Doug Knesek
Agile is Dead, and it Died in Infancy
Is “agile” the right word?

“effective”

“rapid”

“incremental”

“adaptive”

“collaborative”
What does “agile” really mean?

agile
adjective ag·ile \ˈa-jəl\ 
1 : marked by ready ability to move with quick easy grace · an agile dancer

How agile is your delivery?

The quickest your organization can change direction without having to shelve work is:

a) 5+ months  
   0 points

b) 3 months  
   1 point

c) 1 month  
   2 points

d) 1 week  
   3 points

e) 1 day  
   4 points
What is your cultural potential?

*Most* people in your organization:

a) Avoid debate with anyone.  
0 points

b) Debate a few trusted peers.  
1 point

c) Openly debate anyone on their team, including the Product Owner.  
2 points

d) Openly debate others, including managers, outside their team.  
3 points

e) Openly debate anyone they disagree with, including executives.  
4 points
What is your technical potential?

Our approach to technical debt is that we:

a) Don’t think about it.  
0 points

b) Treat known tech-debt like financial debt, accepting a certain level all the time.  
1 point

c) Expect zero known tech-debt by the time we release, or shortly thereafter.  
2 points

d) Expect zero known tech-debt at the end of each sprint.  
3 points

e) Expect zero known tech-debt with each frequent commit (every hour)  
4 points
Where do you fall?
Who the heck is this guy?
I’m Doug Knesek

Director of Agile Development & Coaching at Flexion.

In software development business for almost 30 years.

Agile apprentice for 19 years.
Why do agile frameworks fall short?
Christmas lights
A graph might look like this.

You get nothing...

until everything is fully aligned.
agile “lights” a.k.a. “practices”

Note: I use Extreme Programming practices in this presentation because this diagram provides an illustrative example. But the concepts apply to other agile frameworks.

human-built systems

human-designed

emergent/organic

F R A G I L E

By The Opte Project (2005) [CC BY 2.5], via Wikimedia Commons
Which can you do without?

- testing
- continuous integration
- collective ownership
- simple design
- short releases
- ...
hyper-productivity from frameworks

Some individual practices provide benefits.

But the “engine” doesn’t really start till it’s all there.
inverting the causal arrow

the creators...

frame/mindset → agile methods

deframe/mindset

the followers...

agile methods ← frame/mindset
the components you don’t see
So much can go wrong.
This has happened before.

Toyota Kata
By Mike Rother
©2010
“implementation mode”

Visible behaviors: 1937

Just implement these behaviors.

Benchmark: ~1990

2018

B.S.

 kanban
 heijunka
 poka-yoke
 jidoka
 andon
 Toyota Way
It gets worse.
mastery

How many of you have a master’s degree or PHD?

What if I told you that there were places in this very town where you could go to class for two to four days, take a one hour quiz, and walk away as a <discipline>-Master.
Caused shallow understanding of agility.

Caused the rapid expansion of “agile” awareness.
How well did these CSMs convey this?
seen through a PMI frame

LO³

WTF!? Two people at one keyboard?

Gold plating.

Our PO says we'll need it all anyway.

40-hour work week
simple-design

on-site customer
pair-programming
metaphor

refactoring
short-releases
continuous-integration
coding-standards
planning-game
testing

collective-ownership

40-hour work week
simple-design

on-site customer
pair-programming
metaphor

refactoring
short-releases
continuous-integration
coding-standards
planning-game
testing

collective-ownership

Nope. Gotta know who to blame.

LOL
What happens after a decade?

We think 2X is “success.”

...and we are blind to this gap.
It gets even worse.
We are being sucked into a black hole.
SAFe® is just more of the SAME.

We now think this is “success” at scale, too!
another lost decade

1. SAFe is growing fast and there’s no sign that it will get beyond “implementation mode.”
2. The number of people that professionally matured to think mediocre agile is the norm far exceeds those that have experienced high-performance (10X) agility.
3. The small proportion of authentic agilists is shrinking, and they do not have the power and influence to rescue true agility.
R.I.P.

AGILE SOFTWARE DEVELOPMENT
“Implemented” by many.
Achieved by few.
Is there a way forward?
Focus on first principles.

