Scrum
Are organisations missing the point?

My views
Devils advocate
The beginning

“The New New Product Development Game”
Hirotaka Takeuchi and Ikujiro Nonaka
Harvard Business Review - 1986

6 Characteristics from leading companies

Fuji-Xerox - Canon - Honda - NEC - Epson - Brother - 3M - Xerox - Hewlett-Packard

Built-in instability
Self-organising project teams
Overlapping development phases
Multilearning
Subtle control
Organisational transfer of learning

Built-in Instability
Top management provides goals the challenge to the team to find solutions. This is known as dipping in to the chaos domain in Complexity Theory to allow innovation to emerge.

Self-organising project teams
Agile Manifesto - Scrum - Lean .... it's everywhere....
- Autonomy
- Self-transcendence
- Cross fertilisation

Overlapping Development Phases
Non-sequential

Multilearning
Two dimensions
- Multiple levels
- Multiple functions

Subtle Control
Team are largely on their own but not uncontrolled. Management establishes enough check points to prevent instability turning in to chaos. But avoids rigid control which impairs creativity and spontaneity.

Organisational transfer of learning
Transferring outside of the team
“Left to stew, the process begins to create its own dynamic order. The project team begins to operate like a start-up company - it takes initiatives and risks, and develops an independent agenda. At some point, the team begins to create its own concept.”

–Takeuchi & Nonaka

Quote from the publication
“The Honda team, for example, consisted of hand-picked members from R&D, production, and sales. The company went a step further by placing a wide variety of personalities on the team. Such diversity fostered new ideas and concepts.”

–Takeuchi & Nonaka
That was
1986
30 years ago
In February 2001, Jeff and Ken “Manifesto for Agile Software Development.” in Utah

In 2001, much inspired by Kent Beck, Ken Schwaber co-authored the first book on Scrum with Mike Beedle, Agile Software Development with Scrum.

In 2002, Ken Schwaber founded the Scrum Alliance with Mike Cohn and Esther Derby, with Ken chairing the organization. In the years to follow the highly successful Certified ScrumMaster programs and its derivatives were created and launched.

In 2006, Jeff Sutherland created his own company, Scrum.inc, while continuing to offer and teach the Certified Scrum courses.

Ken left the Scrum Alliance in the fall of 2009, and founded Scrum.org to further improve the quality and effectiveness of Scrum, mainly through the Professional Scrum series.

With the first publication of the Scrum Guide in 2010, and its incremental updates in 2011 and 2013, Jeff and Ken established the globally recognised body of knowledge of Scrum.
First published guide to deal with the misunderstandings of Scrum
Iterative development was called Scrum
Development sprints / Testing Sprints was called Scrum
Waterfall projects with 15 minute stand-ups was called Scrum
Various iterations of the guide was published, 2010, 2011, 2013, and each time getting lighter
Guide adapted to the needs of the market
Scrum has changed

http://agileforest.com/2012/02/19/scrum-evolution-over-time-part-1/
Reality

• Stories are the new specification document
• Business analysts being sole requirement gatherers
• Scoping and estimating entire project in detail
• Overtime and weekend working to meet commitments

Not having conversations to emerge requirements
3 amigos at minimum to help find solutions not just the BA
We can’t know everything in advance, so why try to design and estimate everything
Non-negotiable scope leads to overtime and burnt out teams
PM's head on the block, so they will not allow teams to adapt the process to find better ways of working. This is mainly caused by fear, and risk to the single person accountable who has their reputation at risk.
Iterative Only

Incremental development alone means we are not incrementally building a useable product.
This does not exist, but this is what big organisations want. A package they can simply buy and get results from. It does not work this way.
This is result.
Scrum is not want part of the overall management of the project, Scrum is the way to manage the overall project.
Gateways and sign-offs
Rigid process controls to retain consistency across multiple teams...kills process innovation.
People are at the heart of our work….human systems. Understand Cognitive Science, Neuroscience, sociology.
Agile Manifesto value - “People and interactions over process and tools”
People, mastery, expertise, discipline OVER TOOLS
Interactions create process
Definition of Ready

Should be enough to start.

As long as you are confident you will get what is needed to finish.

Can lead to prescriptive processes....use with caution!
Sprint length is fixed - end of!
But they want to flex to fit time slots....and at the same time they want predictability. Then you need to look beyond simply velocities.
Cargo Cult

Being agile is different to doing Agile

http://www.jamesshore.com/Blog/Cargo-Cult-Agile.html
Doing Agile
Being agile
Scrum is a recipe

We need to chefs who know how to cook!
Scrum - 2013
Scrum - 2013 revision

• Artefact Transparency

• Sprint Planning is now one event and Sprint Goal

• The Product Backlog is refined rather than groomed

• The importance of the Daily Scrum as a planning event is reinforced

• The concept of value is reinforced to use in the Sprint Review
What it is

• Framework for building complex products with high levels of uncertainty

• For building a potentially shippable increment in less than a calendar month that satisfies a goal

• For delivering goals instead of requirements

• Uses inspection & adaption from team through to product to evolve the future state based on empirical evidence

• With motivated, self-organising teams that are left to manage how they turn a backlog item into releasable increment

• Business and IT uncertainty
• Get something done in 30 days
• Goals not requirements
• Inspect & Adapt - Standups, Planning part 1, Sprint Review, Team Retrospective
• Team Autonomy - let them find a natural way to manage their work
Scrum Theory

“Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known.”

