The 12 Step Program to Becoming Agile

Leigh Griffin

Engineering Manager, Red Hat
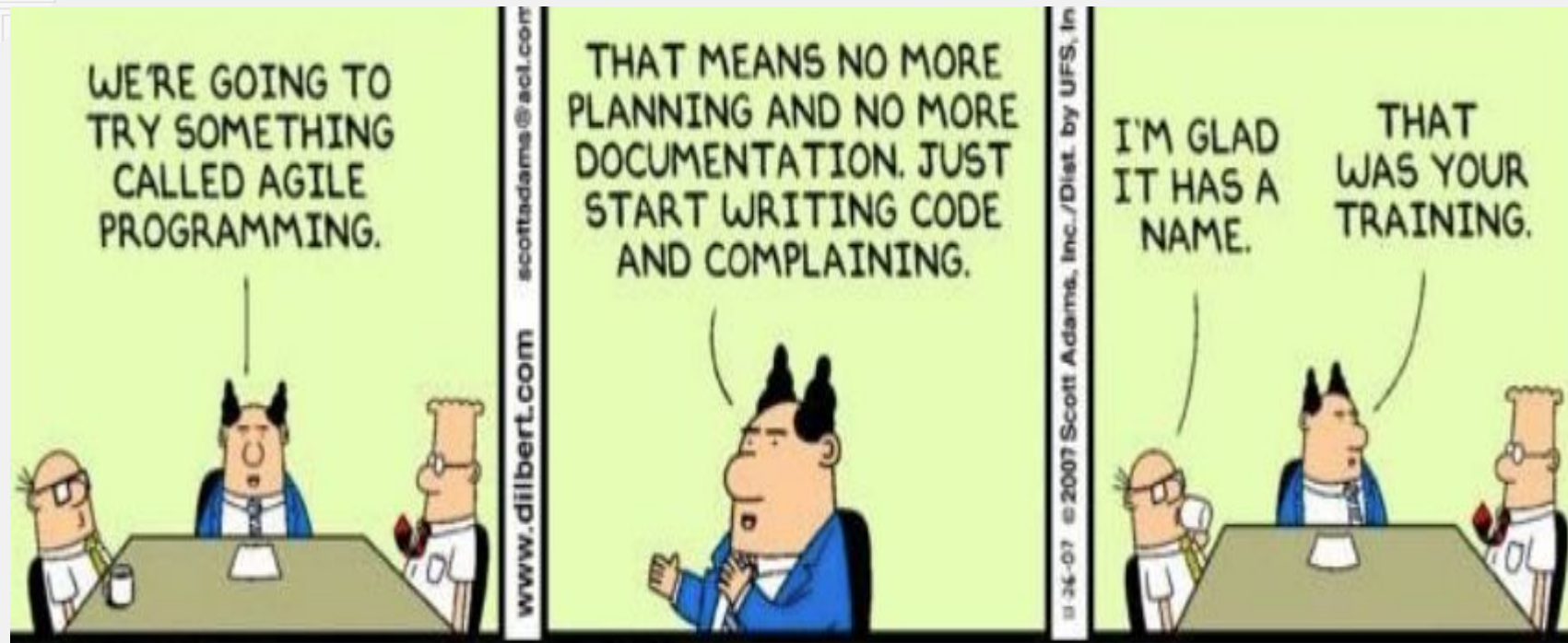
WATERFORD
TECH MEETUP

Agile Conversions: The Paths to Failure

Nobody is really trained

- There is a level of knowledge needed

- There is a level of coaching needed

- It's an experienced based learning Vs a 'read a book' learning

- Bottom line is it costs $$$

Agile Conversions: The Paths to Failure

Mandated to go Agile

- Somebody thought it would be a good idea

- Let's train 1 single person and trust them with the overall strategy

- Someone picked a flavour and sent someone on a 2 day training course

Congratulations you are now a Certified Agile Practitioner

Agile Conversions: The Paths to Failure

The ~~right~~ wrong person drives it

- Fully understands what Agile actually is (and is not)

- Passionate about it

- Strong communicator
  – strong listening skills

- Empathy for the team

- A defender of the team and the process

BE-AGILE-ZEBUB

ScrumMaster+PO+Manager

Undercover command & control
Lack of transparency
Over commitment
Long hours

redhat.

Agile Conversions: The Paths to Failure

Mandated to follow someone else's way of doing it

- Those guys over there are doing it right

- It works for them it must work for us…...

