to_bring_complexity_below_threshold +
cost_to_cut_cycles_at_package_level

Estimate

The Sonar Formula



Estimate

But how much is it costing us?

The Simple-ish Formula

Amount of the “loan” = cost to fix

“Interest Rate” = Adverse impact on development

Estimate

Quantifying is hard.
OK to use other techniques.

Estimate

Use estimation technique that works for you
Get input and buy-in from the team

Break Down

What do we do with it?

Every task breaks down into 2 categories:

1. Debts where we continue paying interest
2. Debts where we pay the principal

Task and Track

How can we make it visible?

Add it to bug tracker, task board, etc.

Paying it Off

Focus on higher interest rate items first.

Paying it Off

Dedicate time for technical debt reduction.

- **Use time right after a release**
- **Tech debt iteration**
- **Time each day for tech debt**
- **Refactor as you go**
- **Use off-season time**

Paying it Off

Keep track of your progress.

Preventing Technical Debt

- **Break past habits**
- **Group consensus & commitment**
- **Teach your “children” well**
- **Never file for bankruptcy**
- **Live within your means**
- **Anticipate your needs**

Make sure to see:

PHP UnderControl:

<http://phpundercontrol.org>

Sonar:

**<http://www.sonarsource.com/>
(plus tech debt plugin)**

Want to contact me?

<http://naramore.net/blog>

@ElizabethN

elizabeth@naramore.net

Freenode IRC: ElizabethN

THANKS!

Image used with permission from Martin Fowler