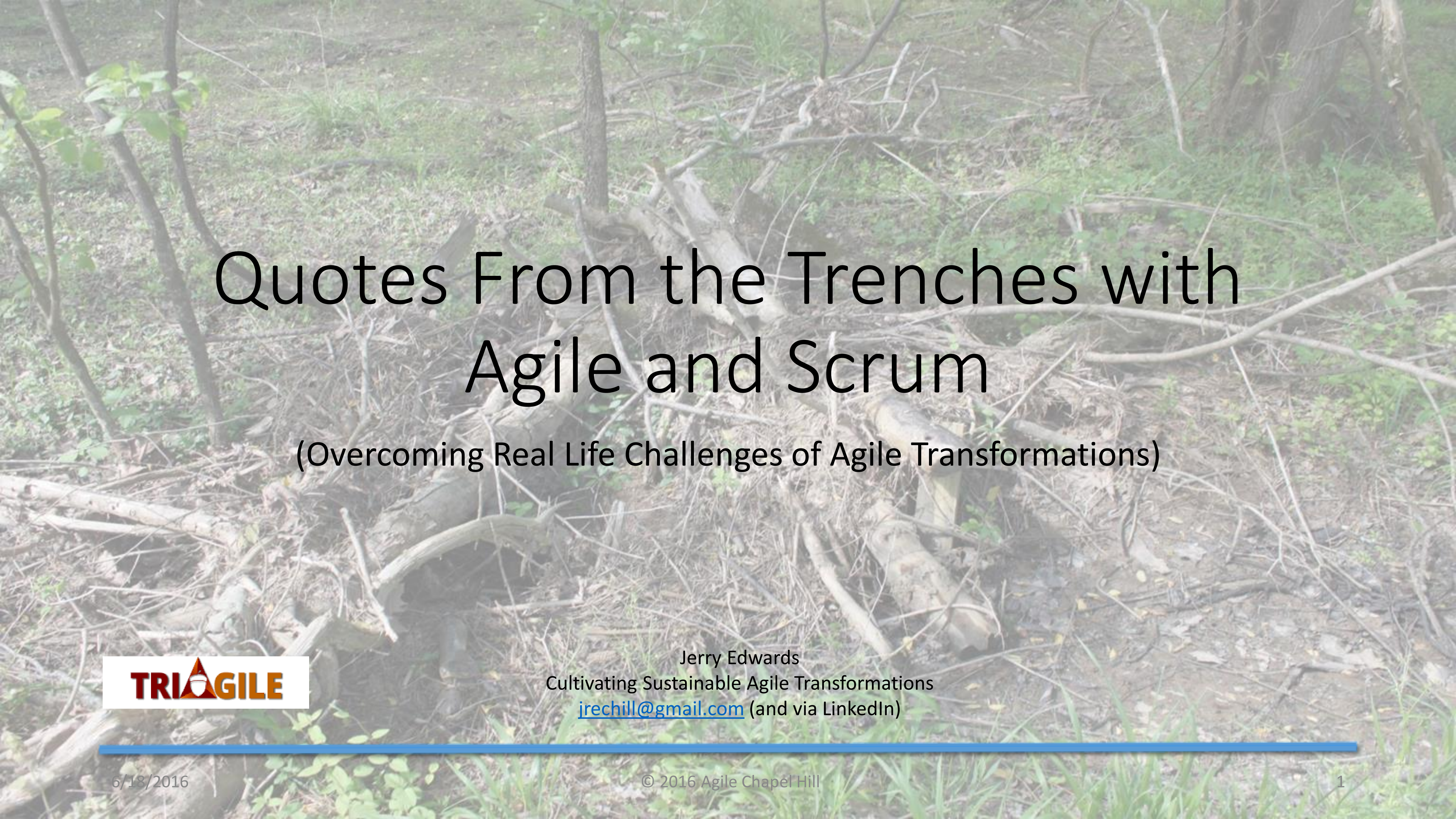


Quotes From the Trenches with Agile and Scrum

(Overcoming Real Life Challenges of Agile Transformations)

TRIAGILE

Jerry Edwards
Cultivating Sustainable Agile Transformations
jrechill@gmail.com (and via LinkedIn)

Some Past Trenches

- Software Trainer for NetApp
  - Development and delivery of Agile Training Curriculum (1000+ engineers, execs)
  - Coaching and advising new Agile teams in an Enterprise environment
- IT Strategist for Sony Ericsson
  - Standardization of SW processes and tools for geographic dispersed teams
  - Global coordination that included Agile transformations
- Manager for SW Configuration and Build Release Teams for Ericsson
  - Training and rollout of new build and automation tools
  - Implementation of lifecycle processes and ways of working globally with a special focus on improving SW quality
- Peace Corps Volunteer University Instructor (Liberia, West Africa)
  - Appreciation for training and working within different cultures
- Certifications
  - CSM, CPO, SAFe ™ Agilist

Many Agile Misconceptions

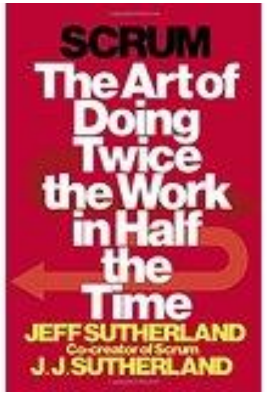
Background

“I hear Agile means no more documentation.”

Many Agile Promises

Background

“Scrum: Art of doing twice the work in half the time.”

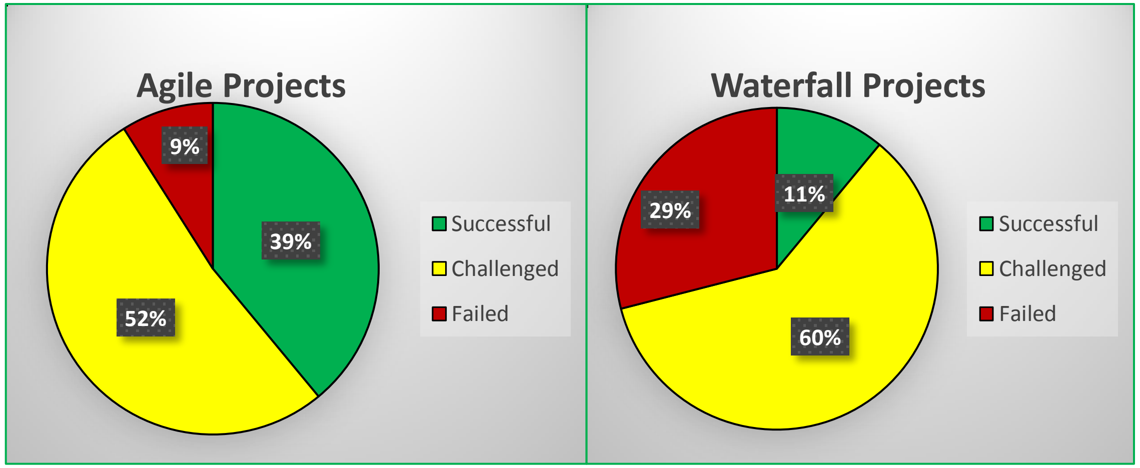


SCRUM
The Art of Doing Twice the Work in Half the Time
JEFF SUTHERLAND
Co-creator of Scrum
J.J. SUTHERLAND

