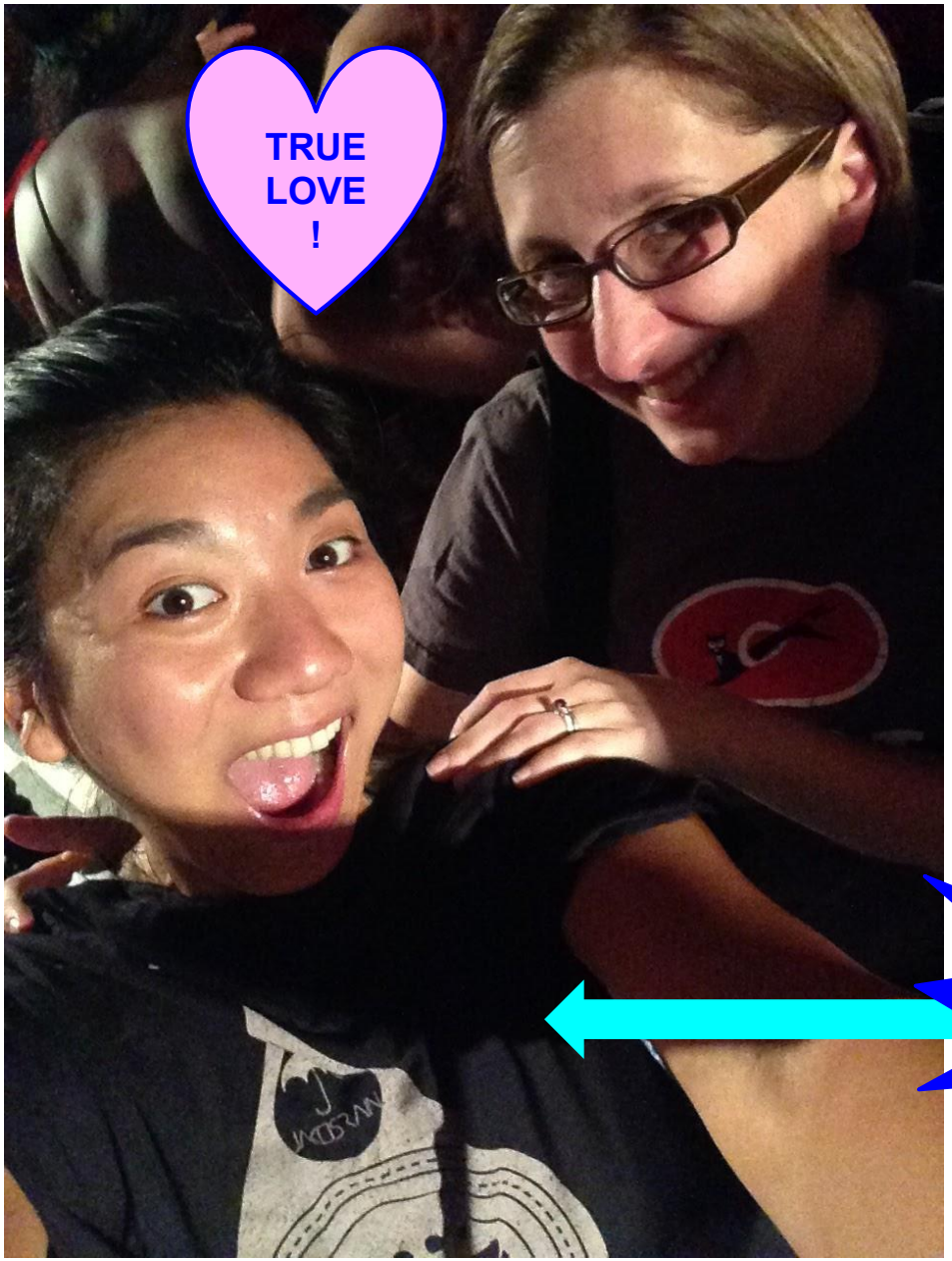
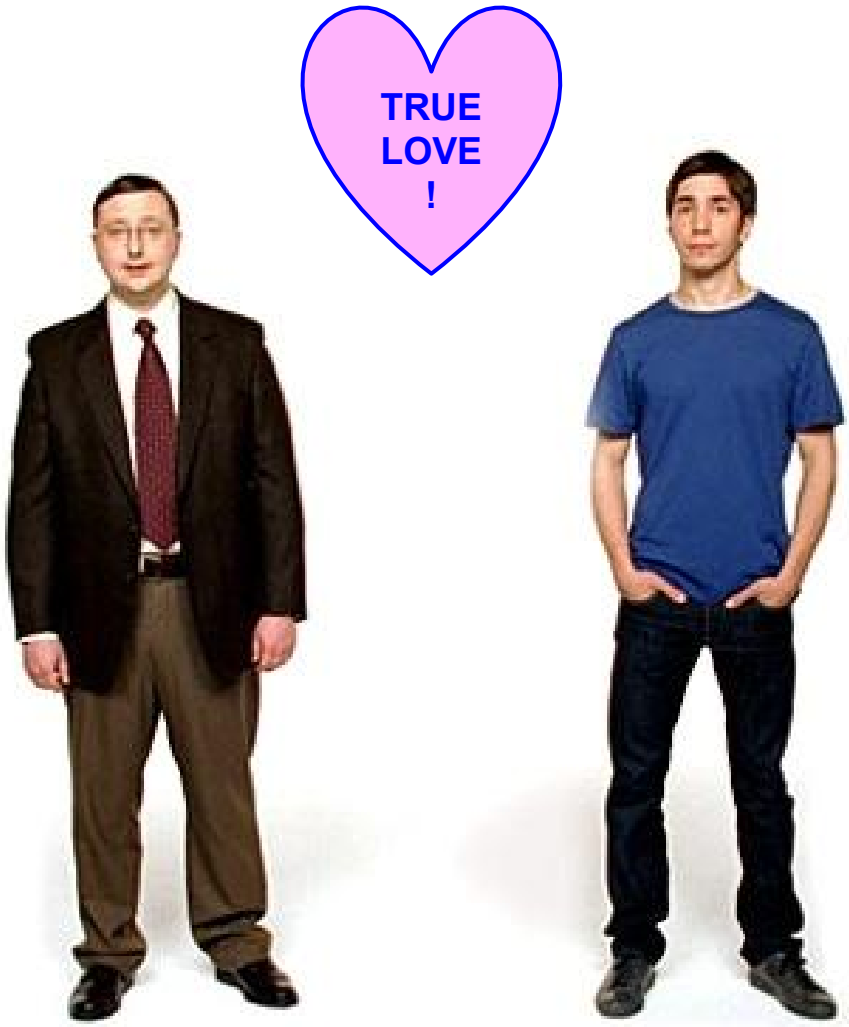
I Don’t Know What the F*ck That Means!

Jenny Liu and Carol (Nichols || Goulding)

@liumonade and @carols10cents

Carol and Jenny, sweaty-faced, at a concert




TRUE
LOVE
!

The Agenda

- A Communication Problem
- How it happens (storytime!)
- Meta-Lessons
- What can YOU do about it
- Cat pictures (purr purr)

So what's the problem?

