Project
Management
Institute.
Dallas

Agile and Scrum Fundamentals

May 16, 2020

William "Red" Davidson

PMP PMI-ACP CSM
CSPO ICP-ACC SAFe-SA CDAP

reddavidson@yahoo.com
www.linkedin.com/in/reddavidson

Agile and Scrum Fundamentals

Agile

- What is Agile, what it is not
- Agile Values and Principles
- Frameworks

Scrum

- Scrum Roles
- Scrum Artifacts and Events
- Scrum Calendar

Some Other Day:

- Story Writing
- Estimation
- Release Planning
- Software Engineering Practices
- Soft Skills
- Leading an Agile Organization
- …

AGILE AND THE MANIFESTO

Origins of Agile Techniques

In the 1990s, new development methodologies were developed as an “alternative to documentation driven, heavyweight software development processes”.

- Adaptive Software Development
- Dynamic Systems Development Method - 1994
- Scrum - 1995
- eXtreme Programming (XP) - 1996
- Crystal Methods - 1996
- Feature Driven Development - 1997

Requirements

Design

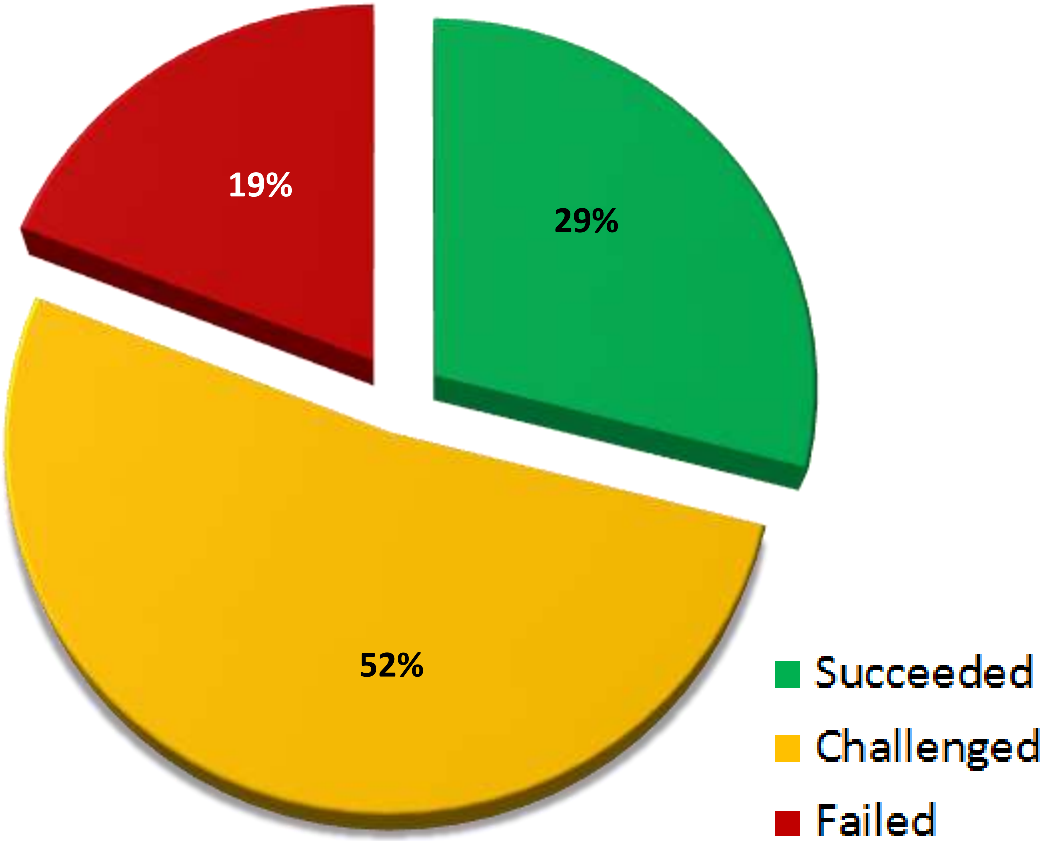
Implementation

Verification

Maintenance

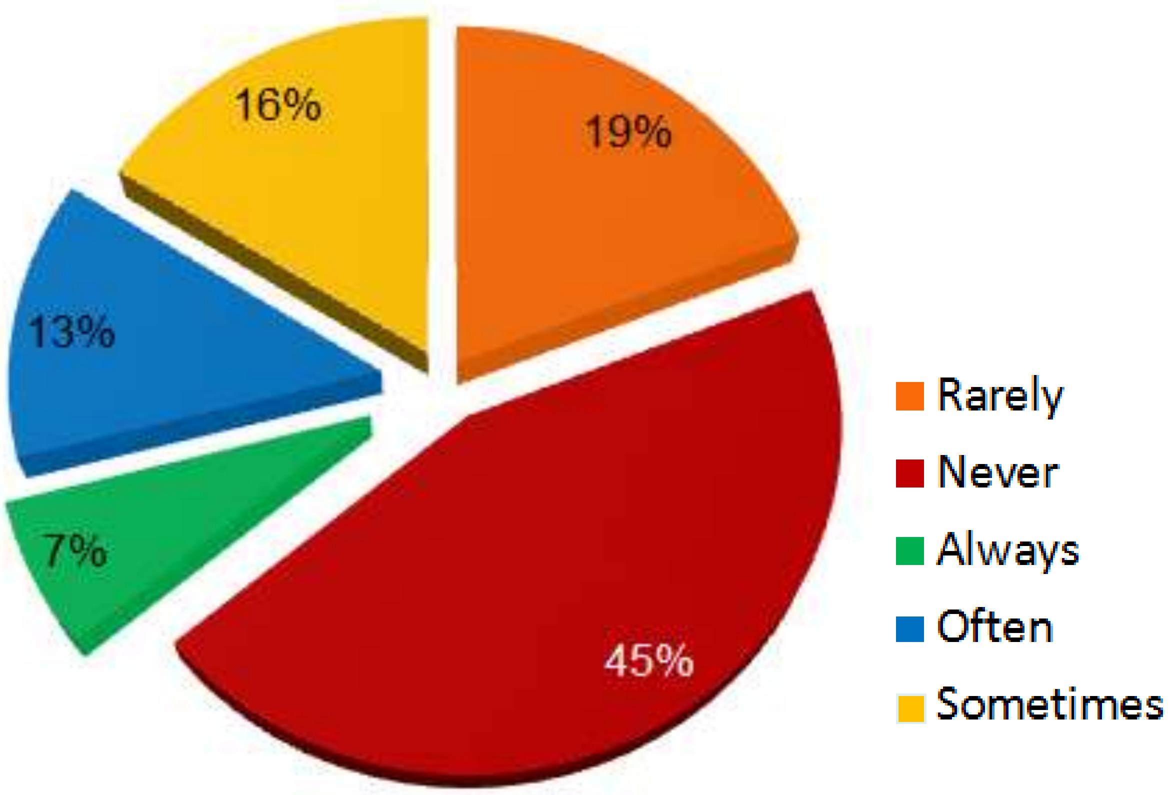
THE STANDISH GROUP

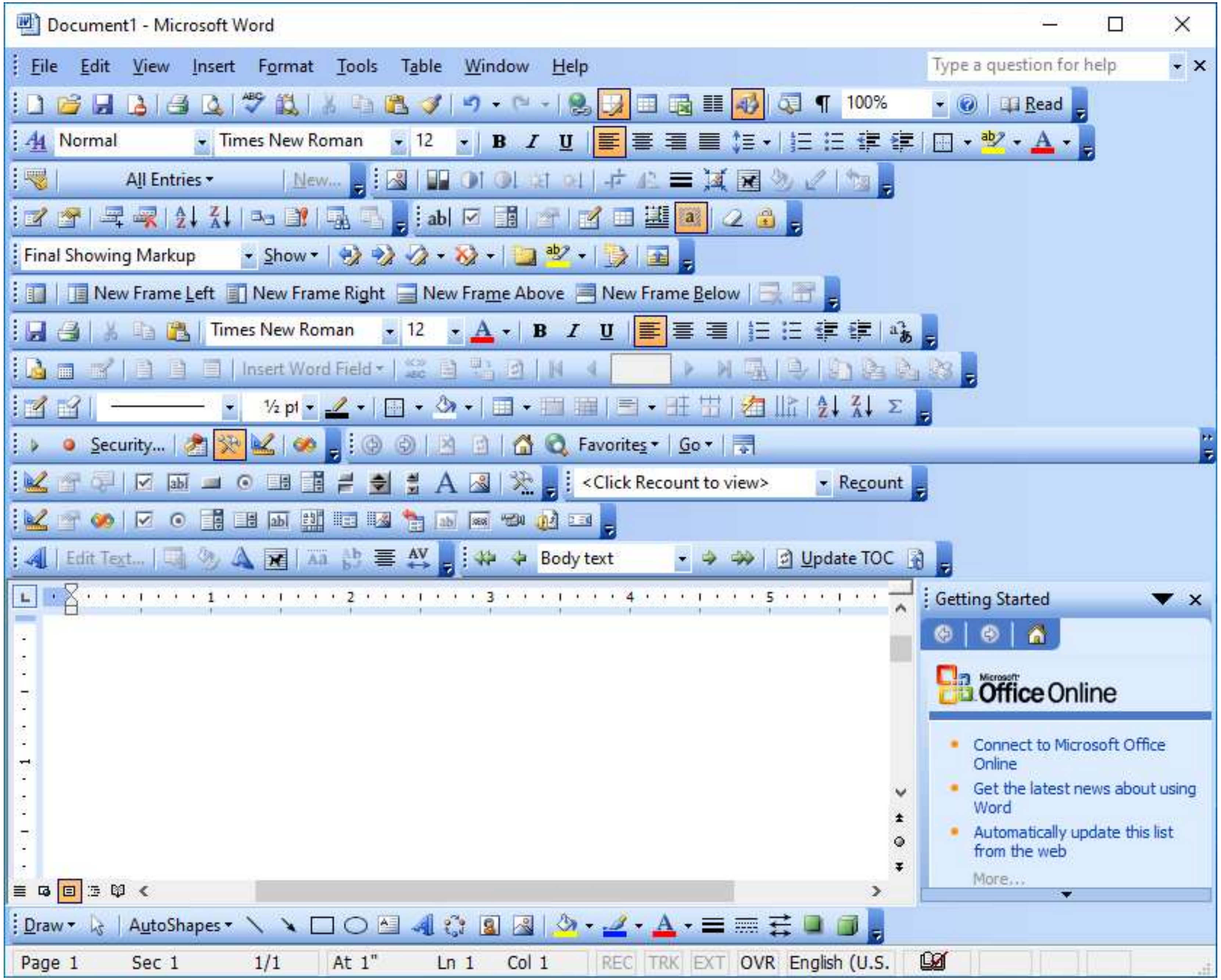
The Standish Group's “CHAOS Report” gathered data from projects done across a variety of industries and presented them in buckets of “**successful**, **challenged**, or **failed**" projects.



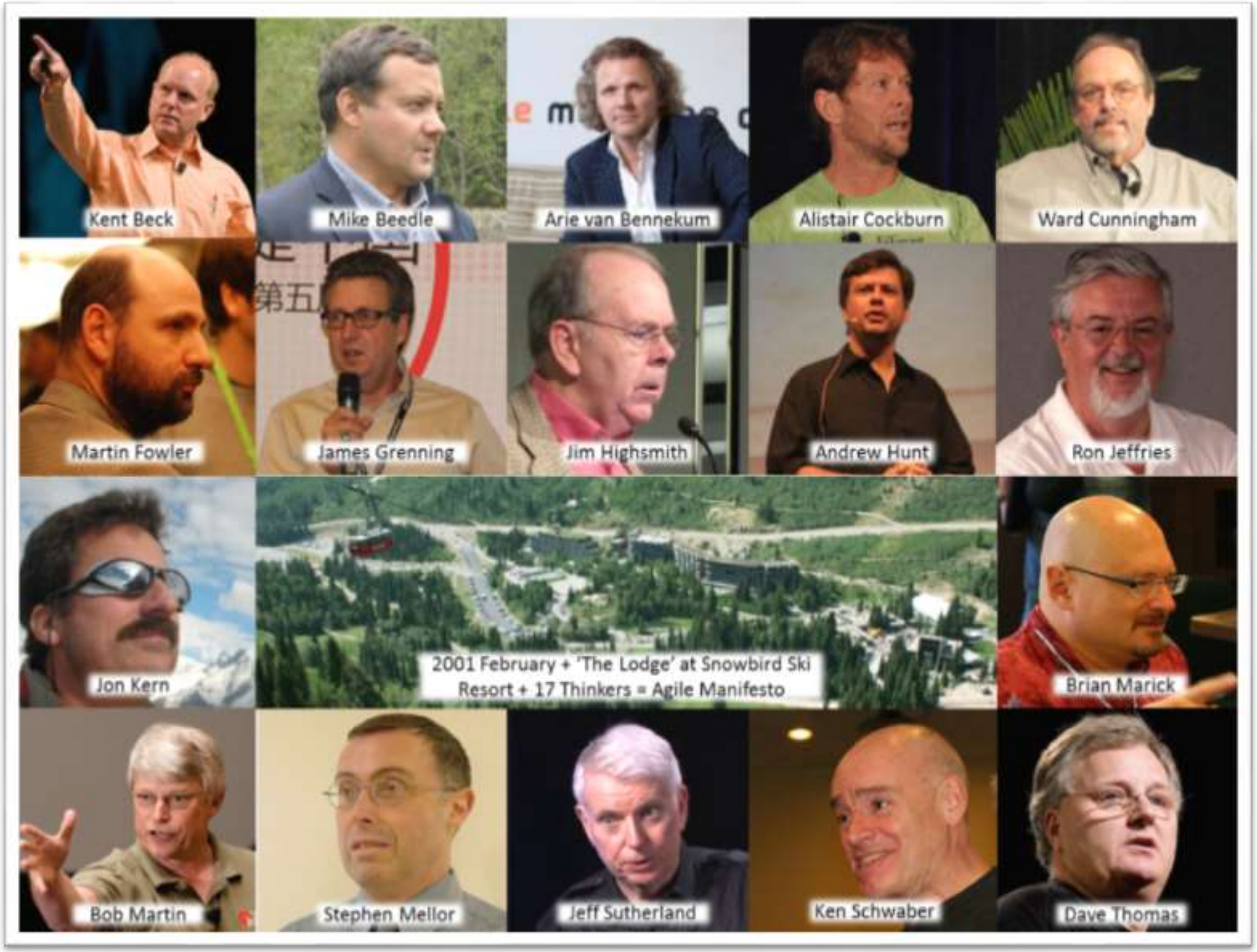
Standish Group Study Reported at XP2002

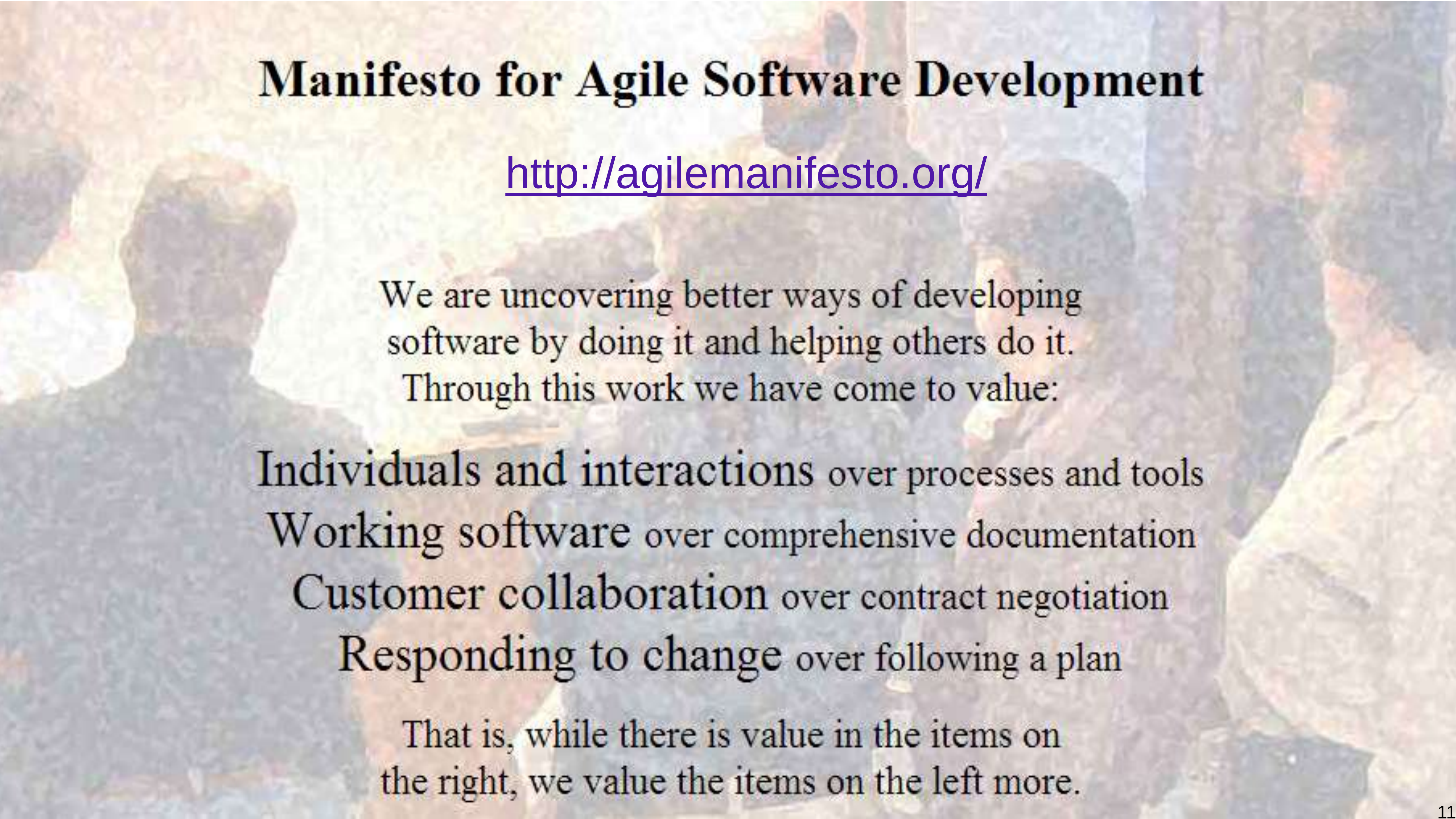
Delivered Feature Utilization




Document1 - Microsoft Word
File
Edit
View
Insert
Format
Tools
Table
Window
Help
Type a question for help
100%
Read
Normal
Times New Roman
12
All Entries
New...
Final Showing Markup
Show
New Frame Left
New Frame Right
New Frame Above
New Frame Below
Times New Roman
12
Insert Word Field
½ pt
Security...
Favorites
Go
<Click Recount to view>
Recount
Edit Text...
Body text
Update TOC
Getting Started
Microsoft Office Online
Connect to Microsoft Office Online
Get the latest news about using Word
Automatically update this list from the web
More...
Draw
AutoShapes
Page 1
Sec 1
1/1
At 1"
Ln 1
Col 1
REC
TRK
EXT
OVR
English (U.S.