- Let's follow them

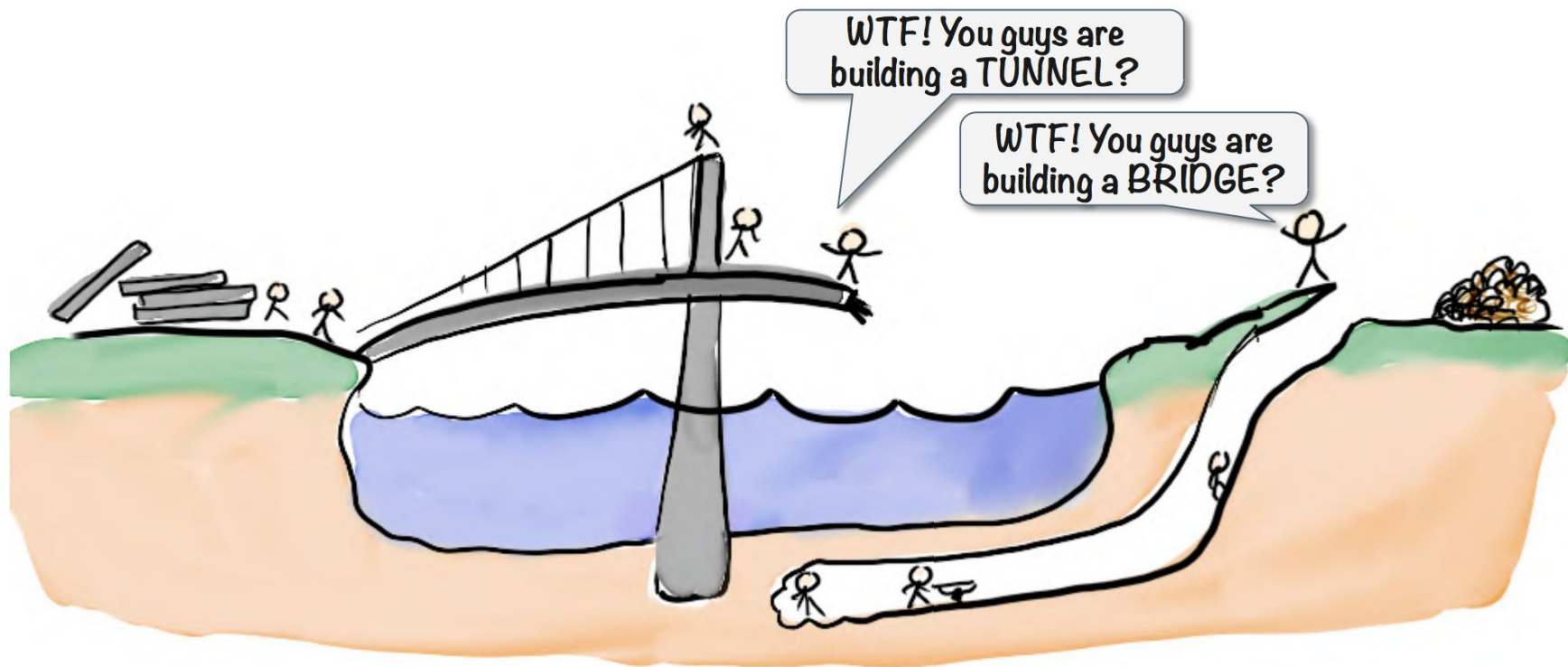redhat.

