10th IFPUG International Software Measurement & Analysis (ISMA10) Conference
Charlotte, N. C., USA – April 30, 2015

Implications and Opportunities Relevant to "Agile" Measurement

Joe Schofield

joescho@joejr.com

Questions to be Answered . . .

- **The use of “agile” is widespread as advertised, but what does that actually mean?**
- **An abbreviated survey of “agile” methods—how do these align with the “Manifesto”?**
- **What are some current numbers for “agile” usage?**
- **Story points, use case points, function points – what’s the point?**
- **Suggestions for enabling organizational learning for “agile” use and estimating**

Abstract

"Agile" has been changing the world of software development for 20 years (as of 2015). Today, at least a dozen different methodologies are included under the umbrella known as "agile." Many of these methodologies are limited in their alignment with the oft cited "Agile Manifesto." Implementation variants can predictably render unpredictable results. Meaningful measurement for organizational planning and especially benchmakring becomes increasingly elusive. The CMMI Institute and others are working with the agile community to address some of these opportunities. As measurement professionals, we share an interest and responsibility—not only for functional and non-functional measurement—but also for process, improvement, and estimation.

XP

Feature-Driven Development

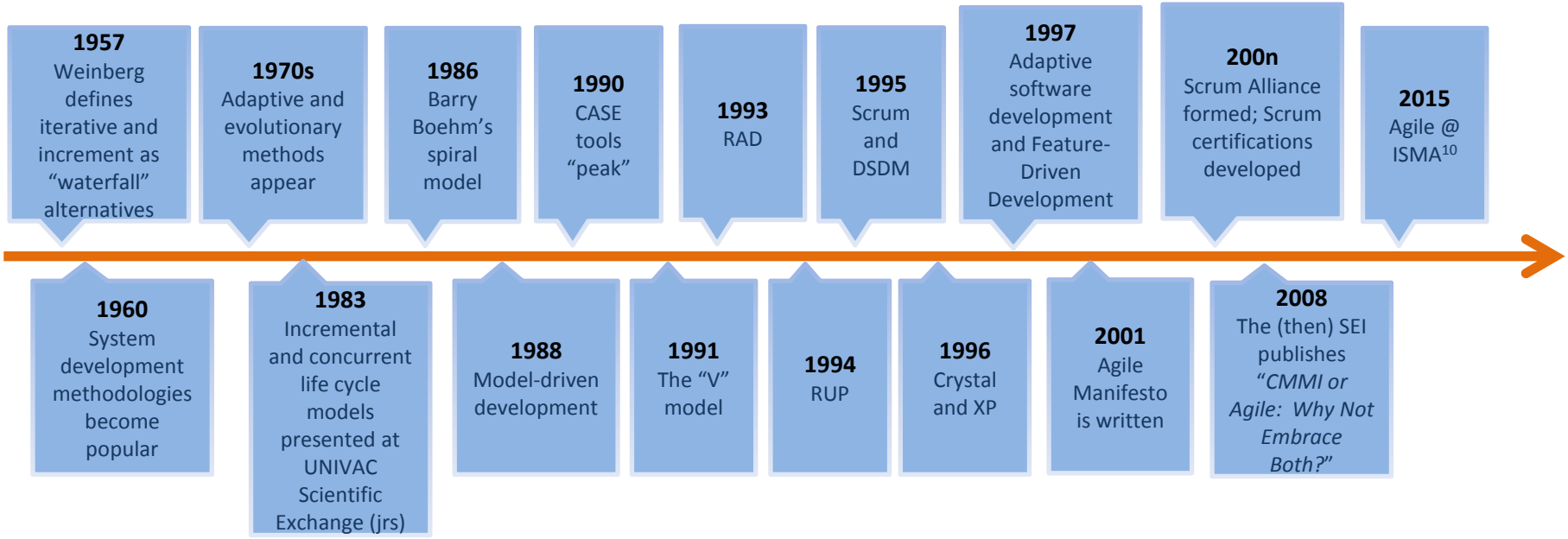
The Manifesto

Crystal Clear

Kanban

Disciplined Agile Delivery

Abbreviated History of the World
(all dates circa)



Well-known Agile Development Methods

- Adaptive Software Development (ASD)
- Agile Modeling
- Agile Unified Process (AUP)
- Crystal Methods (Crystal Clear)
- Disciplined Agile Delivery
- Dynamic Systems Development Method (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Lean software development
- Kanban (development)
- Scrum
- Scrum-ban

retrieved Wikipedia, 8/30/2014

"Agile" Quirks

Did You Know?