The Agile Manifesto was written in February, 2001 by 17 independent-minded software practitioners.





Manifesto for Agile Software Development

http://agilemanifesto.org/

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

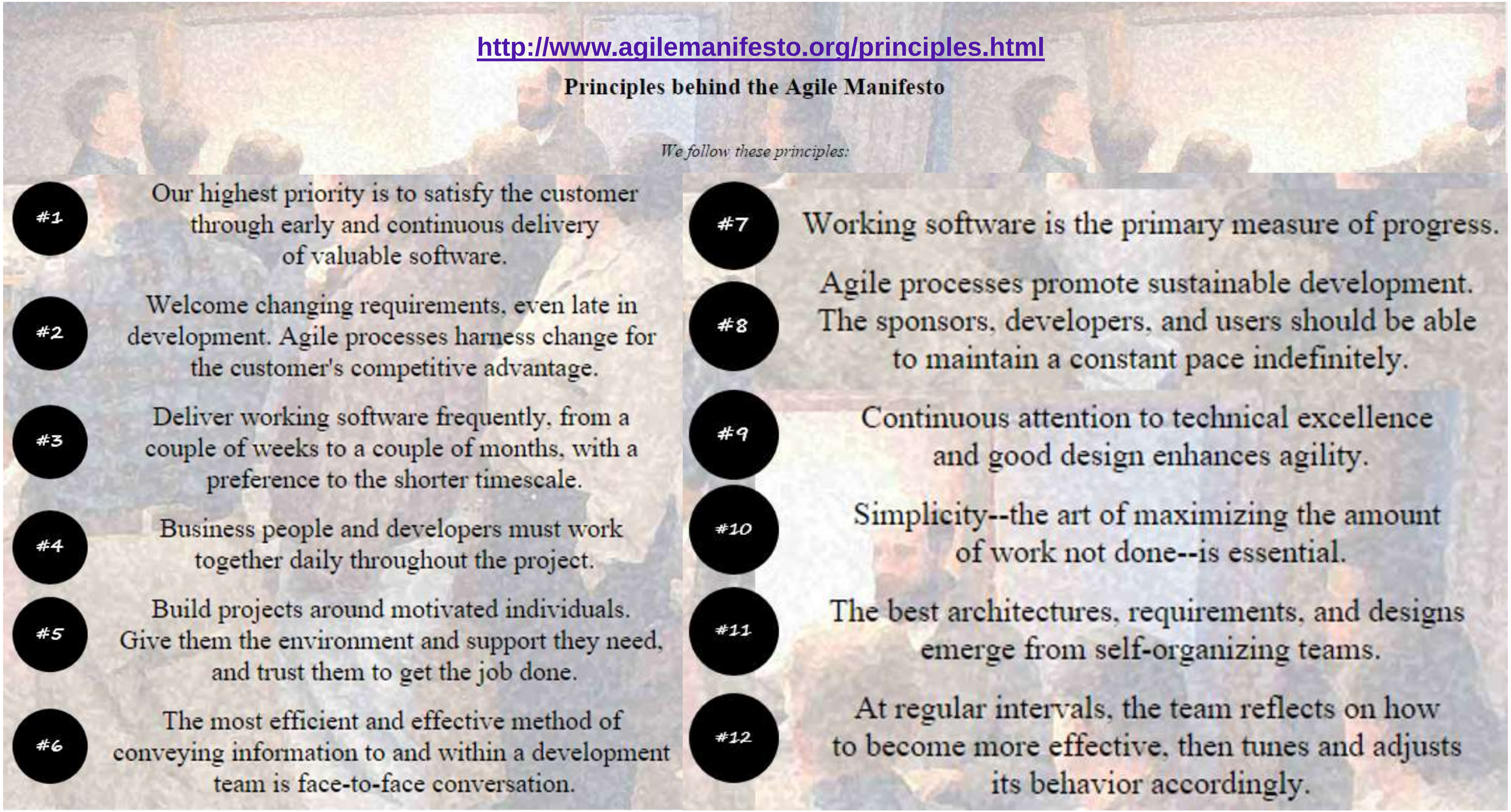
False Dichotomy

As Scott Ambler elucidated:

- Tools and processes are important, but it is more important to have competent people working together effectively.
- Good documentation is useful in helping people to understand how the software is built and how to use it, but the main point of development is to create software, not documentation.
- A contract is important but is no substitute for working closely with customers to discover what they need.
- A project plan is important, but it must not be too rigid to accommodate changes in technology or the environment, stakeholders' priorities, and people's understanding of the problem and its solution.

No CEOs, just passionate people!

2001 February + 'The Lodge' at Snowbird Ski Resort + 17 Thinkers = Agile Manifesto



http://www.agilemanifesto.org/principles.html

Principles behind the Agile Manifesto

We follow these principles:

#1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

#2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

#3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

#4 Business people and developers must work together daily throughout the project.

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

#6 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

#7 Working software is the primary measure of progress.

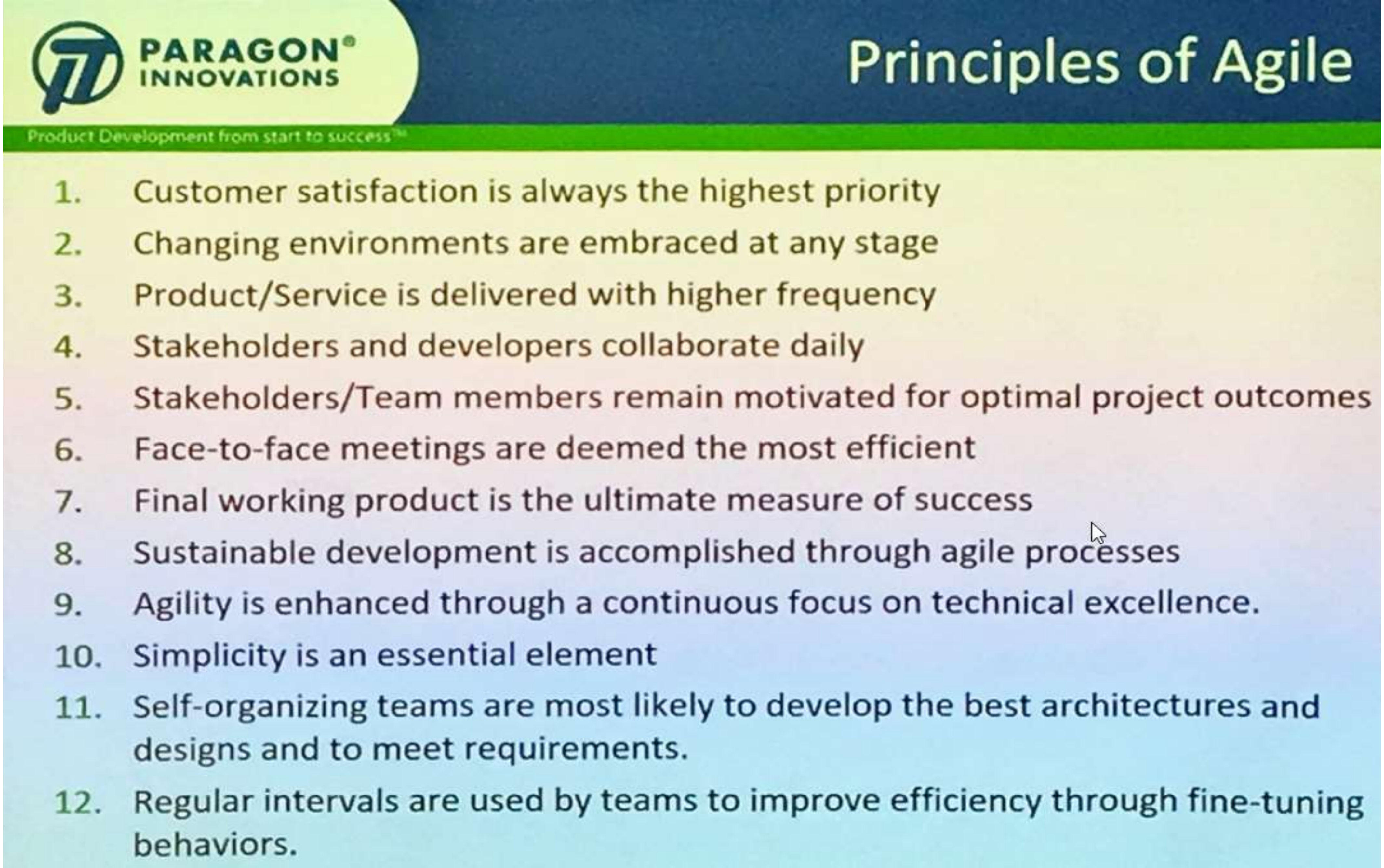
#8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

#9 Continuous attention to technical excellence and good design enhances agility.

#10 Simplicity--the art of maximizing the amount of work not done--is essential.

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

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



PARAGON® INNOVATIONS

Product Development from start to success™

Principles of Agile

1. Customer satisfaction is always the highest priority
2. Changing environments are embraced at any stage
3. Product/Service is delivered with higher frequency
4. Stakeholders and developers collaborate daily
5. Stakeholders/Team members remain motivated for optimal project outcomes
6. Face-to-face meetings are deemed the most efficient
7. Final working product is the ultimate measure of success
8. Sustainable development is accomplished through agile processes
9. Agility is enhanced through a continuous focus on technical excellence.
10. Simplicity is an essential element
11. Self-organizing teams are most likely to develop the best architectures and designs and to meet requirements.
12. Regular intervals are used by teams to improve efficiency through fine-tuning behaviors.

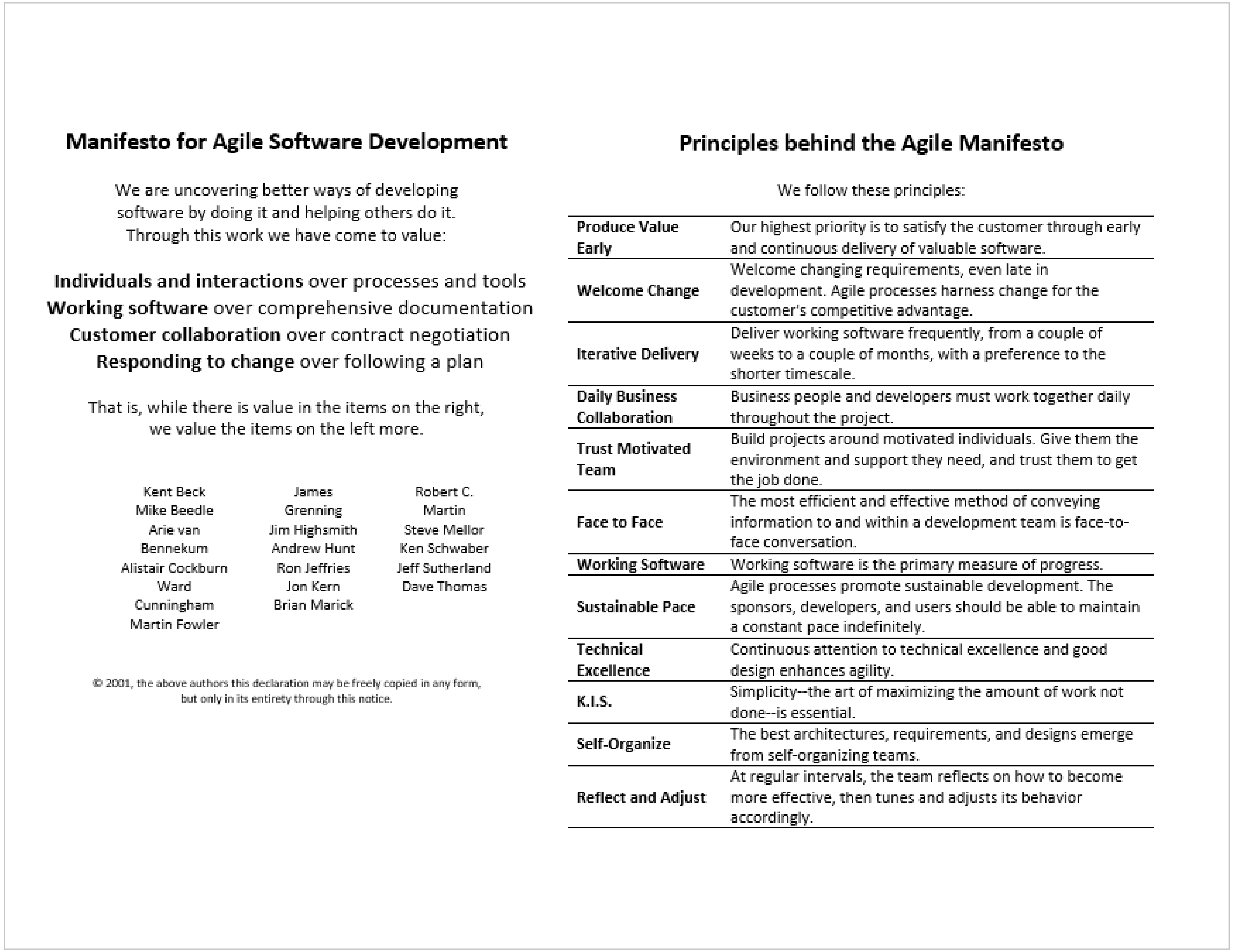
Michael R. Wilkinson, Chief Executive Officer and Founder of Paragon Innovations

What is Agile?

- Agile is
  4 values (more correctly, value pairs)
  12 principles
- Agile is not a framework (a lightweight set of rules)
- Agile is not a methodology (a complete set of rules, procedures)

- However, there are frameworks (Scrum) and methods (Kanban) that, properly implemented, are Agile in nature!

What is Agile?



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right,
we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

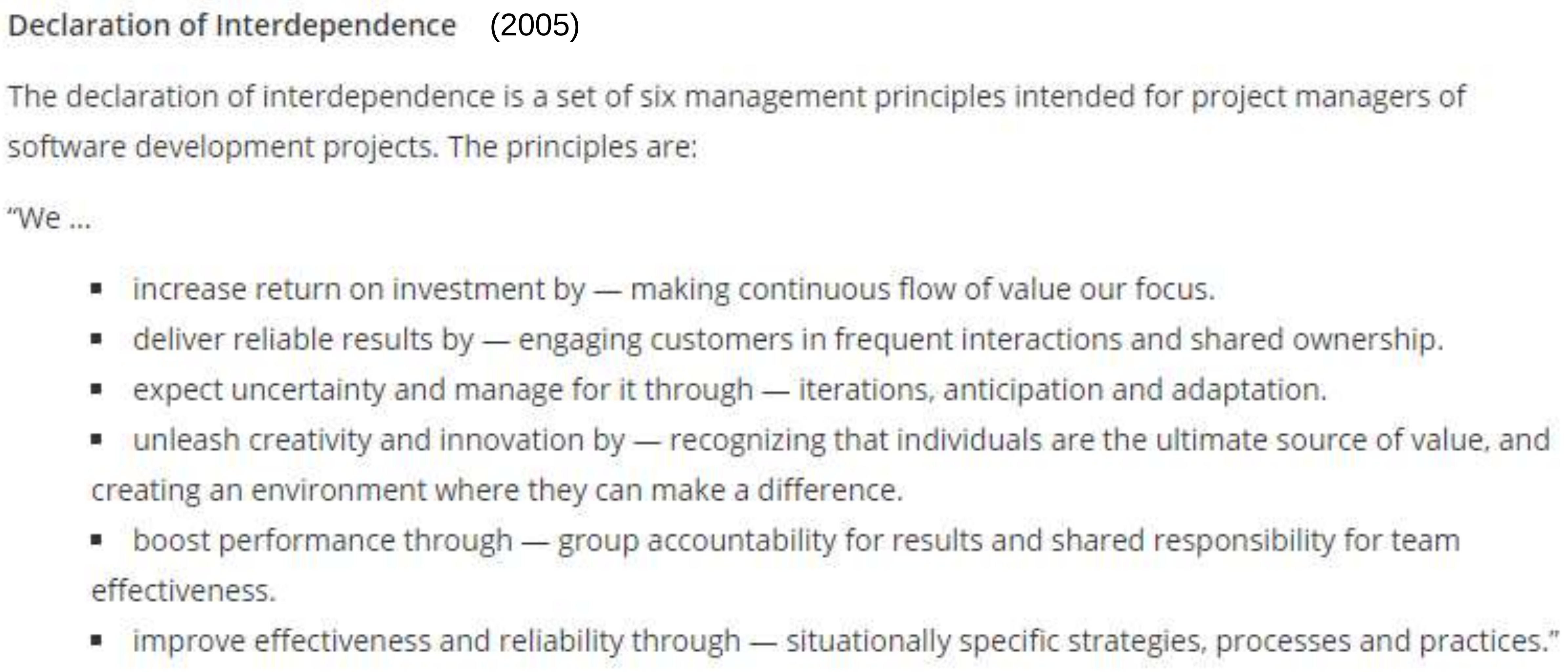
Principles behind the Agile Manifesto

We follow these principles:

Produce Value Early	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Welcome Change	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Iterative Delivery	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Daily Business Collaboration	Business people and developers must work together daily throughout the project.
Trust Motivated Team	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
Face to Face	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Working Software	Working software is the primary measure of progress.
Sustainable Pace	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Technical Excellence	Continuous attention to technical excellence and good design enhances agility.
K.I.S.	Simplicity--the art of maximizing the amount of work not done--is essential.
Self-Organize	The best architectures, requirements, and designs emerge from self-organizing teams.
Reflect and Adjust	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

QUESTIONS?