The way forward lies not in agile practices themselves, but in the principle that underlies them all.
the key insight

The power of “agile” practices stems from the options they enable and create.
Nature is all about the exploitation of optionality; it illustrates how optionality is a substitute for intelligence.”

optionality

asymmetric payoff that stems from the ability to cheaply create, cull, and exercise options

learning organization → optionality
TDD → optionality
testing → optionality
collective code ownership → optionality
simple design → optionality
psychological safety → optionality
refactoring → optionality
short releases → optionality
pair programming → optionality
feedback → optionality
planning game → optionality
architecture → optionality
use story format → optionality
enterprise hypothesis

An enterprise transforms from fragile to agile to antifragile by learning to efficiently create, cull, and exercise *options* at all scales. This is “Option-Based Thinking”.
Option-Based Thinking
uncharted territory

- Developing a product
- Becoming more agile
you could...

A) Draw your own map based on your vantage point, plot a path, and follow it.
alternatively...

B) Follow a compass heading and bring along trained frontiers-people and contingency supplies.
or maybe...

C) Do B plus throw in Star Trek communicators to enable remote communication and teleportation (to another communicator).
option-based thinking principles

1. Expand the option-space.
2. Increase the number and longevity of options.
3. Increase option dispersion and independence.
4. Reduce option cost.
5. Share options.
agile vs antifragile
Stages
five stages of optionality

blind

aware

enabling

seeking

generating
What stages do these achieve?

- generating...
- learning organization
- TDD
- testing
- simple design
- psychological safety
- use story format
- refactoring
- short releases
- pair programming
- architecture
- feedback
- planning game
- seeking
- enabling
TDD: option seeking

Use automated tests as experimental controls to search for a suitable software design option.
psychological safety: option enabling

Eliminate fear so that people will share options of all kinds.
architecture: option enabling

“A good architecture makes the system easy to change, in all the ways that it must change, by leaving options open.”

feedback: option seeking

Cull options and influence the next options to try.
user story format: option enabling

As a customer, I want a “sale” badge to appear on sale items.

As a customer, to easily see what’s on sale, I want a “sale” badge to appear on sale items.

Clearly separate the “why” from the “what” so that the supplier understands the goal and can propose different options for achieving it.
“Q → C”: option seeking

Quarantine then Combine

1. Expand the option-space.
2. Increase the number and longevity of options
3. Increase option dispersion and independence.
4. Reduce option cost.
Mechanism
improvement kata

visible
- poka-yoke
- kanban
- value-stream mapping
- heijunka
- andon
- 5S
- visual factory

invisible

eliminate waste +

Go and see

A
C
D
P
a similar pattern

**lean**

- poka-yoke
- kanban
- value-stream mapping
- jidoka
- visual factory
- heijunka
- andon
- 5S

**agile**

- simple design
- TDD
- refactoring
- short releases
- Q → C
- feedback
- sprint planning

**doing**

eliminate waste +

**searching**

improve optionality +

OBK
Option-Boundary Kata (OBK)

- kata - Rother
- optionality - Taleb
- (boundary) refactoring - Fowler

© 2018 Flexion, Inc.

www.flexion.us
Nassim Nicholas Taleb

Created the theory of antifragility based on optionality.

“The option is the agent of antifragility.”

https://youtu.be/B2-QCv-hChY
Martin Fowler

Pioneered emergent design through
● smells (symptoms)
● Refactorings

OBK refactors the whole system including the organization.

https://youtu.be/vqEg37e4Mkw
Mike Rother

Popularized Toyota’s Improvement and Coaching Kata

- problem solving framework
- learned by everyone
- permeates culture

https://youtu.be/fYoppX3VHIY
the high-level approach

Goal: Transform each boundary through the stages of optionality.

Process
1. Use “smells” to identify the problem.
2. Identify the associated boundary refactoring.
3. Use kata to problem-solve the refactoring.
a familiar analogy

1. Anatomy and physiology affects your health.
2. Your health manifests symptoms (or the lack thereof).
4. The selected treatment (and other factors) informs a procedure.
5. The procedure improves your anatomy and physiology.
1. Boundary and activities affect optionality.

2. The level of optionality manifests in a smell.

3. A smell suggests a refactoring via the "OBK Refactoring Catalog."

4. The refactoring informs the kata which is shaped by the specific boundary context.

5. Application of the kata improves the boundary and activities.

https://www.antifragileenterprise.org/option-boundary-kata
There are no smells remaining at your boundaries.
spontaneous agility

Imagine an organization full of people that spend a small portion of every day improving the optionality at boundaries that they develop or manage.

What would happen over the course of months? Years? Decades?
Call to arms (figuratively)
5 not-so-simple steps

1. Abandon the word “agile.”
2. Adopt option-based thinking.
4. Shift to improving optionality.
5. Spread the idea.
Thank you. Now it’s time to ask me *anything*.

Contact me at dknesekek@flexion.us if you want to learn more.

Access this presentation at: http://bit.ly/agile-rip