Their way works for them and only them

Do **not** do what works for another team. **Do what is right for your team.**

Agile Conversions: The Paths to Failure

Only one team moves to Agile

- Splendid isolation

- The rest of the org might be a Waterfall or another Agile Process

- Boundaries and interactions need to be well defined

Misaligned Expectations

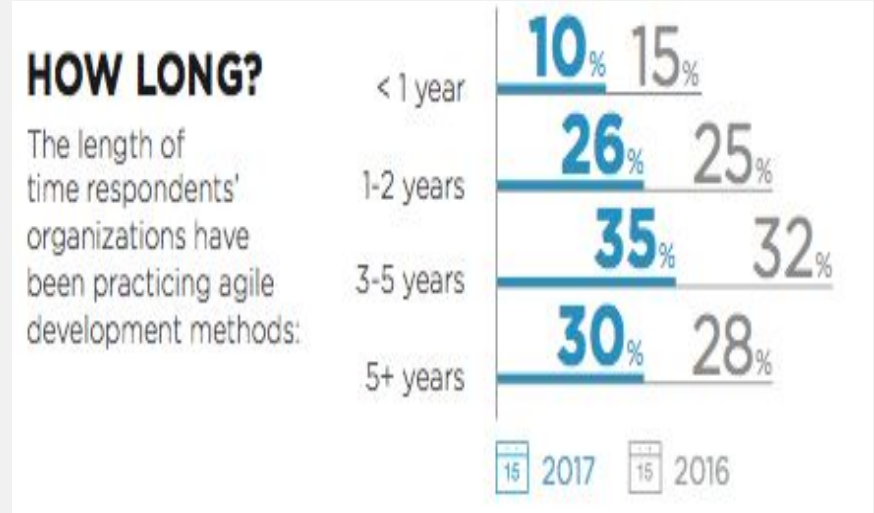# Agile Conversions: The Paths to Failure

Scaling and rolling out to all parts of the org on Day 0

- Begin at Scale, not with Scale in mind
  –Lets move the entire company as one

- Assumes everybody is mentally Agile, not just mechanically Agile
  –Mechanically == the ceremonies
  –Mentally == self organising

- Depends highly on a huge team of coaches and consultants available to guide it

Get the basics right, take your time, then Scale

HOW LONG?

The length of time respondents' organizations have been practicing agile development methods:

	< 1 year	1-2 years	3-5 years	5+ years
2017	10%	26%	35%	30%
2016	15%	25%	32%	28%

Version One State of Agile 2018 Report
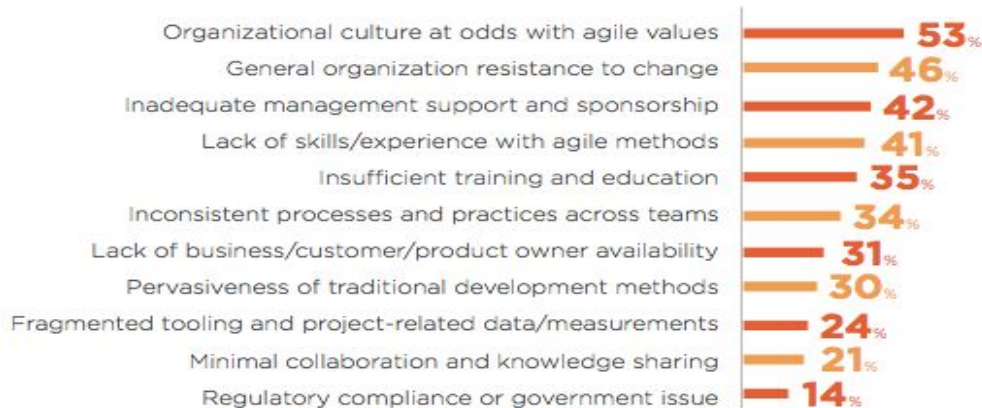
redhat.

Agile Conversions: The Paths to Failure

- No management buy in, the dev teams moved silently
- No time to allow processes bed in
- Excuses about the org structure, we are matrixed we can't do it
- Too narrow a view on what Agile is, they don't see the Umbrella

It's not just me saying this……..