Inspired by Manifesto for Agile Software Development



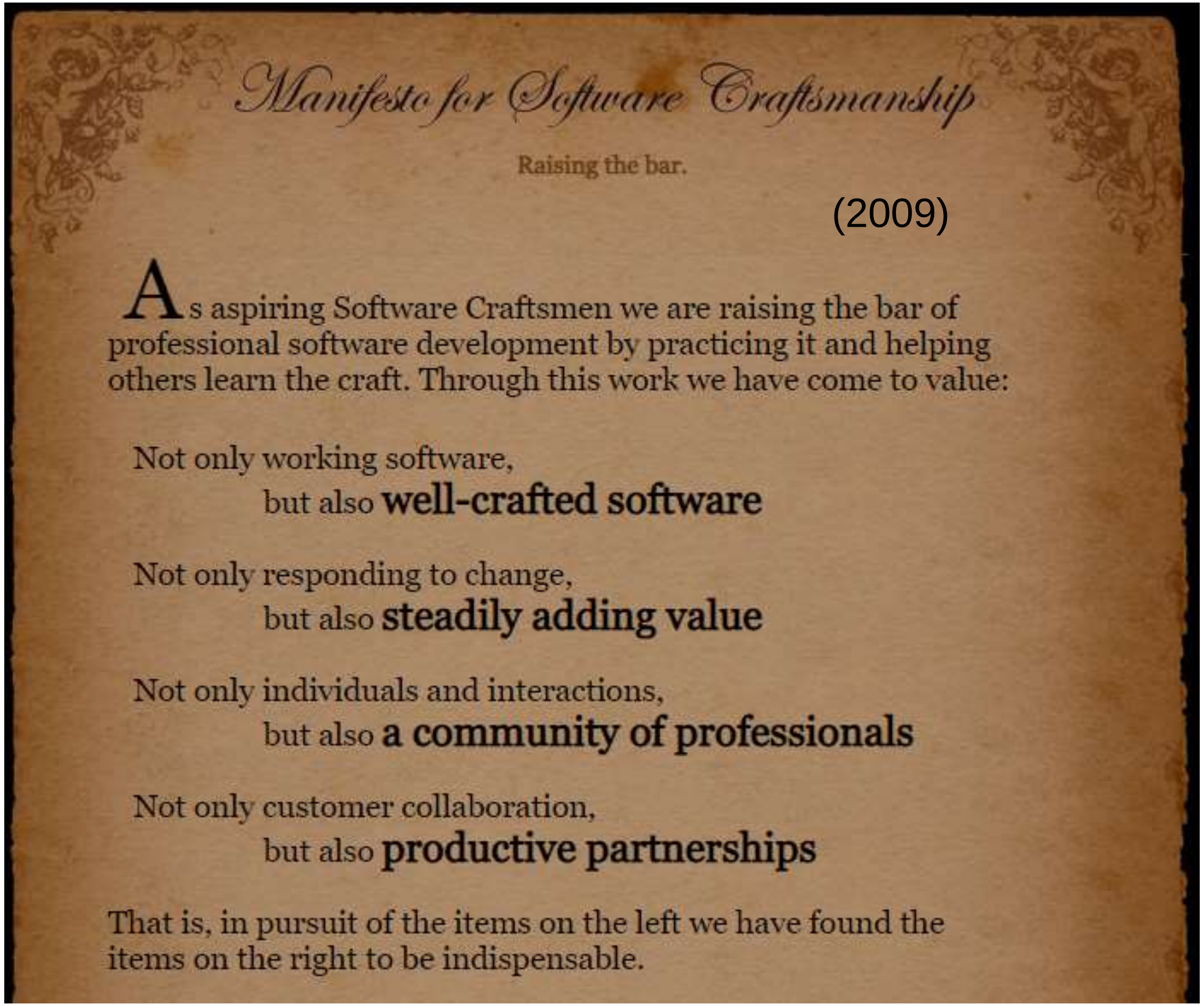
Declaration of Interdependence (2005)

The declaration of interdependence is a set of six management principles intended for project managers of software development projects. The principles are:

“We ...

- increase return on investment by — making continuous flow of value our focus.
- deliver reliable results by — engaging customers in frequent interactions and shared ownership.
- expect uncertainty and manage for it through — iterations, anticipation and adaptation.
- unleash creativity and innovation by — recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
- boost performance through — group accountability for results and shared responsibility for team effectiveness.
- improve effectiveness and reliability through — situationally specific strategies, processes and practices.”

Inspired by Manifesto for Agile Software Development



Manifesto for Software Craftsmanship

Raising the bar.

(2009)

As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software,
but also **well-crafted software**

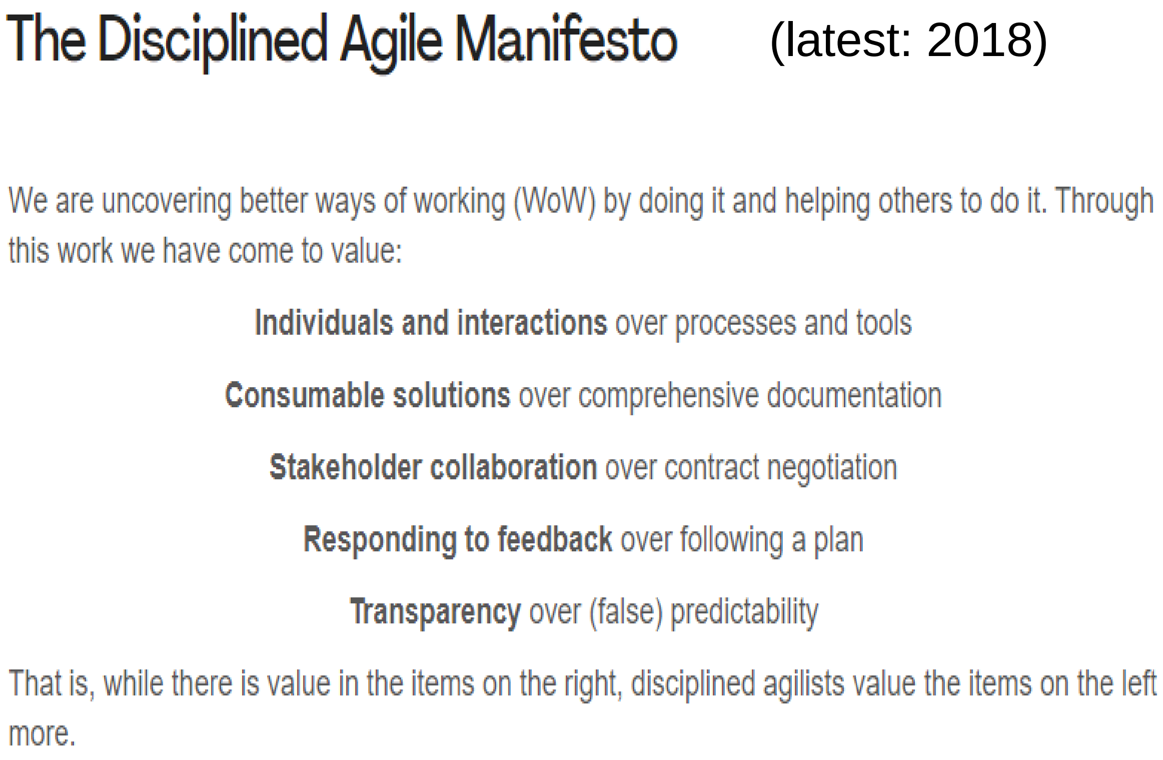
Not only responding to change,
but also **steadily adding value**

Not only individuals and interactions,
but also **a community of professionals**

Not only customer collaboration,
but also **productive partnerships**

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

Inspired by Manifesto for Agile Software Development



The Disciplined Agile Manifesto

(latest: 2018)

We are uncovering better ways of working (WoW) by doing it and helping others to do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Consumable solutions over comprehensive documentation

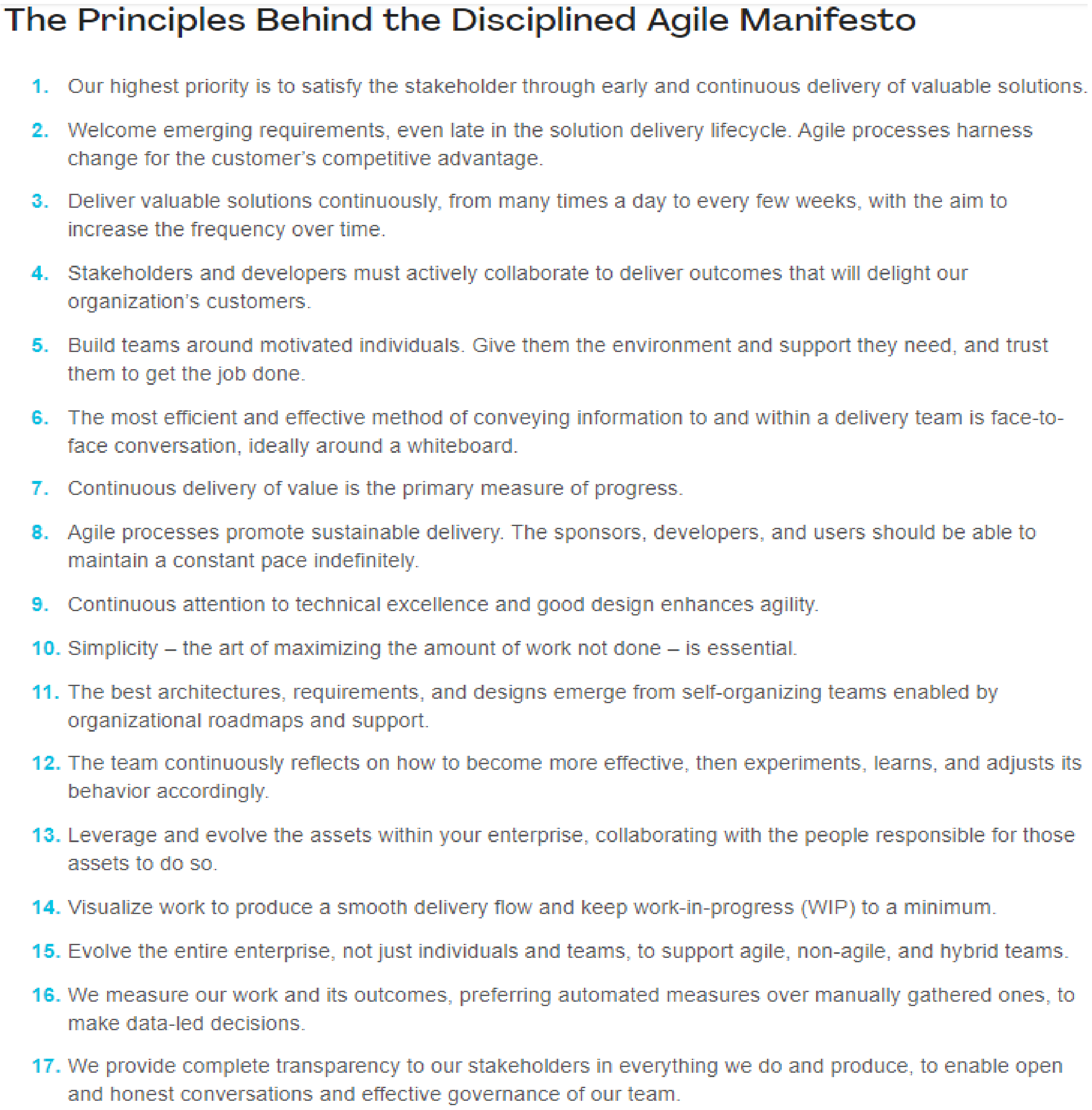
Stakeholder collaboration over contract negotiation

Responding to feedback over following a plan

Transparency over (false) predictability

That is, while there is value in the items on the right, disciplined agilists value the items on the left more.

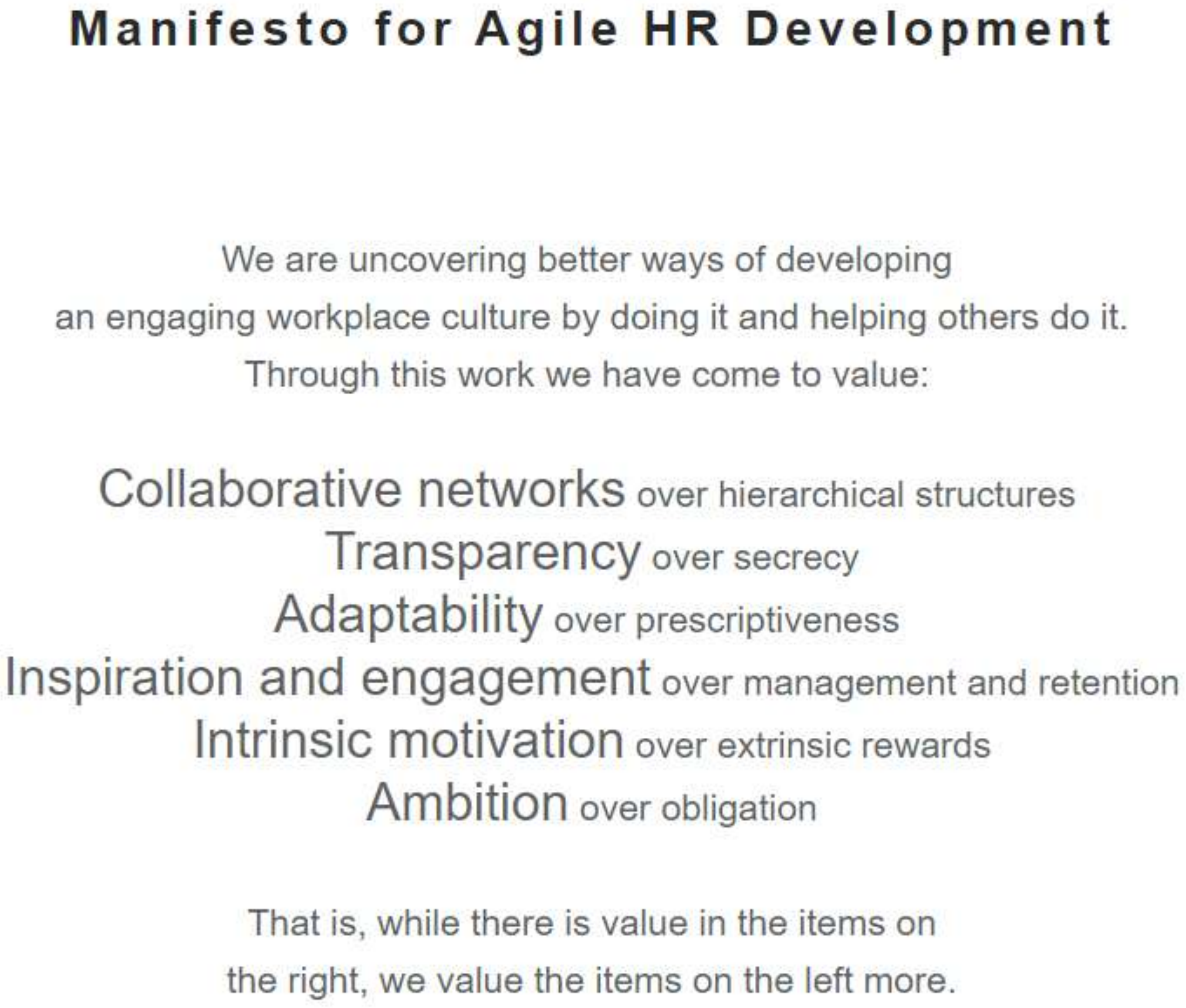
The Disciplined Agile Manifesto is an extension of the original Manifesto … that reflects the philosophies behind the Disciplined Agile (DA) toolkit.



The Principles Behind the Disciplined Agile Manifesto

1. Our highest priority is to satisfy the stakeholder through early and continuous delivery of valuable solutions.
2. Welcome emerging requirements, even late in the solution delivery lifecycle. Agile processes harness change for the customer's competitive advantage.
3. Deliver valuable solutions continuously, from many times a day to every few weeks, with the aim to increase the frequency over time.
4. Stakeholders and developers must actively collaborate to deliver outcomes that will delight our organization's customers.
5. Build teams around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a delivery team is face-to-face conversation, ideally around a whiteboard.
7. Continuous delivery of value is the primary measure of progress.
8. Agile processes promote sustainable delivery. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams enabled by organizational roadmaps and support.
12. The team continuously reflects on how to become more effective, then experiments, learns, and adjusts its behavior accordingly.
13. Leverage and evolve the assets within your enterprise, collaborating with the people responsible for those assets to do so.
14. Visualize work to produce a smooth delivery flow and keep work-in-progress (WIP) to a minimum.
15. Evolve the entire enterprise, not just individuals and teams, to support agile, non-agile, and hybrid teams.
16. We measure our work and its outcomes, preferring automated measures over manually gathered ones, to make data-led decisions.
17. We provide complete transparency to our stakeholders in everything we do and produce, to enable open and honest conversations and effective governance of our team.

