Technical Debt a Business Perspective

Michael Vax
CTO/CPO , WebInterpret

Based on work of Prof. Philippe Kruchten and others

About me

- ~15 years Developer, tech. lead, architect
- 15+ years CTO, VP of Development, Head of Product Management
- Practicing Agile since 2005
- One of organizers of Agile Vancouver

- Born and raised in Moscow
- In Canada since 1992
- 4 years in Munich
- 1 month in Warsaw

About Webinterpret

Webinterpret

"Open worldwide commerce"

$3 Billion+

GMV generated by retailers using WebInterpret

29,333

Professional retailers using Webinterpret

54 Million+

Product listings localised

2007

Enabling and growing Cross Border Trade for over 8 years

250+

Over 250 employees

5 Offices

USA, UK, France, Germany & Poland

Partners : ebay PayPal amazon Google

Technical Debt
Definitions

Technical Debt

Introduced by Ward Cunningham

- Drags long-lived projects and products down
- Technical debt is more a rhetorical category than a technical or ontological category.
  - The concept resonates well with the development community and the business community
  - Both sides “get” the metaphor.
- Technical debt is a concept that bridges the gap between:
  - Business decisions makers
  - Software designers/developers

Technical Debt

Steven McConnell

- Type1 - unintentional, non-strategic
  - Poor design decisions
  - Poor coding
  - No tests
- Type 2 - intentional and strategic
  - Optimize for the present not for the future
  - Not moveable release date
  - ➢ 2a Short term: paid off quickly with refactoring
    - Large chunks: easy to track
    - Many small bits: cannot track
  - ➢ 2b Long-term
    - Stays there forever

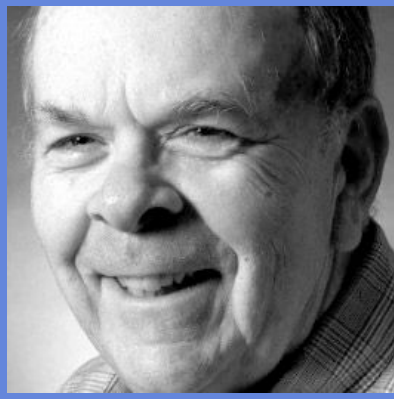
Technical Debt

Martin Fowler

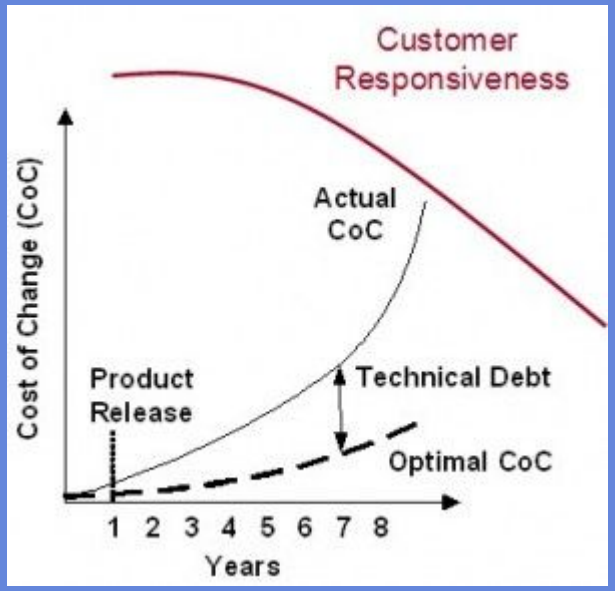
	Reckless	Prudent
Deliberate	"We don't have time for design"	"We must ship now and deal with consequences"
Inadvertent	"What is Layering?"	"Now we know how we should have done it"

Fowler 2009, 2010

Technical Debt as Cost of Change, Jim Highsmith



- Once on the far right of the curve, all choices are hard
- If nothing is done, it is just gets worse
- In application with high technical debt estimations are practically impossible



Technical Debt
Causes

TD Causes
People

- Making bad assumptions
- Inexperience
- Poor leadership and team dynamics
- "Superstars" - egos get in the way
- No push back against customers /sales
- Staff turn over with no knowledge transfer
- Subcontractors

Lim at al. 2012

TD Causes
Process

- Little consideration of code maintenance
- Unclear requirements
- Cutting back on process (code reviews)
- Little or no history of design decisions
- Not knowing or adopting best practices

Lim at al. 2012

TD Causes
Technology

- Technology limitations
- Legacy code
- Changes in technology
- Growing complexity of the product

Lim at al. 2012

TD Causes
Product

- Schedule and budget constraints
- Poor communication between developers and management
- Changing priorities
- Lack of vision, plan, strategy
- Trying to make every customer happy
- Unclear goals, objectives, and priorities
- Consequences of decisions is not clear

Lim at al. 2012

TD Causes Business

- Changes in Business models
- Targeting different customer segments
- Merges and acquisitions
- Extending product line

Technical Debt
What is it exactly

What we consider a Technical Debt?