Journey

Agile Improves Odds Of Success

Journey

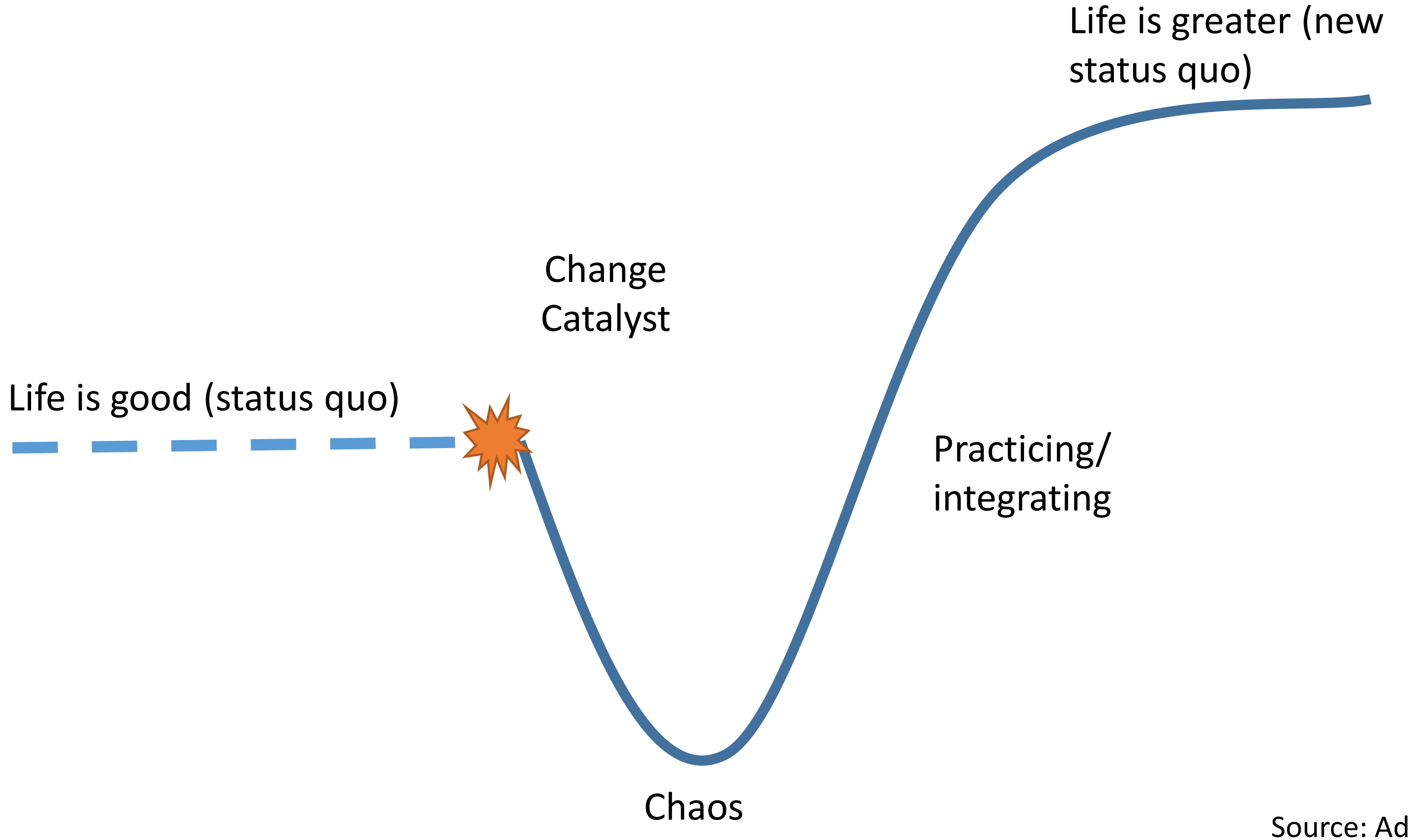


But does not guarantee it!

Source: Standish Group 2015 Chaos Report Q&A with Jennifer Lynch
http://www.infoq.com/articles/standishchaos2015

Change Requires “Effort”

Journey


Life is greater (new status quo)

Change Catalyst

Life is good (status quo)

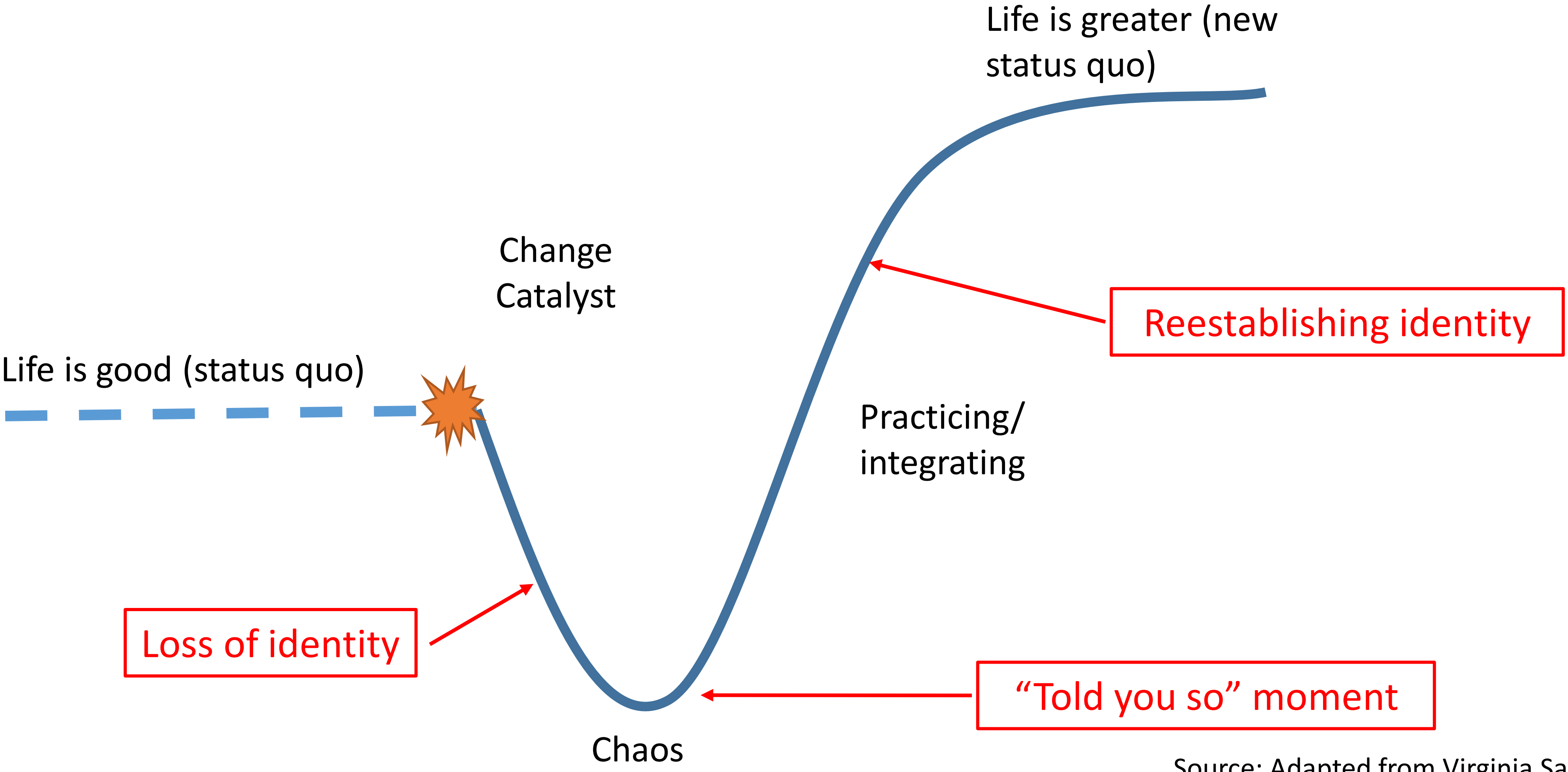
Practicing/ integrating

Chaos

Source: Adapted from Virginia Satir Model

Change Requires "New Identities"

Journey



Source: Adapted from Virginia Satir Model

According to Scrum Guide

Journey

“Lightweight, simple to understand but difficult to master.”