Inspired by Manifesto for Agile Software Development

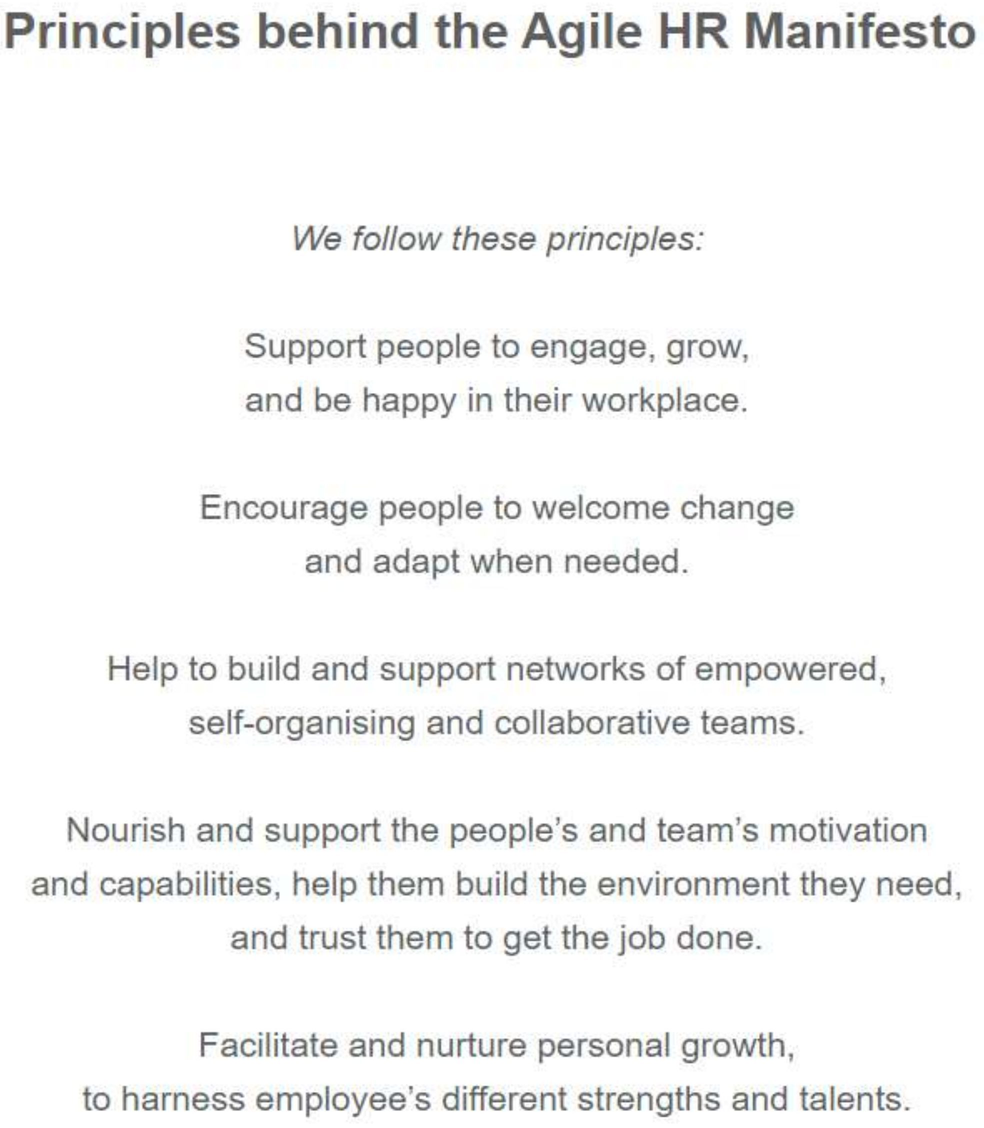


Manifesto for Agile HR Development

We are uncovering better ways of developing
an engaging workplace culture by doing it and helping others do it.
Through this work we have come to value:

Collaborative networks over hierarchical structures
Transparency over secrecy
Adaptability over prescriptiveness
Inspiration and engagement over management and retention
Intrinsic motivation over extrinsic rewards
Ambition over obligation

That is, while there is value in the items on
the right, we value the items on the left more.



Principles behind the Agile HR Manifesto

We follow these principles:

Support people to engage, grow,
and be happy in their workplace.

Encourage people to welcome change
and adapt when needed.

Help to build and support networks of empowered,
self-organising and collaborative teams.

Nourish and support the people's and team's motivation
and capabilities, help them build the environment they need,
and trust them to get the job done.

Facilitate and nurture personal growth,
to harness employee's different strengths and talents.

LEAN, DEVOPS, AND PRODUCT CENTRICITY

What is Lean?

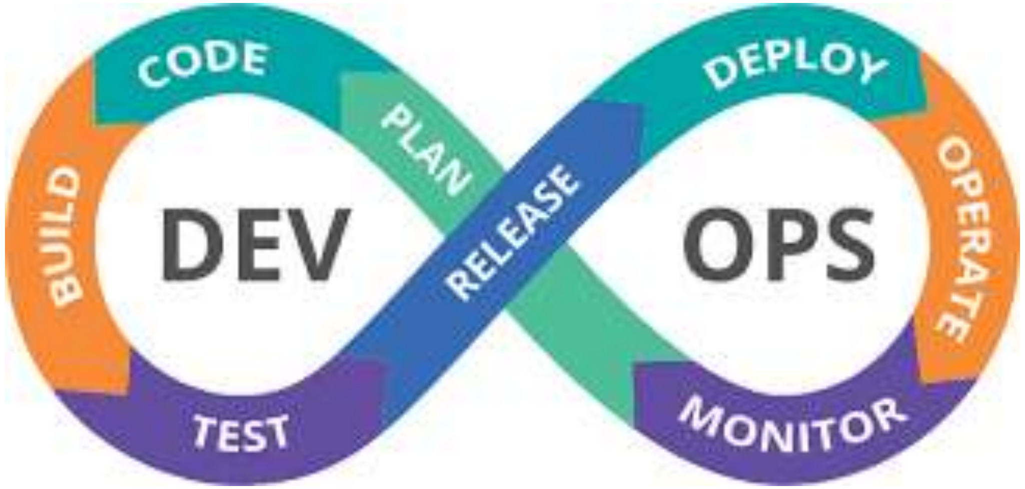
- Based on manufacturing - Toyota Production System
- "...a way to do more and more with less and less … while coming closer and closer to providing customers exactly what they want"
- Five key principles
  - Value
  - The Value Stream
  - Flow
  - Pull
  - Perfection

- Term “Lean-Agile” or “Agile-Lean” is becoming common

What is DevOps (DevSecOps, Dev____Ops)

- Fairly recent – mid-2000’s
- A set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality

Product Increments
Static Code Analysis
Automated Testing

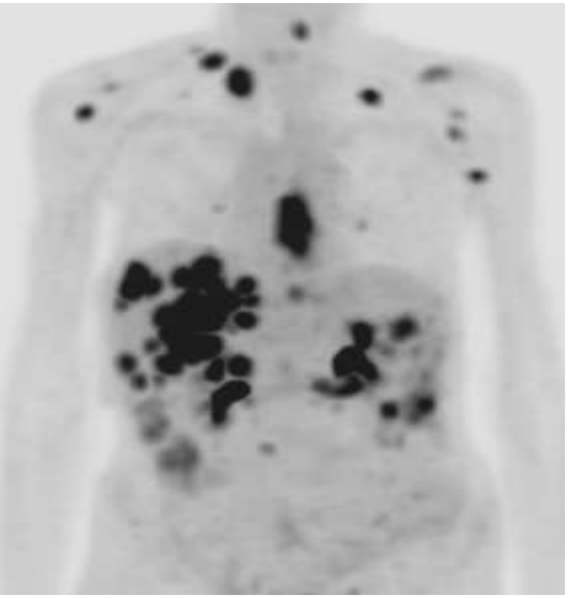


Packaging
Automated promotions
Audit
Logging
Monitoring and Alerting

Product Centric Organizations

- Evolution from Project centric to Product centric organizations
- Focus on customer experience, evolving requirements, and the strategic differentiation
- Long term play:
    - Relationship to the customer
    - Managers of product vision and direction
    - Teams that support the product

EMPIRICISM



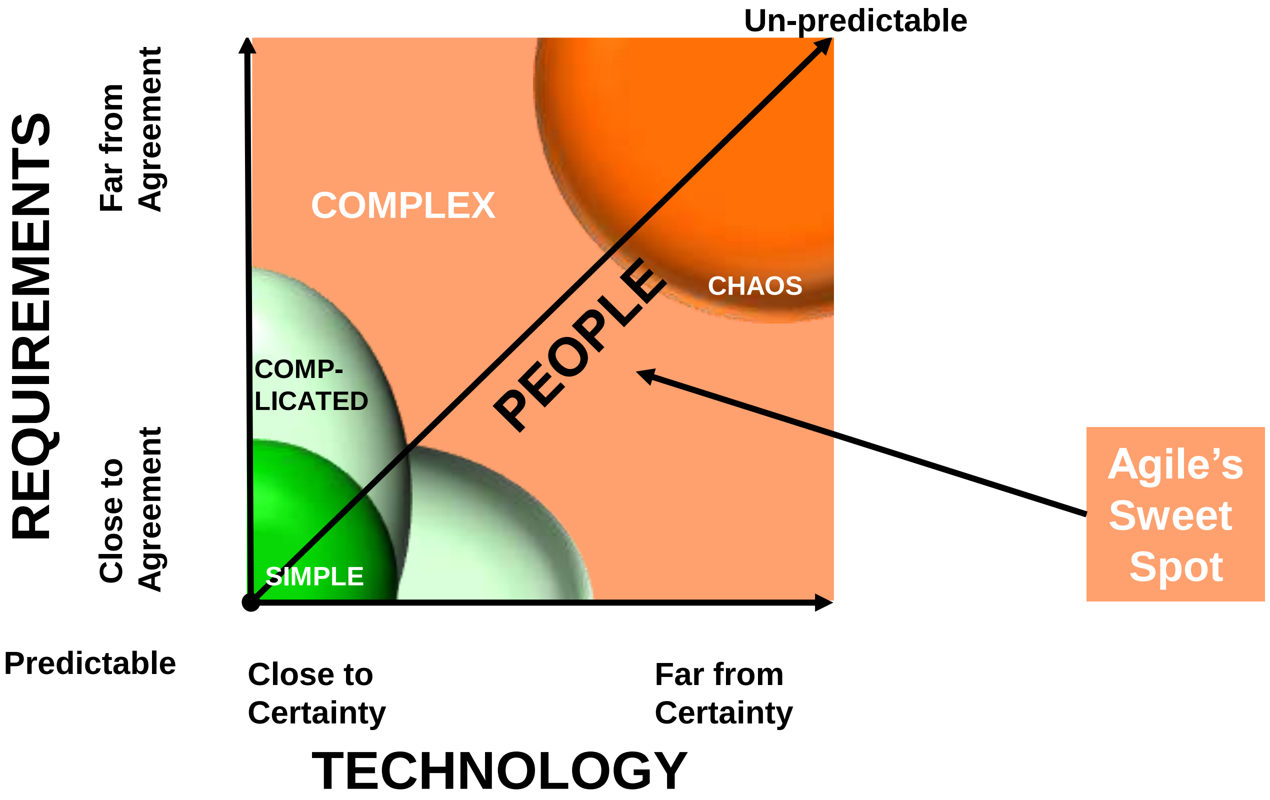
Predict and Plan (Predictive)

- Similar problems to solve
- Certainty in desired outcomes, technologies, people, effectiveness

Sense and Respond (Adaptive)

- Uncertainty in desired outcomes, technologies, people, effectiveness
- Empirical Process Control (only decide based on observed data)

Complex Software Delivery

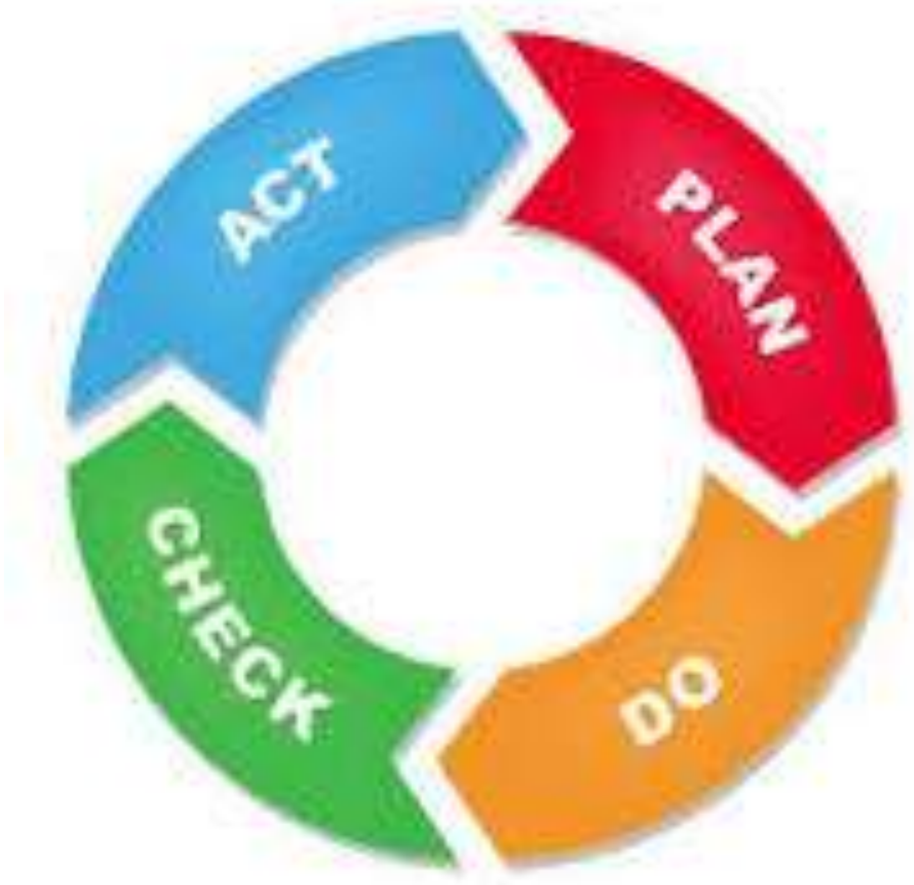


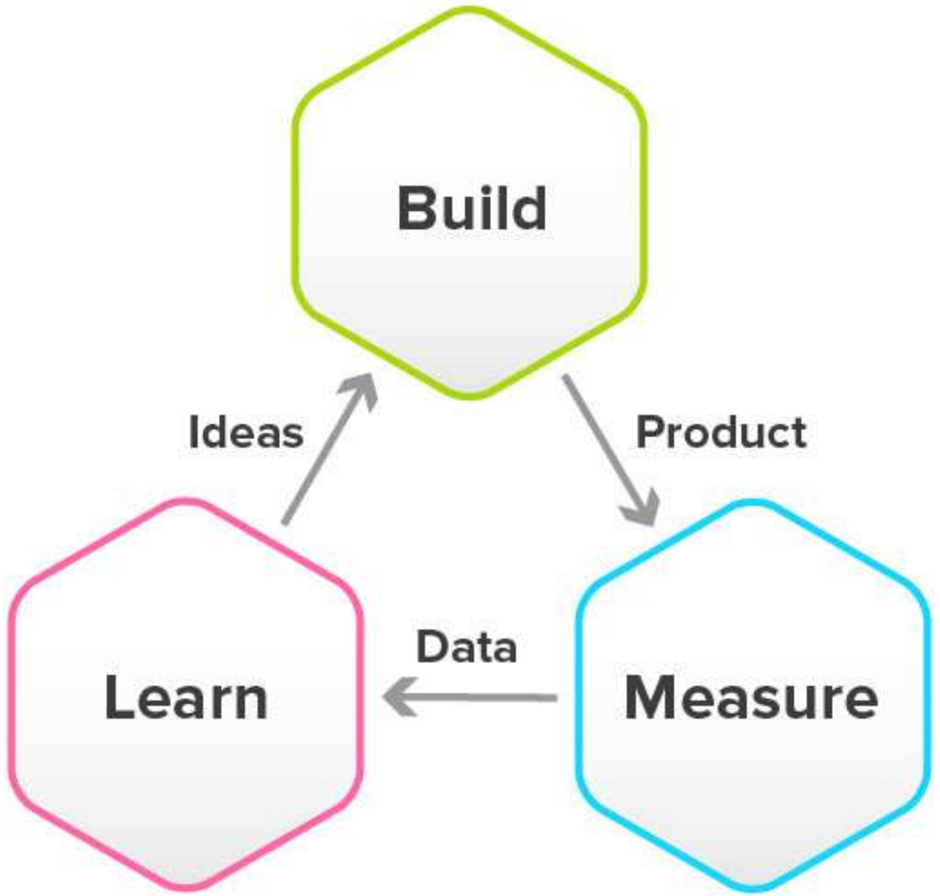
Adapted Stacey Matrix –
Strategic Management and Organizational Dynamics,
Ralph D. Stacey

Making Complex Simple

Sashimi is a Japanese delicacy consisting of very fresh raw meat or fish sliced into *thin pieces*.

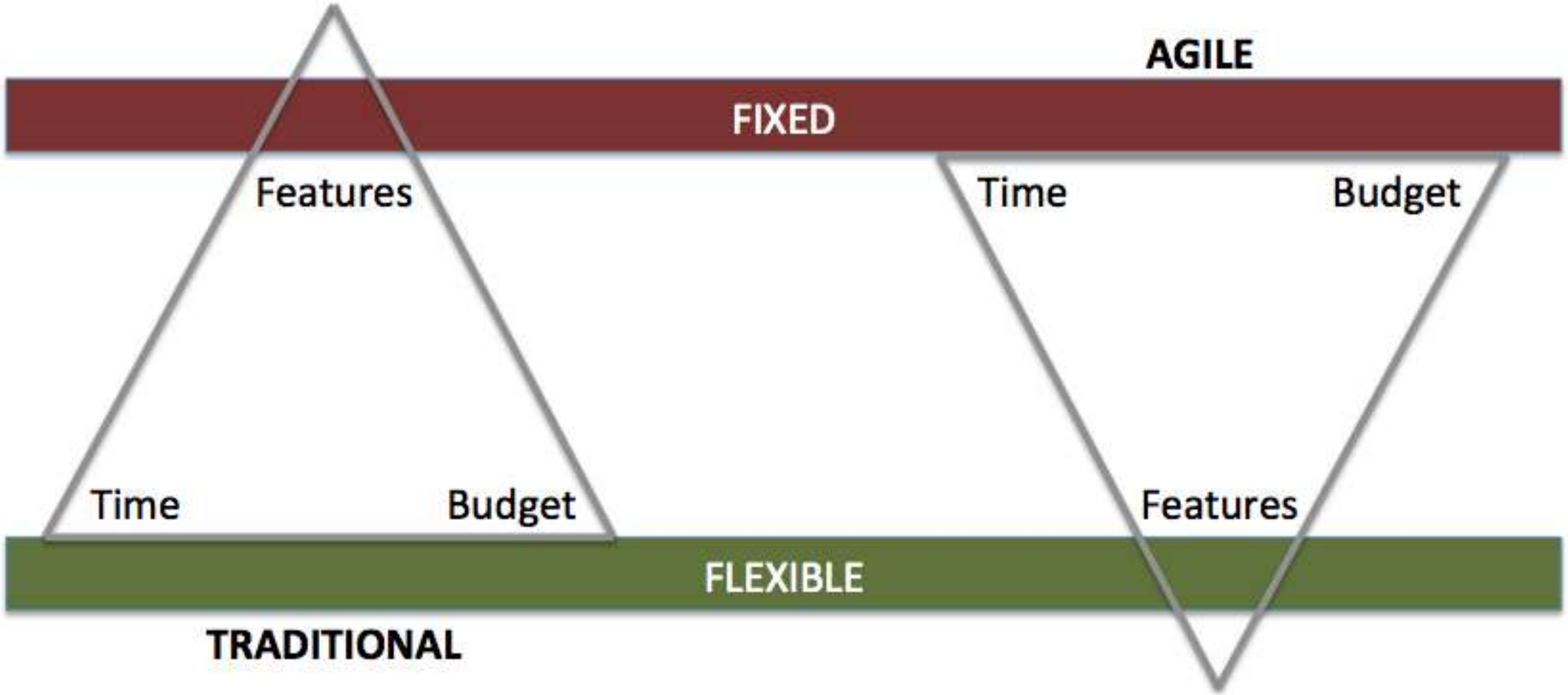
Making Complex Simple





calltheone.com/en/management-goeroes

Paradigm Shifts



Project Scope is set first, followed by time and cost.