- Badly written, poorly structured code
- Code without test coverage
- Duplicated code
- Code that is not used
- Poorly named classes & modules
- Fragile tests

as well as

- Lack of product documentation
- Incorrect documentation is even worse
- Upgrade debt (using outdated versions of libraries and 3rd party s/w)
- Manual setup of test environments
- Not automated build and deployment
- Not adequate monitoring of production environment

Technical Debt
Money Perspective

Time is Money

Think of the amount of money the borrowed time represent - the grant total required to eliminate all issues found in the code

Code smells	167 days
Missing tests	298 days
Design	670 days
Documentation	67 days
Total days	1,202 days
Total cost	$1,202,000

Assuming Cost of one developer day is $1000

Cost of fixing Technical debt is **Borrowed amount (Principal)**

Lost of productivity caused by Technical Debt is **Interest you pay on Principal**

“Interests”

In presence of technical debt:

- Cost of adding new features is higher
- When repaying (fixing), additional cost for retrofitting already implemented features
- Technical debt not repaid => leads to increased cost, **forever!!!**
- Cost of fixing increases over time
- Onboarding new team member takes longer

“Every minute spent on not-quite-right code counts as interest”
Ward Cunningham

Value of Your Product

	Visible	Hidden
Positive Value	**Features**	**Good Architecture**
Negative value	**Bugs**	**Technical Debt**

Value of the software	$100,000
Cost of bug fixing	-$15,000
To address TD	-$25,000
Real value of software	**$60,000**