“Mastery is a journey (not just a sequence of sprints strung together)”

View Of Scrum Flow

Journey

Value & Goals

Requirements
(User Stories)

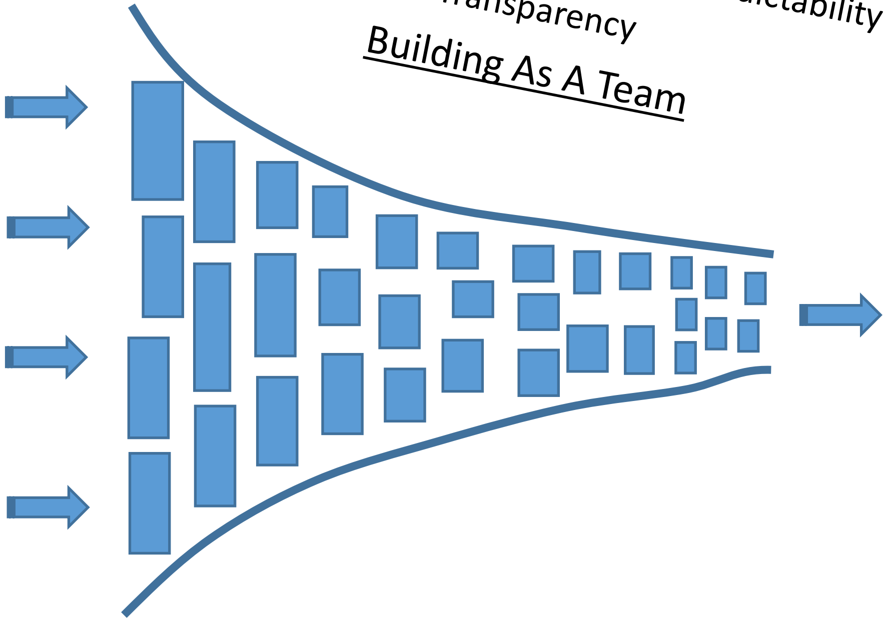
- Smaller
- Priority (value)
- Acceptance criteria
- Size
- “Yes” or “No”

Getting to Ready

Framework & Tools

- Meet team commitments
- Inspect and adapt
- Interruptions & predictability
- Transparency

Building As A Team



Product Backlog

Sprint Backlog

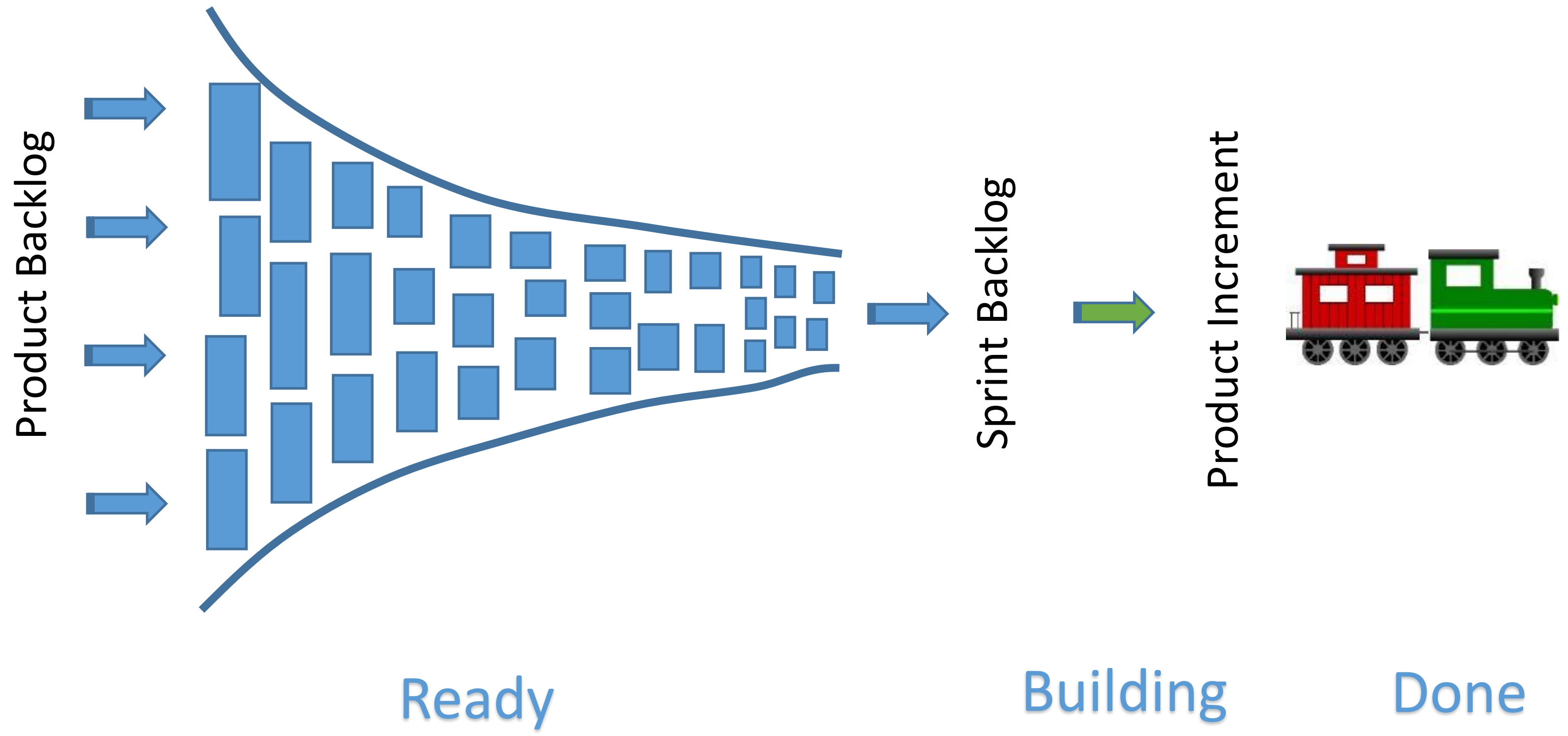
Product Increment

- Working software
- WIP Flow/Throughput
- Definition of Done
- Shippable quality
- Feedback

Achieving Done

Let’s Focus on Three Phases

Journey



Product Backlog

Sprint Backlog

Product Increment

Ready

Building

Done

Room Survey

Journey

- Which phase is your greatest challenge hindering your Agile Scrum success?
  - Ready
  - Building
  - Done

