



rego*U*niversity 2019

SAN DIEGO

# Agile Primer

Your Guides: Rob Greca and Mike Pokorny

# Introductions

- Take 5 Minutes
- Turn to a Person Near You
- Introduce Yourself
- Business Cards



# Agenda

---

- Introduction/overview
- Context: Waterfall vs Agile
- Agile Team Level
- SAFe
- Tools

# Some Context

*rego*University 2019  
SAN DIEGO



# Case Study: FBI's Sentinel project

- Oklahoma city bombing (1995) highlighted challenges of FBI's old technology and processes
- After 9/11, FBI couldn't even securely email photos of hijackers; needed to fax or overnight CD-ROMs
- 2001 Congress approved \$170M for "Virtual Case File"
  - SAIC was the contractor
  - 9/11 Commission: "The poor state of the FBI's information systems...", "no effective mechanism for capturing or sharing its institutional knowledge"
- As late as 2010 the FBI was still operating using 30 year old processes and methods
  - Filing paper reports
  - Print out copies to get approval
  - Index the reports by circling key words with a red pen
- 2004, project was cancelled after spending \$600M with no working software
- 2<sup>nd</sup> attempt was no better
  - Lockheed Martin was the contractor

# Case Study: FBI's Sentinel project

- 2009, FBI Director, Robert Mueller hired an IT executive from Lehman Brothers to help turn this project around
- At this point, 90% of the \$405M budget had been spent and not even 50% complete
- Auditors determined it would require an additional \$350M to complete
- Both previous government contractors (SAIC and Lockheed Martin) had used a traditional waterfall approach
- The internal FBI IT team switched to Scrum (150 resources to about 50)
- With 5% of the budget and in about 20 months the team successfully built and launched Sentinel

# What is it about agile that works?

*rego*University 2019  
SAN DIEGO

# Agile works because

- Dedicated teams (avoid team development stages)
- Self-directed teams
- Whole teams (avoid hand-offs)
- Limited to no multi-tasking
- Few meetings, more work
- Simplicity
- Little time spent on reporting (information radiators)
- Individuals appreciate this way of working and are therefore more motivated
- More motivated individuals are more effective and productive



# Big “A” vs small “a” agile

## Big “A” Agile

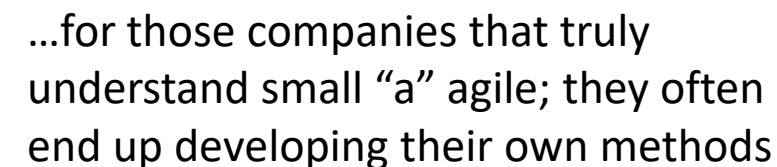
*(noun): relating to or denoting a method of project management, used especially for software development, that is characterised by the division of tasks into short phases of work and frequent reassessment and adaptation of plans*



## Small “a” agile

*(adjective): able to move quickly and easily; able to think, understand and respond quickly*





# Other disciplines

## Lean

- Focus on value (why are you doing this?, what's the value stream?)
- Goal is to deliver value in the shortest sustainable lead time
- Cost of delay
- Small batch sizes
- Eliminate waste and delays
- Release a MVP to get fast feedback with least amount of effort
- Embracing “kaizen” = continuous improvement

## Systems Thinking

- “A system must be managed. It will not manage itself.” - Deming
- Optimizing a component does not optimize the whole system (the opposite may even be true)
- Systems thinking acknowledges that two entities must be considered:
  - The system built for the benefit of the customer (i.e. the PPM tool, processes, governance)
  - The organization building the system (i.e. the PMO) – *the organization building the system is itself a system and therefore see the first bullet*

# Compare Waterfall and Agile



# The assumptions we make

	<b>WATERFALL</b>	<b>AGILE</b>
<b>REQUIREMENTS</b>	With enough time, <b>can</b> figure out well-defined requirements	Customers <b>cannot</b> fully understand or describe requirements up front
<b>CHANGES</b>	Changes will be small, and therefore manageable	Changes will be constant, so deliver in smaller increments
<b>SYSTEM INTEGRATION</b>	Will go smoothy	Integrate from beginning and integrate constantly
<b>INNOVATION</b>	<b>Can</b> be done on a predictable schedule	<b>Cannot</b> be done on a predictable schedule, but can deliver most valuable features earlier to get fast feedback
<b>PROCESSES AND TOOLS</b>	Processes and tools will capture data and metrics providing key information to decision-makers to avoid challenges	Focus more on individuals and interaction rather than processes and tools; can usually do this with post-it notes
<b>AUTHORITY</b>	Command and control structures will ensure everyone stays on track	The teams themselves are self-directed and determine the best way to work
<b>MEASURE OF SUCCESS</b>	Adherence to the original plan is the primary measurement	Working, tested software is the primary measurement



