From Agile Teams to Agile Organizations

agile tour 2016

Agile Québec lightspeed

Steve Mercier, Director of Software Engineering Practices

Who am I ?

Steve Mercier

- 20 years+ of software development experience, 10 years+ of using Agile methodologies daily, 5 years+ of using DevOps philosophy daily
- Specialized in
  - Best practices: Continuous Integration, Continuous Delivery/Deployment, Software Production Lines, Infrastructure As Code, Continuous Improvement, Lean engineering
- Currently Director of Software Engineering Practices at Lightspeed, responsible of DevOps, Test Automation, QA, Security and Documentation practices

The (ongoing/chaotic) journey from Agile Teams to Agile Organization

The Agile Organization Journey

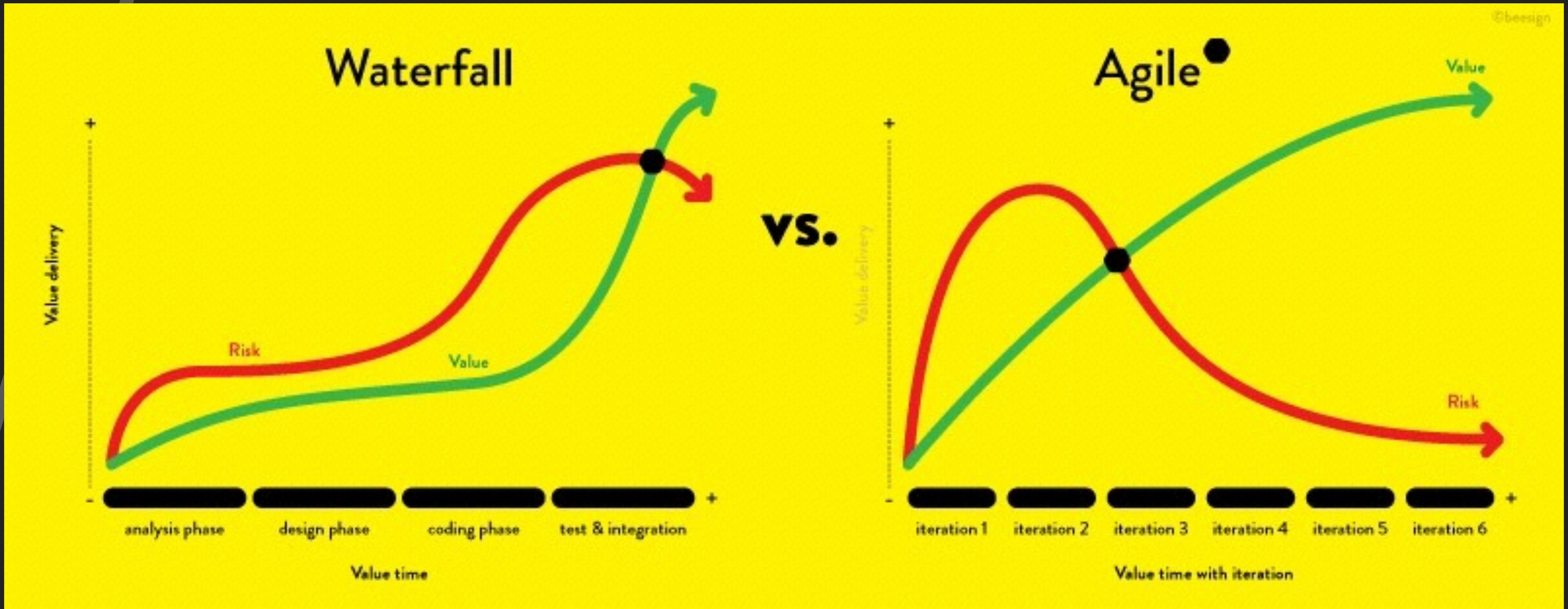
- Promises
- Challenges
- Questions
- (Tentative) Answers
- Conclusion
- Q&A

The Agile Promises

The Agile Promises

- Faster time to market
- Development costs reduction
- Quality improvement
- Business value driven, aligned with customers needs
- Better team work, better focus
- Technical debt reduction
- No useless architecture and documents
- Only good code adding business value!