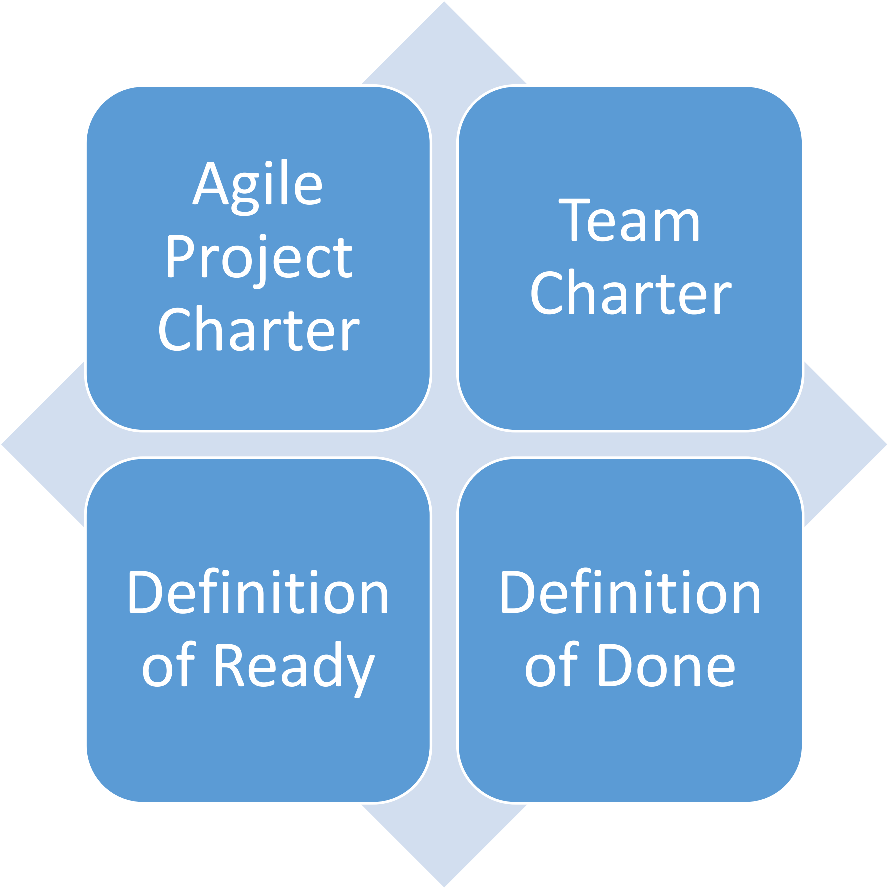
Planning Your Journey

- What problems are you trying to solve?
- What does success look like?
- Are you committed to making changes?

Charters & Guidelines

(Some First Steps To Success)

Journey



Agile Project Charter

Team Charter

Definition of Ready

Definition of Done

Ready

Oops!

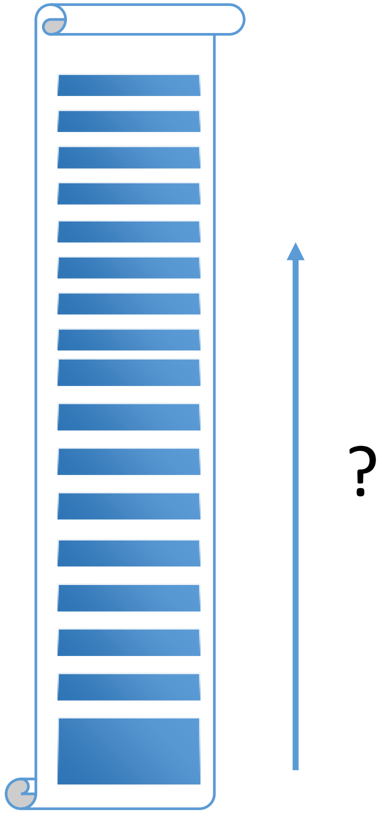
Ready

"Our Agile teams were more productive and efficient, we built the wrong product in half the time."

Product Backlog

Ready

- Critical to create the “right” product backlog
  - Consider techniques such as story maps
- Generally written as user stories
  - INVEST
  - 3 C’s (card, conversation, confirmation)
  - Sliced into testable deliverables
- Prioritized & Refined
  - Small to fit within sprint and even smaller yet
  - Acceptance Criteria
  - Sized
  - Meets Definition of Ready (DOR)