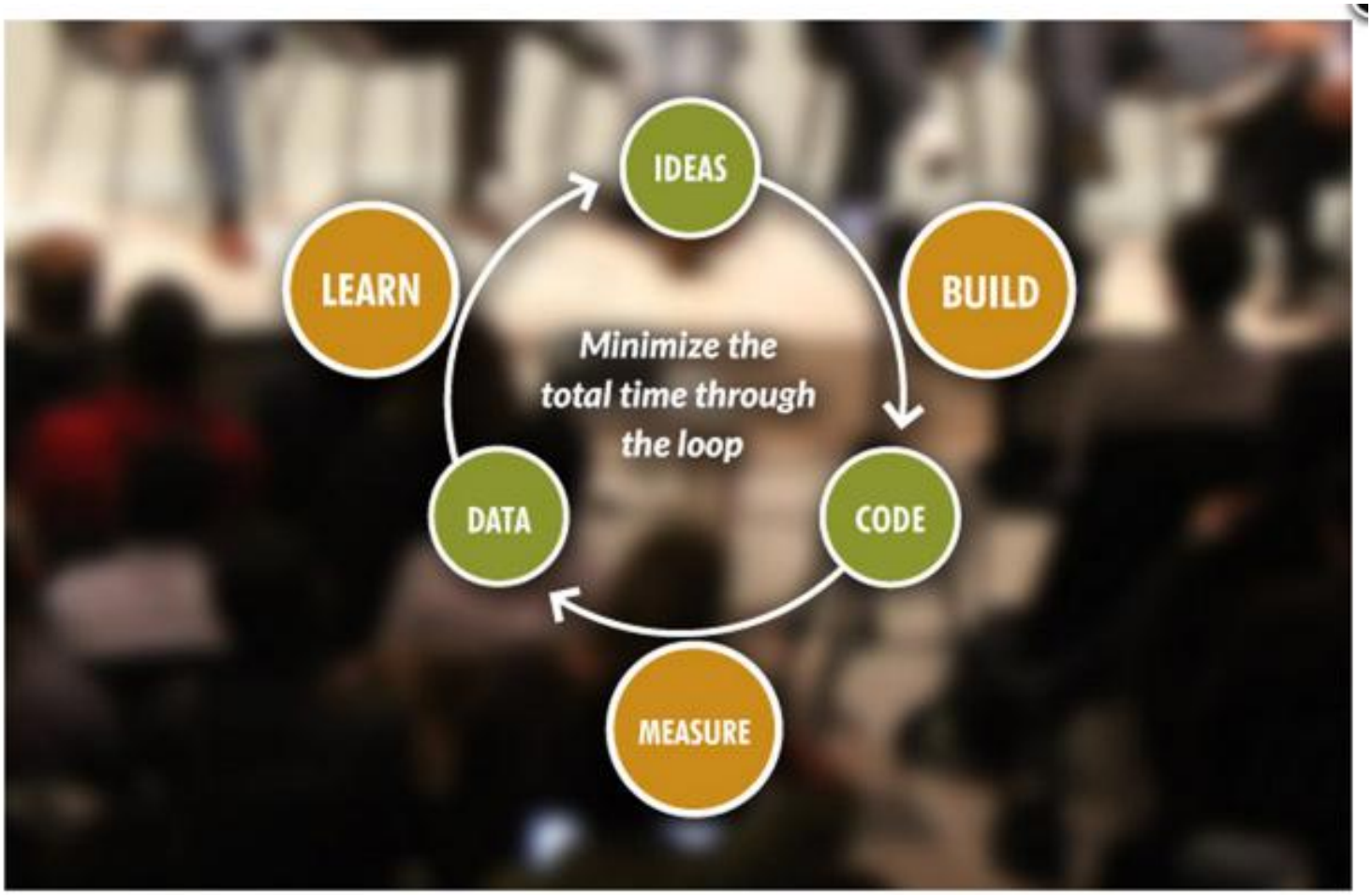
Exhibit A:

MVP

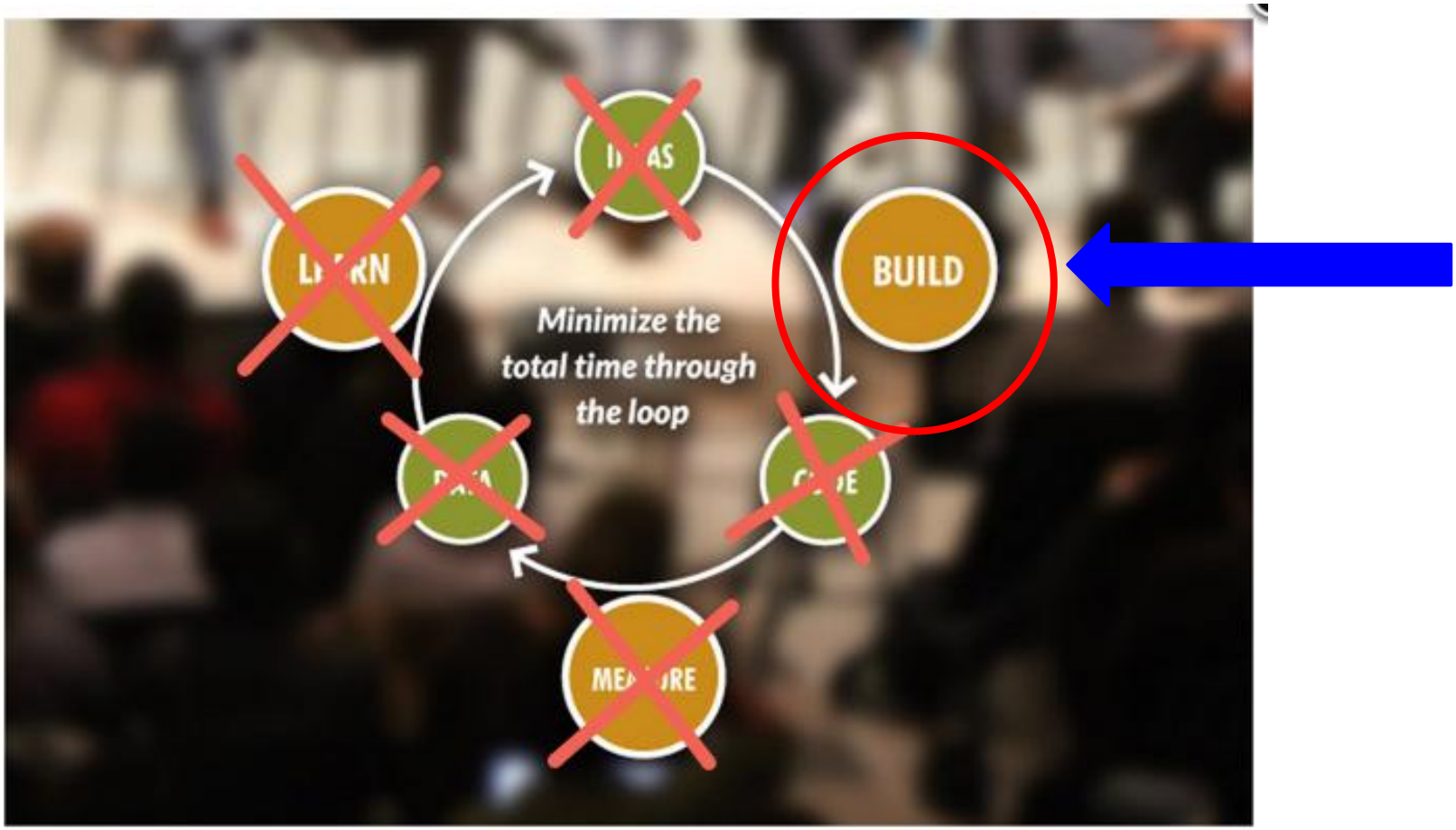
aka ...