- **Feature-Driven Development – Individual ownership of code (vs team)**
- **Dynamic Systems Development Methodology – not lightweight, not undemanding; phases and system-oriented; demonstrating control is a core value**
- **Test-Driven Development – includes no learning curve; documentation consists of test cases and code ONLY**
- **Crystal Clear red and maroon (levels of rigor) are highly-defined and rigorous**
- **Kanban – has little to fulfill manifesto principles 5, 6, 9, 10, 11, 12 (ScrumBody BoK pg. 105 – 106)**

"Agile" – It's all good— wait what?

CrossTalk; May / June, 2013

All agile methods make it easy to oversimplify complexity. In fact, agile's strength of eliminating complexity might be better stated as "ignoring complexity."
***Why Agile isn't enough (and why it doesn't work);* Zacharias Beckman; April 23, 2010**

(thus) Disciplined Agile Delivery (DAD)

Background:

- **Developed by Scott Ambler, chief methodologist at IBM Rational (Summer 2006 to Summer 2012)**
- **Intended to address the process gaps in Scrum**
- **Scalable to the enterprise**

WHY?

"Many agile methodologies — including Scrum, XP, AM, Agile Data, Kanban, and more — focus on a subset of the activities required to deliver a solution from project initiation to delivery. Before DAD was developed, you needed to cobble together your own agile methodology . . .” – Scott Ambler

Disciplined Agile Delivery Meets CMMI (Cutter IT Journal, November 2013)

"Agile" – needs a context for meaningful communication

Please don't tell me you're doing "agile." Really, what does that mean to anyone who actually knows something about agile?

At least twelve different development approaches now claim a spot under the "agile" umbrella.[1] Many of these are incompatible. Many of these defy articles of the Agile Manifesto. Some of these are as heavy-weighted as any traditional methodology[2].

As comforting as those words might be to conventionalists, they portray a dichotomy with those who claim "if it's written down, it's not agile."

[1] retrieved Wikipedia, Agile Software Development, 8/30/2014: Adaptive Software Development (ASD); Agile Modeling; Agile Unified Process (AUP); Crystal Methods (Crystal Clear); Disciplined Agile Delivery; Dynamic Systems Development Method (DSDM); Extreme Programming (XP); Feature Driven Development (FDD); Lean software development; Kanban (development); Scrum; Scrum-ban

[2] Crystal (maroon) and DSDM, as examples

Some worst practices

Any reason to doubt that a survey of organizations using "agile" found 85 percent of those organizations had experienced some level of failure with "agile"?[5]

[5] VERSIONONE; 8th Annual State of Agile™ Survey; 2014

Wrong Motives, Bad Practices

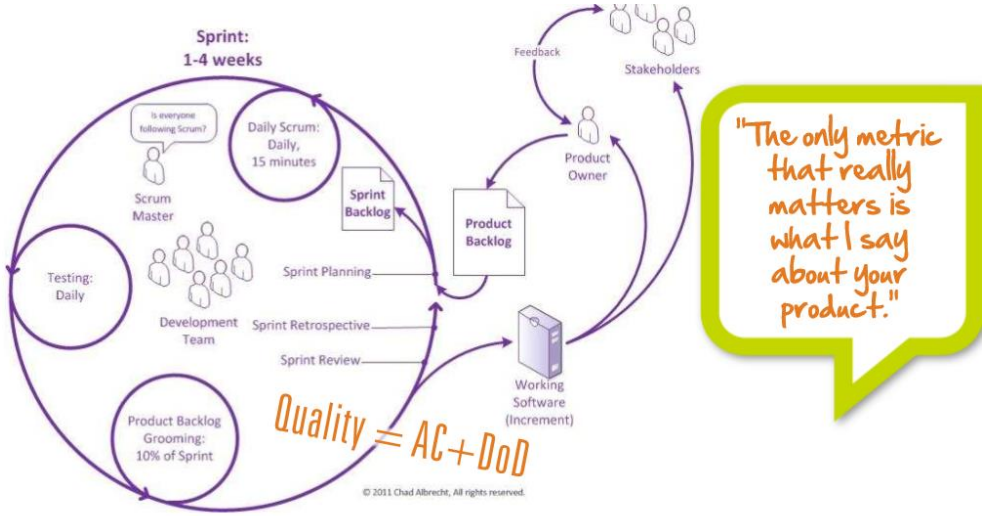
Use "agile" because:

- **Everybody else is doing it**
- **"so and so" is doing everything with agile . . .**
- **Use "it" because "waterfall" didn't work**

Sprint practice:

- **Move sprint demonstrations until sprint stories are ready for prime time**
- **Let sprints run until the work is done**
- **Stack test sprints after development sprints – ignore the "potentially shippable product"**

Is the customer the *only* "judge" of goodness?