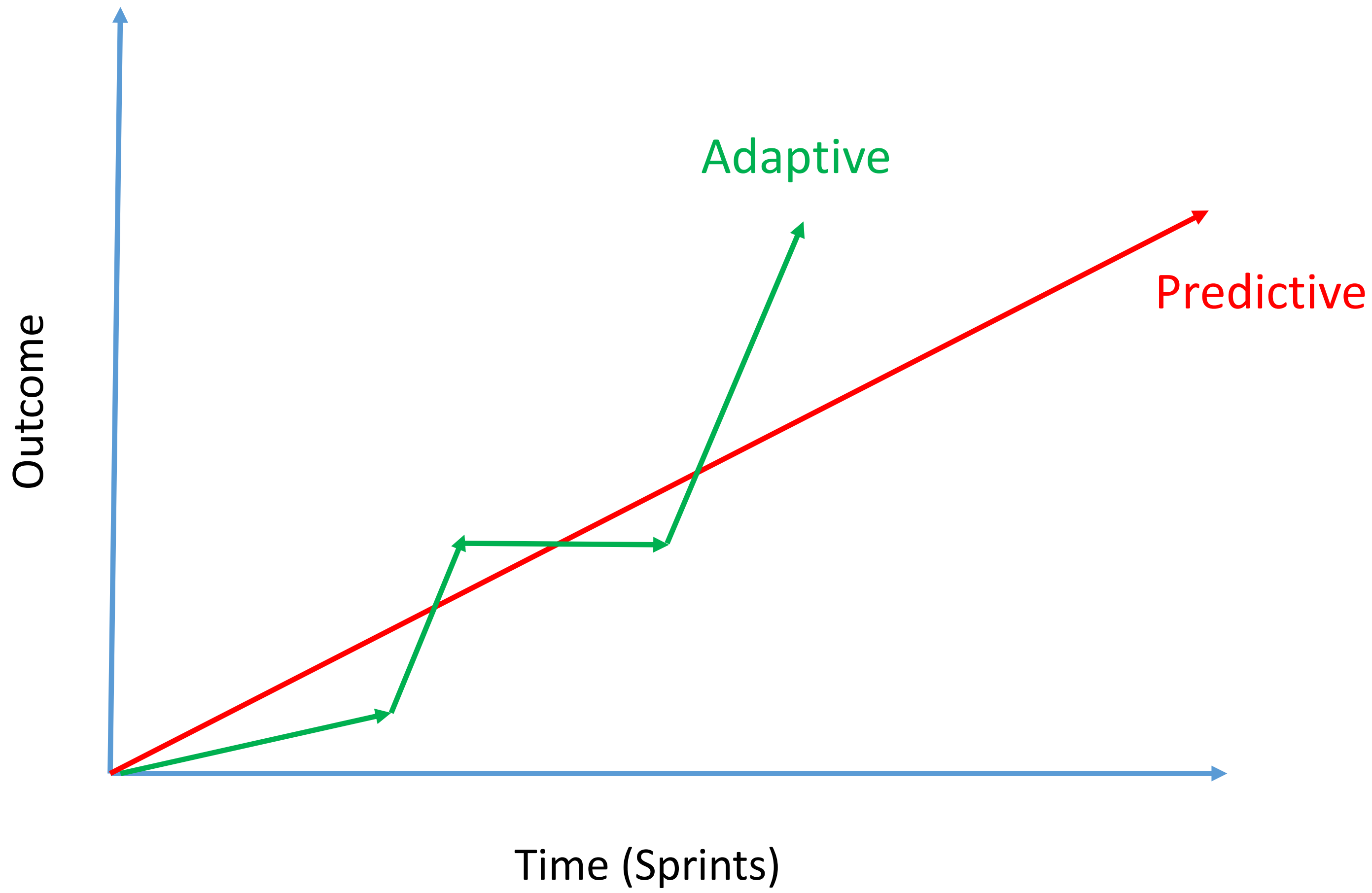
Room Survey

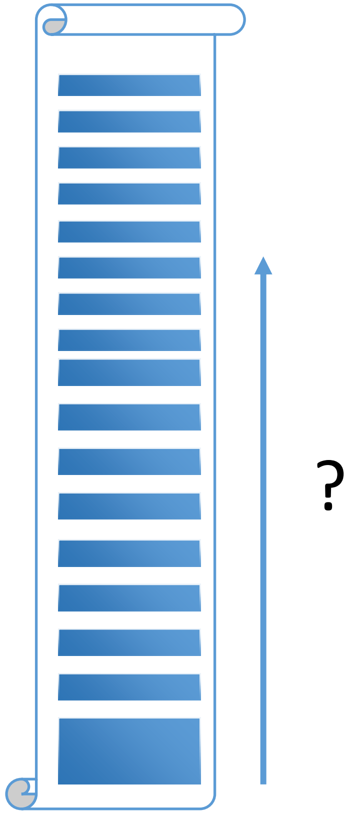
Ready

- How many of you have a formalized Definition of Ready (DOR)?

Predictive Vs Adaptive

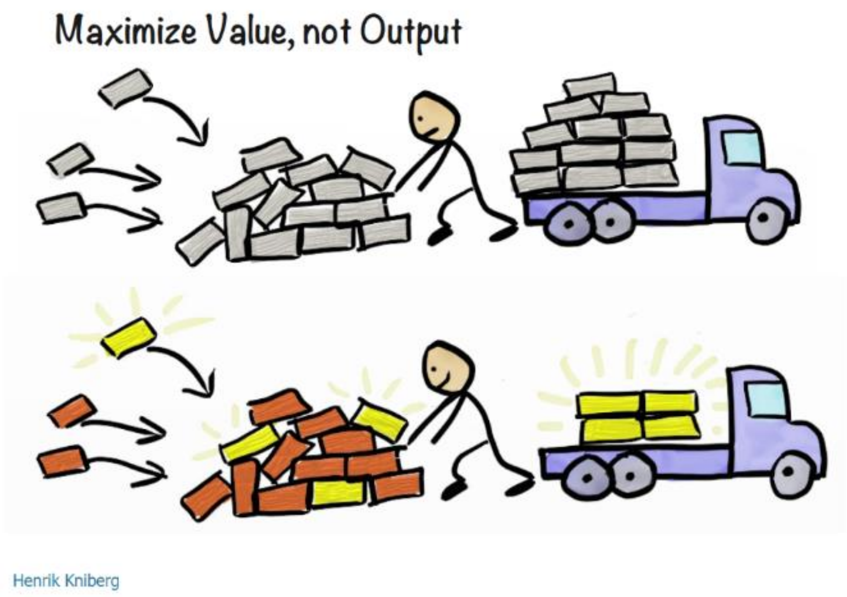
Ready



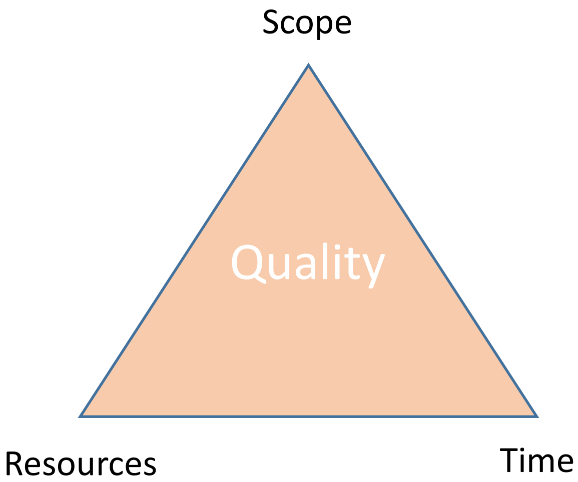


Value Driven

Ready



Source: Henrik Kniberg, "Stop Starting, Start Finishing", Crisp's Blog, 14 March 2013



Value driven requires a good understanding of minimal viable product (MVP) or minimal business product (MBP)

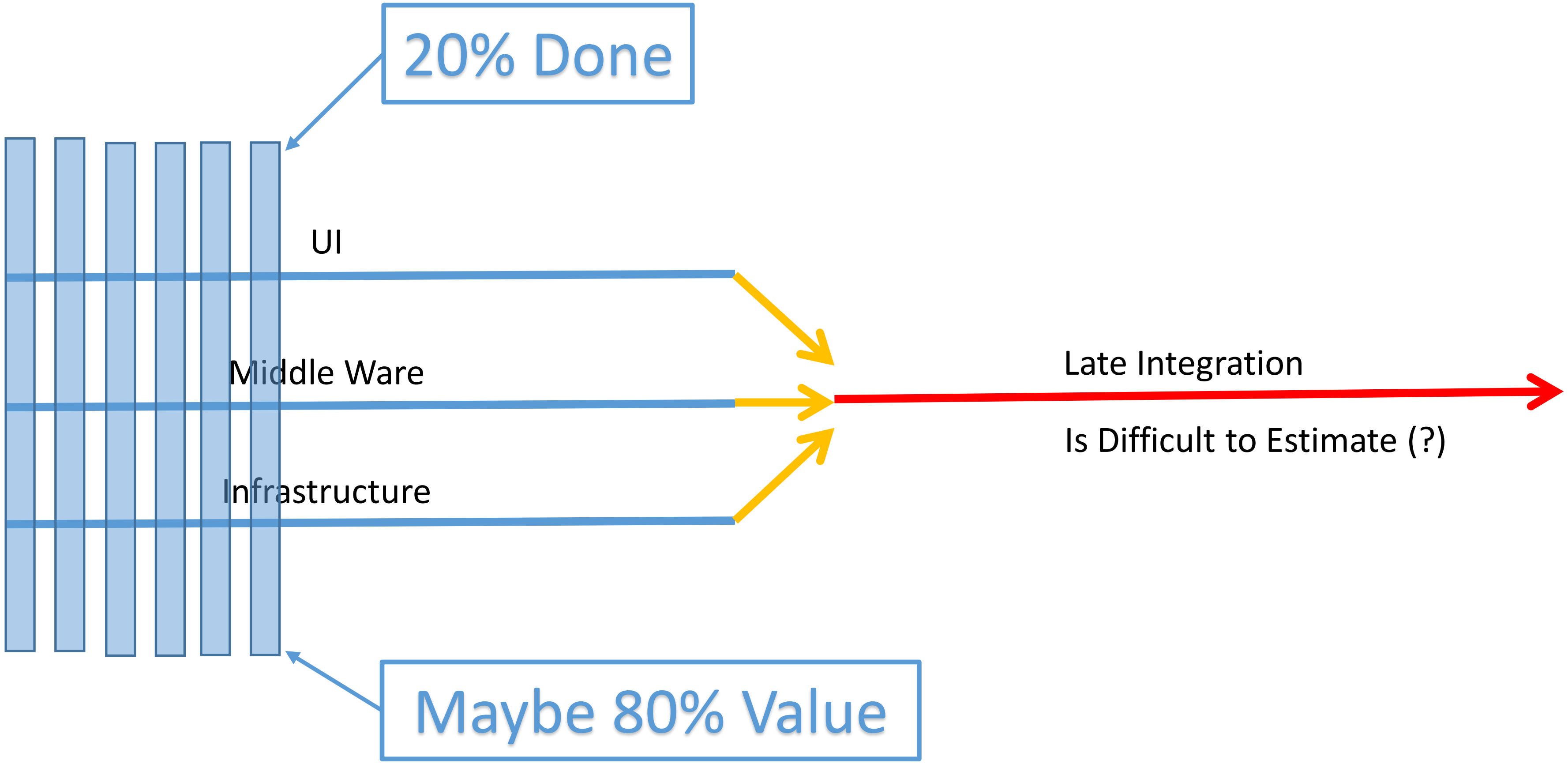
Early Problem And Risk Exposure

Ready

"I know things in a project are going to change, but in my Agile projects, I know this information much sooner which allows for good decision making."