# The approaches we take for Resources

	<b>WATERFALL</b>	<b>AGILE</b>
<b>TEAM STRUCTURE</b>	Non-dedicated teams (Have to go through Forming, Storming, Norming, Performing, every single time)	Dedicated teams (Once Performing, always Performing)
<b>PUSH vs PULL</b>	Push approved projects right away to individual resources	Dedicated teams pull the next high priority item
<b>SCOPE</b>	“Get it All Done”	Break down to smaller deliverables and finish highest value ones rapidly (“Stop Starting and Start Finishing”)
<b>ESTIMATING</b>	Assumes we estimate and can forecast accurately “Absolute estimating”	Understands that human are horrible at absolute estimating, but very good at relative estimating
<b>FLOW</b>	Starting projects is a measure of success—leading to many projects in-flight	Work in process (WIP) is limited based on throughput
<b>STAFFING</b>	Project managers have to beg/borrow/steal resources	Work comes to the teams that are stable/dedicated
<b>MEASURE OF SUCCESS</b>	Full resource utilization	Velocity and value delivery

# What are the fundamental elements

- How humans really work rather than how we say we work
- A motivated/happy team
  - Self-directed teams: Team decides how it can best be productive and effective (tools, methods)
  - The team typically stays together (Forming, Storming, Norming, Performing)
  - Servant-leadership
  - Multi-tasking eliminated: “Stop starting and start finishing”
- Continuous improvement baked-in (“Kaizen”)
- Definition of done is clear
- Collaboration and feedback throughout

# Questions

---

- Why have you chosen traditional waterfall at your organization? Was there a reason?
- How effective is it? Has this ever been measured?
- How often do you review your PPM processes (Plan-Do-Check-Act)? Do you even do that?
- How often are stakeholders involved in project reviews?

# Agile at the Team Level

*rego*University 2019  
SAN DIEGO

# Agile development

---

- Team of 3 to 9 people
- A whole team; all members are able to complete all activities to complete all work
- Self-directed
- Leaders are “servant leaders”
- Inspect and Adapt



# Agile Manifesto

---

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 process 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.

# Agile Principles

- **Customer satisfaction through early and continuous software delivery** – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
- **Accommodate changing requirements throughout the development process** – The ability to avoid delays when a requirement or feature request changes.
- **Frequent delivery of working software** – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
- **Collaboration between the business stakeholders and developers throughout the project** – Better decisions are made when the business and technical team are aligned.
- **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.
- **Enable face-to-face interactions** – Communication is more successful when development teams are co-located.
- **Working software is the primary measure of progress** – Delivering functional software to the customer is the ultimate factor that measures progress.
- **Agile processes to support a consistent development pace** – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
- **Attention to technical detail and design enhances agility** – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
- **Simplicity** – Develop just enough to get the job done for right now.
- **Self-organizing teams encourage great architectures, requirements, and designs** – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
- **Regular reflections on how to become more effective** – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

# Scrum Concepts

- Roles

- Product owner
- ScrumMaster
- Development team

- Activities

- Sprint Planning
- Daily scrum
- Sprint execution
- Sprint review
- Sprint retrospective
- Product backlog grooming

- Artifacts

- Product backlog
- Sprint backlog
- Stories
- Potentially shippable product increment

