The New Methodology

Review on Agile Methods

CSCW 2015

Paper Author:
Martin Fowler
@ThoughtWorks

Review by:
Jingxiang Gou
M2 HCID @UPS XI

Why the New Methods

- reasons for agile methods
- adaptive nature
- people-first orientation

Paper Contents

1. From Nothing, to Monumental, to Agile
2. Predictive Vs. Adaptive
3. Putting People First
4. The Self-Adaptive Process
5. Flavors of Agile Development
6. Should you go agile?

Paper Contents

From Nothing, to Monumental, to Agile

Predictive Vs. Adaptive

Putting People First

The Self-Adaptive Process

Flavors of Agile Development

Should you go agile?

From Nothing, to Agile

Engineering methods

plan-driven methodologies

Predictive?

Process oriented

Document oriented

Agile methods

test-driven methodologies

Adaptive!

People-oriented

Code-oriented

Paper Contents

From Nothing, to Monumental, to Agile

Predictive Vs. Adaptive

Putting People First

The Self-Adaptive Process

Flavors of Agile Development

Should you go agile?

Predictive Vs. Adaptive

Separation of Design & Construction

Design:
difficult to predict
expensive & creative people

Construction:
easier to predict

Predictive Vs. Adaptive

Crucial QUESTION !

Can you get a design:

1 that is capable of turning the coding into a predictable construction activity?

2 If so, is cost of doing this sufficiently small to make this approach worthwhile?

look very good on paper, yet be seriously flawed when you actually have to program the project

False Conclusion on Engineering

Jack Reeves: “anything that you can treat as construction can and should be automated.”

- In software: construction is so cheap as to be free
- In software: all the effort is design, and thus requires creative and talented people
- Creative processes are not easily planned, and so predictability may well be an impossible target.
- be very wary of the traditional engineering metaphor for building software.

It's a different kind of activity and requires a different process

Unpredictability of Requirements

Requirements, are always Changing !

- understanding the requirement is though
- software dev is a design activity
- basic materials keep changing rapidly
- difficult to see value of a feature until use for real
- no stable requirements = no predictable plan

Question: what we do about it ?

Misunderstanding Requirements

leads to people using a methodology
in the wrong circumstances, such as:

using a predictable methodology in
an unpredictable situation

What to Do?

a process that
can give you control over an unpredictability.

Adaptivity

Control Unpredictable Process

Adaptivity with iterations

TEST !

Why tests?

1 bugs

2 misunderstood requirements

Control Unpredictable Process

adaptivity with iterations

TEST !

Why tests?

1 bugs

2 misunderstood requirements

Key question:

How long the iteration should be ?

Depending on each agile method

With Adaptive Customer

Agile approach:

fix time, fix money, vary scope

1 constant reworking of plan

2 risk control

3 a late change in requirement: competitive advantage

With Adaptive Customer

Business success?

On-time? on-cost?(predictive methods)
Business Value ! (Agile methods)

The customer get the software
with more value than the cost

Paper Contents

From Nothing, to Monumental, to Agile

Predictive Vs. Adaptive

Putting People First

The Self-Adaptive Process

Flavors of Agile Development

Should you go agile?

Putting people first

synergy:

adaptivity require a strong team

most good developers prefer an adaptive process

Putting people first

plug-compatible programming units Engineering

- Frederick Taylor's Scientific Management approach.
- “Achilles heel” of measurement based management.
- leads to high levels of measurement dysfunction.

responsible professionals Agile

- accepting the process rather than imposition of a process
- developers make all technical decisions

+ Business Leadership

Paper Contents

From Nothing, to Monumental, to Agile

Predictive Vs. Adaptive

Putting People First

The Self-Adaptive Process

Flavors of Agile Development

Should you go agile?

Self-Adaptive Process

each iteration:

What did we do well?
What have we learned?
What can we do better?
What puzzles us?

leads to ideas to change for the next iteration

Paper Contents

From Nothing, to Monumental, to Agile

Predictive Vs. Adaptive

Putting People First

The Self-Adaptive Process

Flavors of Agile Development

Should you go agile?

Flavors of Agile Development

1 Extreme Programming

Communication, Feedback, Simplicity, Courage, Respect

14 principles, 24 practice

2 Scrum

sprints

daily scrum meetings

3 Crystal

family of methods

safety (in project outcome), efficiency, habitability (developers can live with crystal)

4 Context Driven Testing

questioning mainstream testing thinking

5 Lean Development

overlaps & inspirations between lean production and software development

6 Unified Process

a process framework

Use Case Driven

problem: its infinite variability

Paper Contents

From Nothing, to Monumental, to Agile

Predictive Vs. Adaptive

Putting People First

The Self-Adaptive Process

Flavors of Agile Development

Should you go agile?

Go Agile?

NOT for everyone, BUT could be used by more

To start with:

1 suitable projects to try agile

team & customer want to work with it

project more critical than comfortable with

2 find a mentor experienced in agile

3 follow their advice try methods first

4 hard to tell where the boundary

WHERE NOT to use? never impose on a team who doesn’t want it

Thank you !