Slicing User Stories Across SW Layers

Ready



Building

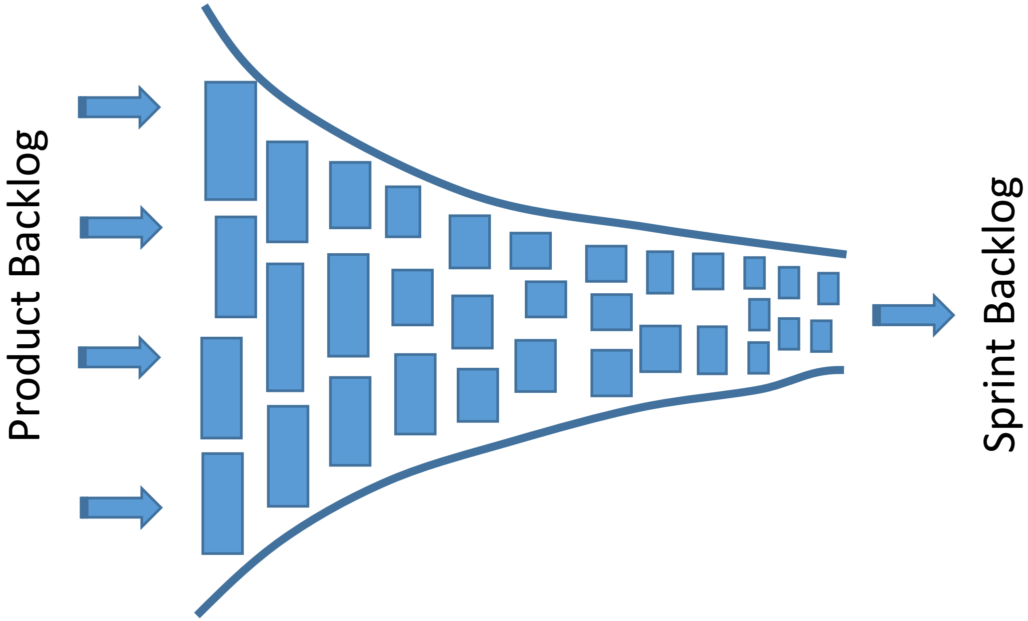
Limiting WIP

Building

> *“Do a few things insanely great.”*

Value Completed Work

Building



Room Survey

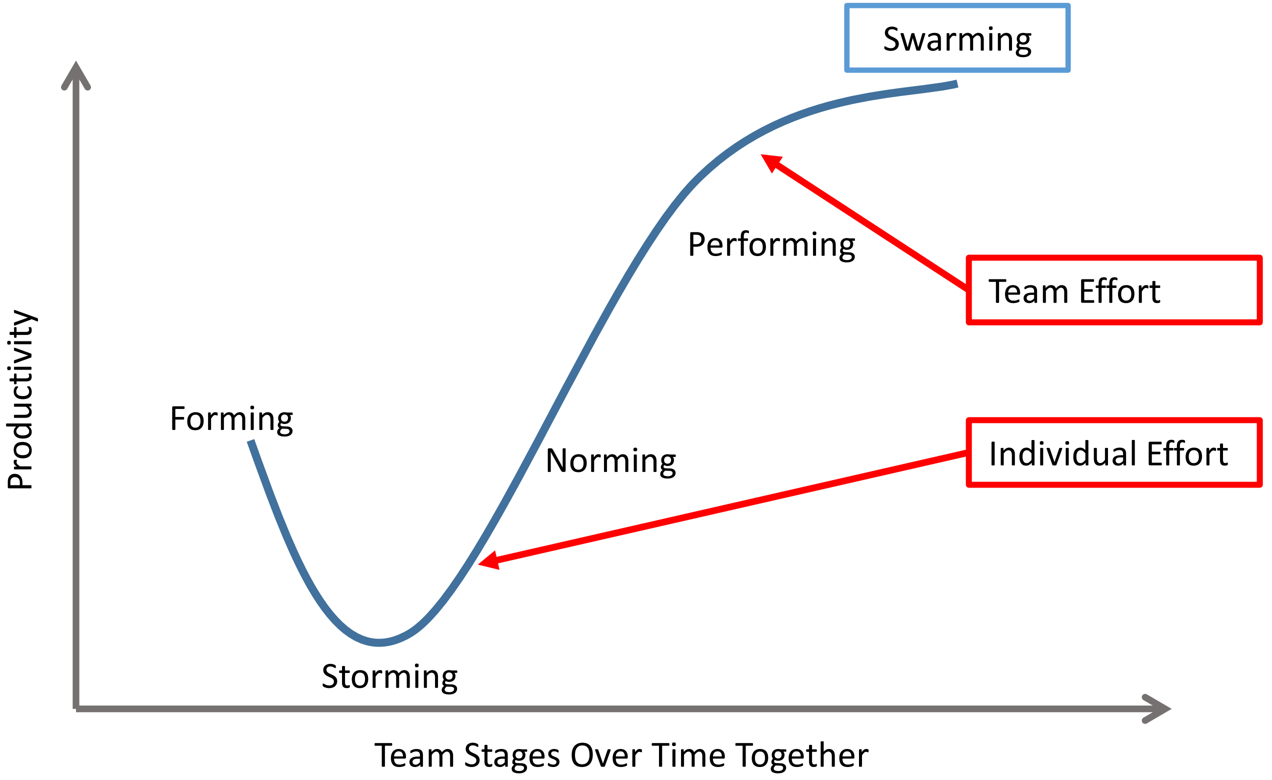
Building

- On average, what percentage of sprint backlog user story points are accepted each sprint?
    - 0 to 50%
    - 50 to 75%
    - Greater than 75%