Minimum Viable Product

Eric Ries' Build-Measure-Learn Loop



Not ideal, but at least it's honest

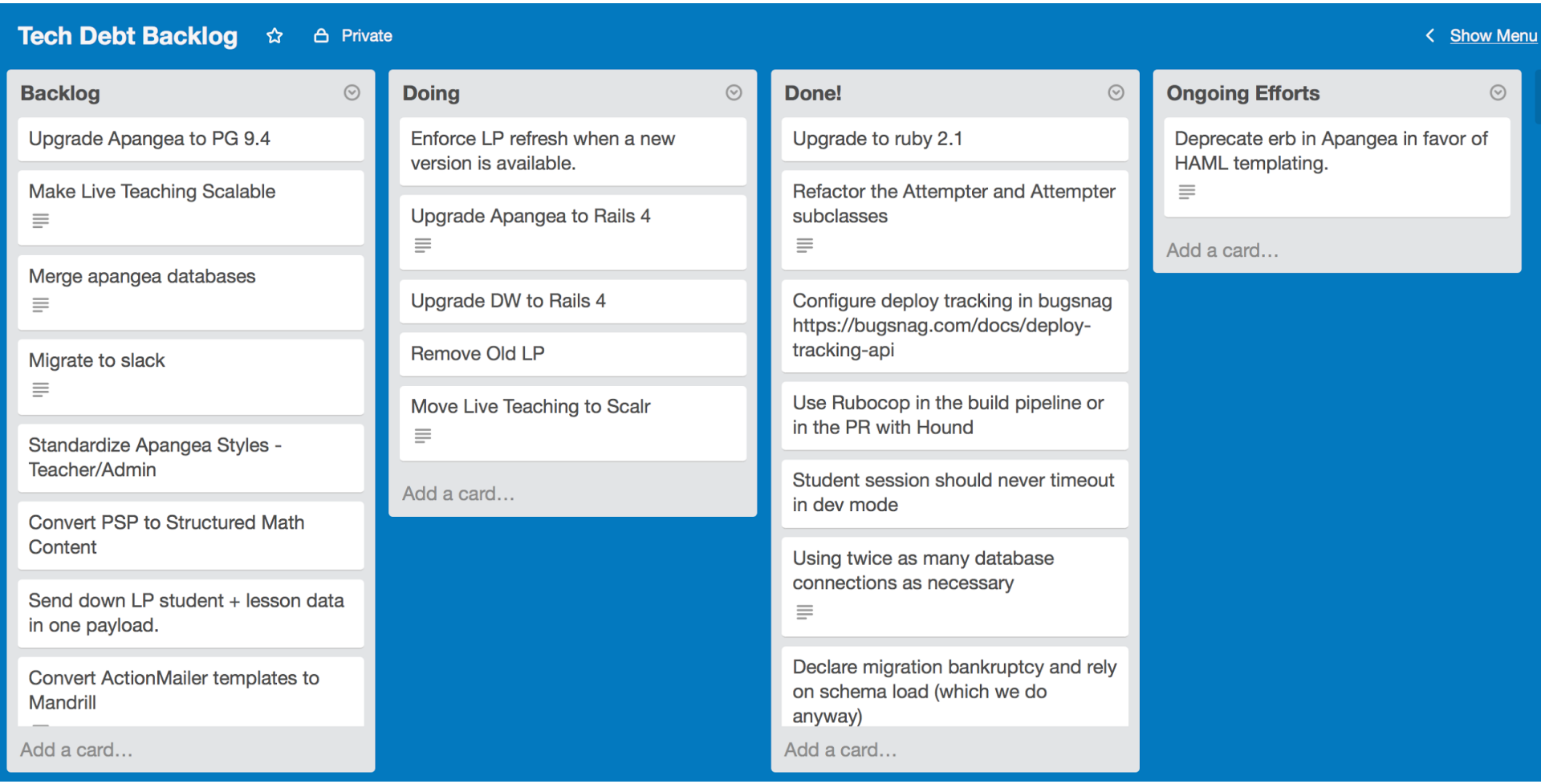


https://dl.dropboxusercontent.com/u/1018963/Articles/HowSpotifyBuildsProducts.pdf

Exhibit B:

Technical Debt

Dat Technical Debt Backlog



Is it Technical Debt?

Ask yourself... **If you said no to even one…**

Is the code clean?

Is the code tested?

Is there a learning objective?

Is there a plan for payback?

Is the business truly informed?

Then you don't have Technical Debt

http://www.slideshare.net/LeanDog/the-technical-debt-trap-michael-doc-norton

Or is it cruft?

