Agile Beyond Software

Jake Calabrese
@jcalabrese
July 25th 2016
Jake Calabrese

For any updated slides, blogs, and worksheets (e.g. A3) – Go to:
www.agileforall.com/agile-beyond-software-resources

- Coach & Trainer
- Focused on Enterprises, Leadership, and Human Systems
- Organization & Relationships Systems Certified Coach (ORSCC)
- jake@agileforall.com
- @jcalabrese
- agileforall.com/blog/jcalabrese
Organizational Challenges

- Complexity
- Speed
- Socialization
- Globalization
- Humans
Are We Limiting Agile?

Software Agile vs. Agile
Imagine...

You are at a conference

and you get a text message
I keep hearing about this agile thing...

Our IT group is using it and appears successful – people keep talking about how much they are delivering.

I asked them for help, but they said agile is only for software???
I keep hearing about this agile thing…
Our IT group is using it and appears successful – people keep talking about how much they are delivering.
I asked them for help, but they said agile is only for software???
I remember you saying agile can work anywhere… can you help?
Uh… sure… let’s talk
What else do you know?
Your friend is Chief Operating Officer
Her company creates books, training courses, and manuals. They have customers worldwide.
Her role is involved in operations, products, distribution, sales, marketing, and HR
What will you tell her?
You start thinking

- You know a lot about agile in software
- You could talk about the manifesto or agile practices
- You start getting excited!
Has anyone ever asked you:

“Do you think you are too enthusiast about agile?!”

Your Reply:

“I don’t understand the question?”
Let’s Explore Common Failure Patterns
Failure Pattern: Brain Dump

the agile manifesto... that was in 2001 so agile is not a fad and it started before that in like the 90s with scrum or even with a Harvard business review paper in ’86... before that there was Toyota and before that... well now you get my point... it’s been around a long long time... and there is scrum like I mentioned and kanban... people like to battle between them, but they can get along. But they are all part of agile... We should talk about estimating as well cause it can be problematic... we use story points based on Fibonacci or modified Fibonacci cause why use 21 and 39 or whatever... those are too specific right... and there are people talking about #noestimates, but they are not all saying to not estimate... we should probably talk about Taylorism and scientific management and why that does not work... oh and I guess we never covered stories... stories are PBIs or parts of features, really of MMFs. They should have the 3 C’s and using INVEST helps with splitting...
Result:
People pass out
Failure Pattern: Overly critical
Result: Disengagement
Failure pattern: Too idealistic

Have you tried this?

- Dude!!! Agile is SOOO Kool!
- A manifesto man!
- Sustainable pace – sprints dude!
- Scrum is so cool... in scrum you try to use the word scrum as many times as you can... Scrum, Daily Scrum, Scrum of Scrums... I know you’re like NO WAY... but WAY!
- And Scaling is radical... like people are packaging up practices and selling it ... WHOW
- Dude... did I mention the manifesto?

tag:calabrese
Result: People look at you like you’re this guy.

@jcalabrese
Failure pattern:

Forgetting the Value of Agile

1. Inspect and adapt
2. Team empowerment
3. Collaboration
4. Working software
5. Customer involvement
6. Embrace complexity
7. Embrace change
8. Sustainable pace
9. Self-organizing
10. Incremental delivery
11. Trust
12. Simplicity
13. Deliver fast
14. Respect people
15. Eliminate waste
16. Optimize flow
17. Decide as late as possible
18. Limit WIP
19. Optimize the whole
20. Empiricism
21. Feedback
22. Respect
23. Focus
24. Transparency
25. Openness
26. Commitment
27. Courage
28. Mitigate risks
29. Antifragile relationships
30. Relentless learning
31. Minimum Viable Product
32.
Result: Confusion & Loss of Focus
Avoid!
• Brain dump
• Overly critical
• Too idealistic
• Forget the value
Learning Objectives

• Identify recommended actionable steps to help your friend get started with agile (in any industry).

• Start your list of talking points explaining agile and it’s value, that don’t mention software or weird agile terms.

