Why Agile?

The Economics, Psychology, and Science of Agile’s Success

@MatthewRenze

#PrairieCode
Purpose

Explain why Agile practices are so successful
Insights from Economics, Psychology, and Science
Top 7 most important ideas
Ideas that are not typically covered
Overview

1. The World after Midnight
2. Inverted Constraints
3. Prioritizing Value
4. Embracing Change
5. Self-Organization
6. Effective Communication
7. Feedback
About Me

Independent software consultant

Education
- B.S. in Computer Science
- B.A. in Philosophy

Community
- Public Speaker
- Pluralsight Author
- Microsoft MVP
- ASPInsider
- Open-Source Software
A Brief Review of Agile
What is Agile?

Agile Manifesto
- 4 value propositions
- 12 principles

Common practices

What is Agile?

Agile is *not*:  
A methodology itself  
A magic silver bullet

Source: http://www.best-story.net/userfiles/silver-bullets.jpg
Agile 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

Source: http://agilemanifesto.org/
12 Principles of Agile

1. Continuous delivery of value
2. Embrace changing requirements
3. Frequent deployment
4. Customer collaboration
5. Motivated individuals
6. Face-to-face conversation
12 Principles of Agile

7. Working software as measure of progress
8. Sustainable development
9. Technical excellence
10. Simplicity
11. Self-organization
12. Continuous improvement
Agile Methodologies

Scrum
XP
Kanban
Lean

Source: http://parkertoddloesch.files.wordpress.com/2011/09/umbrella.jpg
Is Agile More Successful?

Original Source: The Standish Group, The CHAOS Report 2012
Agile = Good
Waterfall = Bad
Waterfall

Agile
Waterfall vs Agile
1. The World after Midnight
The World After Midnight

About fifteen years ago all the ‘Rules’ about how to run a business, organization, or government successfully, were changed or deleted and a completely new set of ‘Rules’ has been in operation ever since, which means that we keep acting rationally in response to a world we recognize and understand... but which no longer exists!

- Eddie Obeng
The World After Midnight
Source: http://www.ted.com/talks/eddie_obeng_smart_failure_for_a_fast_changing_world.html
The World after Midnight

Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg
The World after Midnight

Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg
The World after Midnight

Old World

- Identify problem
- Gather info
- Design product
- Build product
- Get to market
- Capture revenue

Midnight

- Build prototype
- Get it to market
- Capture feedback
- Capture revenue
- Iterate on design
- Adapt over time

New World

Past

Time ->

Now

Rate of Learning
Rate of Change

Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg
The World after Midnight

Original Source: http://pentacle.co.uk/Downloads/InnovationResources/Old%20New%20World.jpg
Laminar Flow vs. Turbulent Flow

Source: http://www-mdp.eng.cam.ac.uk/web/library/enginfo/aerothermaldvd_only/aero/fprops/pipeflow/node8.html
Laminar Flow vs. Turbulent Flow

Laminar Flow vs. Turbulent Flow

Ebbs, Flows, and Residual Impact of Business Fads

Why is this important?

**Problem**
World has changed
Markets change rapidly
Requirements change rapidly
High degree of uncertainty

**Solution**
Adapt to new physics
Faster time-to-market
Better response to change
Continuous and rapid feedback
Agile is very well suited to operate in the physics of this new world!
2. Inverted Constraints
Four Levers of Software Development

Scope
Resources
Schedule
Quality

Source: http://farm6.staticflickr.com/5300/5521479079_36815225e4_z.jpg
Four Levers of Software Development

Working software
Max value
Min cost

Source: http://farm6.staticflickr.com/5300/5521479079_36815225e4_z.jpg
Constraints

Restriction on freedom
Prevents achieving goal

Examples
  Time
  Money
  Talent

Source: http://www.myspaceantics.com
Constrained Optimization

ISO-Values objective function $5x_1 + 3x_2$

optimal vertex $x_1 = 10, x_2 = 20$

$x_1 + 2x_2 = 50$

$2x_1 + x_2 = 40$

Source: http://home.ubalt.edu/ntsbarsh/business-stat/opre/partVIII.htm
Too much to math!
Waterfall Constraints

- Scope
- Fixed
- Estimated

Waterfall (plan driven)

- Resources
- Schedule
Waterfall Constraints

Waterfall (plan driven)

Scope
Resources
Schedule

Estimated

Fixed
Fixed
Waterfall Constraints

Waterfall (plan driven)

Scope

Resources

Schedule

Fixed
Agile Constraints

Waterfall
(plan-driven)

Agile
(value-driven)
Agile Constraints

Fixed team size
Fixed releases
Estimated features
Team controls quality

Resources

Schedule

Fixed

Estimated

Scope

Agile
(value-driven)
### Why is This Important?