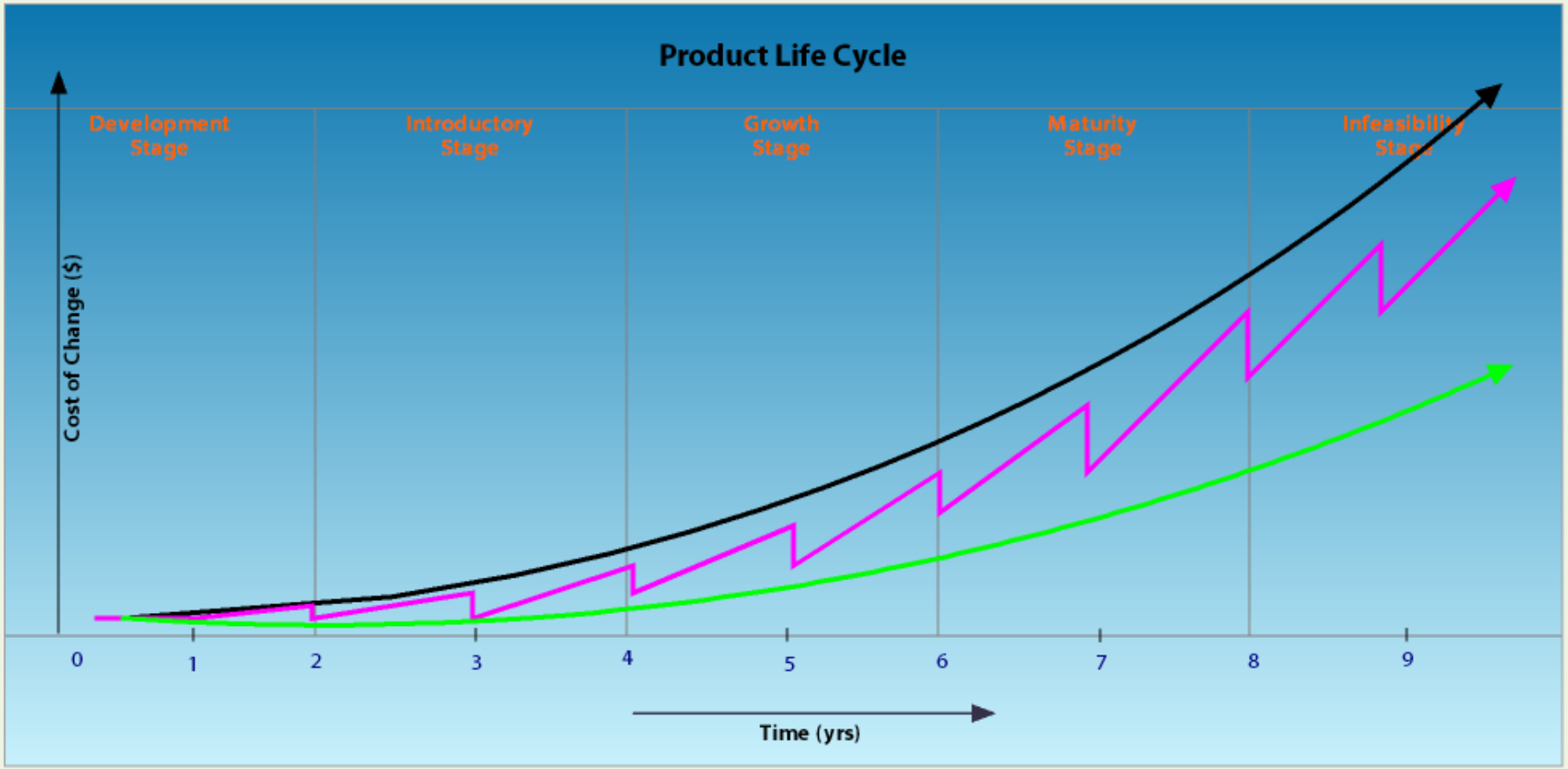
Intentional Cruft

(naughty naughty!)

vs

Unintentional Cruft

(oops, i did it again)



Legend:

- Traditional Application - No Refactoring (High Cost of Change)
- Once-a-year Technical Debt Refactoring (Lower Cost of Change)
- Continuous Refactoring (Lowest Total Cost of Change)

https://web.archive.org/web/20080112220939/http://www.jacoozi.com/blog/?p=11

Exhibit C:

And for our last term …

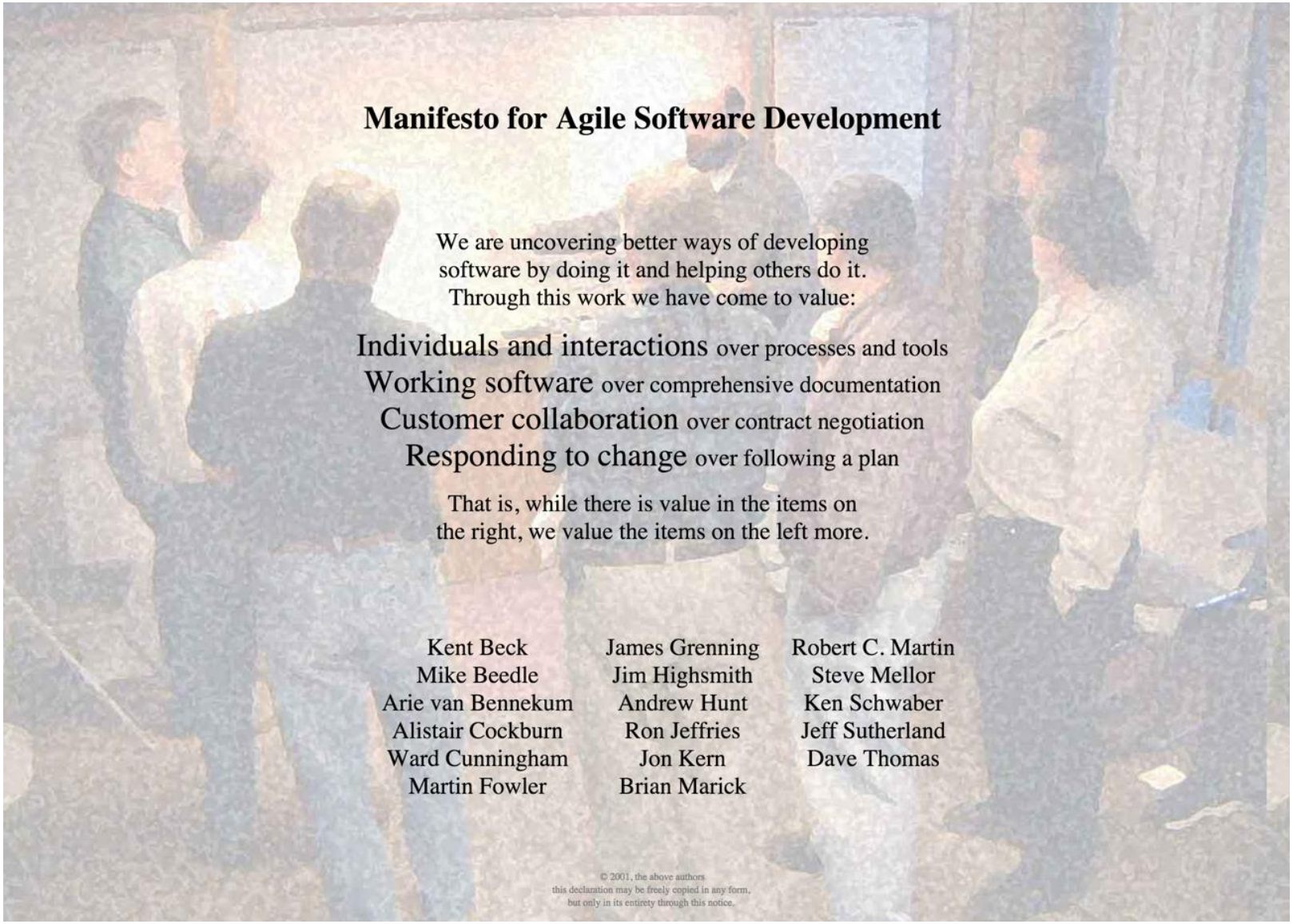

Agile

How have YOU heard agile defined?

Here's what we've heard:

- Agile is when you do standups
- Agile is pair programming
- Agile is measuring progress with points and burndown charts
- Agile is the opposite of waterfall
- Agile is Scrum
- Agile means no planning
- Agile means no documentation

Who doesn't love an origin story?



(this screenshot is probs too difficult to read)

Ah, that's better!

The Agile Manifesto

Individuals and interactions	over	Processes and Tools
Working Product	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to change	over	Following a plan

That is, while there is value in the items on the right, we value the items on the left more.