Build the team and short iteration schedule, then deliver features (highest value first).

Can leverage the Pareto Principle!

QUESTIONS?

SCRUM

Implementations

The following are processes that embody the agile principles to deliver consistent value in varies ways:

Scrum
(Framework)

- Most popular
- Team focused
- Utilizes iterative development cycles
- Standard roles, artifacts and events
- Incremental value delivery

Extreme Programming
XP (Practices)

- Software development
- Focus is customer satisfaction
- Coding is key
- Test driven development
- Direct communication between customer and programmer
- Collective Ownership

Kanban
(Method/Framework)

- Based in Manufacturing
- Focus on the Big Picture
- Built in value
- Engage everyone
- Continuous learning and improvement

Best for Product Development
e.g. Weekly Sitcom

Best for Procedural or Interrupt Driven work
e.g. Day Surgery, ER

Origins of Scrum

- "Scrum" first appeared in a 1986 HBR article as a metaphor for a team-based product development process.
- Jeff Sutherland developed Scrum for software product development in 1993.
- Ken Schwaber formalized the process in presentation at OOPSLA 1995.
- The Scrum framework is defined in a 19-page guide.
- Written by Ken Schwaber and Jeff Sutherland.
- It‘s free.
- Available for download from www.scrumguides.org
- Last updated: November 2017

The Scrum Guide™

The Definitive Guide to Scrum:
The Rules of the Game

November 2017

Developed and sustained by Scrum creators: Ken Schwaber and Jeff Sutherland

Framework vs. Detailed Manual

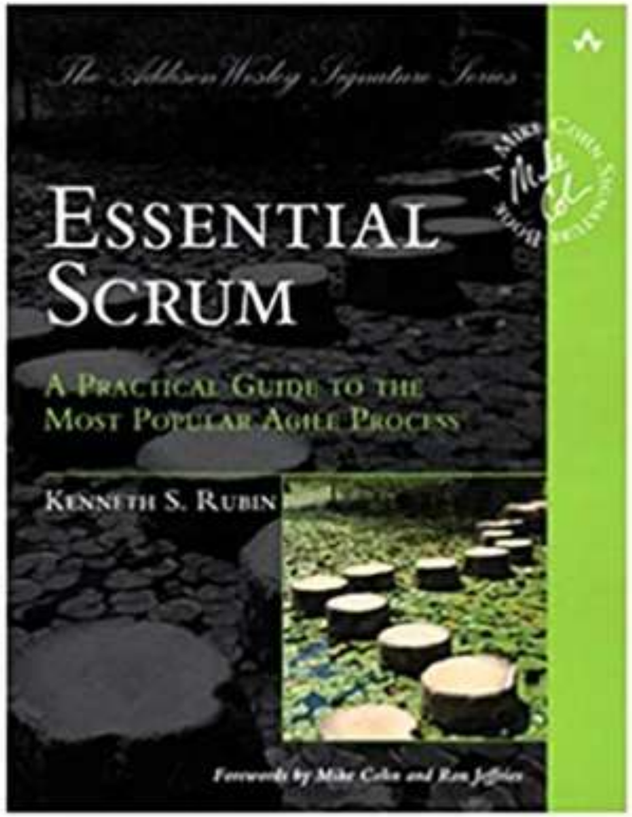
The Scrum Guide™

The Definitive Guide to Scrum:
The Rules of the Game

November 2017

Developed and sustained by Scrum creators: Ken Schwaber and Jeff Sutherland

19 pages



The Addison-Wesley Signature Series
A Mike Cohn Signature Book
Essential Scrum
A Practical Guide to the Most Popular Agile Process
Kenneth S. Rubin
Forewords by Mike Cohn and Ron Jeffries

500 pages

Other Than Software Scrum

wikispeed.org/

www.scruminc.com/wp-content/uploads/2015/09/Release-version_Owning-the-Sky-with-Agile.pdf

Nederlands
English
German
Spanish
Internatio
eduScrum
HOMEPAGE / ABOUT US / NEWS / TRAINING / FAQ / CONTACT / BLOGS
ABOUT EDUSCRUM / EDUSCRUM IN CLASS / TITLES AND LINES
ENJOYABLE

With eduscrum cooperating teams will function optimally. Students enjoy this and will turn into increasingly better team players who are valued for their qualities.

eduscrum.nl/en/

www.scrumyourwedding.com/

How is Scrum Different

Predict and Plan

Certainty in desired outcomes, technologies, people, effectiveness in processes employed

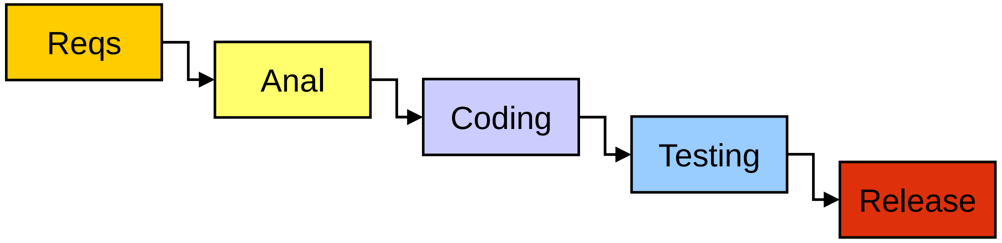
Sense and Respond

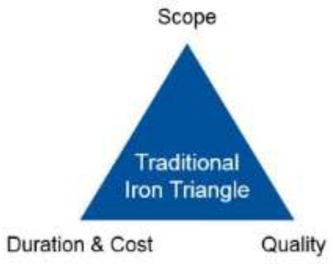
Uncertainty with any…

- Desired outcomes
- Technologies
- People
- Processes

Waterfall Model

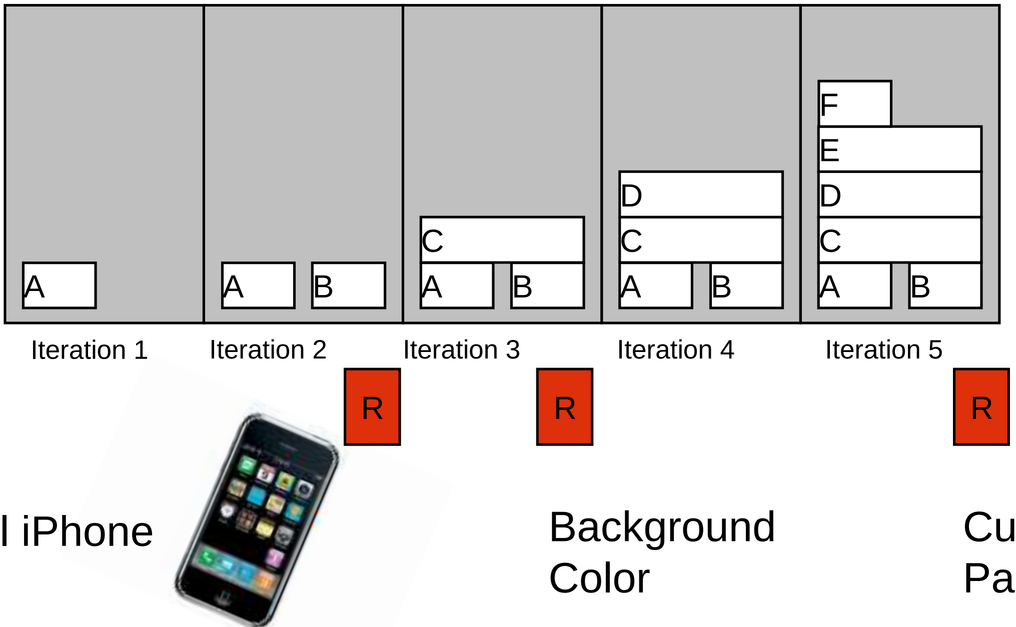
Phased or Gated



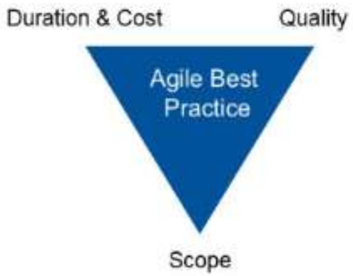


Scrum Framework

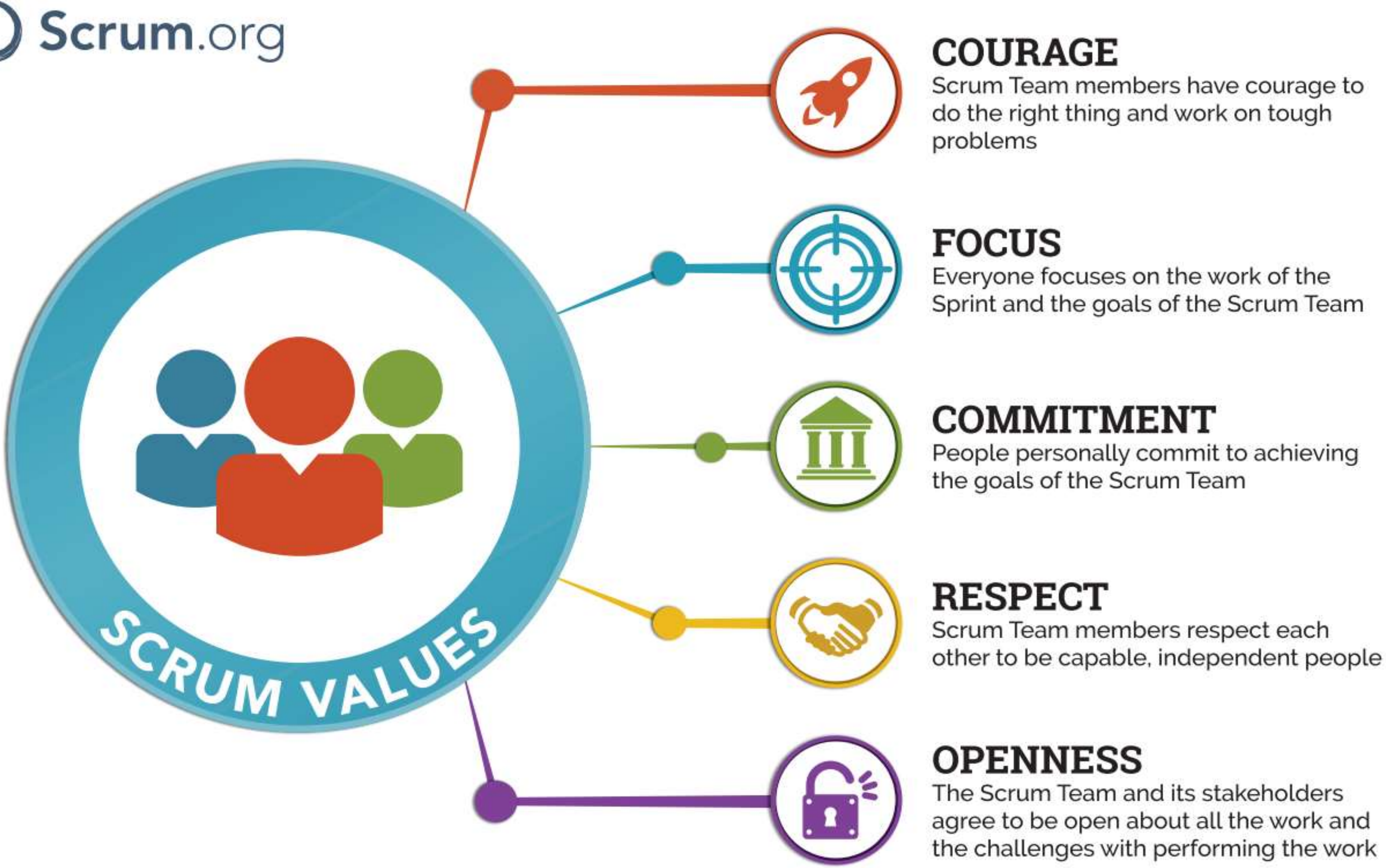
Iterative & Incremental



Original iPhone



App Store
3g Support
Pic over SMS
Notifications



SCRUM VALUES

COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

© Scrum.org

Manifested in how people treat each other – and in the Team Working Agreements

Three Pillars of an Empirical Control Process

Transparency

- Nothing is hidden; all requirements are available to everyone responsible for completing them

Inspection

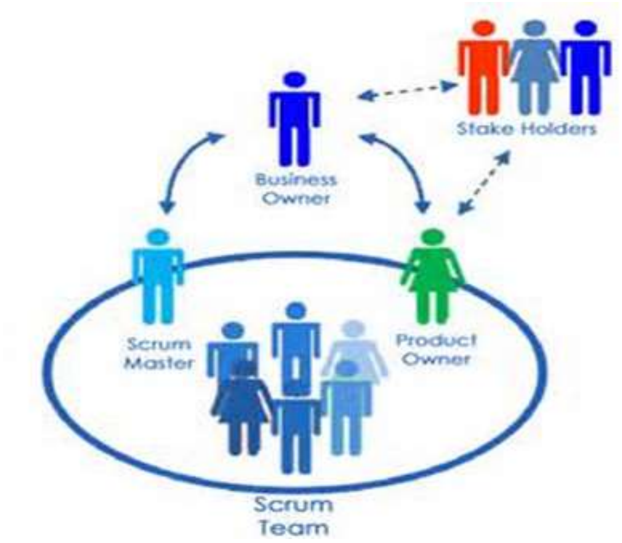
- Constantly examine artifacts to correct undesirable outcomes

Adaption

- Constantly make update to processes and artifacts based on lessons learned

Make plans based on observations,
not what you expect (hope) will happen.

Scrum Roles



Product Owner

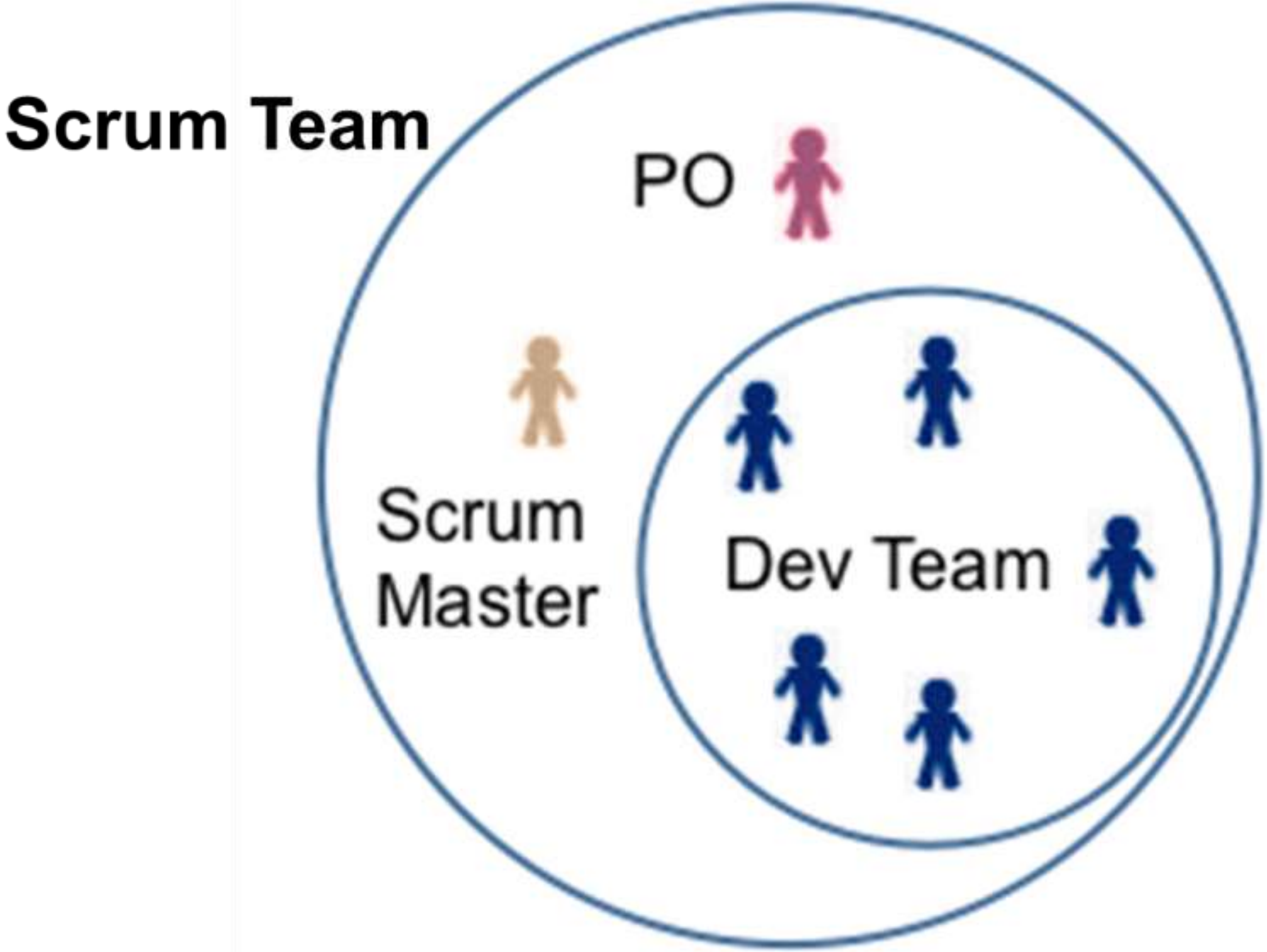
- Own the Product Vision
- Prioritizes work according to market value
- Adjusts features and priority every iteration, as needed
- Defines the features of the product and accepts/rejects work
- Decides when to release

Development Team

- Dedicated, self-organizing, cross-functional team with all the skills needed to complete the work
- Determines the amount of work to bring into the Sprint
- Determines how to turn backlog into an increment of functionality
- Size of team: 6 plus or minus 3

Scrum Master

- Removes impediments and shields the team from interferences
- Ensures that the team is fully functional and productive
- Enables close cooperation across all roles and functions (a Facilitator)



Scrum Team

PO

Scrum Master

Dev Team

Decisions Made by an Empowered Development Teams

- Estimates the size of their work as a team; backlog items, user stories, tasks, etc.