Productive Teams Rise To Swarming

Building

Tuckman Model



Transparency

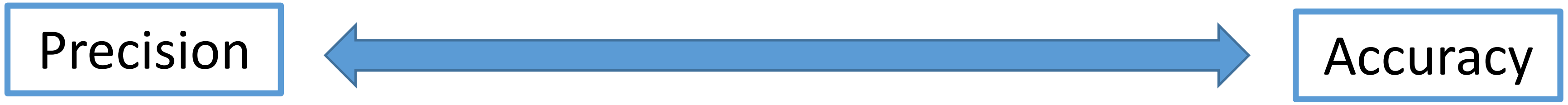
Building

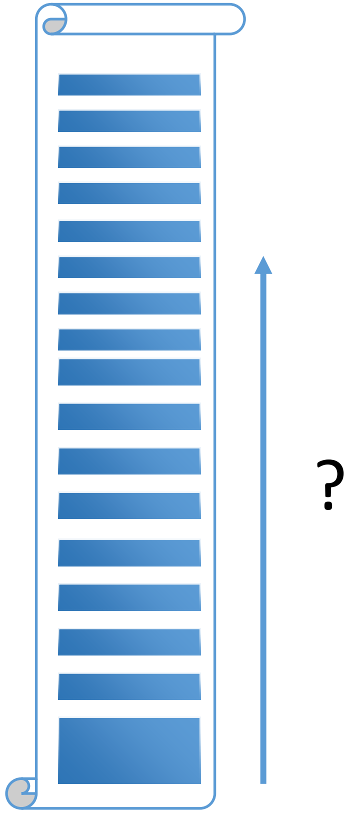
> *“Know where you are every day with Scrum or think you know where you are on your well-formed plan and discover that you are very wrong, very much later.”*

Source: *Ken Schwaber, A Playbook for Adopting the Scrum Method of Achieving Software Agility*

Relative Sizing: Accuracy Over Precision

Building



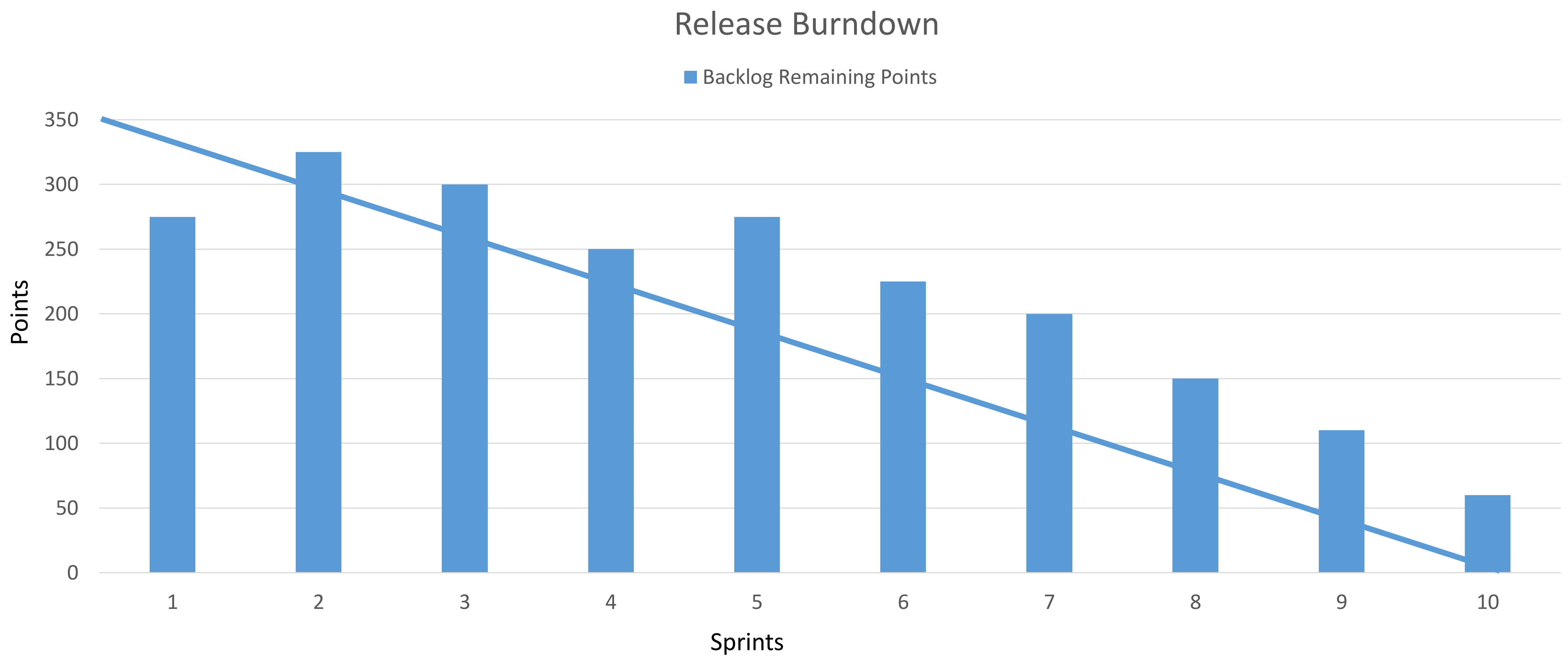


Keys to measuring realistic progress are:

1. Vertically sliced small user stories
2. Relative estimates which provide a quicker and “good” average accuracy
3. Maintaining tested working software (shippable quality)
4. Consistent teams over time
5. Using empirical data from time boxing (velocity)

Release Transparency - Burndown

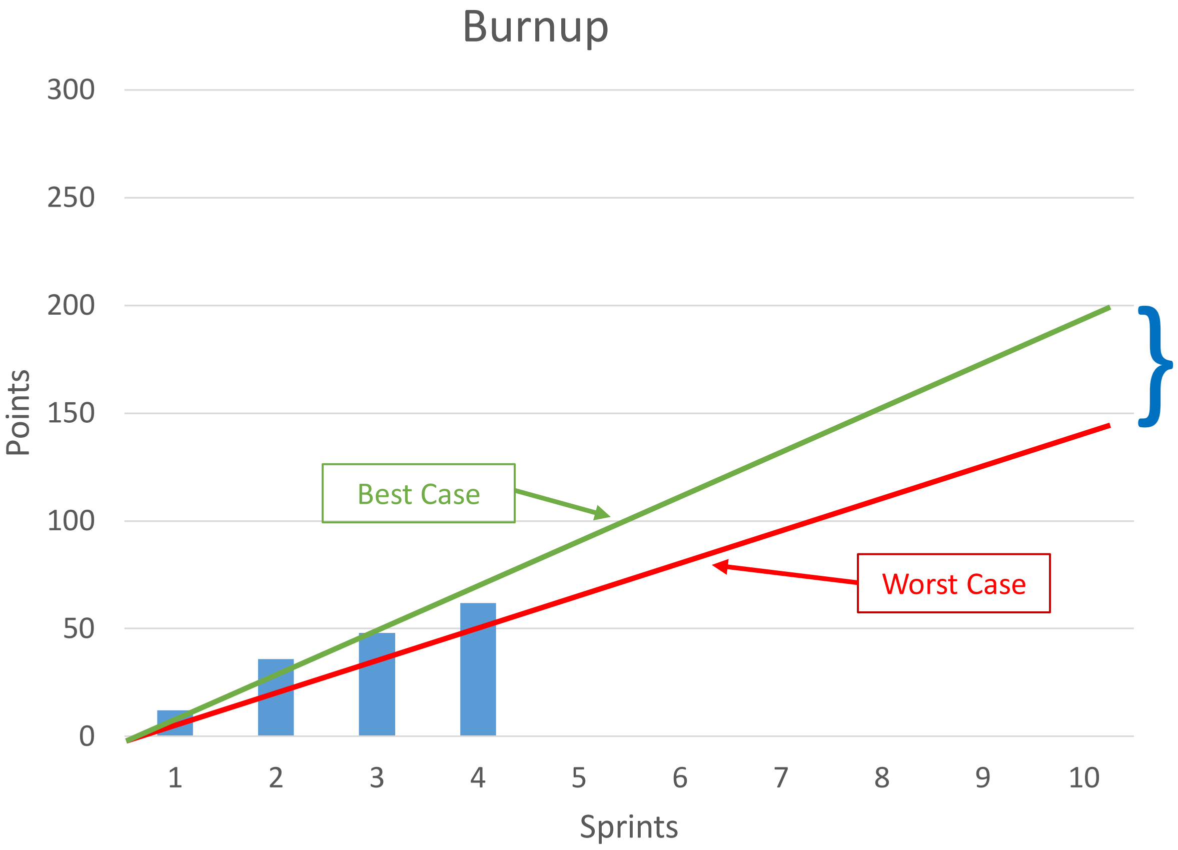
Building



Release Transparency - Burnup

Building

- Velocity = 15-20 points per sprint
- Forecasting with best and worst case scenarios



Ignoring Transparency

Building

“Management wanted us to release in 5 sprints, but our velocity charts showed we needed at least 10 sprints.”