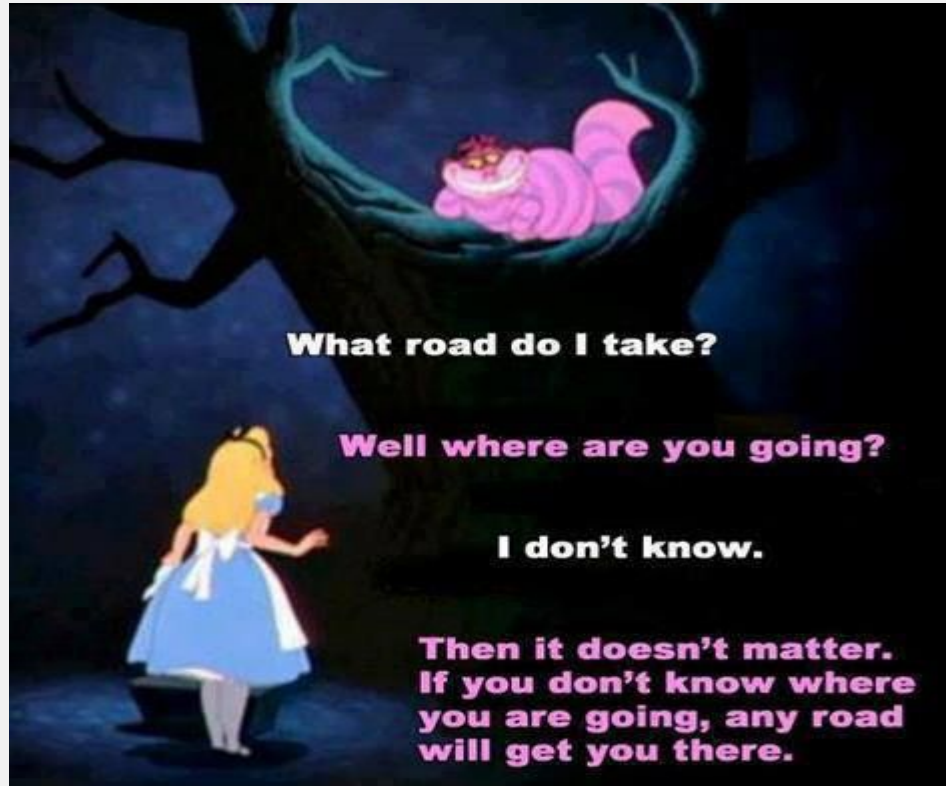## Challenges Experienced Adopting & Scaling Agile

From last year to this year we saw a decrease in respondents citing "organizational culture at odds with agile values" and "lack of business/customer/product owner availability" as challenges for adopting and scaling agile. Barriers that were cited more this year include "fragmented tooling", "inconsistent processes across teams" and "general resistance to change".

Challenge	%
Organizational culture at odds with agile values	53%
General organization resistance to change	46%
Inadequate management support and sponsorship	42%
Lack of skills/experience with agile methods	41%
Insufficient training and education	35%
Inconsistent processes and practices across teams	34%
Lack of business/customer/product owner availability	31%
Pervasiveness of traditional development methods	30%
Fragmented tooling and project-related data/measurements	24%
Minimal collaboration and knowledge sharing	21%
Regulatory compliance or government issue	14%

*Respondents were able to make multiple selections.

Version One
State of Agile
2018 Report

redhat.

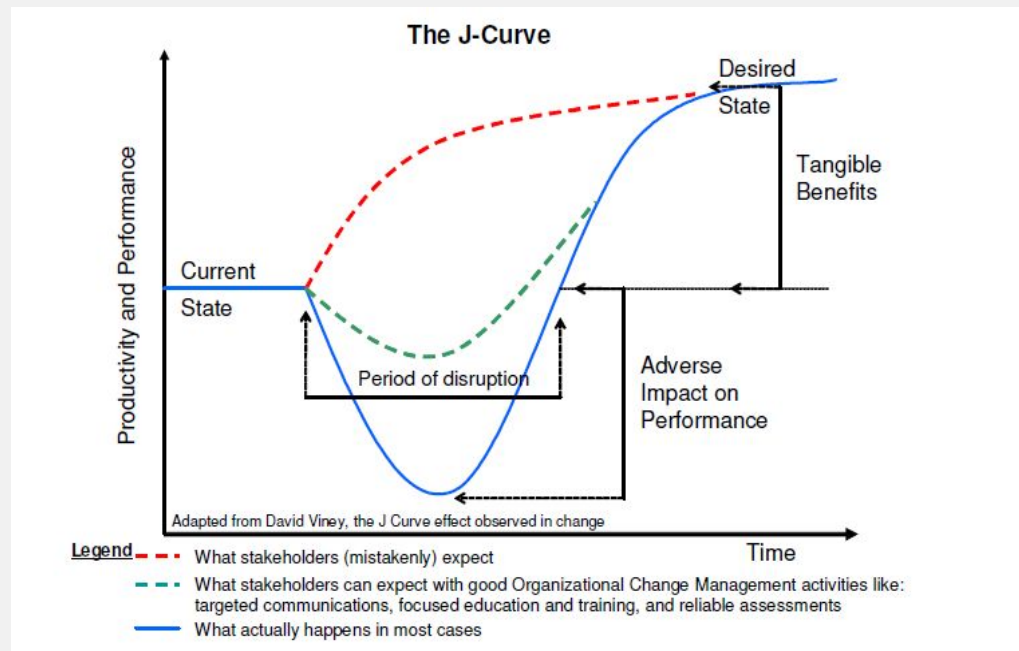The Destination is important: 12 Steps to get you there

Step 1: Why Agile?