www.agilemanifesto.org

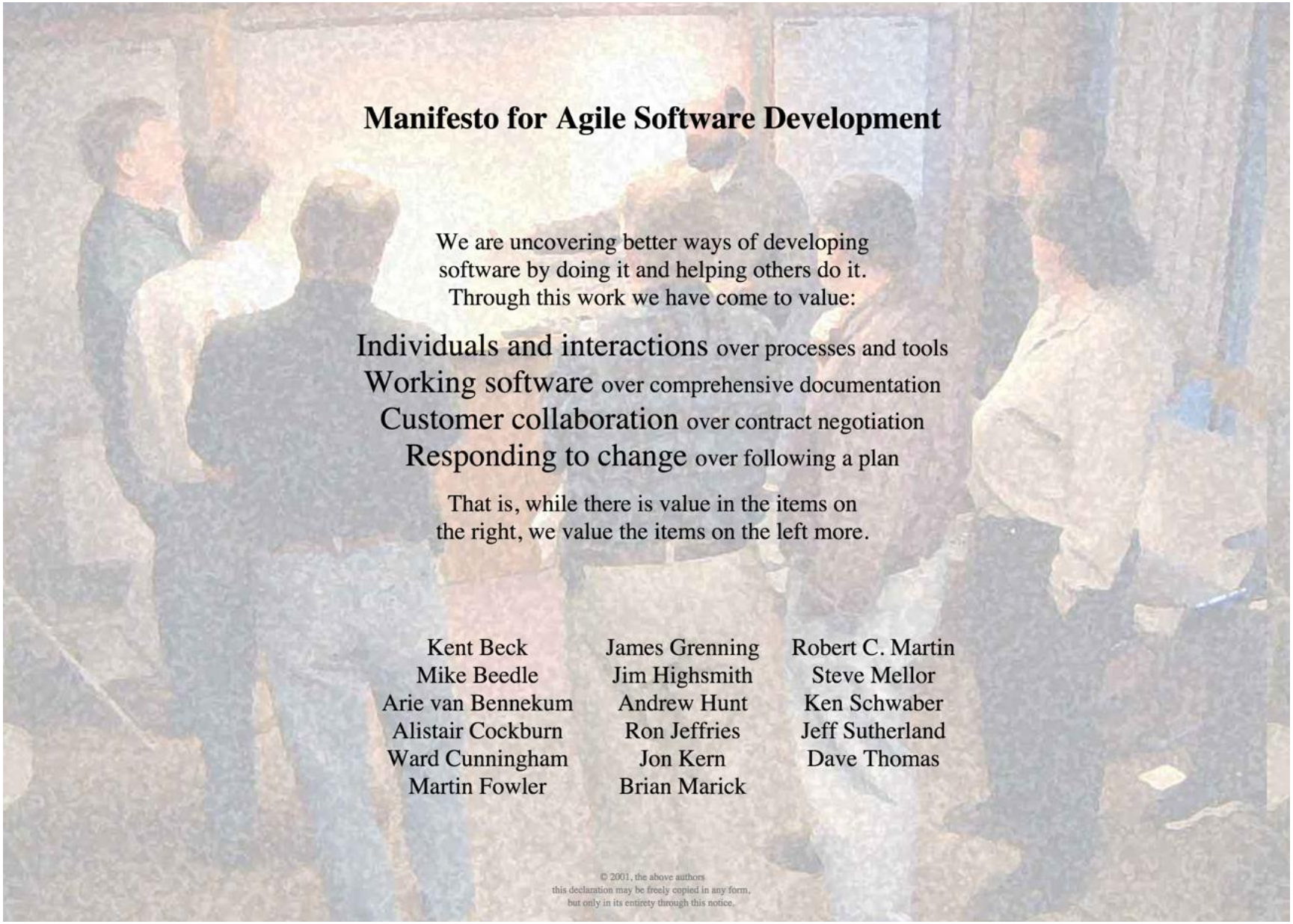
Meta-Lessons; or, the Big Picture

So what are we *really* saying?

Brace yourselves

FOR SOME DISCOURSE

Meta-Lesson I: Consensus < Plurality



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Meta-Lesson I: Consensus < Plurality

Excerpts from the Half-Arsed Manifesto:

Customer collaboration

over contract negotiation

(within the boundaries of strict contracts, of course, and subject to rigorous change control)

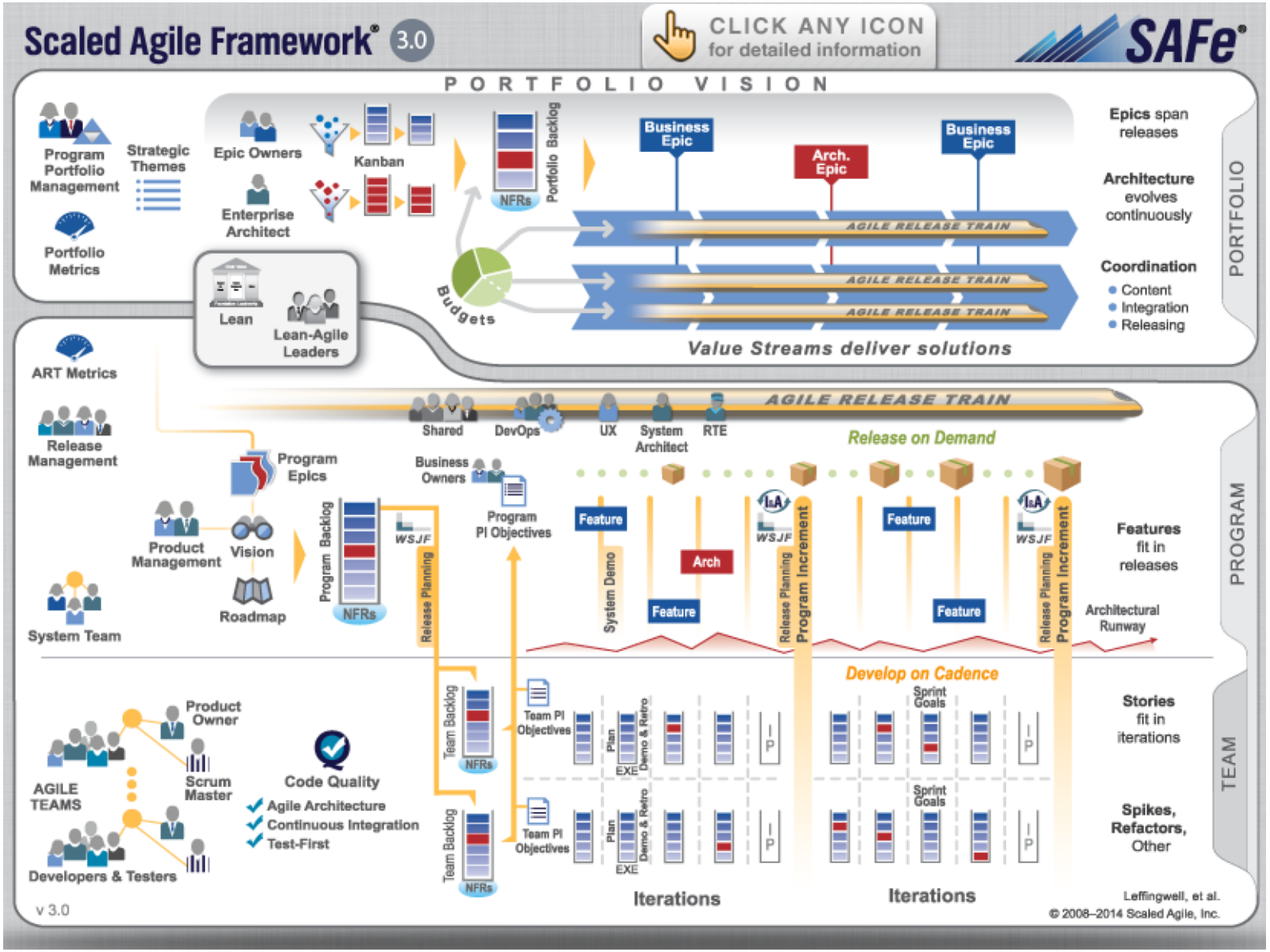
Responding to change

over following a plan

(provided a detailed plan is in place to respond to the change, and is followed precisely)