• Assess differences and similarities between agile in IT (software) and agile outside of IT.
Why I’m passionate about agile

Agile is about helping people embracing change and complexity, in a way that enhances peoples lives.
Why I’m passionate about agile

Agile is a set of ideas that together provide a way to create an environment where people are happier, more creative, more innovative, more productive.
So. . .

What will you tell her?
Your Friends Questions

A. What concepts only apply to software?

B. What are the top 2-3 most important agile concepts to you?

C. Why are these concepts important to you – why are you so passionate about these ones in particular? (what’s your story?)
Failure pattern:

Forgetting the Value of Agile

1. Inspect and adapt
2. Team empowerment
3. Collaboration
4. Working software
5. Customer involvement
6. Embrace complexity
7. Embrace change
8. Sustainable pace
9. Self-organizing
10. Incremental delivery
11. Trust
12. Simplicity
13. Deliver fast
14. Respect people
15. Eliminate waste
16. Optimize flow
17. Decide as late as possible
18. Limit WIP
19. Optimize the whole
20. Empiricism
21. Feedback
22. Respect
23. Focus
24. Transparency
25. Openness
26. Commitment
27. Courage
28. Mitigate risks
29. Antifragile relationships
30. Relentless learning
31. Minimum Viable Product
32. Core Concepts
Let’s Explore MORE Common Failure Patterns
Failure pattern: Jumping to solutions
Result: Solution Overload – No Connection to Real Issues
D. Ask: What challenges are you running into? (this is about them, not you!)

E. Ask: What are your goals?

F. Ask: What have you tried?

G. Looking back on questions A, B, and C, which concepts might be important to them?
Listen first for challenges – Pair with concepts
Failure Patterns:

- Only using software examples
- Fear of breaking the “agile rules”
Learn the rules so you know how to break them properly
Learning Objectives

• Identify recommended actionable steps to help your friend get started with agile (in any industry).

• Start your list of talking points explaining agile and it’s value, that don’t mention software or weird agile terms.

• Assess differences and similarities between agile in IT (software) and agile outside of IT.
Roles, Practices, Activities, Frameworks

1) Retrospective
2) Release planning
3) Sprint review
4) Daily scrum
5) Daily standup
6) Sprint planning
7) Value stream maps
8) Product backlog
9) Product roadmap
10) A3
11) Definition of done
12) Definition of ready
13) Framework
14) Scrum
15) Theory of constraints
16) BDD, TDD, and ATDD
17) Kanban
18) Five focusing steps
19) Extreme programming
20) Lean
21) Sprint
22) Epic
23) User story
24) Minimum marketable feature
25) Story points
26) Relative estimating
27) Pairing
28) Cross training
29) Unit test
30) Automated builds
31) Continuous integration
32) Specification by example
33) Small releases
34) ScrumMaster
35) Product owner
36) Onsite customer
37) Cross functional team
38)
What are your options?

H. What “agile rules” prevent you from using agile practices, roles, activities and frameworks outside of software?

I. Can you break the rule and maintain the intent? How?
Managers Needs

J. When her boss asks how you track progress in agile, what ideas would be useful to point out?

K. What agile concepts are valuable to your manager and their challenges?

L. Do you respond differently than you would if they were “in software”? 
How Can You Safely Break the Rules?
Learning Objectives

• Identify recommended actionable steps to help your friend get started with agile (in any industry).
• Start your list of talking points explaining agile and its value, that don’t mention software or weird agile terms.
• Assess differences and similarities between agile in IT (software) and agile outside of IT.
“Simple” Steps to Get Started

I. Break “agile rules” while honoring the essence of the rule*

* Use at any time!!