<table>
<thead>
<tr>
<th><strong>Problem</strong></th>
<th><strong>Solution</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mythical man-month</td>
<td>Limit team size</td>
</tr>
<tr>
<td>Slipping release dates</td>
<td>Fix schedule</td>
</tr>
<tr>
<td>Scope creep</td>
<td>Estimate scope</td>
</tr>
<tr>
<td>Technical debt</td>
<td>Protect quality</td>
</tr>
</tbody>
</table>
Agile is more flexible
3. Prioritizing Value
Quick Lesson in Economics

1. Return on Investment
2. Pareto Principle
3. Opportunity Cost

Source: http://myhomeworkhelp.com/economics-homework-help/
Return on Investment

\[ ROI = \frac{Value - Cost}{Cost} \]

High ROI => lots of value  
Low ROI => some value  
Neg. ROI => lost value
Return on Investment

\[ ROI = \frac{Value - Cost}{Cost} \]

High ROI => lots of value
Low ROI => some value
Neg. ROI => lost value

ROI Curve for an Investment

Time

Break even
Return on Investment

Each feature has ROI
  Cost to develop
  Value to business

Project ROI is sum of feature ROIs
Goal is to maximize ROI
Pareto Principle

80/20 rule

Power law function

Diminishing marginal returns
Pareto Principle

80/20 rule
Power law function
Diminishing marginal returns
Pareto Principle of Software Feature Usage

Features
20% of features
80% of value

Traditional software is
20% high-value features
80% low-value features

Source: Standish Group
Opportunity Cost

Opportunity Cost

Cost of foregone alternative options
True cost = explicit cost + implicit cost
Must be included in cost-benefit analysis

Source: http://www.stus.com/
Prioritizing Features by Business Value

Product backlog
List of features
Ordered by business value
Highest priority on top
Create and deliver in order
## Why is This Important?

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Need to maximize ROI</td>
<td>Prioritize features by ROI</td>
</tr>
<tr>
<td>Low-value features</td>
<td>Deliver highest-value first</td>
</tr>
<tr>
<td>Opportunity cost</td>
<td>Prioritize features relative</td>
</tr>
</tbody>
</table>
Agile Produces More Value

Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/
4. Embracing Change
Waterfall’s Key Assumption

Source: Doug DeCarlo – eXtreme Project Management
Waterfall’s Key Assumption

Plan:
Start ——> Finish

Actual:
Start ——> Finish

Source: Doug DeCarlo – eXtreme Project Management
Waterfall Assumptions

Users actually know what they want
Markets will not change during development
There is nothing new or unknown
Technology is stable and mature
All of the pieces will fit together in the end
Waterfall Reality

Requirements are not stable
Requirements are just assumptions
Cost of Fixing Defects in Waterfall

Finding Defects in Waterfall

http://www.agilemodeling.com/essays/costofchange.htm
Finding Defects in Agile

http://www.agilemodeling.com/essays/costofchange.htm
Cost of Change in Agile

Source: http://www.agilemodeling.com/essays/costOfChange.htm
Why is This Important?

**Problem**
- Requirements change
- Fixing defects late is costly
- Late changes are costly

**Solution**
- Embrace change
- Fix defects early
- Build in flexibility
Agile is More Adaptable

Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/
5. Self-Organization
How do you determine the price to charge for a loaf of bread?
Market Economy

Market makes decisions
Produces and consumers
Supply and demand
Millions of decisions

Source: Britannica
Market Economy

Goal: Maximize social welfare
Competitive market equilibrium
Extremely efficient
“Chaotic success”

Source: https://content.dodea.edu/VS/HS/DVHS_Courses/Economics/syllabus.html
Complex Adaptive Systems

System
  collection of interconnected things

Complex
  dynamic network of interactions

Adaptive
  changes in response to environment
to increase survivability

Source: http://integral-options.blogspot.com/2013/03/peter-fryer-brief-description-of.html
Inversion of Control

Top-down
Command and Control
Bureaucracy

Inversion of Control

Top-down
Command and Control
Bureaucracy
vs.
Bottom-up
Self-organization
Adhocracy

Source: http://funnyasduck.net/post/10458
Wisdom of the Crowd

Collective guesses of crowd
Aggregate better than expert
Only some types of knowledge
Not all crowds are wise!
Why is This Important?