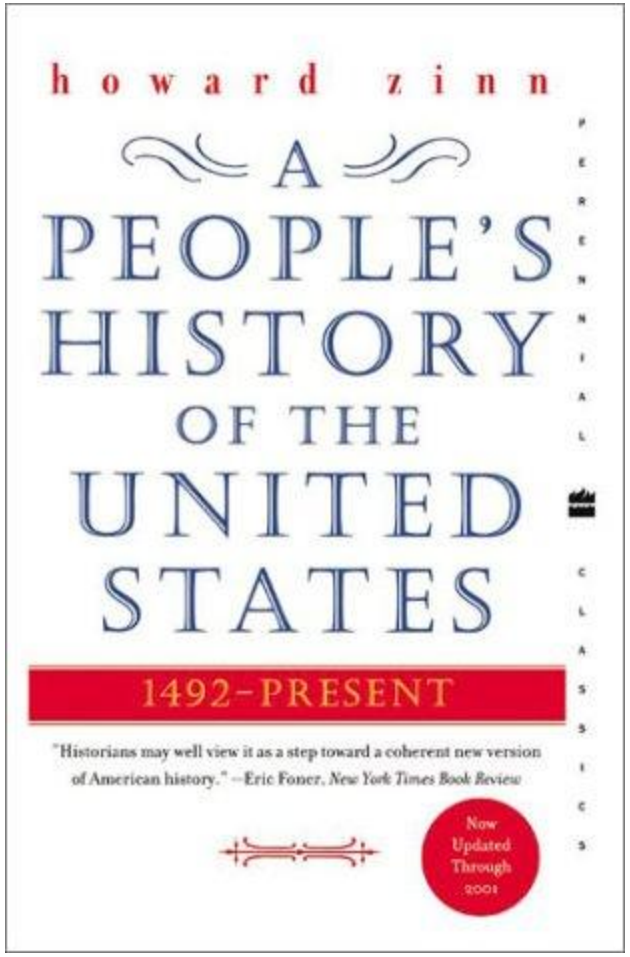
That is while the items on the left sound nice in theory, we're an enterprise company and there's no way we're letting go of the items on the right

“As Scrum is to the Agile team,

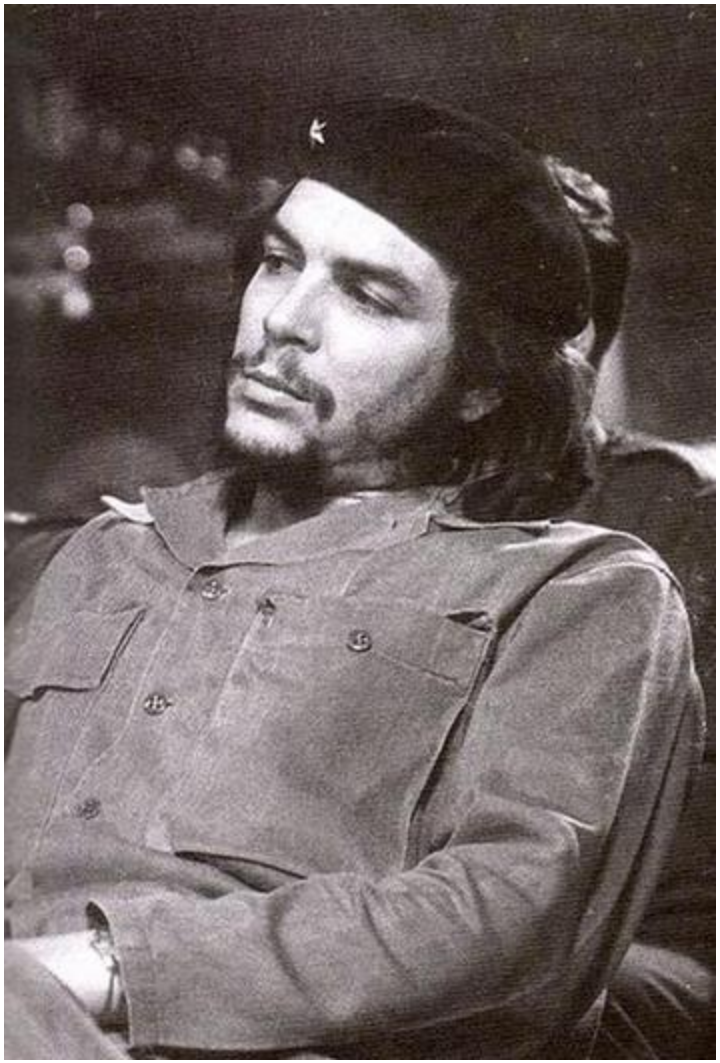


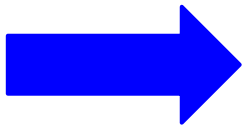
SAFe is to the Agile enterprise”

To be clear, plurality is GOOD

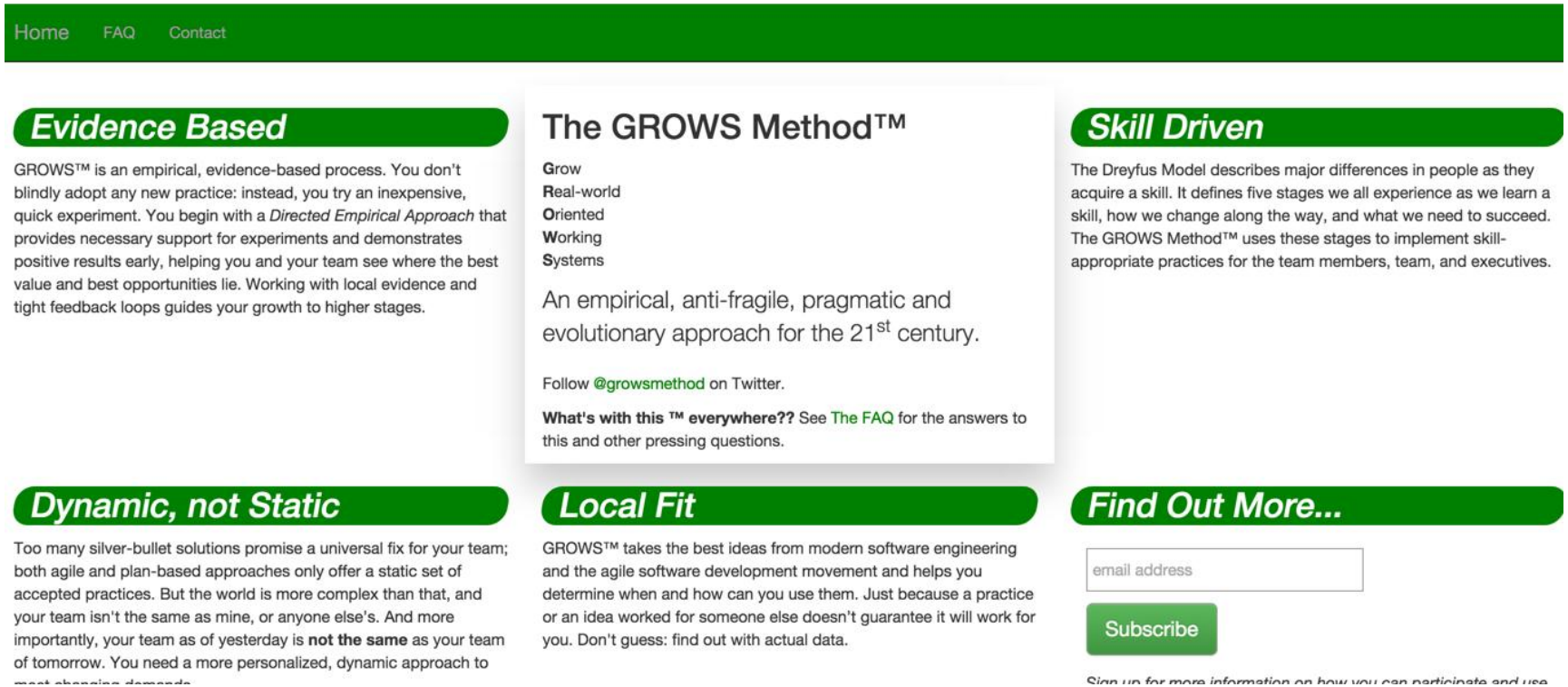


Meta-Lesson II:
Mainstream adoption is unavoidable





… and the cycle continues



A noble, but ultimately futile effort

from the GROWS website:

What's with the ™ all over the place?