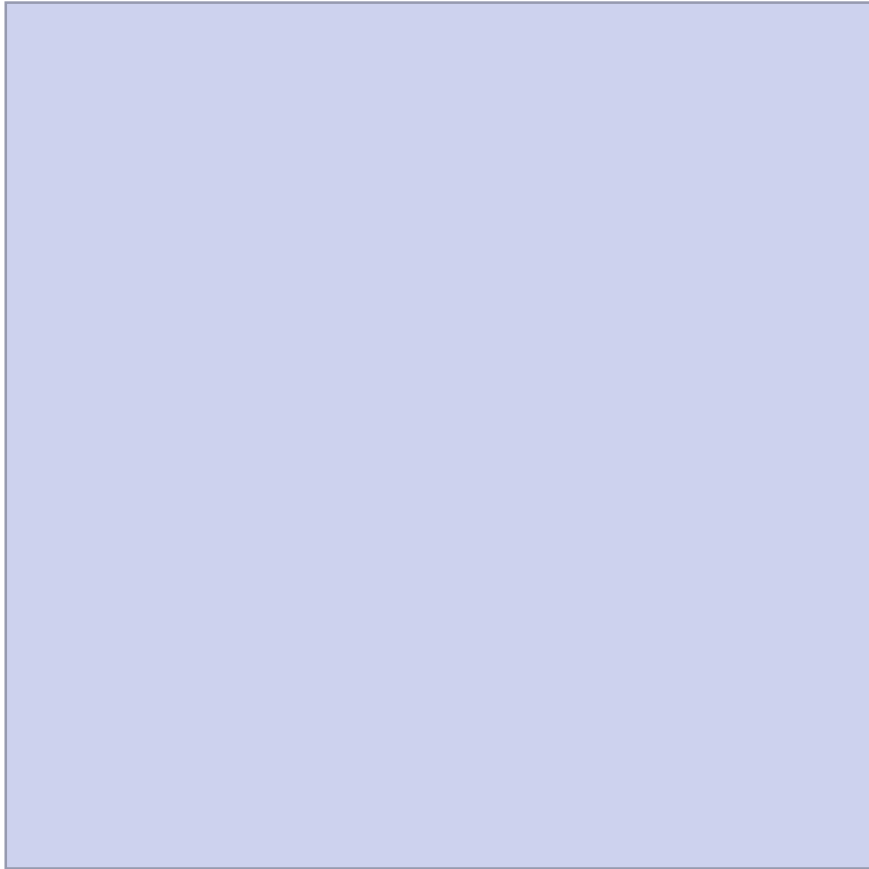
- Rules

# Story points vs hours

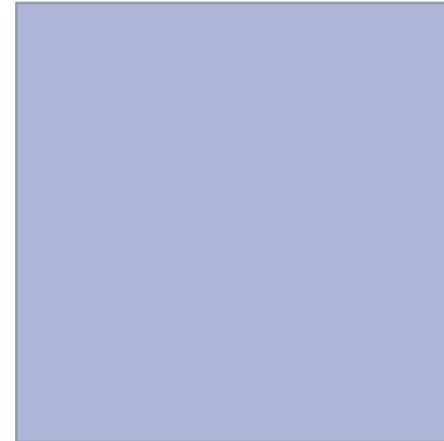
- Traditional estimating gives us “false precision”
  - Clients will try to hold us to the number of hours we estimate
- Estimating is really “guessing”
- We need to acknowledge the complexities and uncertainties of software development (ask the client how well they estimate—so why do they keep doing it)
- In the first iteration, estimates could vary widely
- Avoid giving “false precision”
- Use Fibonacci sequence: 1, 2, 3, 5, 8, 13, 21, etc.
  - This helps avoid false precision
  - Highlights large differences
- Determine what a “1” story point will be and then compare everything else against that using the Fibonacci sequence

# Hours vs Points exercise

- Estimate absolutely; what is the area of each square



$X \text{ in}^2$

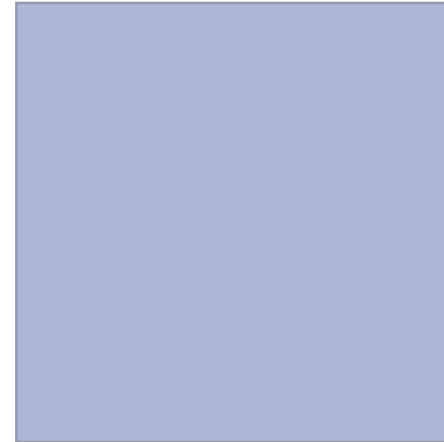
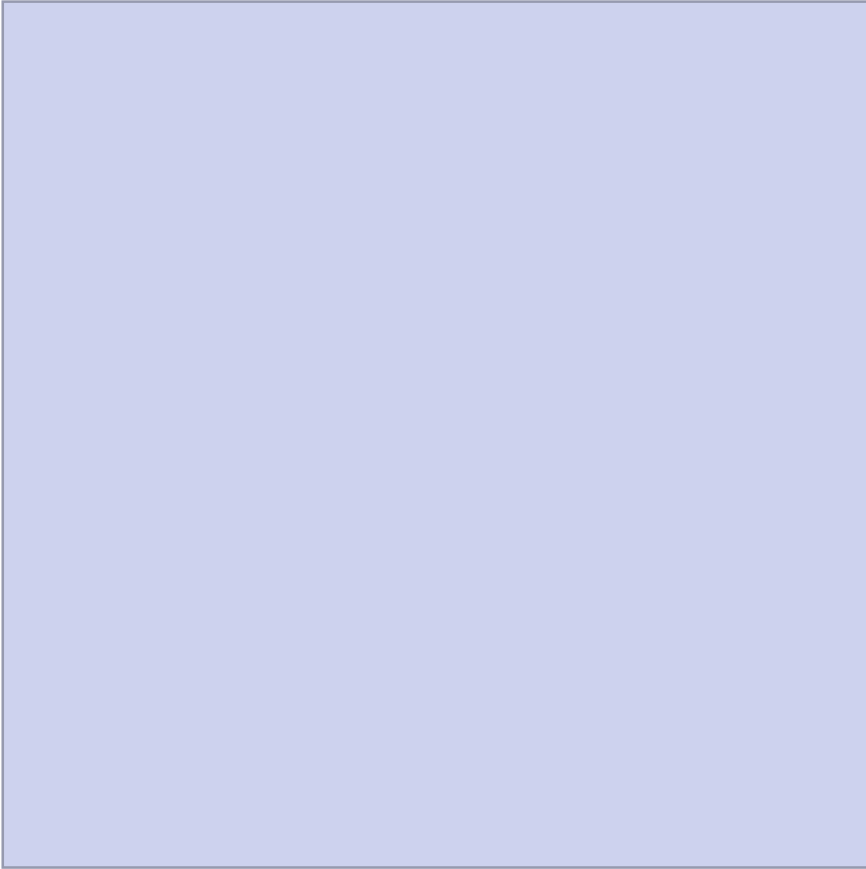