II. Define your Organizational Goal

III. Create a Product (or Project) Map

IV. Create a Ranked Backlog(s) – Product & Project

V. Visualize Your Work
“Simple” Steps to Get Started

VI. Use A3s - to target challenges - with countermeasures - to break the rules

VII. Identify facilitators

VIII. Start retrospectives
M. How are these steps different or the same as your agile with software approach?

N. What is the NEXT STEP you will take?
<table>
<thead>
<tr>
<th></th>
<th>1. Background: What are you talking about and why?</th>
<th></th>
<th>5. Recommended Experiments (countermeasures)?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>What are the facts and data?</td>
<td></td>
<td>What experiments do you recommend to reach the goals or target conditions? (2 or 3 options)</td>
</tr>
<tr>
<td></td>
<td>What visualizations do you have?</td>
<td></td>
<td>For each experiment:</td>
</tr>
<tr>
<td></td>
<td>Have you visited the gemba (the ‘real place’ that work is taking place)? Who are the people that know the most about the issue?</td>
<td></td>
<td>- What is your hypothesis?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Are you targeting the root cause?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- How will you evaluate the experiments hypothesis?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- What unintended consequences may occur?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Which experiment will you start with? (remember that gathering data is not an experiment.)</td>
</tr>
<tr>
<td></td>
<td>What is the problem?</td>
<td></td>
<td>What are the actions you will take to implement your experiment?</td>
</tr>
<tr>
<td></td>
<td>What is the current state?</td>
<td></td>
<td>How will measure the experiment?</td>
</tr>
<tr>
<td></td>
<td>What are the facts and data?</td>
<td></td>
<td>Who will be involved in the experiment?</td>
</tr>
<tr>
<td></td>
<td>What visualizations do you have?</td>
<td></td>
<td>Is everyone aware of their role?</td>
</tr>
<tr>
<td></td>
<td>Have you visited the gemba (the ‘real place’ that work is taking place)? Who are the people that know the most about the issue?</td>
<td></td>
<td>Who else needs to be aware of the experiment and how will they be and stay aware?</td>
</tr>
<tr>
<td></td>
<td>3. Goals or Target Conditions - What specific outcome is required and why?</td>
<td></td>
<td>7. Follow-up:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>How will you share what you learned?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>What changes are need to change the current standard to incorporate the experiments results?</td>
</tr>
</tbody>
</table>
|   |   |   | How will you know what the long term success and impact is?
4. Analyze the root cause – why does this problem or issue exist?

*Do you REALLY know what the problem is? Not what you assume, what you prove.*
- Can’t prove it, how can you learn?
- Do you need to gather data as part of this step to determine what to do next?
- How can you limit the time that you require to gather data to 1-2 weeks or less?

<table>
<thead>
<tr>
<th>4. Analyze the root cause – why does this problem or issue exist?</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Do you REALLY know what the problem is? Not what you assume, what you prove.</em></td>
</tr>
<tr>
<td>- Can’t prove it, how can you learn?</td>
</tr>
<tr>
<td>- Do you need to gather data as part of this step to determine what to do next?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. Follow-up:</th>
</tr>
</thead>
<tbody>
<tr>
<td>How will you share what you learned?</td>
</tr>
<tr>
<td>What changes are need to change the current standard to incorporate the experiments results?</td>
</tr>
<tr>
<td>How will you know what the long term success and impact is?</td>
</tr>
</tbody>
</table>
2. Current Conditions: What are you observing?

What is the problem?
What is the current state?
What are the facts and data?
What visualizations do you have?

Have you visited the gemba (the ‘real place’ that work is taking place)? Who are the people that know the most about the issue?

4. Analyze the root cause – why does this problem or issue exist?

Do you REALLY know what the problem is?
Not what you assume, what you prove.
- Can’t prove it, how can you learn?
- Do you need to gather data as part of this step to determine what to do next?

7. Follow-up:

How will you share what you learned?
What changes are needed to change the current standard to incorporate the experiments results?
How will you know what the long term success and impact is?
Recapping

✓ Know your passion for agile
✓ Avoid failure patterns
✓ Align core concepts with challenges
✓ Learn intent of rules to break rules
✓ Incrementally improve
Don’t let “Software” or “Agile Rules” stop you!
Thanks!

- jake@agileforall.com
- @jcalabrese
Possible Books

• *The Goal* (audio book is great)
• *Team of Teams* by Stanley McChrystal, Tantum Collins, David Silverman, Chris Fussell,
• *Turn the Ship Around* by David Marquet
• *Leading Lean Software Development* by Mary & Tom Poppendieck (software, but great ways to view the concepts)
• *Lean Enterprise* by Jez Humble, Joanne Molesky, Barry O’Reilly
• *Scrum* by Jeff and JJ Sutherland (audio book is good)