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-down is inefficient</td>
<td>Self-organizing teams</td>
</tr>
<tr>
<td>Poor information flow</td>
<td>Invert control to bottom-up</td>
</tr>
<tr>
<td>Ineffective decisions</td>
<td>Wisdom of the Crowds</td>
</tr>
</tbody>
</table>
Self-organizing Agile teams are more efficient
6. Effective Communication
Cost of Poor Communication

Cost is enormous
Hard to quantify
Hidden cost
Expense is real

Source: http://www.cathy.willman.com/2012/06/what-boys-need.html
Cost of Poor Communication

17.5 hrs / person / week

Top 5 issues identified:

1. Waiting for information
2. Unwanted communication
3. Inefficient coordination
4. Barriers to collaboration
5. Customer complaints

Total estimated annual cost of poor communication per enterprise knowledge worker: $50,562

Communication Structures

Fully-connected graph
   Nodes = people
   Edges = channels

Edges increase by $O(n^2)$

Becomes inefficient very fast

Effectiveness of Communication
Signal-to-Noise Ratio

SNR = $P(\text{signal}) / P(\text{noise})$

Signal = message

Noise = everything else

Goal is to maximize SNR

Source: http://uber.la/2012/05/signal-to-noise/
Visibility

Waterfall hides problems
High visibility at start
Low visibility at middle
High visibility at end

Original source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/
Visibility

Agile provides visibility
On the surface with visibility
Problems have no where to hide

Original source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/
<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication overload</td>
<td>Small teams</td>
</tr>
<tr>
<td>Cost of poor communication</td>
<td>Maximize signal-to-noise ratio</td>
</tr>
<tr>
<td>Lack of transparency</td>
<td>Increase visibility</td>
</tr>
</tbody>
</table>
Agile teams communicate more effectively
7. Feedback
Slightly less frigid, giving the user false hope

Back to frostbite

Perfect!

The temperature at which you would boil a lobster to death

Tundra

Arctic

Antarctic

Cone of Uncertainty

Feedback and Learning

Learning reduces uncertainty
Feedback is necessary
Continuous and rapid feedback

Source: http://www.icanhascheezburger.com
Agile Feedback

Continuous and rapid feedback
Multiple timescales

Powerful for:
- Learning
- Reducing risk
- Eliminating Uncertainty

Source: http://www.agile-process.org/communicate.html
Smart Failure

Short and frequent experiments
Low cost and high value
Old world vs. new world
Requires mindset change

It’s Not OK to Fail BIG!

Source: http://t4toby.files.wordpress.com/2008/07/epicfail1.jpg/
Know When to Pivot

Pivot = change direction
Assumptions incorrect => pivot
Pivot early, not late
Minimize cost to pivot

# Why is This Important?

<table>
<thead>
<tr>
<th><strong>Problem</strong></th>
<th><strong>Solution</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cone of uncertainty</td>
<td>Feedback</td>
</tr>
<tr>
<td>Avoid epic failure</td>
<td>Embrace smart failure</td>
</tr>
<tr>
<td>Difficulty changing course late</td>
<td>Minimize cost to learn</td>
</tr>
</tbody>
</table>
Agile Teams Use Feedback to Reduce Risk

Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/
Know When to Stop

Marginal Return on Investment (Feature)

mROI = 0

mROI > 0

mROI < 0

Time / Cost / Effort

Agile stops here

Waterfall stops here

Original Source: http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/
Know When to Stop

- Everything else:
  - The Cost of Complexity
  - Eliminating Waste
  - Inventory Hides Problems
  - Metrics Have Consequences
  - Embracing Human Factors
  - Information Gain / Entropy
  - Embedded Documentation
  - Kanban and Queuing Theory
  - TDD, Dopamine, and Crack
  - Sustainable Development
  - Agile is an Emergent Property
  - and much more...

Conclusion
Why is Agile so Successful?

1. It is well adapted to the world after midnight.
2. It inverts its constraints to be more flexible.
3. It maximizes ROI by prioritizing features by value.
4. It is more adaptable by embracing change.
5. It utilizes the efficiencies of self-organization.
6. It produces more effective communication.
7. It reduces risk by continuous and rapid feedback.
My Website

Articles
Courses
Presentations
Source Code
Videos

www.matthewrenze.com
Feedback

Feedback is very important to me!
One thing you liked?
One thing I could improve?
Contact Info

Matthew Renze
Data Science Consultant
Renze Consulting

Twitter: @matthewrenze
Email: info@matthewrenze.com
Website: www.matthewrenze.com

Thank You! : )