Technical Debt
How to address

Common Approaches to Technical Debt Reduction

We are way too busy to deal with it now

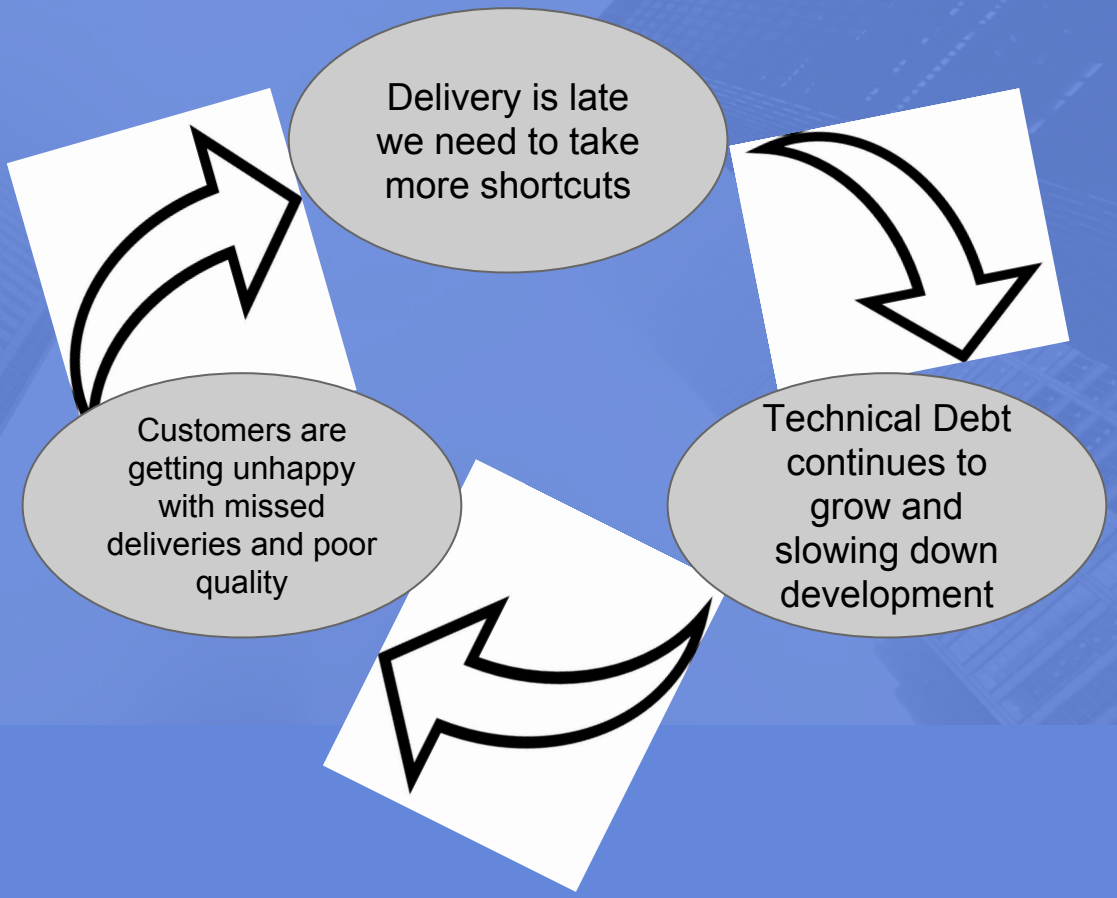
Next release, next year for sure

Let's stop developing new features until we fix **ALL** this mess

Three Strategies

1. **Do nothing**
2. Replace
3. Incremental Refactoring

It will get worse



Three Strategies

1. Do nothing
2. **Replace**
3. Incremental Refactoring

- High-cost
- High-risk

Rewrite projects have the highest failure rate

Three Strategies

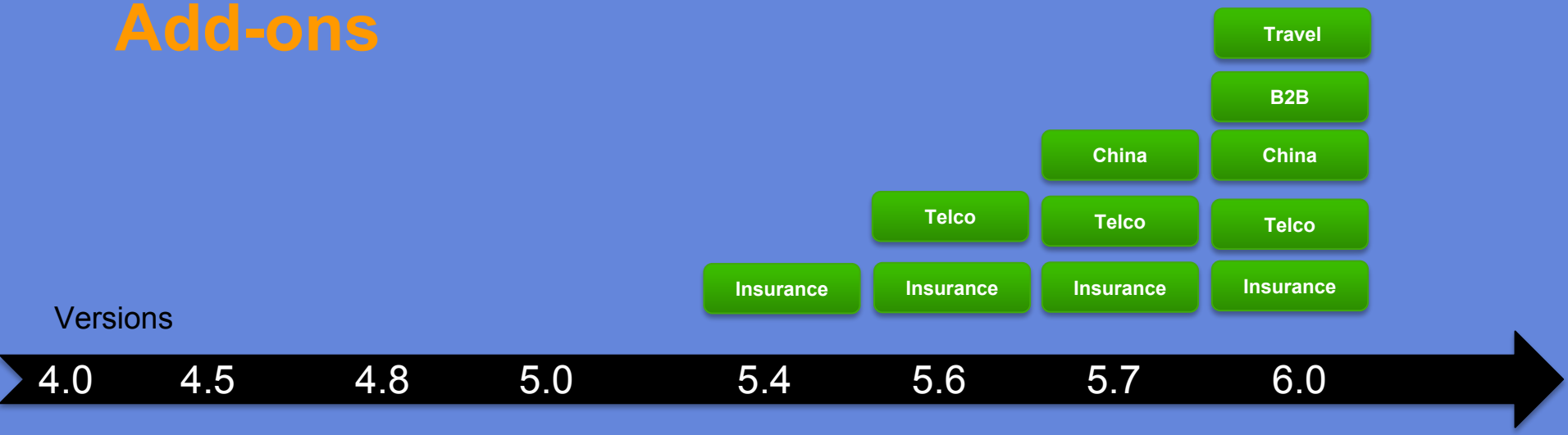
1. Do nothing
2. Replace
3. **Incremental Refactoring**

- Debt reduction
- Requires commitment & discipline
- Control amount of new debt added to the product

Reducing Debt

- Make it visible
- Incorporate debt reduction as a regular activity
- Look for opportunities to combine it with feature development
- Finish what you've started
- Make it quick
- Don't touch if it does not hurt

Technical Debt Reduction Story



Social Debt & Friction

Social debt is a state of a development project which is the result of the accumulation over time of decisions about the way the development team (or community) communicates, collaborates and coordinates.

In other words, decisions about the organizational structure, the process, the governance, the social interactions, or some elements inherited through the people: their knowledge, personality, working style, etc.

Thank You

Staying in touch
https://www.linkedin.com/in/michaelvax