$Y \text{ in}^2$

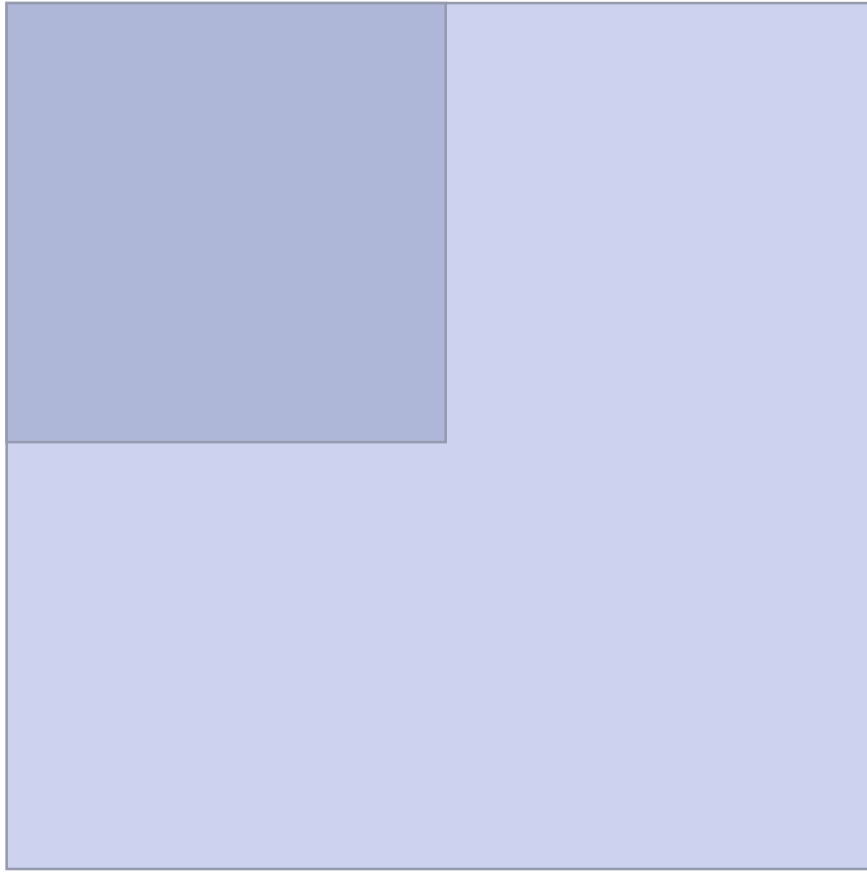


# Hours vs Points exercise

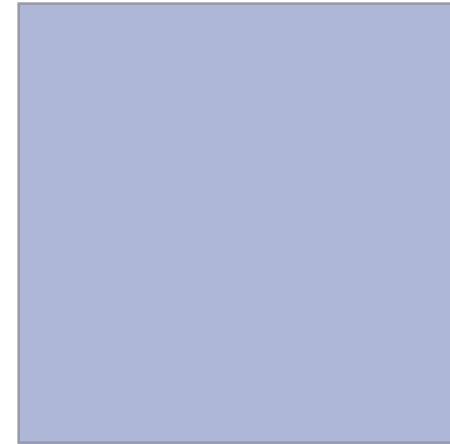
- Estimate relatively; how much bigger is the larger square



