

You're Agile. So What?

David West, Ph.D.

TranscendenceCorporation.com



BERLIN
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE

gotocn.com

What you are about to hear is
deliberately provocative.

My intent is to challenge and
question, not to insult or anger.

If you are moved to think, and to
discuss, with me or amongst
yourselves, this hour will be a success.

Caveats

Agility is a good thing!

Agility has massive potential!

Agility enhances your professionalism!

Agile is 'majority mainstream'

But ...

bottom line (e.g. Chaos Report) shows only marginal improvement - that can be accounted for by the Hawthorne Effect

IT remains “irrelevant”

there is no consensus understanding of what Agile is

customer experience of IT is reluctant
wariness tempered with fear.

when everyone is agile, is anyone?

Exercise

On the Red card write two or three reasons why you wanted to become Agile.

On the Orange card write two or three ways that Agile has made your life better (or worse).

On the Yellow card write two or three ways - based on your experience - that your being Agile has made a difference to your employer/user/customer.

Agile was a protest movement!

“a bunch of Smalltalk programmers that were mad as hell they could no longer use that language.”

It was not the language - but the environment, the iterative, incremental, verified approach that was central to the Smalltalk experience.

Agile was intended to make the programmer's life better, to remove artificial constraints and micro-management - to allow the programmer to feel like and act like a professional.

Did It Work?

(red and orange Cards)

Agile had another motivation - one that requires a bit of history to understand.

in 1968, I became the first non-banker to work in the IT department of a major bank.

coincidentally, 1968 was the year that software engineer became an actual profession.

from the 1970s to the present, the relationship between IT and the enterprise changed dramatically, from business expressing naive trust and gratitude to IT, to business detesting IT and attempting to micro-manage, to mutual distrust and the contractual relationship of today - existing only to fix the blame when things inevitably went wrong.

Agile was intended to provide a means to reconcile and redefine the relationship between business and IT -- and “Crossing the Chasm” was a major focus of interest in the first few years of Agile adoption.

Did It Work?

(yellow cards)

*“But Mousie, thou art no thy lane,
In proving foresight may be vain:
The best-laid schemes o' mice an' men
Gang aft agley,
An' lea'e us nought but grief an' pain,
For promis'd joy!”*

Robert Burns

Things Go Awry (“gang agly”)

Christopher Alexander described what happens when a profession (in his case architecture) is defined and especially when it becomes an academic discipline.

A body of theory emerges based on abstract ‘first principles’. Standing in the profession (student and master) is based on adherence to that theory.

Application of the theory is denigrated (e.g. Software Engineering is NOT REAL Computer Science) and applications are evaluated solely on the basis of conformance to theory and not real world value or effectiveness.

Why Things Went Awry

A failure of nerve:

management resisted
practices like pair
programming and agile
workspaces

developers resisted acceptance of
responsibility, could not say no,
and put on the straightjacket of
practices instead of accepting the
challenge of values and principles.

dominance of old habits

projects and
project
managers

scrum masters
not coaches

production
process - lean

user stories
morphed to
specifications

the agile revolution echoed the object revolution - *“everyone
does it, every does it differently, and everyone does it wrong.”*

--- T. Rensch

A little bit of 'Chasm Irony'

Business' greatest goal is to be Agile (adaptive, innovative, learning, and sustainable) and the biggest single barrier to realizing this goal is the IT department -- including those that are "Agile."

If Agile is to be relevant, if IT is to be relevant, it must not only assure business realizes this goal, but must become foundational to the achievement of that goal.

Making Agile Relevant

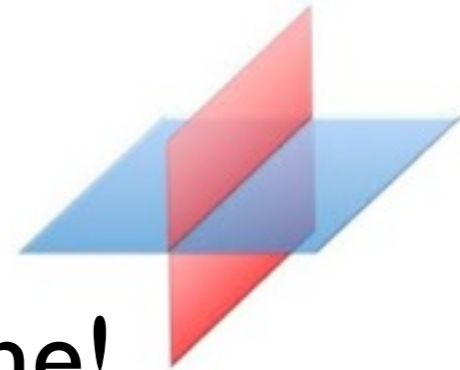
1. Eschew Computer Science, Software Engineering and the Cult of the Machine!

2.



Take the red pill!

Explore the blue plane!



3. leave Agile Software Development behind and adopt Agile Enterprise Evolution!

4. exhibit the courage and accept the responsibility, necessary to change the world!

Retain

1. the “story” as unit of work.
2. “on-site customer” - actually, redefine the “whole team” as a business team with developer craftspeople as integral members of all teams. (Temporarily include DBAs, Network admins, etc. - all the roles that exist primarily for the purpose of telling you what you cannot do.)
3. the original user story as narrative to guide the work (not the “as an X I want the system to Y” specification abomination).
4. the exploratory-iterative-incremental-deploy-immediate feedback rhythm

Agile Utopia

A dynamic, living, enterprise constantly generates stories about itself - i.e. about the elements that comprise the enterprise system and the relationships among those elements. Some of the stories document things to be improved, some innovative and creative ideas about what might be. All stories enter an “enterprise backlog” - visible throughout the organization - that is constantly prioritized, by consensus.

Agile craft-developers, from their position within business working groups, help realize the stories - one at-a-time - in a series of daily iterations capped at 10-15.

Business and IT are completely integrated and the entire organization operates and evolves in real time.

Customer Experience Bliss

You provide the ultimate customer experience - however customer is defined - when you provide no experience at all. You become the 'invisible' yet indispensable