Often, and regrettably, this is the message associated with agile products:

- Products built for use internal to an organization must be sustainable rather than riddled with technical debt issues
- "Customers" have an abundance of negative "things to say" about most of the world's best selling software—sorry, it's just not a valid assertion

Measure Challenges with "Agile"

Estimation spectrum from T-shirt sizes to 1 to 10, to Fibonacci scales – sounds good.

Story points, what value beyond the team? Where is *organizational* learning – the 5th Discipline?

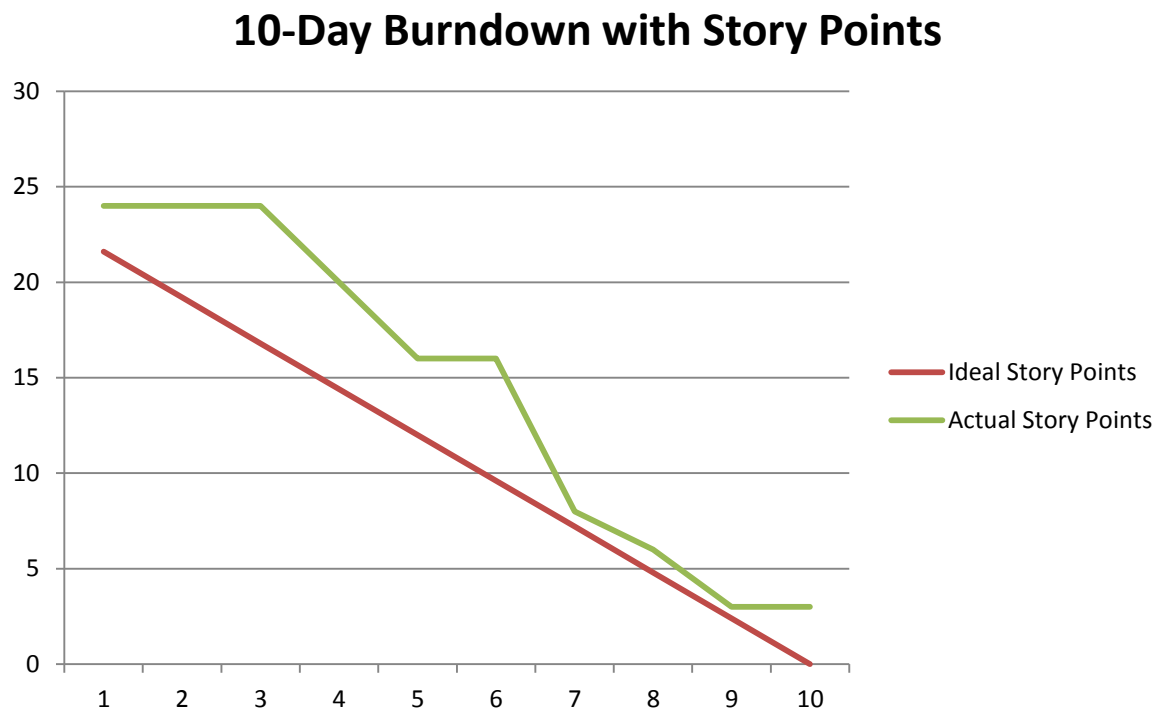
Other sources of estimation error during sprint planning:

- **Inflate story points to give an illusion of increased velocity**
- **Productivity time (%)**
- **Doubling "dipping" on what's included in productive time (e-mails, phone calls, meetings) {count meetings a non-productive time, but then including them and their time in story tasks}**
- **Inflating required tasks for story completion – "Overstuffing" tasks with refactoring**
- **Inflating tasks time for story completion**

Veracity with Velocity

Did the team:

A. Overestimate
B. Under perform
C. Neither
D. Both



An Industry Summary of Story Points as a Benchmark

(across projects and across organizations!)