Done

Move From Throwing Over Wall …

Done

“Developers did not give completed user stories to testers until the last day so there was no time to test before the sprint was over.”

To Team Responsibility

Done

- Team is responsible for quality
- Testers are part of the team
- Everyone tests
- Testing is not a phase

Quality is not equal to test. Quality is achieved by putting development and testing into a blender and mixing them until one is indistinguishable from the other.

Source: How Google Tests

Tester in Agile is "a person whose primary skill is testing", rather than "a person whose role is to do only testing"

Source: Henrik Kniberg, Scrum and XP from the Trenches, www.crisp.se

"There should be as much test activity on the first day of a sprint as on the last day. "

Source: Michael Cohn, Succeeding with Agile: Software Development Using Scrum

Move From Identifying A Defect …

Done

"My testing performance review is based upon how many defects I find and report."

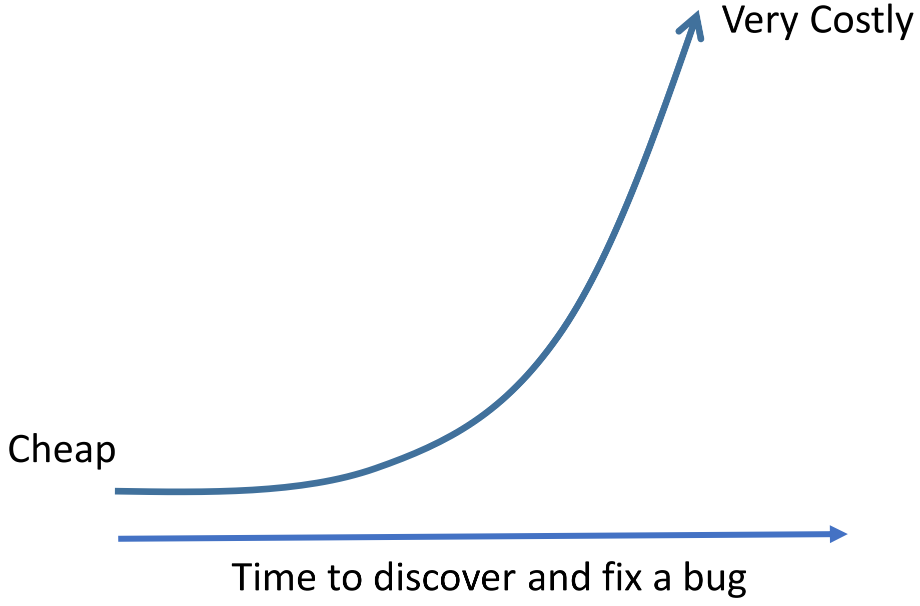
To Preventing a Defect

Done

- Performance based upon quality delivered not defects found
- Fix it!
    - Like finding a hole in a boat
- Cleaning up your "poop" before it smells

Awareness Of Technical Debt Cost

Done



Eliminating Technical Debt

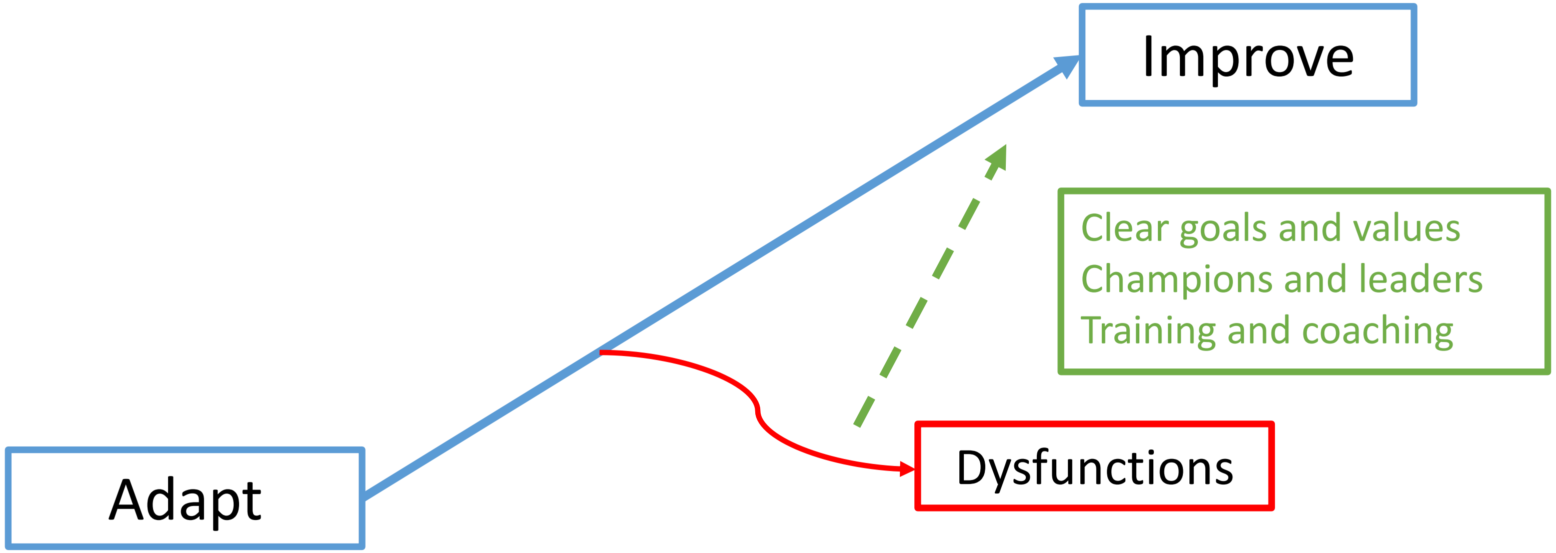
Done

- Avoid accepting user stories with “defects”
  - Cheaper to find and fix bugs early
  - Addressing technical debt is rarely accounted for in backlog which annihilates predictions and release schedules
  - Especially if defect prevents the release
- Some Testing aids to find issues early
  - Formalize testing in Def of Done
  - TDD, ATDD, and BTDD
  - Automation
  - Continuous integration

Cultivating

Adapt To Improve

Cultivating



Clear goals and values
Champions and leaders
Training and coaching

Dysfunctional Example

"Our daily Scrums lasted 2 hours each day, so team members have started skipping."

Finally, Leaving You With …

- Slicing
- Swarming
- Sustaining

Ready

Building

Done

Thank You

Jerry Edwards
Cultivating Sustainable Agile Transformations
jrechill@gmail.com (and via LinkedIn)

TRIAGILE

Supporting Slides