- Admit you have a problem that you are trying to solve

- Share and discuss WHY Agile could solve this problem
  - Choose elements to carefully to fix the problems identified
  - Not because someone told you to
  - Not because you done a 2 day course
  - Not because those guys over there are doing it

- You don't need a full flavour (e.g. Scrum, Kanban, Lean) to answer the WHY
  - It might be part of the story
  - It might never need to happen

Step 2: Setting Expectations

- A proper Agile Conversion is going to take you 9-18 months

- There is a J-Curve Effect

- It needs Management Buy In
  - $$$$
  - Breathing Space
  - Protection

The J-Curve

Adapted from David Viney, the J Curve effect observed in change

Legend
- - - What stakeholders (mistakenly) expect
- - - What stakeholders can expect with good Organizational Change Management activities like: targeted communications, focused education and training, and reliable assessments
—— What actually happens in most cases

Step 3: Identify and Clarify Roles

- Identify and establish a distinction for every Agile Role e.g. if you go Scrum
    - ScrumMaster, dedicated, full time
    - Product Owner, dedicated, full time, politically neutral

- When people know what the role is they are willing to embrace it

- Best practices become easier to adopt and follow

- Accountability exists

- This can be flexible to your Orgs roles, modify the legacy roles over time

redhat.

Step 4: Product Owner Focus

- Focus on the ownership of your Product
  - Have a vision
  - Have an understanding of what is in Scope and out of Scope based on that

- Backlog Item Creation is true product ownership
  - Sets expectations
  - Shows value and ensures focus on value adds for your Product

For: (Our Target Customer)
Who: (Statement of need)
The: (Product/Product name) is a (Product/Product category)
That: (Product/Product key benefit, compelling reason to buy and/or use)
Unlike: (Primary competitive alternative)
Our Product: (Final statement of primary differentiation)

redhat.

Step 5: Proper Agile Meetings

- Correct Purpose of the Meeting

- Correct Attendees

- Agenda driven, no meetings for the sake of meetings

Step 6: Define Release Cycles

- You do NOT have to release EVERY time boxed Sprint
  - Your customer might NOT want it

- Define product release cycles that make sense to you
  - NOT what a methodology says