Story Points, Use Case Points, Function Points—What's the Point?

Characteristic	Function Points	Use Case Points	Story Points
Useful at the project level for estimating or planning	With historical FP data	With historical UCP data	With historical SP data
ISO / Standards based	ISO 20926	no	no
Captures customer view	Expected	Expected	Definitely
Useful for benchmarking outside the company	Could be	Could be	Less so
Easy to calculate	Less so	More so	Yes
Easy to validate for repeatability / consistency	More so	More so	Less so
Objectivity	More so	More so	Less so (team / team member variability)
Technologically independent	Yes	Yes	Maybe
Functional measurement to customer	Yes	Yes	Not exclusively (may include refactoring, design, and other work)

Deeper Reading *Function Points, Use Case Points, Story Points: Observations from a Case Study*; CrossTalk; May / June, 2013

All is NOT lost . . . (keep the baby) – Pt. 1

- **Optics / visualization. "Agile" is often cited for its transparency and visual cues:**
  - ❑ **product roadmap**
  - ❑ **task boards**
  - ❑ **burndown charts**
  - ❑ **velocity charts**
- **The role of the product owner**
- **Definition of *done***
- **Daily stand-ups**
- **"As a" statements "AS a *PERSONA*, I WANT / NEED SO THAT . . .**

Deeper Reading *Keep the Baby*; MetricViews; Winter, 2015

All is NOT lost . . . (keep the baby) – Pt. 2

- **Sprint demos / reviews**
- **Retrospectives. Another MEETING! Don't scream**
- **Strong focus on teaming**
- **Using recent performance to estimate work being planned**
- **More definition and discipline than advertised**

Deeper Reading *Keep the Baby*; MetricViews; Winter, 2015

"Agile" Discipline & Structure is Rarely Promoted

From the earliest days of the "agile" movement "agile" was used as the anti-venom for waterfall reinforcing an "anything but that" mentality. Clearly not all the "agile" world intended that anything non-waterfall would fall into the "agile" bucket. Jim Highsmith once noted:

"The Agile movement is not anti-methodology, in fact, many of us want to restore credibility to the word methodology. We want to restore a balance. We embrace modeling, but not in order to file some diagram in a dusty corporate repository. We embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes. We plan, but recognize the limits of planning in a turbulent environment.[7]

[7] *History: The Agile Manifesto;* Jim Highsmith; 2001; agilemanifesto.org

Re-visit: Questions to be Answered . . .

- The use of “agile” is widespread as advertised, but what does that actually mean?
- An abbreviated survey of “agile” methods—how do these align with the “Manifesto”? (6)
- What are some current numbers for “agile” usage? (13, 14, 15)
- Story points, use case points, function points – what’s the point? (16)
- Suggestions for enabling organizational learning for “agile” use and estimating (18 – 20)

About the presenter . . .

CFPS, SMC, AEC, CSQA, CSMS, Certified CMMI Instructor

http://joejr.com/presentd.htm **(~55)**

http://joejr.com/publishd.htm **(~36)**

- Joe Schofield is President Emeritus of The International Function Point Users Group. He retired from Sandia National Laboratories as a Distinguished Member of the Technical Staff after a 31-year career. During twelve of those years he served as the SEPG Chair for an organization of about 400 personnel which was awarded a SW-CMM® Level 3 in 2005. He served as the migration lead to CMMI® Level 4 until his departure. He continues that commitment today by helping his clients excel in software and business improvement and measurement.
- Joe has facilitated over 100 teams in the areas of software specification, team building and organizational planning by using lean six sigma and business process reengineering. Joe has taught graduate courses over three decades. He was a licensed girl's mid-school basketball coach for 21 seasons--the last five undefeated, over a span of 50 games.
- He has over 80 published books, papers, conference presentations and keynotes—including contributions to the books *The IFPUG Guide to IT and Software Measurement (2012), IT Measurement, Certified Function Point Specialist Exam Guide, and The Economics of Software Quality.* He is a CMMI Institute-certified Instructor for the Introduction to the CMMI®, Certified Software Quality Analyst, Certified Function Point Specialist, a Certified Software Measurement Specialist, a Certified Scrum Master and most recently a Certified Agile Expert.
- Joe completed his Master's degree in MIS at the University of Arizona in 1980. By "others" he is known as a husband, father, son, and grandfather.