“Scrum employs an iterative, incremental approach to optimise predictability and control risk.”
The Scrum Team

“Self-organising teams choose how best to accomplish their work, rather than being directed by others outside the team.”

“Incremental deliveries of ‘Done’ backlog items ensures a potentially useful version of working product is always available.”
The Product Owner

“The Product Owner is one person, not a committee.”

“For the Product Owner to succeed, the entire organisation must respect his or her decisions.”
The Development Team

- **They are self-organising.** No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;

- **Development Teams are cross-functional.** with all of the skills as a team necessary to create a product Increment;

- **Scrum recognises no titles for Development Team members other than Developer,** regardless of the work being performed by the person; there are no exceptions to this rule;

- **Scrum recognises no sub-teams in the Development Team,** regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule; and,

- Individual Development Team members **may have specialised skills** and areas of focus, **but accountability belongs to the Development Team as a whole.**
Sprint Planning

“Work planned for the **first days** of the Sprint by the Development Team is **decomposed** by the end of this meeting, **often to units of one day or less**.”

“The **Development Team self-organises** to undertake the work in the Sprint Backlog, both during Sprint Planning and as needed throughout the Sprint.”

now 8 hours for 1 month sprint
shorter for smaller sprints
No plan survives first contact

Military are some of the best agile teams in the world. They rapidly adapt to all kinds of situations because their repetitive training has taught them to respond by second nature without having to think.

Sprint plans often don’t survive the opening days of a sprint, so why plan out the entire sprint extensively with all stories broken down into tasks and all estimates in hours.
The Sprint Goal

“Created during the Sprint Planning meeting”

“The Sprint Goal is an objective set for the Sprint”

“…gives the development team some flexibility regarding the functionality implemented within the Sprint.”
The Sprint

“The heart of Scrum is a Sprint, a time-box of one month or less during which a ‘Done’, useable, and potentially releasable product Increment is created.”

During the Sprint:

- **No changes are made that would endanger the Sprint Goal**;
- Quality goals do not decrease; and,
- **Scope may be clarified and re-negotiated** between the Product Owner and Development Team as more is learned.

Changes are allowed to happen in a sprint, as long as they do not risk the sprint goal.
“In Scrum you are not finished when you complete the requirements, you are finished when you complete the goal”

- Ken Schwaber
  Keynote Scrum Insights - AgileVancouver - 2008
“This is when you'll see teams artificially slicing work into inappropriate chunks simply for the purpose of fitting the work into the sprint rather than delivering the value needed by the business.”

Ken Schwaber

We are mis-using the purpose of a time-box to meet the needs of a goal
Daily Scrum

It is a planning event not a status meeting
Sprint Review

It’s an inspect and adapt of the Product Increment

It is not just a demo - that is one part of it!
Transparency
Is the backlog hidden?
Is it hard to find items within in?

Is there hidden work in the sprint?

Is the Definition of Done transparent to all?

Are the issues during sprint discussed in the sprint review?
A bad burn down, which is often not made transparent
Velocity is a range, so a single precision delivery date is not making the uncertainty of delivery transparent.
This make the delivery transparent
Probabilistic Forecasting

Monte Carlo Estimation

Completed stories in past sprints
5 7 11 12 4 7 8 10 15 4

How many stories are in the new project? 100

Perform 10,000 simulations

Results

Full range: 7.7 - 18.3 sprints
90% confidence: 9.8 - 14.6 sprints
Median: 12.1 sprints

Likelihood of project completion

http://monte-carlo-estimation.meteor.com/

Try probabilistic Forecasting

http://monte-carlo-estimation.meteor.com/
ScrumBan

Bring flow management into Scrum.
Manage the flow of upstream and downstream activities.

Product refinement session can end up with developments opening up the code to remove uncertainty. This is work, but it's hidden work.
use control charts against story point sizing to see how much variance their is on your story points.

I have recorded 3 point stories over the course of a year and found they range between 2 days to 12 days to complete.

You can do this yourself but making a note of the date the work started and the date it was finished. The difference is the cycle time.
**Using the right tool for the job**

<table>
<thead>
<tr>
<th>Tools</th>
<th>Features</th>
</tr>
</thead>
</table>
| Leankit | - Visualise the workflow  
|        | - Visualising the different work  
|        | - Service policies for work types  
|        | - Control WIP limits  
|        | - Make process explicit  
|        | - See who is working on what  
|        | - Detailed metrics |

Use the right tools for the Job.

Tools like Rally and Jira claim to provide Kanban support, but it is basic and often inadequate.
Agile is about having a toolbox of techniques they can be applied in different situations.
Embrace change!

“It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to change.”

-Charles Darwin, 1809
Scrum

• Used for building complex products development with high levels of uncertainty

• Uses a timebox that should allow for the emergence of new requirements as well as descoping to achieve a sprint goal of working software.

• Uses sprints of up to 30 days which create a potentially shippable product