- For Sprint Planning:
  - Determines their capacity for the upcoming Sprint.
  - Decides the amount of work to bring into the Sprint.

  The team avoids exceeding their capacity. At the end of Sprint Planning, the team believes they can complete a vast majority (>80%) of their work to their Definition of Done by the end of the Sprint.

- Decides who does what to complete the work in the Sprint.

- Decides how to create the solution with minimal oversight or approvals. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality

- Decides what to do if they fall behind or get ahead during a Sprint.

- Decides which process improvement effort(s) they will undertake to improve their skills, the process or tools they use to create the solution or the relationships within the team or between the team and its stakeholders.

What about Other Roles in the Organization?

- Lead people, manage the environment.
- Make it easier for the team to deliver value.
- "What's getting in your way?"
  "How can I help you?"
  "Does our environment allow you do
  to your best work?"
  "Would you recommend…"
- Define "technical excellence" and empower the team to achieve it
  (e.g. allocate capacity to reduce
  tech debt and improve skill sets)
- How do I know what they are doing? Ask PO for Roadmap; attend Sprint Reviews.

Technical Debt is a Velocity TAX

Definition:

A metaphor for doing things in a quick and dirty way resulting in extra effort that we have to do in future development because of the quick and dirty design choice.

Examples:

- Known/existing defects
- Lack of automation
- Complex code
- Lack of Unit Tests
- Highly coupled code
- High Cyclomatic complexity
- Duplicate code
- Unreadable names

Building to your Definition of Done helps prevent *adding* technical debt to the system.

QUESTIONS?

Five Events & An Activity

Sprint

- A fixed period of time (typically 2-4 weeks) in which a small piece of the product is completed.

Sprint Planning

- What will be built in the next iteration.
- Define Sprint Goal.

Daily Scrum (not Stand-up)

- Limited to 15 minutes each day. Planning how the team will get highest priority done ASAP.

Sprint Review

- Customer sees/uses the completed work and provides feedback

Sprint Retrospective

- Reflect on what went well and how to get better.

Backlog Refinement

- Decompose, estimate and make ready (immediately actionable, no stopper) the top items on the Product Backlog.

Three Artifacts

Product Backlog – the desired list of project work

- The items needed to make a complete product
- Evolves and is dynamic
- Prioritized by the Product Owner and frequently reordered

Sprint Backlog – the work items to be completed in the sprint

- Estimated work remaining is updated daily
- Update work remaining as more becomes known

Product Increment

- Sum of all the Product Backlog items completed during a Sprint
- Working, Tested, Potentially Shippable

Artifact Transparency

Definition of Ready

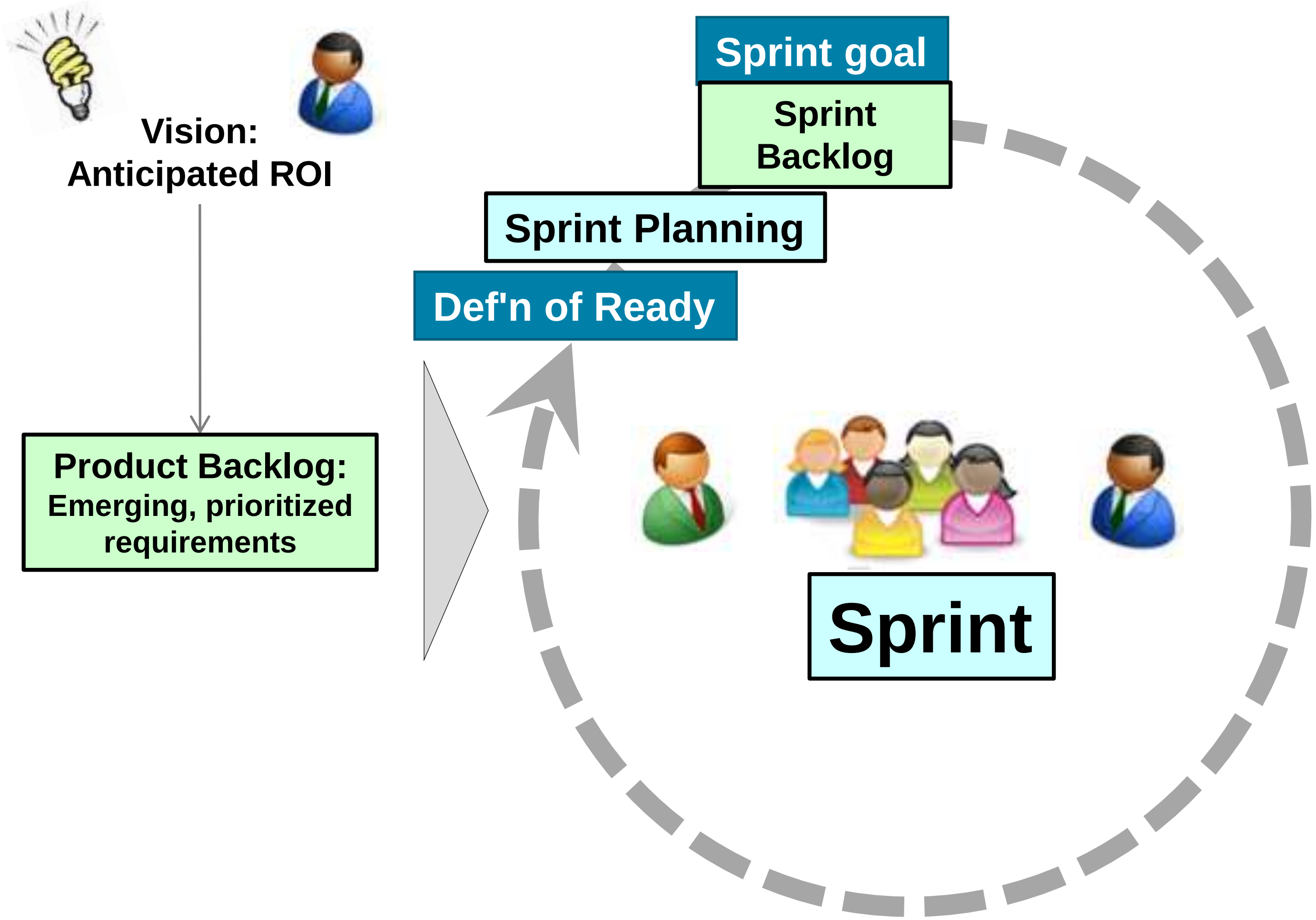
- The prerequisites to achieve to bring a work item into a sprint

Sprint Goal – a SMART short term objective

- Negotiated between the Product Owner and the Development Team during Sprint Planning
- Is met through the implementation of Product Backlog items

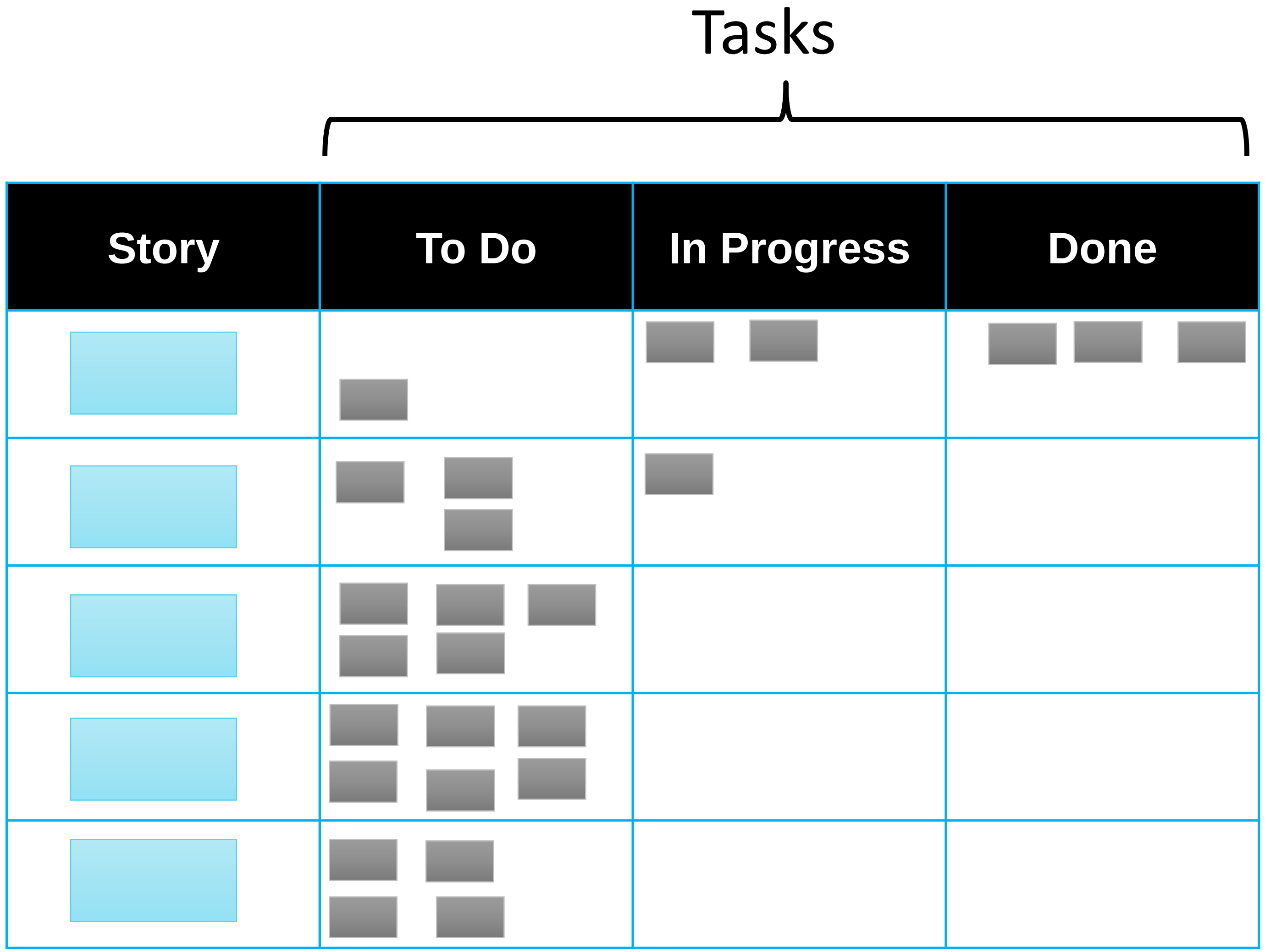
Definition of Done

- All the work needed to create a complete product increment that customers can immediately use


Vision:
Anticipated ROI
Product Backlog:
Emerging, prioritized
requirements
Sprint goal
Sprint
Backlog
Sprint Planning
Def'n of Ready
Sprint

Typical Task Board Updated the Following Day

Sprint Goal: Support payment by Bitcoin.



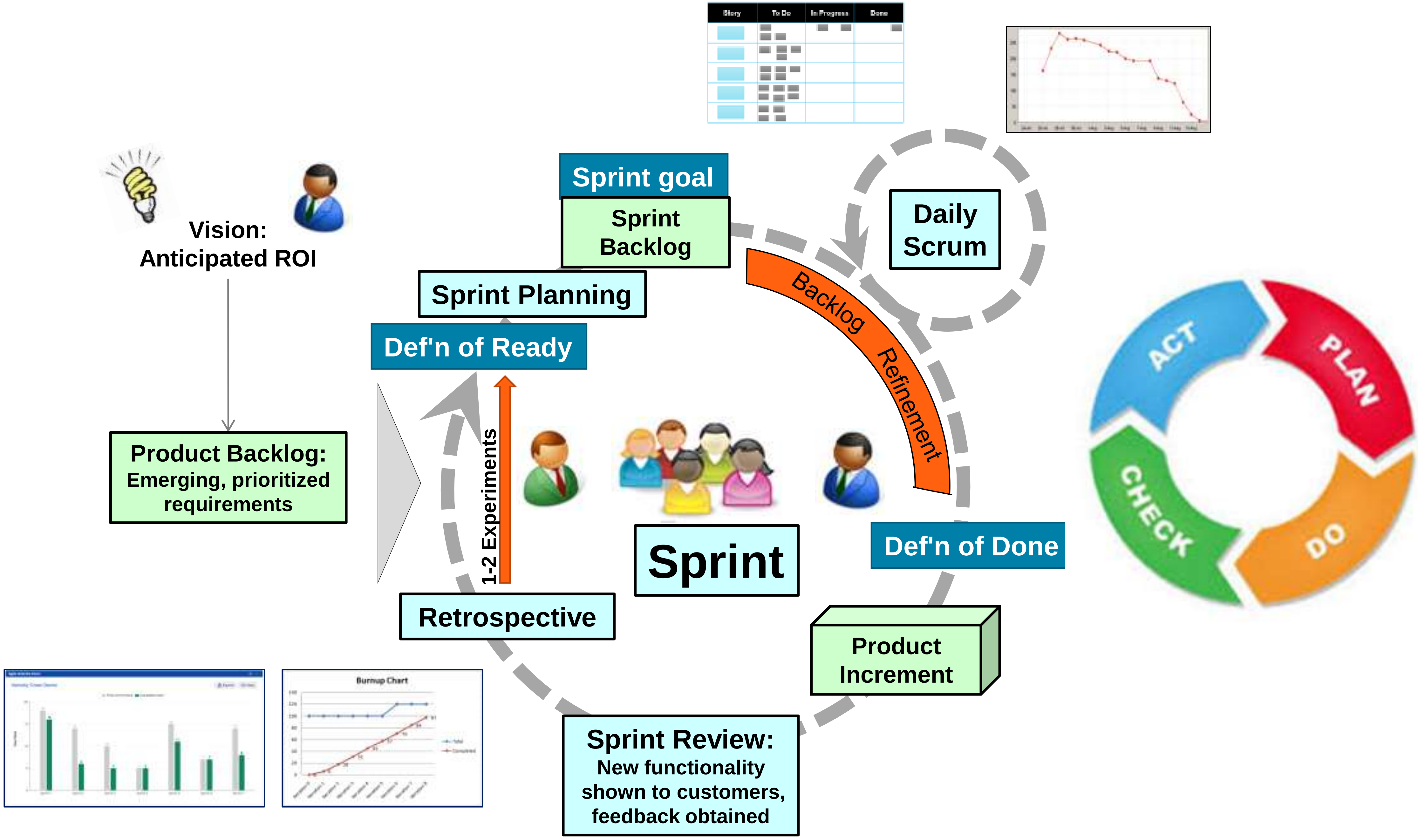
Tasks

Story

To Do

In Progress

Done



Vision:
Anticipated ROI

Product Backlog:
Emerging, prioritized requirements

Def'n of Ready

Sprint Planning

Sprint goal

Sprint Backlog

Daily Scrum

Backlog Refinement

1-2 Experiments

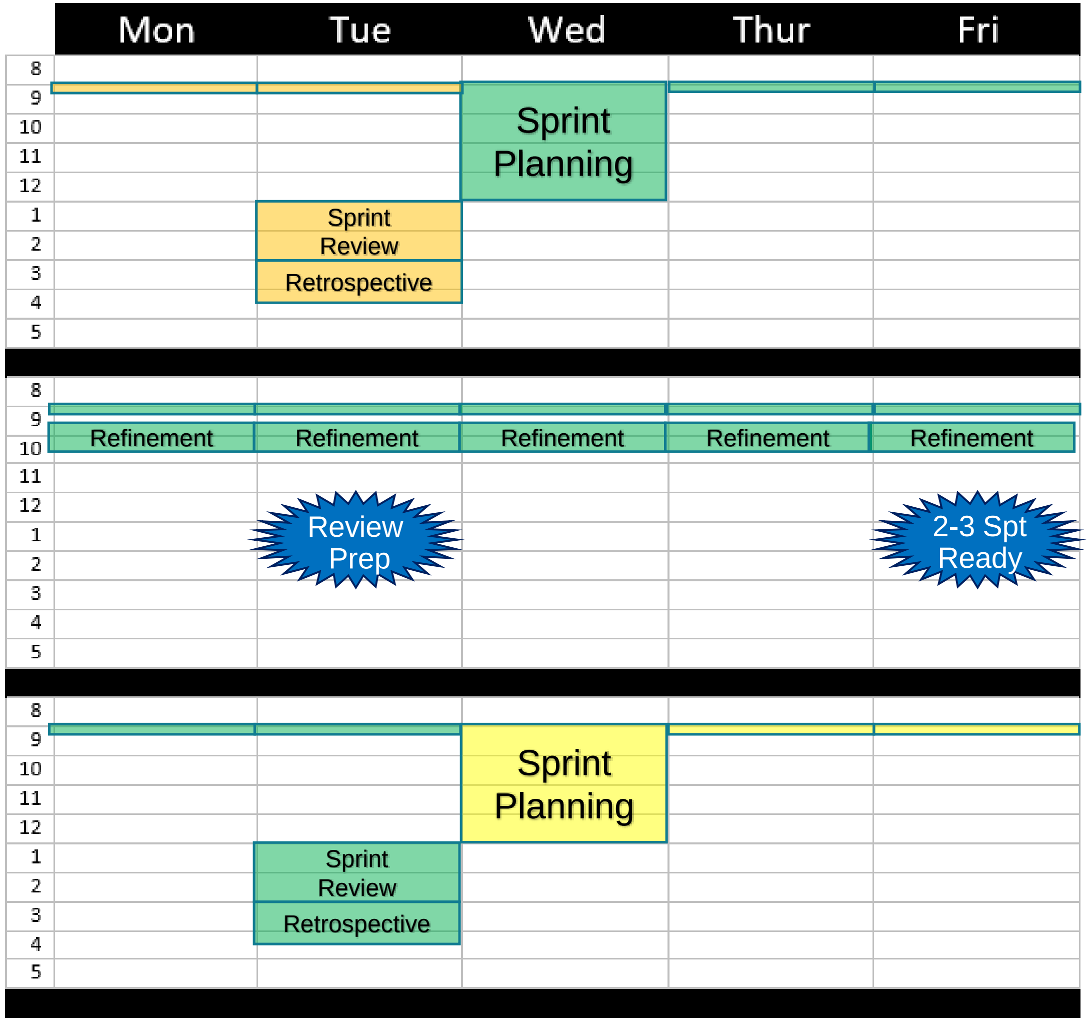
Sprint

Def'n of Done

Product Increment

Sprint Review:
New functionality shown to customers, feedback obtained

Retrospective


Mon
Tue
Wed
Thur
Fri
8
9
10
11
12
1
2
3
4
5
Sprint Planning
Sprint Review
Retrospective
Refinement
Refinement
Refinement
Refinement
Refinement
Review Prep
2-3 Spt Ready
Sprint Planning
Sprint Review
Retrospective

Daily Scrum

Scrum Guide is Silent On…

How to capture and estimate requirements.