- The release cycle needs to be
  - Realistic
  - Tolerable
  - Sustainable
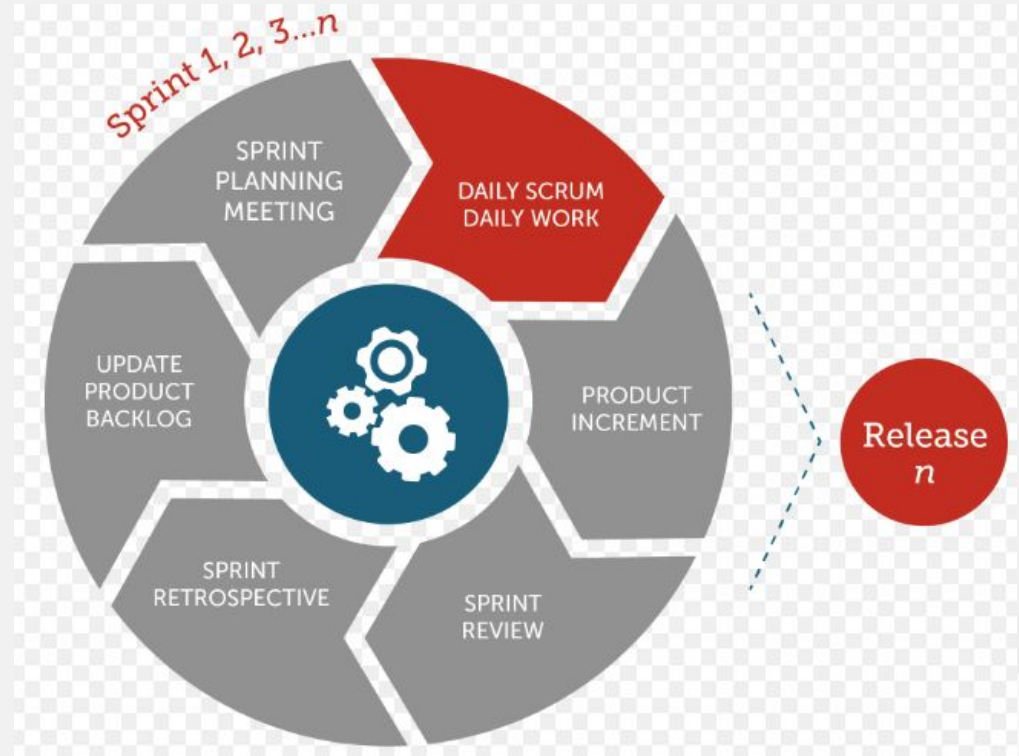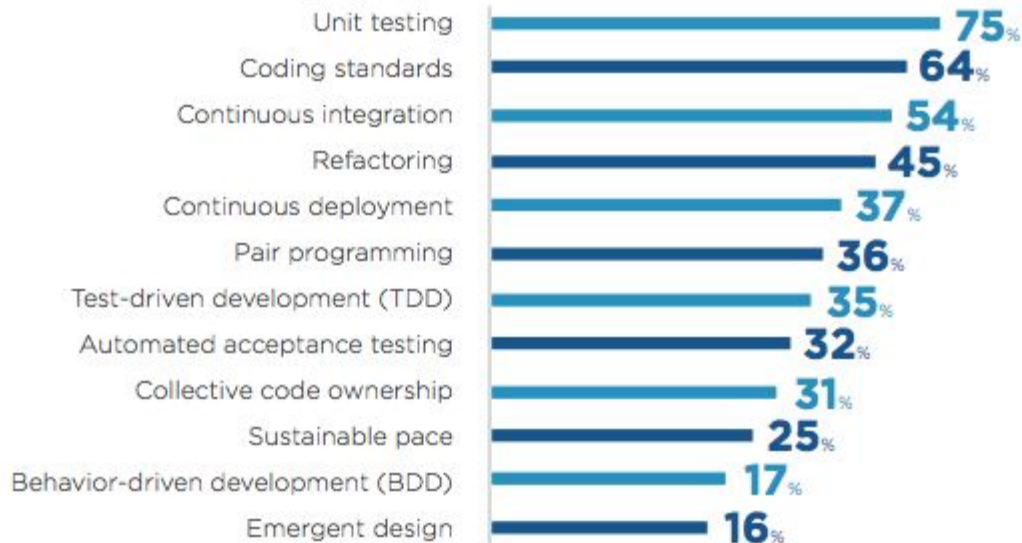
Step 7: Customer Satisfaction

- Drive your work with Acceptance Criteria

- Ensure you get feedback on your releases and on your way of working
    - Is the customer happy?
    - Who is our Customer? Someone paying? The Management? All?
    - Have you made them unhappy?
    - Did they notice a change?

- Frequent demos for the customer or their rep

- Upward and Outward Communication is key

Step 8: Better Programming Practices

- A Definition of Done that makes sense, at a minimum:

  - Unit Tests

  - Docs

- Code Reviews in place

- Technical Debt focus and paydown

- Coding Standards in place and enforced

- If you are not building software pay attention to how your product is built

Engineering Practices Employed

This year's survey demonstrated an increased use of coding standards (64% compared to 56% last year) and the use of continuous integration and refactoring were cited less as practices used.

Practice	Percentage
Unit testing	75%
Coding standards	64%
Continuous integration	54%
Refactoring	45%
Continuous deployment	37%
Pair programming	36%
Test-driven development (TDD)	35%
Automated acceptance testing	32%
Collective code ownership	31%
Sustainable pace	25%
Behavior-driven development (BDD)	17%
Emergent design	16%

*Respondents were able to make multiple selections.

redhat.

Step 9: Better Testing

- Working towards some form of TDD / BDD approach

- Adhering to Acceptance Criteria written ahead of the work

- Strong automated body of
  integration & end-to-end testing

- Moving towards potentially shippable

redhat.