We've seen first-hand how words like "agile" and "scrum" have been used and abused, and we'd like to avoid the same fate. Claiming GROWS as a trademark give us some measure of protection from companies misusing this word too.

We've seen first-hand how words like "agile" and "scrum" have been used and abused, and we'd like to avoid the same fate. Claiming GROWS as a trademark give us some measure of protection from companies misusing this word too.

So what can we do?

(tactically, that is)

TELL EVERYONE HOW WRONG THEY ARE!!!

~~TELL~~

~~EVERYONE HOW~~

~~WRONG THEY~~

~~ARE!!!~~

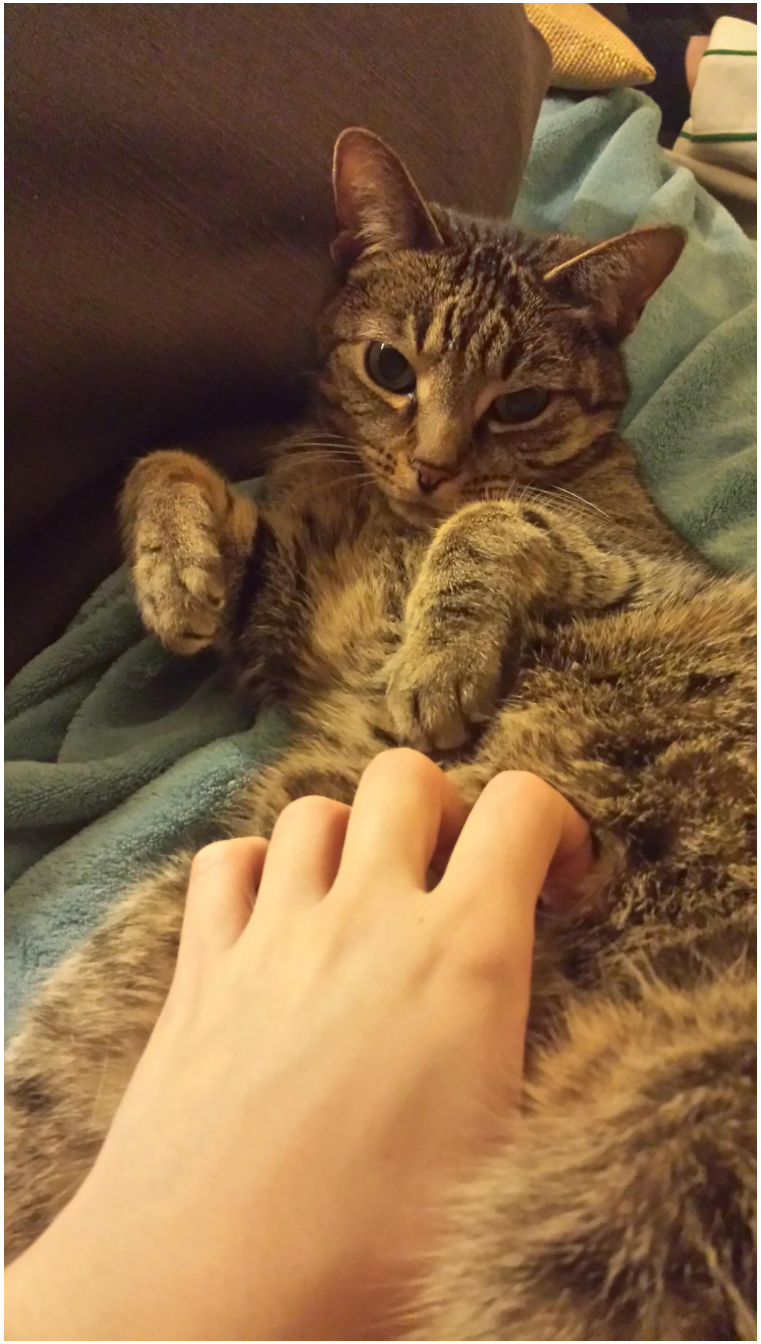
Communicate More Effectively!

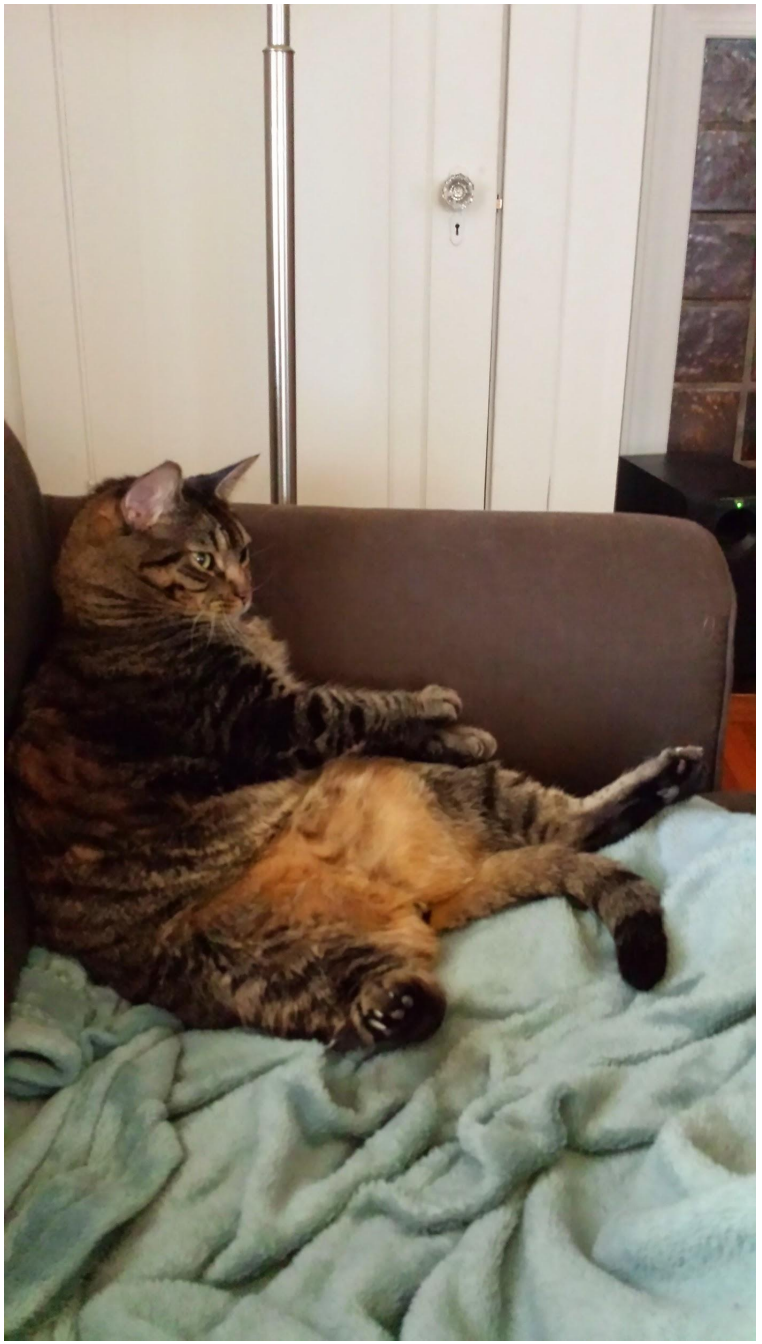
- Ask all the questions!
- Check for understanding, and use active listening
- Switch mediums
- Don’t use buzzwords; just say what you mean
  - If you *do* use them, use responsibly
- Use CRITICAL THINKING