The Agile Promises - graphically



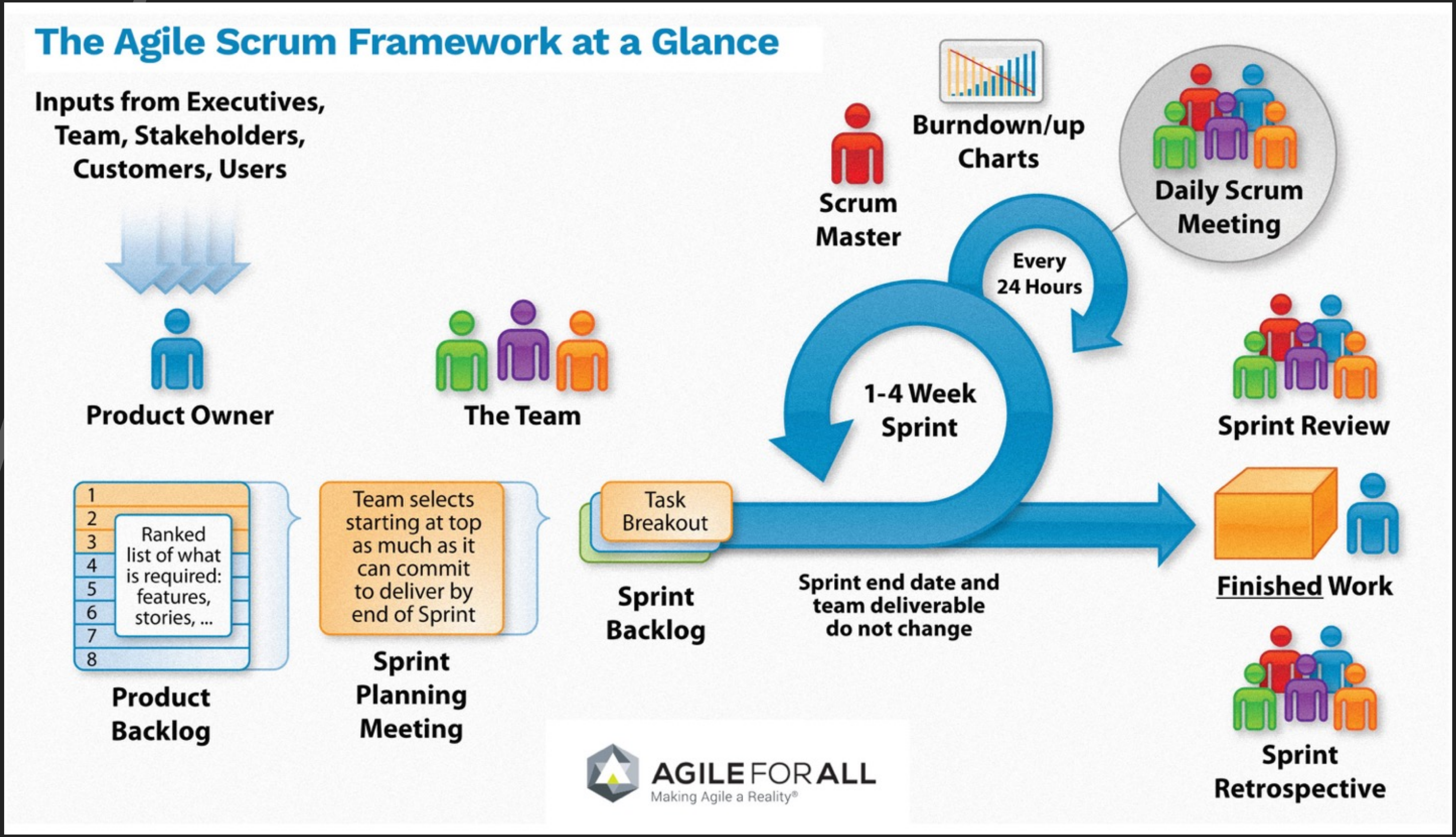
Who would not want that? Maybe a little simplistic…

The Agile Promises - The journey begins

WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
scottadams@aol.com
www.dilbert.com
11-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.

We send the first team(s) to training

Agile SCRUM at its core is quite simple



The Agile Scrum Framework at a Glance

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Scrum Master

Burndown/up Charts

Daily Scrum Meeting

Every 24 Hours

1-4 Week Sprint

Sprint Review

1
2
3
4
5
6
7
8

Ranked list of what is required: features, stories, ...