Step 10: The Right Agile Tool

- Not viable to just work with post-its on a wall

- Understanding the progress at a team level

- Understanding the progress at an organisational level

- Transparency is key to buy-in

- Chance to gather stats and have a historic context

Step 11: Measure Success

- Empirical data is huge to make decisions happen
    - Positive reinforcement in the team
    - Stronger buy in further up the chain

- Identify SLAs for the team internally and externally
    - X% of work committed to completed (as high in the 90s as feasible)
    - X releases in a specific period

- Don't leave the team feel that one sprint can roll into the next Sprint

- Recognise positive accomplishments
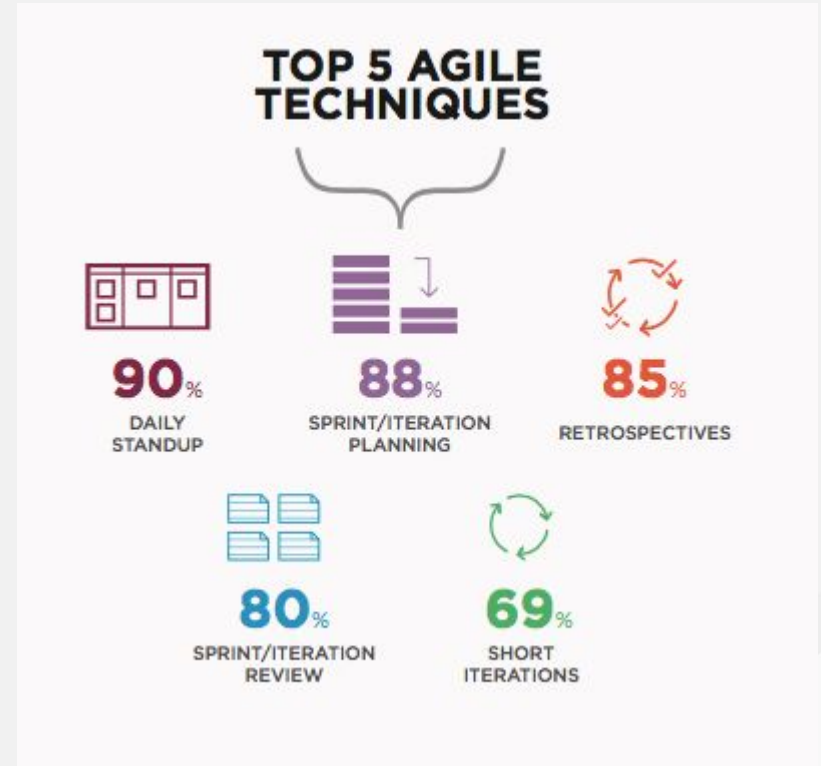   - Build high performing teams

Step 12: Inspect and Adapt

- Run Retrospectives frequently
  –On the team
  –On the process
  –On yourself

- Pick a max of 2 issues to work on and improve for the next iteration
  – Don't flip the table and move too many at once

- Look backwards to look forward

- Celebrate overcoming and dodging a failure
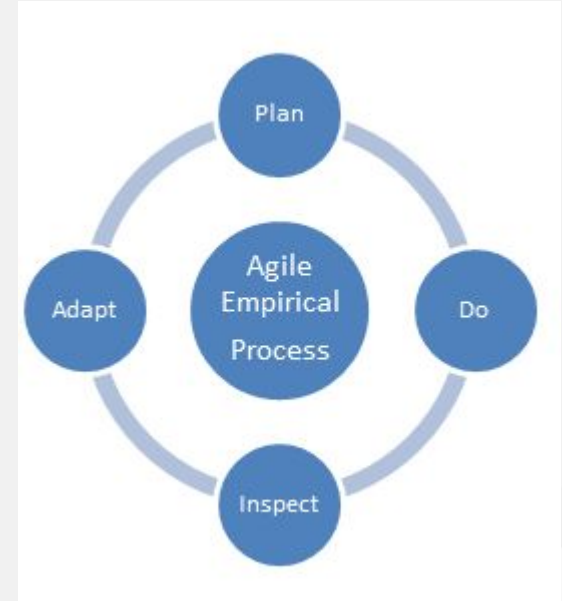
Closing Remarks

- Start small don't jump into a methodology

- Almost too easy to get this wrong

- Agile transformations never truly end

- Identify and invest in an Agile leader within your team

- Employ a Coach to come in
  - Ideally fully time if the skills aren't in house
  - Even part time or remote at Sprint boundaries
  - Huge value in someone looking at your outputs and guide you on the way

TOP 5 AGILE TECHNIQUES

90% DAILY STANDUP

88% SPRINT/ITERATION PLANNING

85% RETROSPECTIVES

80% SPRINT/ITERATION REVIEW

69% SHORT ITERATIONS

redhat.