… and now, some cat pictures!

(purr purr)

***No one* puts Baby in the corner!**






MANUFACTURING CONSENT
THE MORAL LANDSCAPE
SAM HARRIS
Hemingway
Jen Lancaster
KRUGMAN
Gina Kolata
Rethinking Thin
NO LOGO
MACHETE SEASON
OUR KIND
eternity
The Soul of Capitalism
BANANA
NAOMI KLEIN
Barbara Kingsolver
Prodigal Summer
BARBARA KINGSOLVER
THE LACUNA
CHIP KI
GOOD IS DEAD
THE ROA
MAKING COMICS
AL GORE
The ASSAULT ON REASON
blink
OUTLIERS
baby proof
emily giffin
something blue
Ever Since Darwin
BAD SCIENCE
BEN GOLDACRE
One Flew Over the Cuckoo's Nest
Guy Kawasaki
THE ART OF THE START
franz kafka
Ulysses
THE SCANDAL OF ULYSSES
Ulysses Annotated
UNDERSTANDING COMICS
SCOTT McCLOUD
Life of
GEORGE R.R. MARTIN
POLITICS OF PIETY
Torture the Artist
ARTURO CRUZ
SIGMUND FREUD
The Interpretation of Dreams
HALF-ASSED
Paulo Freire
PEDAGOGY OF THE OPPRESSED
FEMININE MYSTIQUE
Fifty Shades Freed
E L James
Fifty Shades Darker
Fifty Shades of Grey
THE ODYSSEY
STRAIN
HITCH-22
FREAKONOMICS
SUPERFREAKONOMICS

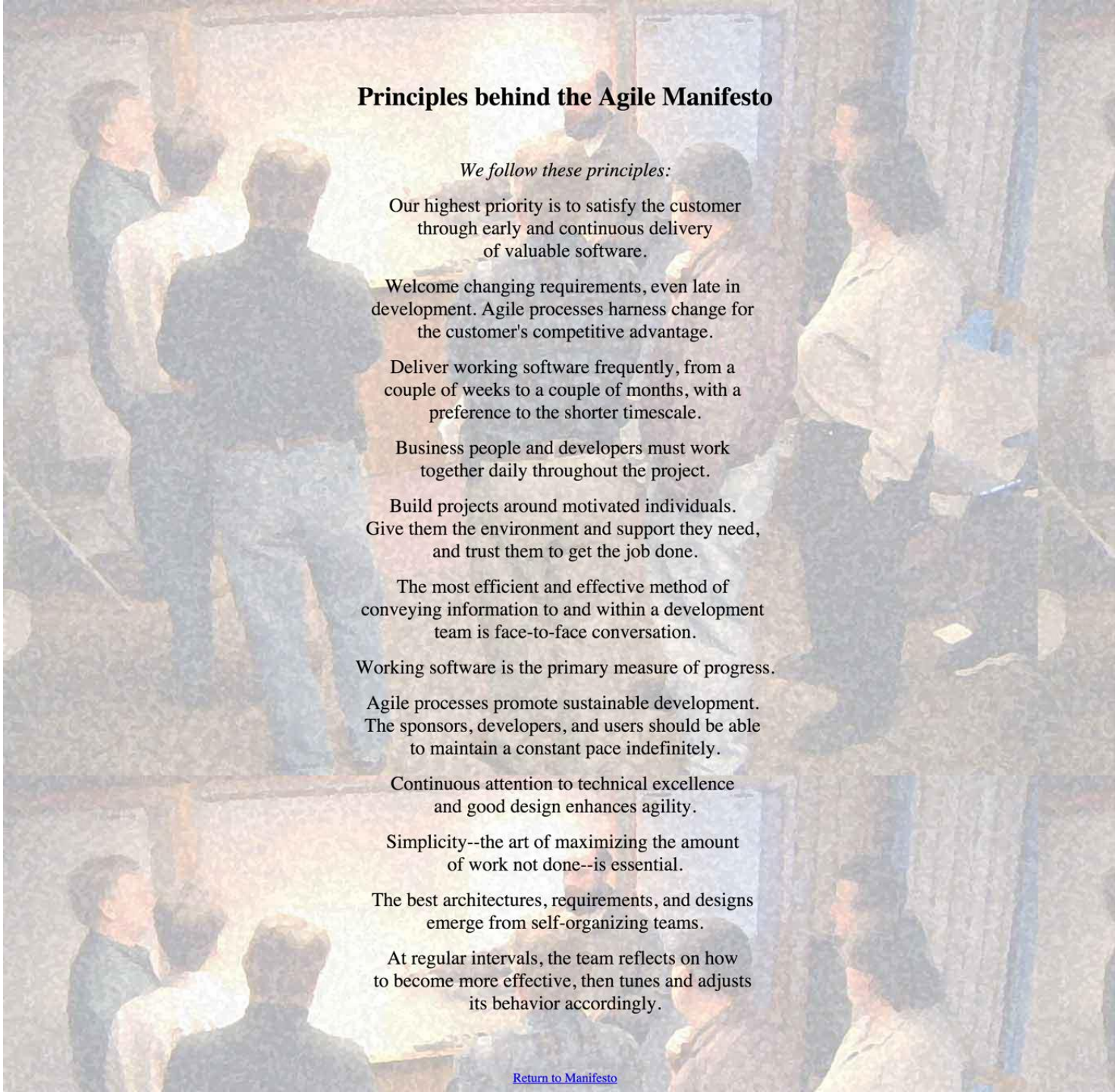
References: Happy Reading!

- How Spotify Builds Products by Henrik Kniberg
- The Lean Startup by Eric Ries
- Ward Cunningham on Technical Debt
- Doc Norton on Technical Debt
- Levent Gurses on Refactoring & ROI
- Manifesto for Agile Software Development
- Scaled Agile Framework
- Manifesto for Half-Arsed Agile Software Development
- Beyond Agile: New Principles? by Ron Jeffries
- The GROWS Method™ by Andy Hunt

Thanks for all the fish!

Carol (Nichols || Goulding) and Jenny Liu

@carols10cents and @liumonade



Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

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

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Return to Manifesto