Experimental Agile

PUTTING PEOPLE OVER PROCESS TO THE TEST

David Wallace

SR. SCRUM MASTER, DELUXE CORPORATION

The Agile Manifesto

What the Agile Manifesto values*:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

*From “Manifesto for Agile Software Development” @ agilemanifesto.org

People over Process

WHAT DOES THAT MEAN?

First, what it DOESN’T mean:

We don’t care about process and tools

It’s the wild wild west – anything goes

Each person can do what works best for them, screw everyone else

Individual desires outweigh team decisions

What “People over Process” DOES mean

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*

The best architectures, requirements, and designs emerge from self-organizing teams.*

There is no prescribed way to practice Agile Development

Not every team will do things the same way

If the team wants to try something that doesn’t violate the other Agile values, they should be free to try it

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.*

*From “Principles behind the Agile Manifesto” @ agilemanifesto.org/principles.html

Putting it into Practice

EXPERIMENTING WITH AGILE

Why to Experiment in Agile

One of the key principles of Agile is to “Inspect and Adapt”

We can’t get better by doing everything the same way all the time

There are no “Best Practices” only “Common Practices”

You don’t know if it will work until you try it

Just because it worked for you before doesn’t mean it will work for you now

Just because it didn’t work for you before doesn’t mean it won’t work for you now

How To Experiment in Agile

Time-boxed experiments

Measurable experiments

Anyone can propose an experiment

The team decides if they want to try the experiment

Retrospectives can be a great place to get ideas for and propose experiments

The team decides if the experiment was a success

Learn from the experiment regardless of the outcome

Examples

No Hours Estimation and Capacity Planning

Problems to Solve

Considerable time wasted doing task estimates

Commitment is as a whole team, estimates are individual

Estimates were introducing an artificial time constraint on the stories

Foundation of Reasoning

The team had a stable velocity

Backlog was mostly new code development

Small team

Proposed Experiment

Plan based on:

- Velocity
- Items
- Team's overall feeling

Keep focus on quality and delivering value

Try this for 2 Sprints

Hypothesis

This experiment would be considered successful if:

- The team maintained their commitments
- Their velocity did not decrease
- The team is satisfied that it will work in the long term

Experiment Results

Sprint 1 Results:

- Team met their commitment of points and items
- The sprint did not turn into a 2 week waterfall with all items in the QA bucket at the end
- Team decided to continue the experiment with next sprint

Sprint 2 Results:

- Team exceeded their sprint commitment of points and items
- Team has decided to keep an eye on this but make it part of their formal process to not do hours estimates
- Risk is that if we add new team members we will have to train/educate on non-hours estimation and no capacity planning; if we have a major change in velocity this could also become an issue

SUCCESS

Tech Debt - Kanban

Problem to Solve

Bug Bashing as part of Tech Debt

Maximizing the amount of Tech Debt completed in a Sprint

Foundation of Reasoning

Team had been working together for quite a while

Team was doing a good job of delivering Product stories to QA early in the Sprint

Team was motivated to:

- Reduce number of bugs
- Complete more Tech Debt

Proposed Experiment

Plan our Product Commitment as usual (80% of available team time)

Have a prioritized backlog of Tech Debt/Bugs

As the team completes the Product Commitment they will pull from the top of the Tech Debt backlog

Complete as much Tech Debt as possible in the remaining 20% of the Sprint

Duration of Experiment to be 4 Sprints

Hypothesis

This experiment would be considered successful if:

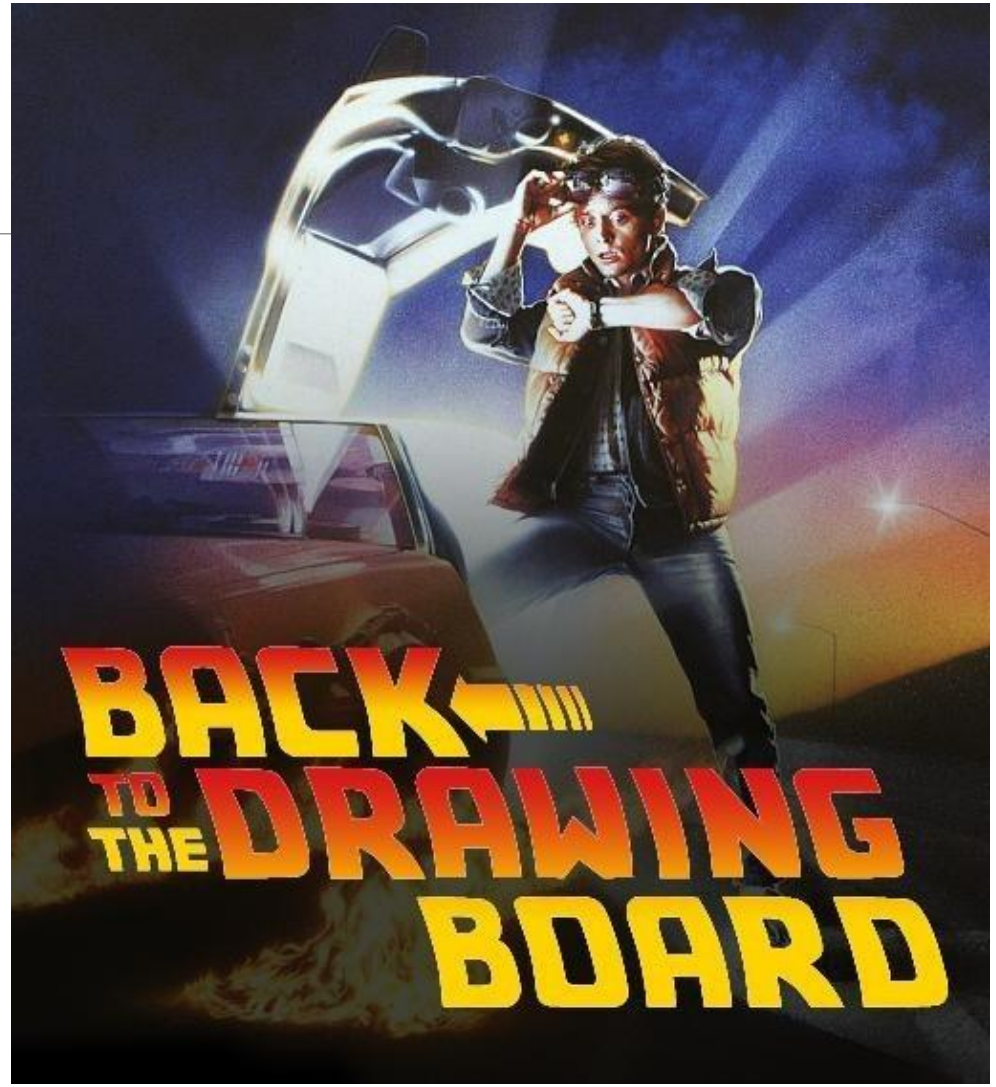
- The number of Tech Debt issues/Bugs Closed increased
- Product stories continue to get to QA early in the Sprint

Experiment: Failure

Actual Tech Debt Percentage:

- First Sprint: 1%
- Second Sprint: 16%
- Third Sprint: 0%
- Fourth Sprint: 0%

Decided to return to planning commitments for Tech Debt from prioritized backlog



A Failed Experiment is not a Failure

You will never know if you can do something better unless you try to do something different than you are already doing

We can learn from our mistakes

A failed experiment may give you ideas for another experiment that will work

Breakout Discussion #1

Two Week Waterfall

Your teams work in two week Sprints

One of your teams is consistently getting their work to QA late in the Sprint making it difficult for them to complete all of the testing in time to meet commitments

What is an experiment you could try to address this issue

How would you execute and measure the experiment to determine success?

Breakout Discussion #2

Real World Issues

Go around the table and discuss real issues you are currently having with teams you are working with

Let others at the table suggest experiments you could try in order to address that issue

David Wallace

HTTPS://WWW.LINKEDIN.COM/IN/DAVIDWALLACE3

DWALLACE1971@GMAIL.COM