# Hours vs Points exercise

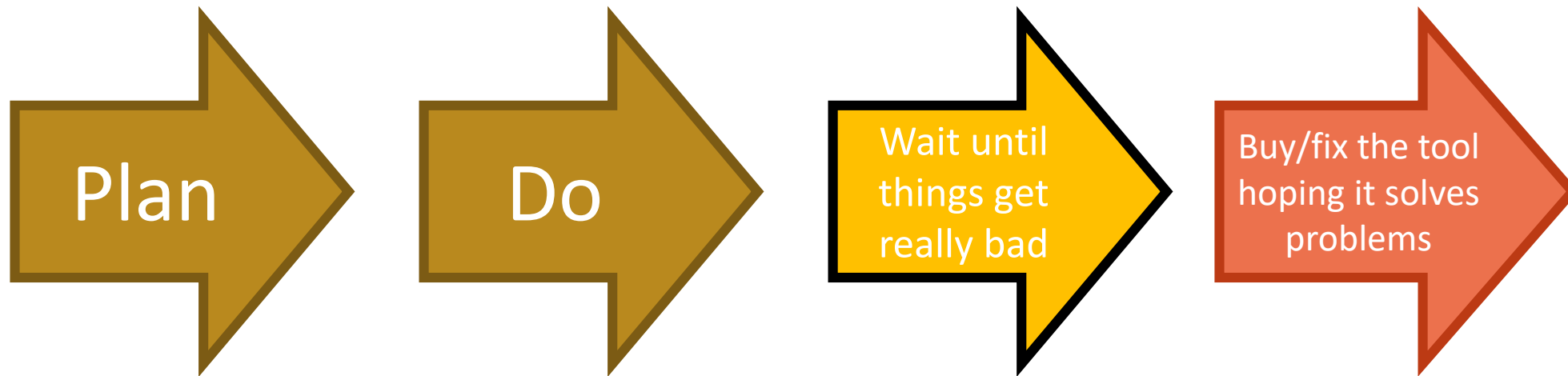


x 4



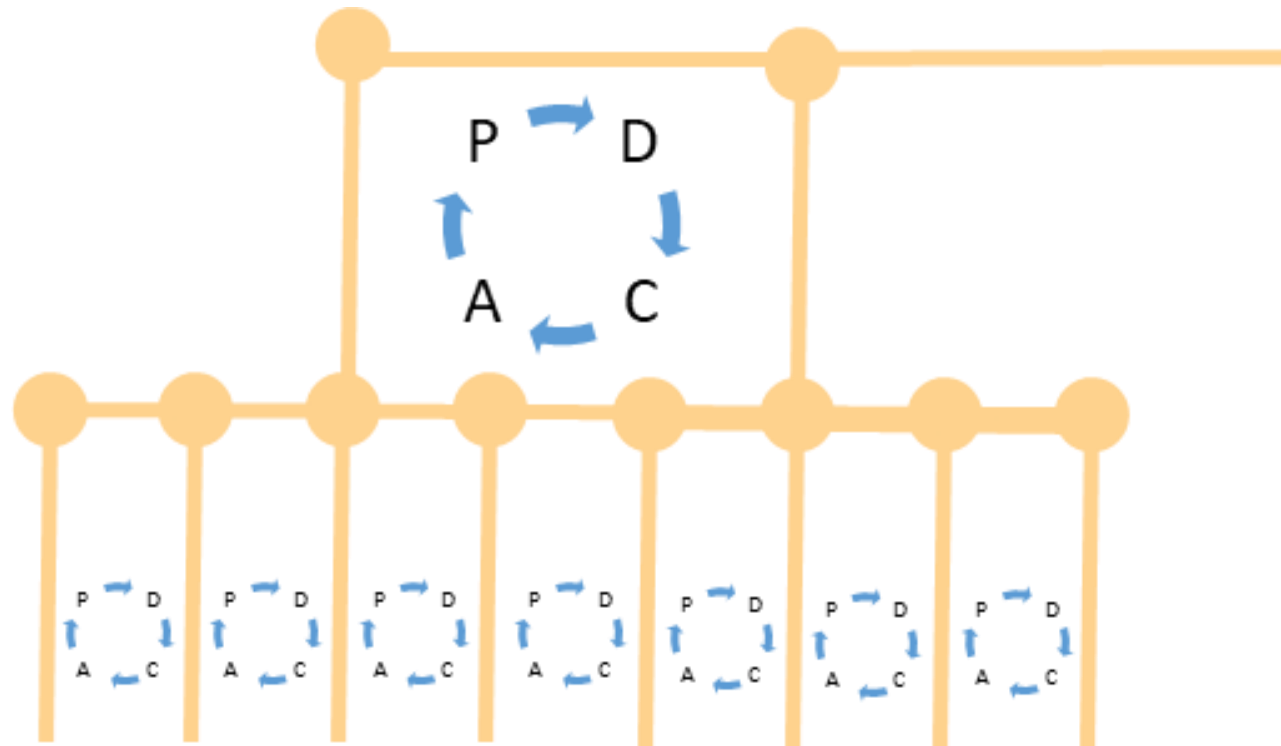
# The approaches we take

Common approach at building waterfall methods



# The approaches we take

With Agile (and SAFe especially), the teams are constantly Inspecting & Adapting



# Agile at Scale

*rego*University 2019  
SAN DIEGO



*Today, no company can make, deliver or market its product efficiently without technology. While smart phones and the internet used to be cutting edge, these days, new application code release dates and time to market for new technologies are shrinking. This is forcing companies accustomed to a four-year release cycle to adopt to change faster. Businesses must learn how to integrate technology release cycles into their production and service cycles.*