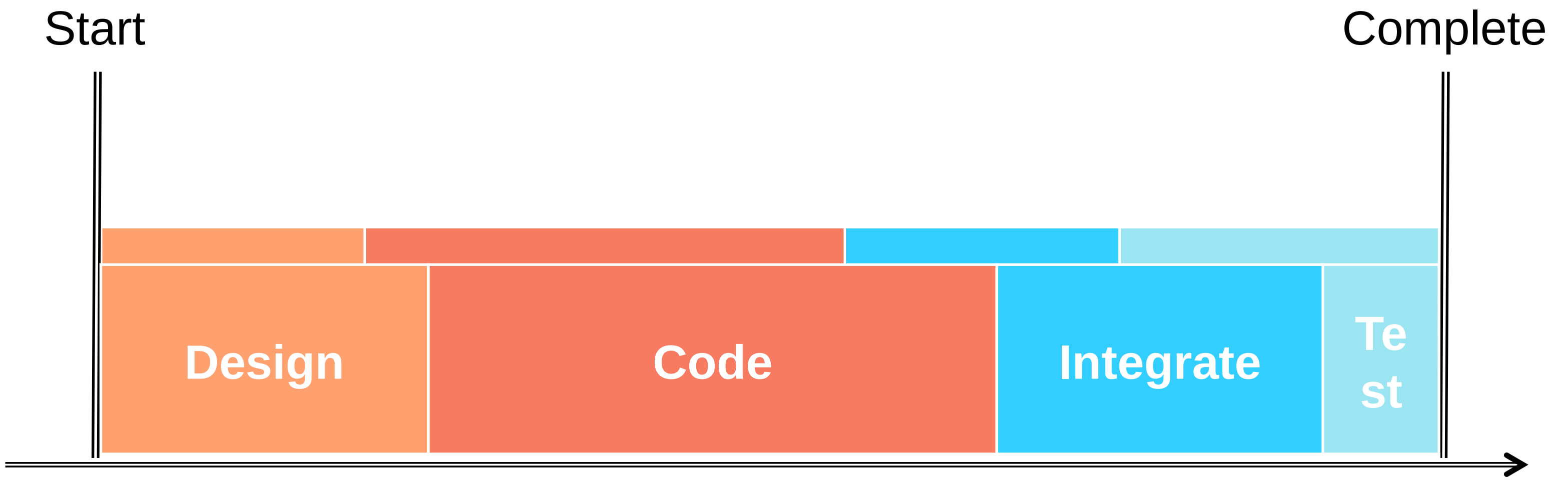
- Most use User Stories and Story Points.

Risk Management.

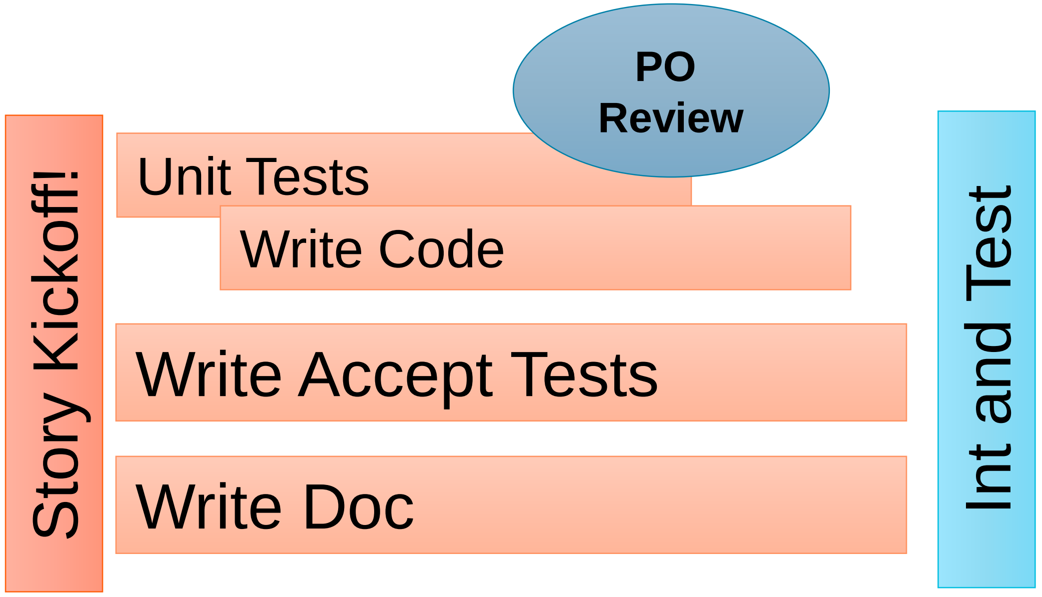
Engineering Practices. This was done on purpose. Scrum only defines a management framework. Engineering practices are domain dependant. In software development, XP practices are assumed:

- Test Driven Development (TDD)
- Paired Programming (instant code reviews)
- Refactoring
- Continuous Integration/Continuous Automated Testing

Typical Execution



Test Drive Development and Swarming: One-Piece Continuous Flow



Story Kickoff!

PO Review

Unit Tests

Write Code

Write Accept Tests

Write Doc

Int and Test

Done!
Next…

QUESTIONS?

METRICS

Beware of Goodhart's Law

Goodhart's law is an adage named after economist Charles Goodhart, which has been phrased by Marilyn Strathern as:

"When a **measure becomes** the **target**, it ceases to be a **good measure**."

- From 2002 to 2016, employees used fraud to meet impossible sales goals.
- Metric: “Eight is Great!”
- 3½ million fraudulent account opened causing harm to customers (fees).
- Regulators began looking at other lines of business.
- Total fine/penalties to date: over $4+ billion for fraud; $21+ billion overall since 2002

Metrics – What are you Optimizing For?

Principle: Use appropriate metrics that are valuable to the observer to inform them of a decision they **will** take

Scrum Team: Velocity, Say/Do Ratio, Cycle Time

Stakeholders:

- Business Value Delivered/Realized
- Customer Satisfaction
- Team Satisfaction
- Quality
- Lead Time

Choose more than one metric!

They are usually opposing forces… key is to find the balance.

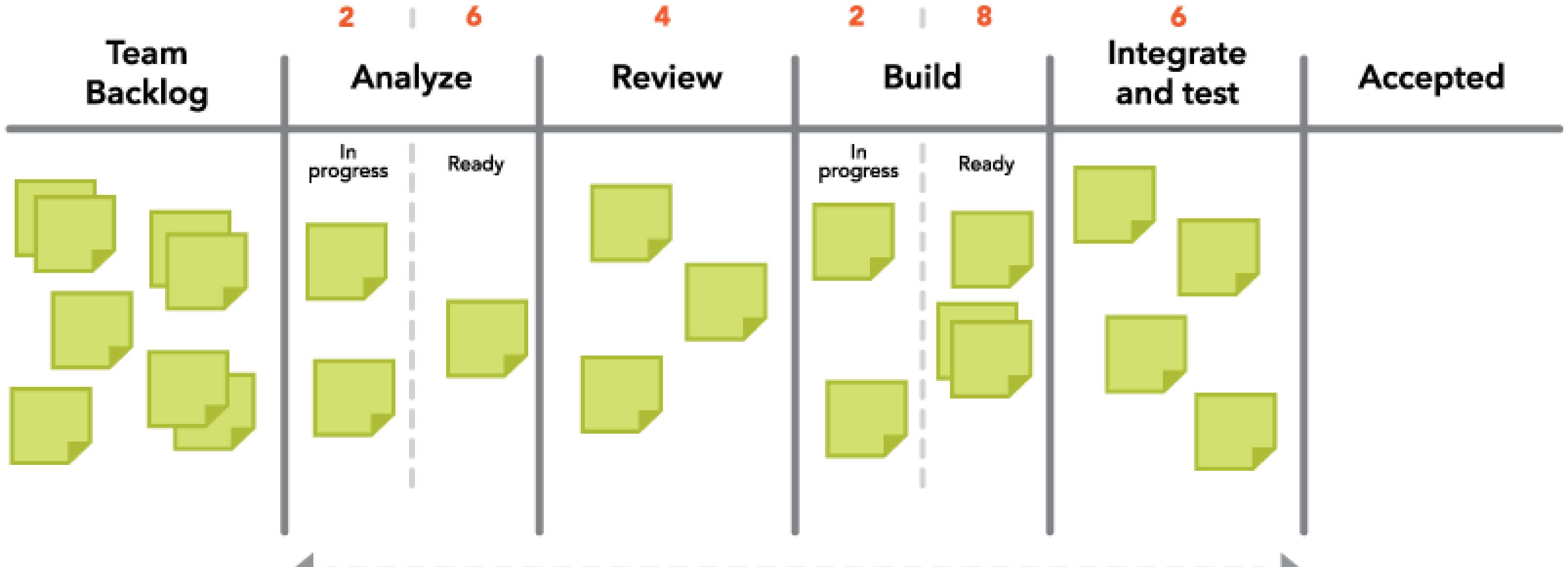
Further Reading: Evidence Based Management, Scrum.org

SCRUM AND KANBAN

SCALING

What is Kanban?

- A set of principles and practices
- Developed by David J Anderson; published in 2010
- Emphasis Focus and Flow (get to DONE ASAP)
- Focus: work an item until complete or blocked
- Flow: achieve end-to-end one piece continuous flow
- Continuous improvement experiments

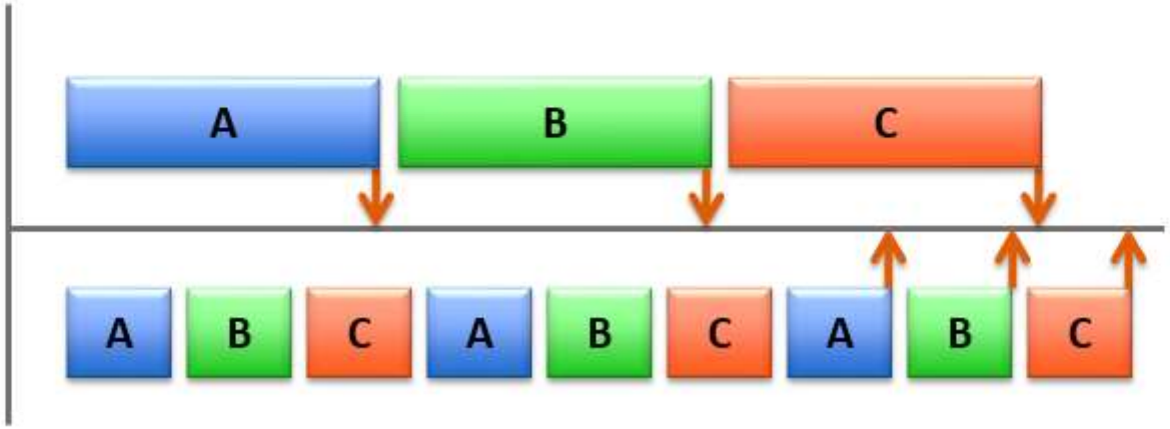


Scrum and Kanban

• Limits work in progress by what can be accomplished in x-weeks.	• Limits work in progress at each state on the Kanban board.
• Has only 3 roles.	• Officially, no defined roles. However, 4 roles have become common: • Customer • Request Mgr (e.g. PO) • Delivery Mgr (e.g. Flow Master) • Team Member
• Daily Scrum (walk Top to Bottom)	• Daily Stand-up (walk Right to Left)

Want to go fast? FOCUS!

Sequential work yields results sooner.



Not multi-tasking but task-switching.

It takes more time to switch tasks than stick with them until you finish.

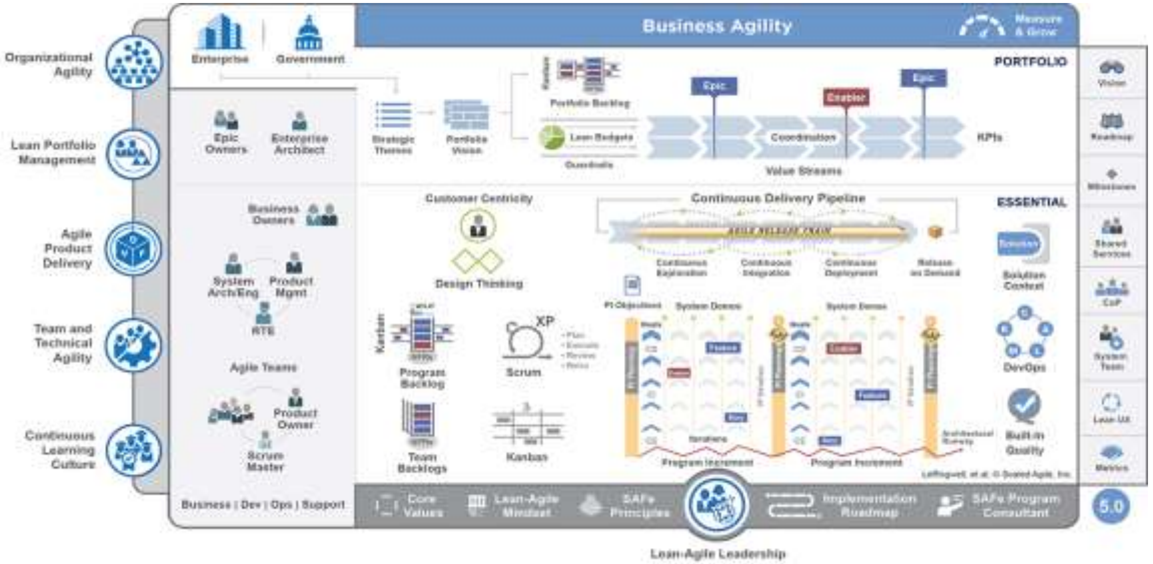
Stop Starting and Start Finishing

Source: NetObjectives

Scaling

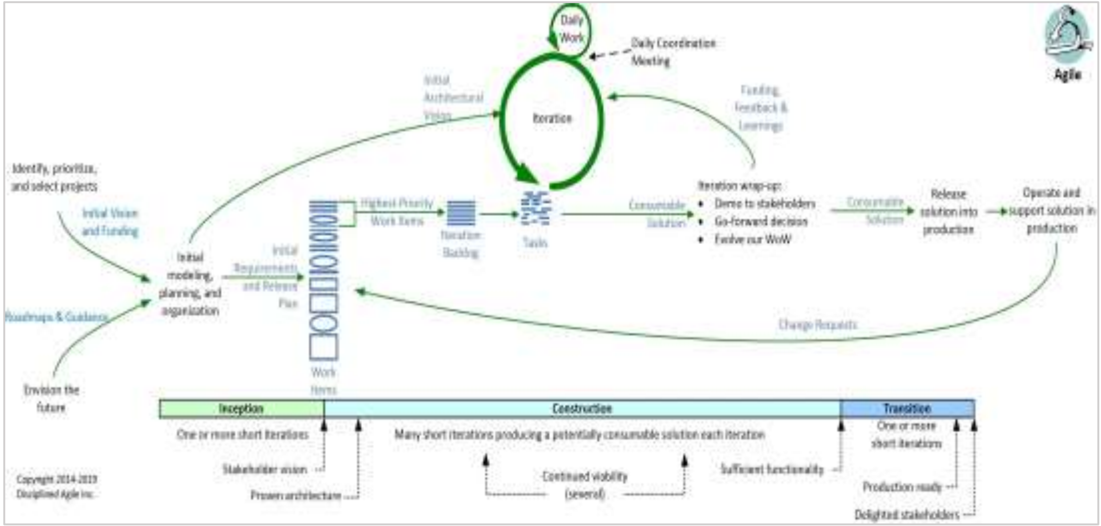
How do you get multiple teams working together on a single product?

Scaled Agile Framework V5



https://www.scaledagileframework.com/

Disciplined Agile



Daily Work
Daily Coordination Meeting
Agile
Initial Architectural Vision
Iteration
Funding, Feedback & Learnings
Identify, prioritize, and select projects
Initial Vision and Funding
Initial modeling, planning, and organization
Initial Requirements and Release Plan
Highest Priority Work Items
Iteration Backlog
Tasks
Consumable Solution
Iteration wrap-up:
• Demo to stakeholders
• Go-forward decision
• Evolve our WoW
Consumable Solution
Release solution into production
Operate and support solution in production
Roadmaps & Guidance
Change Requests
Work Items
Envision the future
Inception
Construction
Transition
One or more short iterations
Many short iterations producing a potentially consumable solution each iteration
One or more short iterations
Stakeholder vision
Proven architecture
Continued viability (several)
Sufficient functionality
Production ready
Delighted stakeholders
Copyright 2014-2019 Disciplined Agile Inc.

https://disciplinedagiledelivery.com/

Others…

LeSS (Large-Scale Scrum)	https://less.works/
Nexus	https://www.scrum.org/resources/scaling-scrum
Scrum@Scale	https://www.scrumatscale.com/

Why Use a Scaling Framework?