Product Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout

Sprint Backlog

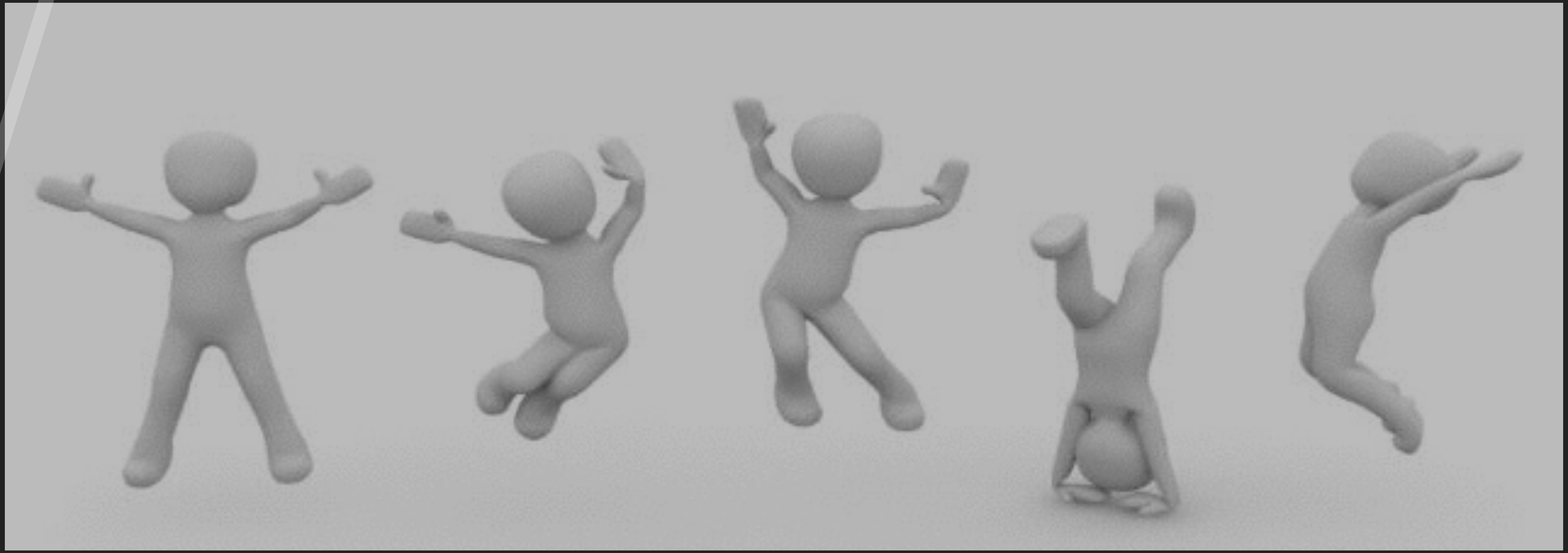
Sprint end date and team deliverable do not change

Finished Work

Sprint Retrospective

AGILE FOR ALL
Making Agile a Reality®

The Agile Promises



This first team comes back, full of good intention

the team starts using Agile, and it works!

The Agile Promises

So it seems Agile works, right?

Question: does it work for you?

The Agile Promises

By experience, Agile typically works well if:

You are working on new software, with small teams and a limited number of teams

Agile tends to work less if:

You are trying to scale Agile to multiple teams on larger projects

The Agile Challenges

Challenge #1 - Scaling to multiple (independent) teams

Based on early successes, other teams are asked to try it

With possibly less training, less passion, less mentoring

possibly even resisting the transformation

Challenge #2 - Scaling it to dispersed (independent) teams

Not colocated teams, across time zones

Teams have different cultures, values

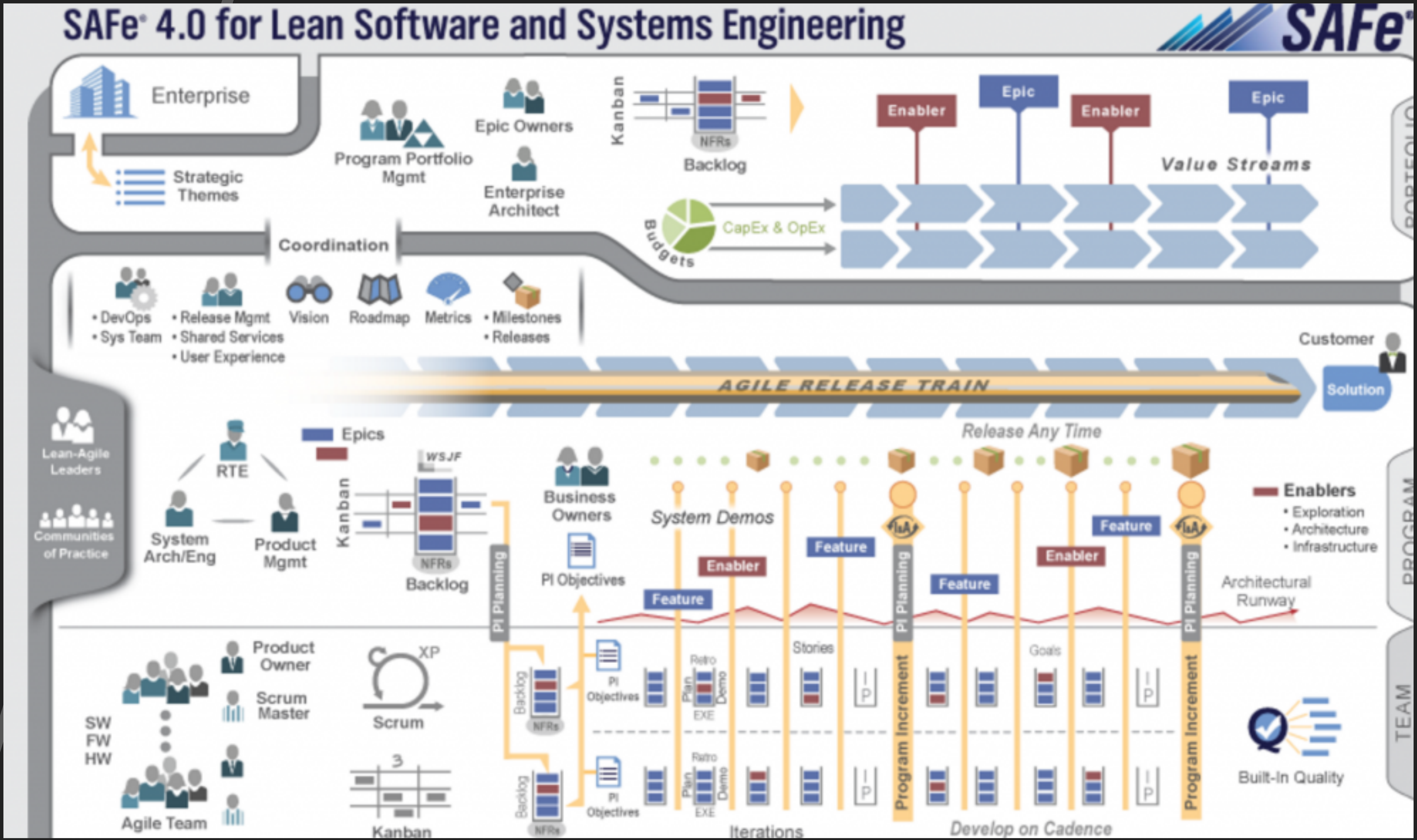
Teams do not all see Agile in the same way

Challenges #3 - Scaling it to dispersed dependent teams

Individual teams, OK, dispersed independent teams, also OK

But if the business requires different teams to deliver a common product across continents… more challenging!

Scaling Agile to multiple teams is complex…


SAFe® 4.0 for Lean Software and Systems Engineering
SAFe®
Enterprise
Strategic Themes
Program Portfolio Mgmt
Epic Owners
Enterprise Architect
Kanban
NFRs
Backlog
Budgets
CapEx & OpEx
Enabler
Epic
Enabler
Epic
Value Streams
PORTFOLIO
Coordination
• DevOps
• Sys Team
• Release Mgmt
• Shared Services
• User Experience
Vision
Roadmap
Metrics
• Milestones
• Releases
Customer
Solution
AGILE RELEASE TRAIN
Lean-Agile Leaders
Communities of Practice
RTE
System Arch/Eng
Product Mgmt
Epics
WSJF
Kanban
NFRs
Backlog
Business Owners
PI Objectives
Release Any Time
System Demos
Feature
Enabler
Feature
Feature
Enabler
Feature
PI Planning
PI Planning
Enablers
• Exploration
• Architecture
• Infrastructure
Architectural Runway
PROGRAM
Product Owner
Scrum Master
XP
Scrum
SW
FW
HW
Agile Team
3
Kanban
Backlog
NFRs
PI Objectives
Retro
Plan
Demo
EXE
Stories
Goals
IP
Program Increment
Program Increment
Iterations
Develop on Cadence
Built-In Quality
TEAM

Challenge #4 - Wrong team composition

QA, Ops not part of the Agile teams

PO/PM not part of the teams or not available

Challenge #5 - Not having an end to end Agile process

Having handoffs between the Agile teams and Ops for example

Definition of Done not including Shipping It

Challenge #6 - Too much manual process

Red tape / Various Authorizations

Agile is about empowering teams

Challenge #7 - Old school management

“New” Agile management should focus on:

Creation of a “safe” environment for trying things, enforcing the fail fast / fail differently model

Rewarding the right behaviours

Fostering a learning organization culture

The Questions

Is SCRUM enough to obtain Agile organizations? No. Does it help? Sure!