<https://www.forbes.com/sites/forbestechcouncil/2017/01/23/why-every-company-is-a-technology-company/#12a3fbbb57ae>

## Challenges of System Development

- Digital World
- Fast Pace Markets
- Need for Quick Adaption to Technical Changes
- Complexity – Bigger Systems, More Integration, Large Failure Impact

## New Bodies of Knowledge

- Agile Development
- System Thinking
- Lean Product Development

# SAFe Lean-Agile Principles

---

#1 - Take an economic view

---

#2 - Apply systems thinking

---

#3 - Assume variability; preserve options

---

#4 - Build incrementally with fast, integrated learning cycles

---

#5 - Base milestones on objective evaluation of working systems

---

#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

---

#7 - Apply cadence, synchronize with cross-domain planning

---

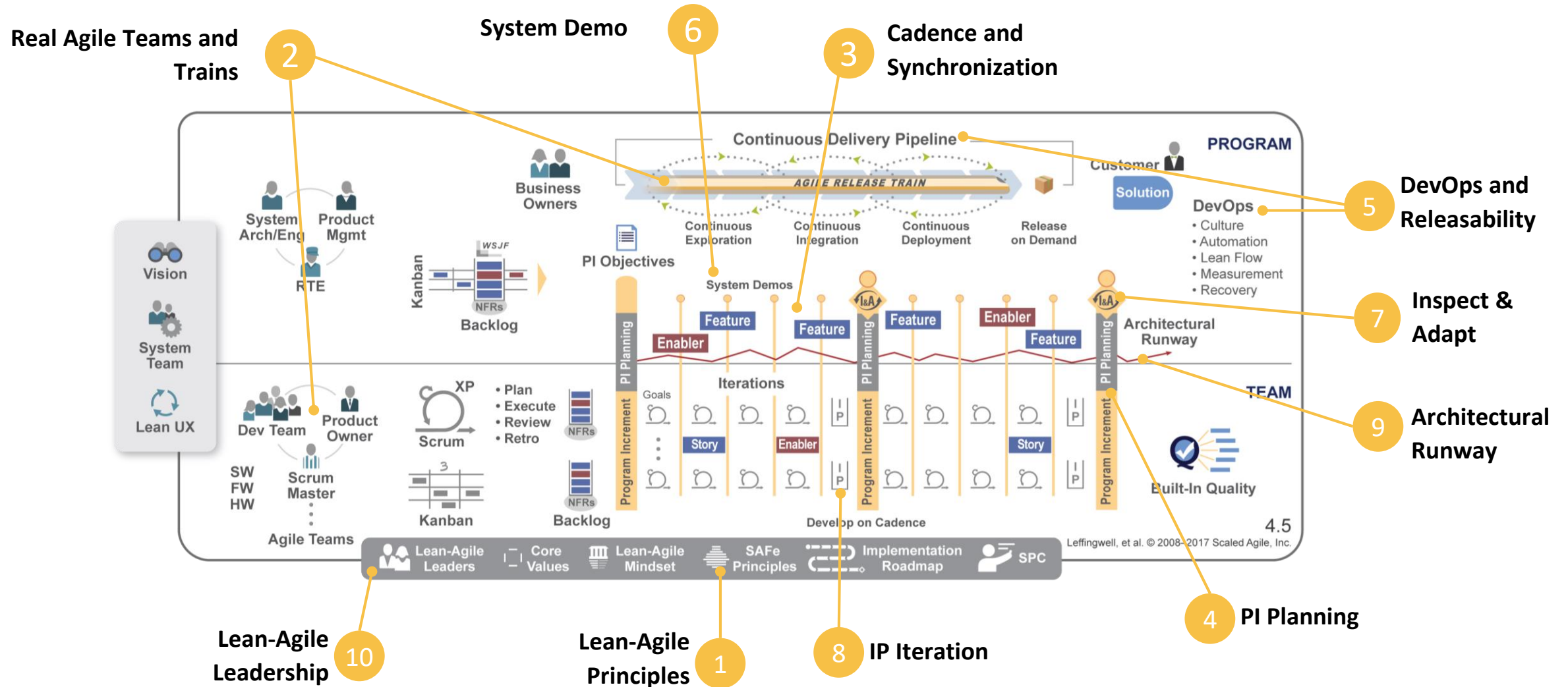
#8 - Unlock the intrinsic motivation of knowledge workers