- Because it solves a problem.

- Fix before you dive in:
  - Fix your test suite before automating it
  - Fix your team’s execution before scaling

- Tools (e.g. JIRA) won’t make you Agile,
  living the Values and Principles will.

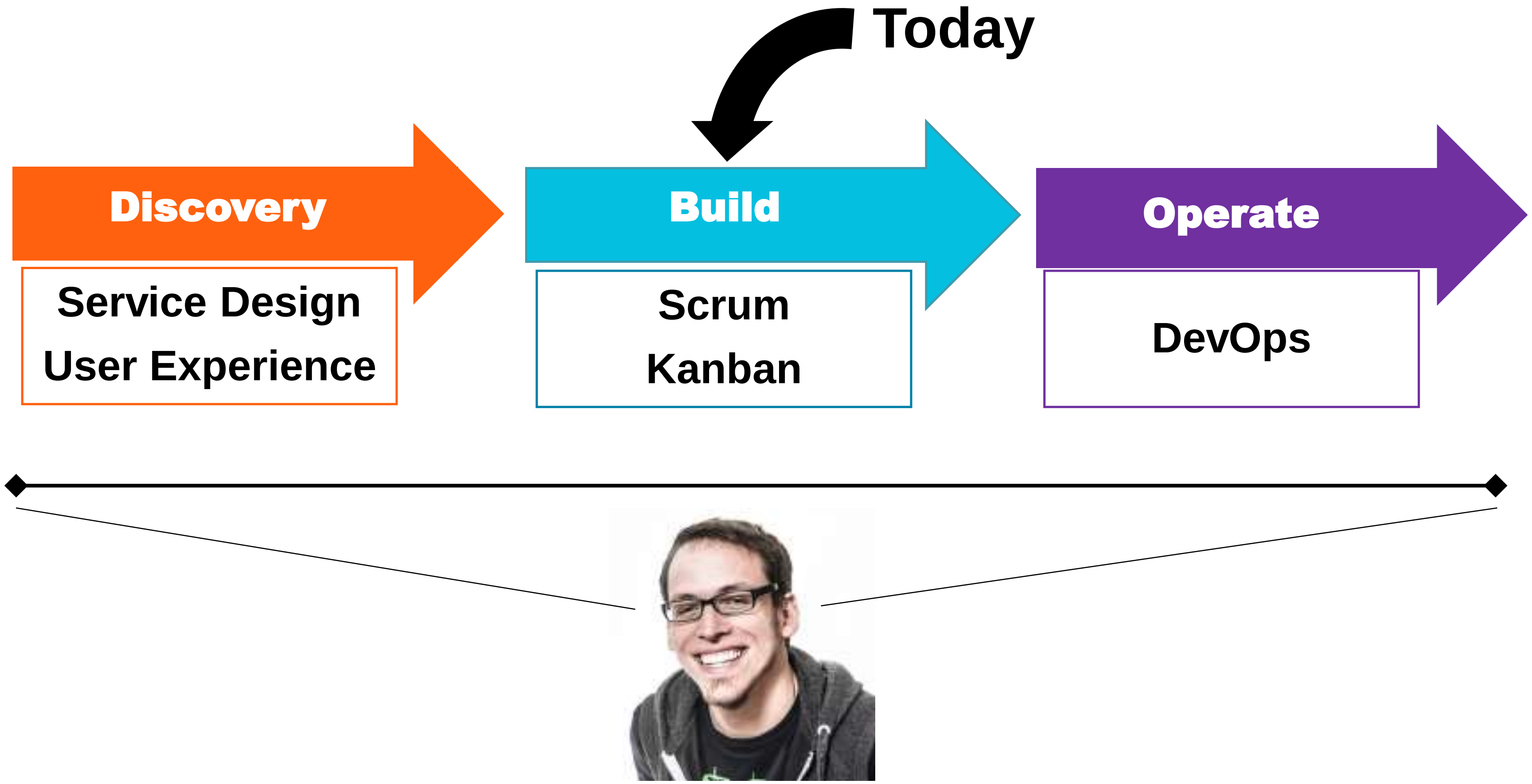
WRAP UP

Desired Outcomes

- Delight the customers
- Delivering the highest value items first
- Frequent, incremental deliveries
- Seeking and responding to feedback
- Outstanding quality from the customer’s point of view
- Engineering excellence par excellence
- Continuous improvement
  (pushing the status quo; “no problem” is the problem)
- Team joy (purpose, empowerment, sustainable pace, growth)

Agile, Lean, Scrum, Kanban, etc.
These are NOT the End.

It is a Means to an End!



Today

Discovery

Service Design
User Experience

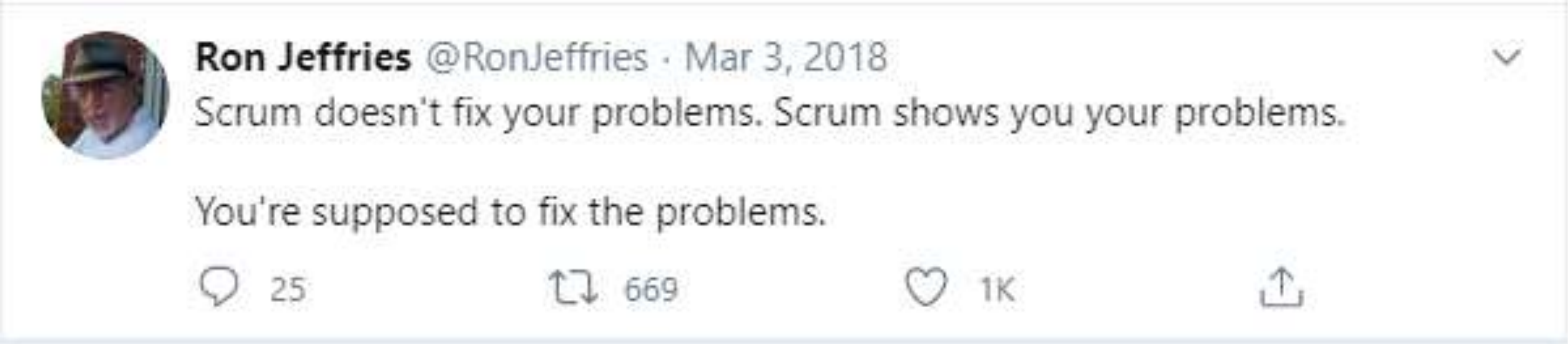
Build

Scrum
Kanban

Operate

DevOps

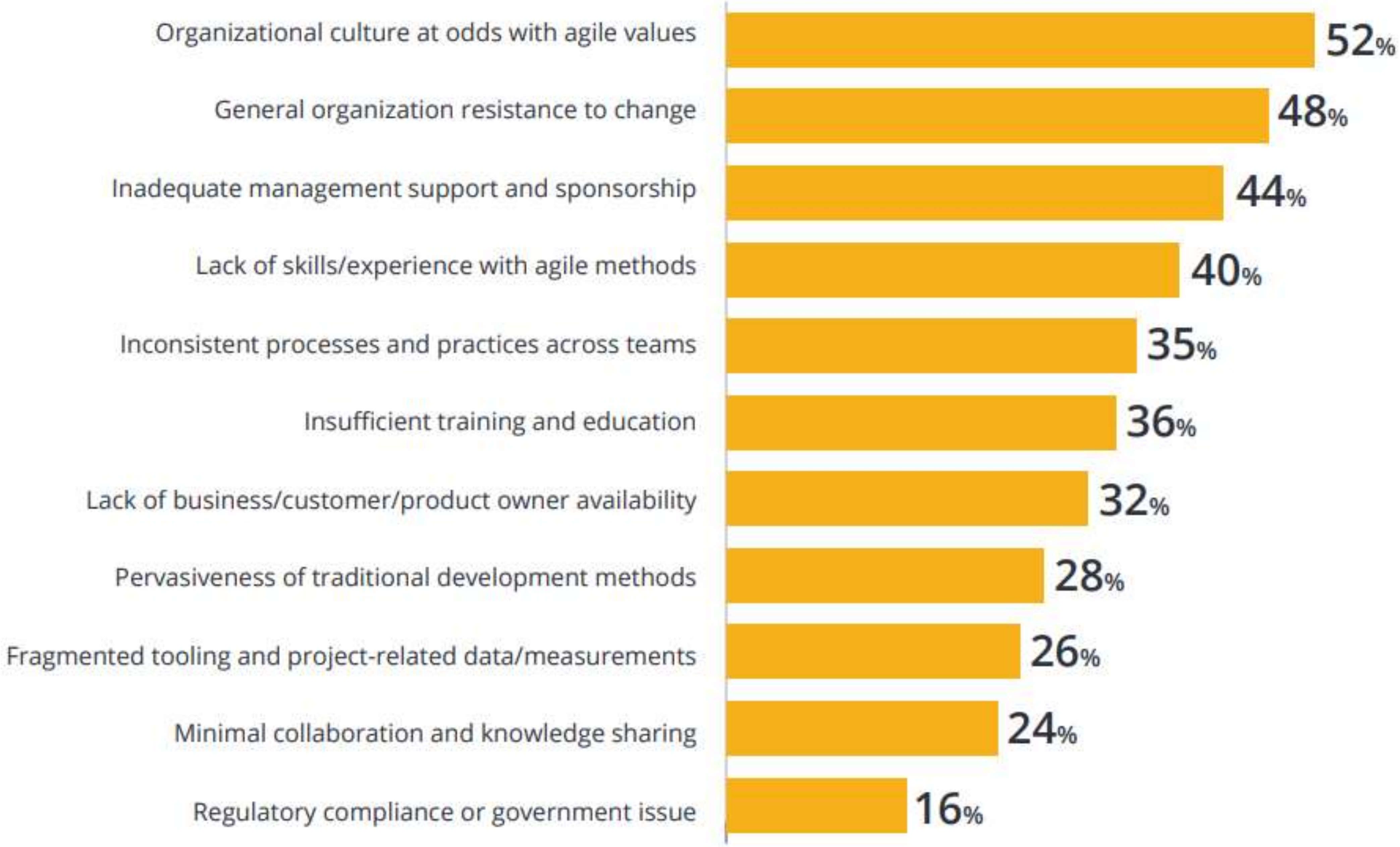
Need identification
Stakeholder Map
Personas
Value Proposition Canvas
Customer Journey Map
Service Blueprint
Vision/Mission Statement
Story Mapping
Prioritization
Road Mapping
Story Writing
Story Splitting
Backlog Refinement
Release Planning
Sprint Execution
Sprint Review with customers!
Customer Satisfaction
Organizational Change Mgmt
Operations
Audit and Complaince
…



Ron Jeffries @RonJeffries · Mar 3, 2018

Scrum doesn't fix your problems. Scrum shows you your problems.

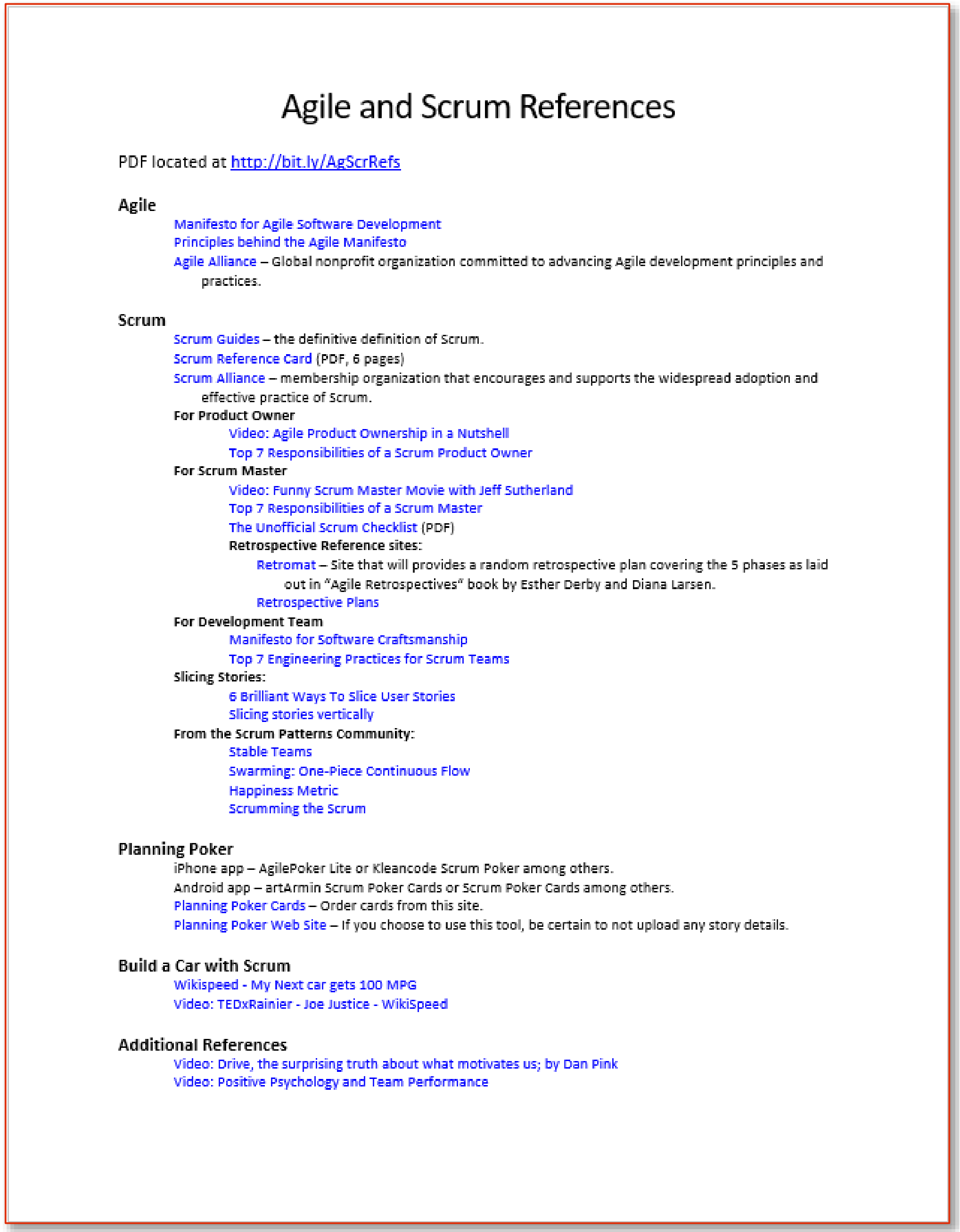
You're supposed to fix the problems.


Organizational culture at odds with agile values
52%
General organization resistance to change
48%
Inadequate management support and sponsorship
44%
Lack of skills/experience with agile methods
40%
Inconsistent processes and practices across teams
35%
Insufficient training and education
36%
Lack of business/customer/product owner availability
32%
Pervasiveness of traditional development methods
28%
Fragmented tooling and project-related data/measurements
26%
Minimal collaboration and knowledge sharing
24%
Regulatory compliance or government issue
16%

References

http://bit.ly/AgScrRefs

Download the PDF to utilize the embedded links.



Agile and Scrum References

PDF located at http://bit.ly/AgScrRefs

Agile

- Manifesto for Agile Software Development
- Principles behind the Agile Manifesto
- Agile Alliance – Global nonprofit organization committed to advancing Agile development principles and practices.

Scrum

- Scrum Guides – the definitive definition of Scrum.
- Scrum Reference Card (PDF, 6 pages)
- Scrum Alliance – membership organization that encourages and supports the widespread adoption and effective practice of Scrum.
- **For Product Owner**
  - Video: Agile Product Ownership in a Nutshell
  - Top 7 Responsibilities of a Scrum Product Owner
- **For Scrum Master**
  - Video: Funny Scrum Master Movie with Jeff Sutherland
  - Top 7 Responsibilities of a Scrum Master
  - The Unofficial Scrum Checklist (PDF)
  - **Retrospective Reference sites:**
    - Retromat – Site that will provides a random retrospective plan covering the 5 phases as laid out in "Agile Retrospectives" book by Esther Derby and Diana Larsen.
    - Retrospective Plans
- **For Development Team**
  - Manifesto for Software Craftsmanship
  - Top 7 Engineering Practices for Scrum Teams
- **Slicing Stories:**
  - 6 Brilliant Ways To Slice User Stories
  - Slicing stories vertically
- **From the Scrum Patterns Community:**
  - Stable Teams
  - Swarming: One-Piece Continuous Flow
  - Happiness Metric
  - Scrumming the Scrum

Planning Poker

- iPhone app – AgilePoker Lite or Kleancode Scrum Poker among others.
- Android app – artArmin Scrum Poker Cards or Scrum Poker Cards among others.
- Planning Poker Cards – Order cards from this site.
- Planning Poker Web Site – If you choose to use this tool, be certain to not upload any story details.

Build a Car with Scrum

- Wikispeed - My Next car gets 100 MPG
- Video: TEDxRainier - Joe Justice - WikiSpeed

Additional References

- Video: Drive, the surprising truth about what motivates us; by Dan Pink
- Video: Positive Psychology and Team Performance

Future Learnings

Sponsored by Project Management Institute:

- PMI-ACP – (Agile Certified Practitioner)
- Disciplined Agile (CDA, DALSM, CDAP, …)

Sponsored by Scrum Alliance:

- Certified Scrum Master (CSM)
- Certified Scrum Product Owner (CSPO)
- CSP, CSD, CTC, CEC, CST, CAL

Sponsored by Scrum.org:

- Professional Scrum Master (PSM1, PSM2, PSM3)
- Professional Scrum Product Owner (PSPO1, PSPO2)
- More… PSD, SPS

Sponsored by Scaled Agile:

- Scaled Scrum Master (SSM); Scaled Advanced Scrum Master (SASM)
- Scaled Scrum Product Manager Product Owner (SPMPO)
- More… SA, SPC, SP, SPCT,

www.meetup.com

Community meetings on just about any subject.

www.meetup.com/find

Small fee or ***free***, most provide pizza and drinks.

Groups include:

- DFW Scrum (two locations)
- Dallas Agile Leadership Network
- North Dallas Agile Product Owners Meetup
- ProductTank DFW
- Dallas User Experience Group
- …and so many more

Final Q&A

Please complete the survey!