Not a Silver Bullet

Scrum is simple

...implementing Scrum is not

Is Scrum of Scrums a solution?

- How could we frame the common work across multiple teams?
- How to structure the whole software development effort of many teams?
- Scrum of Scrums can help; sufficient?

What could be this structuring frame?

The (Tentative) Answers

Hint: Ever heard of a Software Delivery Pipeline?

Step #1 - Leverage Software Engineering Practices

- Use Software Engineering Best Practices as a frame to constrain how software is developed and connected together
- Helps mostly with structuring the How
- Communities of Practices can be helpful

Step #2 - Develop/Use a Delivery Pipeline System

Engrain those defined practices into a

single Software Delivery Pipeline system

Step #3 - Feed your system with the real customers needs

Ensure you feed your delivery pipeline with the right things -

do the right thing for your customers

The best pipeline system in the world will not help your agility if

you do the wrong thing with it!

Step #4 - Apply Continuous Improvement to your pipeline

Use Lean / Plan-Do-Check-Act principles and

Continuously reflect on the system to optimize it to your business

Why a Delivery Pipeline system?

“Average leaders have quotes.

Good leaders have a plan.

Exceptional leaders have a system.”

- Urban Meyer

Your Automated Delivery Pipeline is your system

But what should be in a typical pipeline?

What is the scope of such a system?

Delivery Pipeline Elements

- Starts with a feature file -like input (i.e. a clear customer need)
- Code Commit (everything should be under SCM)
- CI - Continuous Build / Unit tests / Continuous Testing / System tests
- Continuous Delivery / Deployment
- Continuous Monitoring of all systems

How to measure progress - The (true) Agility KPIs

- Total Lead time for any improvement
- Number of deployments per day
- Number of incidents in production
- Impact of the incidents, duration
- The time to onboard a new developer

Agility KPIs - top DevOps performers

	Before	After
Lead time	Months	Days / Minutes
# of deployments	Quarterly	Multiple Daily
# of incidents	Multiple per deploy	Almost none
Incidents impact	Days of downtime	0 downtime
On-boarding time	Months	Days

How to get there?

Use SCRUM and Agile principles, values, processes, yes. But also:

- Put in place the feedback loops, Continuous Improvements, Lean processes in place
- Apply the Plan-Do-Check-Act approach on small process improvements
- Find your waste, using Value Stream Mapping analysis, reduce your batch size

Use your Pipeline to make the issues visible

A global Continuous Delivery pipeline for all the company’s software would help highlighting the issues, challenges, areas requiring improvements

Reduce cycle time by enforcing Automation

- Continuous Integration with automatic tests at unit, system and system of systems levels
- Continuous Delivery or Deployment using Infrastructure As Code

Keep the focus on the global system, not on small teams

Company Continuous Delivery pipeline help keep the focus on the company delivered business value to external customers, reducing the natural silos barriers impacts.

What more

- Teams’ composition is key - all the required roles must be fulfilled within the teams
- Complement Agile and Scrum with other compatible approaches such as LEAN and DevOps to optimize global organization and not just a small team work
- Ensure an environment permitting trials and failures is in place; create a safe environment for contributions; learn from failures, i.e. Fail fast and fail differently each time

Conclusion

Conclusion

Having an Agile organization is a journey that can certainly start with Scrum, but cannot really stop until all the software you produce and operate is continuously delivered to your end customers

The key here is to deliver faster, faster than your competition, to disrupt yourself before your competitors do disrupt your business

The Agile philosophy, values and tools are only a partial answer

The DevOps/Lean philosophy, values and tools are only a partial answer

Ask yourselves what prevents you from delivering value faster?

Conclusion

Break Silos, Work end-to-end, in small batches of work

Empower your teams, Evolve your management style

Remove all your red-tape and manual processes, one by one

Measure your true Agility KPIs

Put in place a system delivering customer’s value!

And be cautious…

“There is nothing quite so useless as doing with great efficiency something that should not be done at all”

- Peter Drucker (the inventor of modern management)

Q&A

Questions and answers - What do you think about this?