---

#9 - Decentralize decision-making

---

# Start with Essential SAFe



# Synchronizing with PI Planning

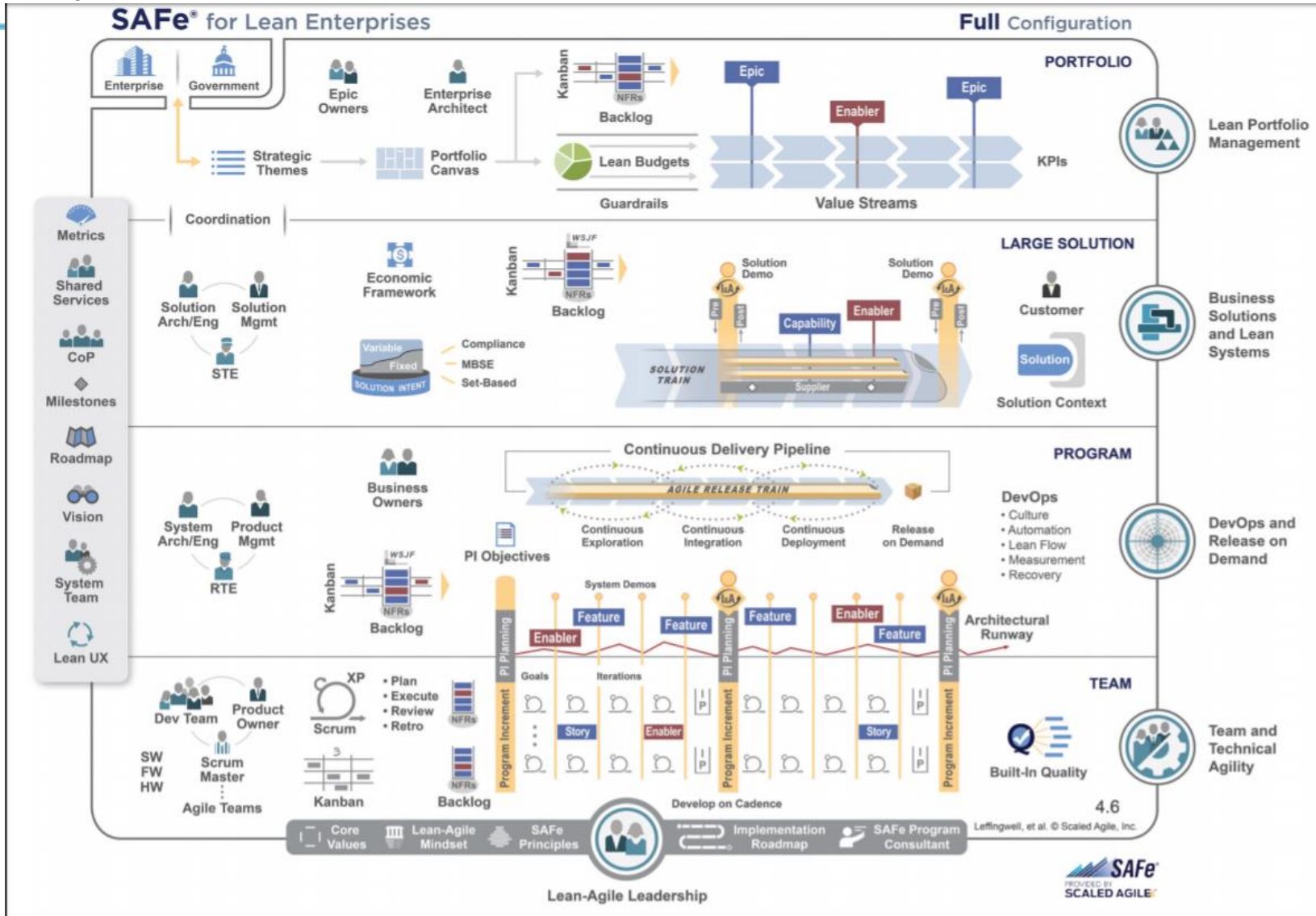
*Future product development tasks can't be pre-determined. Distribute planning and control to those who can understand and react to the end results. —Michael Kennedy, Product Development for the Lean Enterprise*

- ▶ All stakeholders face-to-face (but typically multiple locations)
- ▶ Management sets the mission, with minimum possible constraints
- ▶ Requirements and design emerge
- ▶ Important stakeholder decisions are accelerated
- ▶ Teams create—and take responsibility for—plans



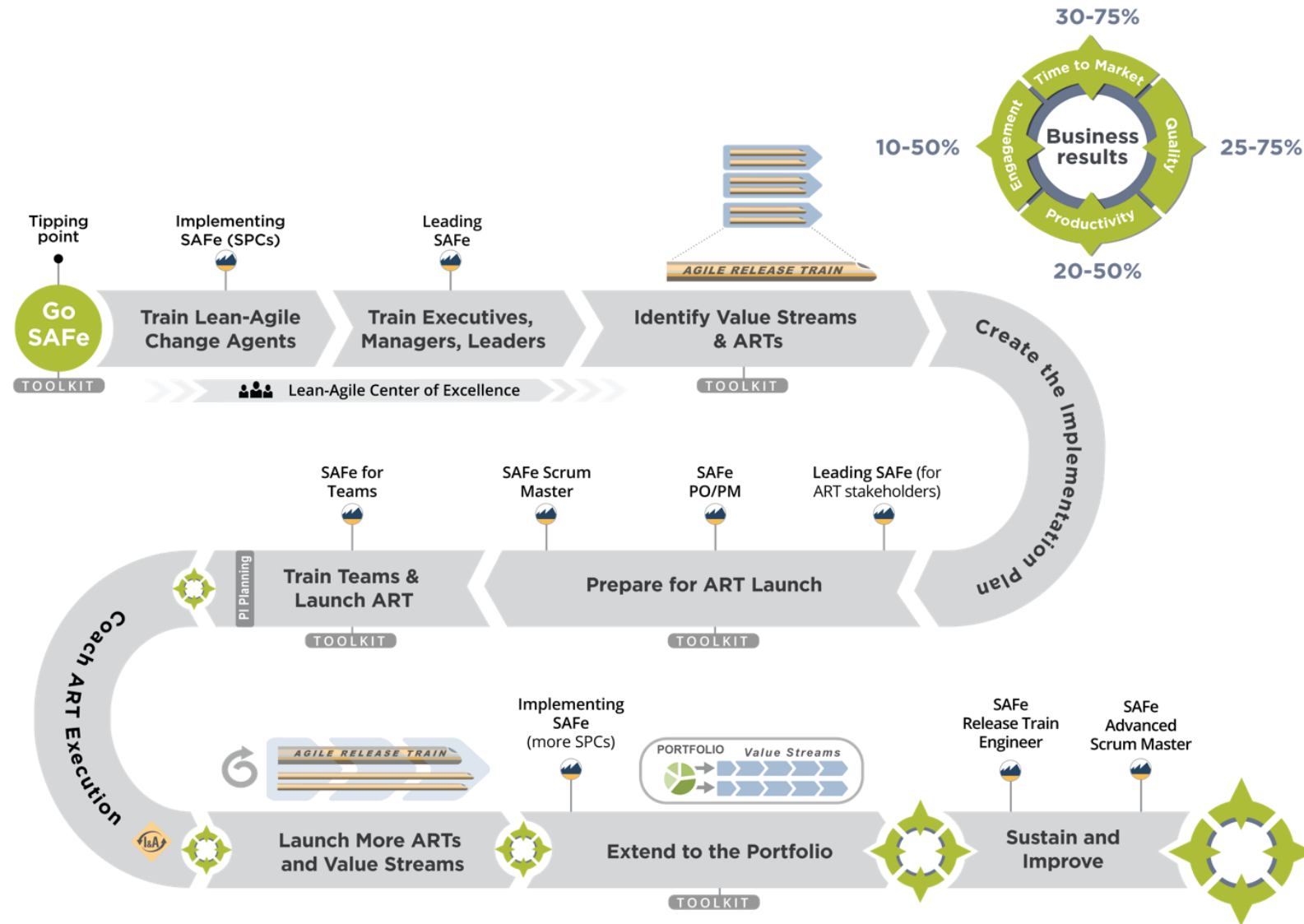
For a short video PI planning example, see: <https://youtu.be/ZZAtI7nAB1M>

# Most Comprehensive, Adds Portfolios – Full SAFe





# Follow the SAFe Implementation Roadmap



# Common topics

*rego*University 2019  
SAN DIEGO

# Resource management

- If doing Agile, there really isn't resource mgmt. in traditional sense
  - Teams are dedicated
  - Scope changes (bring the work to the team)
- Agile funds teams
- However, hybrid project management will continue
- We have some experience, but we continue to have solution emerge
- The challenge is the many still want to do forecasting (estimating)
  - Refer to the points vs hours topic above



# Timesheets

- Many of the Agile Manifesto architects reject timesheets
- While we have some methods, some companies are beginning to do away with timesheets (credit card company)
- Some of our solutions
  - Pitney Bowes: timesheets without time entry
  - Our common Agile connector approach

# Financials

- SAFe refers to “Lean Budgeting”
- Very much theory; most Finance groups are not adopting this
- Traditional project cost accounting will still prevail for the foreseeable future
- We are looking to solve this

# Questions?



*rego*University 2019  
SAN DIEGO

Let Rego be your guide.

# Thank You For Attending regoUniversity

## Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Name = **regoUniversity**
- Course Number = **Session Number**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**



Let us know how we can improve!  
Don't forget to fill out the class survey.



### Phone

888.813.0444



### Email

[info@regouniversity.com](mailto:info@regouniversity.com)



### Website

[www.regouniversity.com](http://www.regouniversity.com)