



# Agile in Practice

Real-world implications for the BA role

Kyle Stock - [kyle.stock@parivedasolutions.com](mailto:kyle.stock@parivedasolutions.com)

Danny Bakker - [danny.bakker@parivedasolutions.com](mailto:danny.bakker@parivedasolutions.com)

Intros & Expectations

The State of Agile

Textbook Definitions

Agile in Real-Life

Common Questions and Challenges

The BA Role in Agile

Closing



# Pariveda Overview

Pariveda is a consulting firm dedicated to solving complex technology and business problems by aligning our people development focus with the mission of our clients.

As an employee-owned company, our naturally curious people have strong technical, business and strategic skills to help our clients identify, architect and develop custom solutions. We believe in challenging our clients' thinking, and we are comfortable solving problems without a clear solution. Our solutions create new opportunities for our clients, and we manage the change each solution brings to their company.

## Our Clients

Pariveda solves the complex problems of clients ranging from Fortune 100 to Global 2000 to startup companies and spanning multiple industries.

Clients partner with us because we specialize in solving complex technology and business problems with no clear solution, our excellence in delivery and our focus on building lifetime relationships. We have served over 400 clients since inception.

## Key Details



### On-shore Only

100% on-shore;  
in-person matters



### Curious Teams

Small teams to solve  
complex problems  
efficiently



### Trusted Relationships

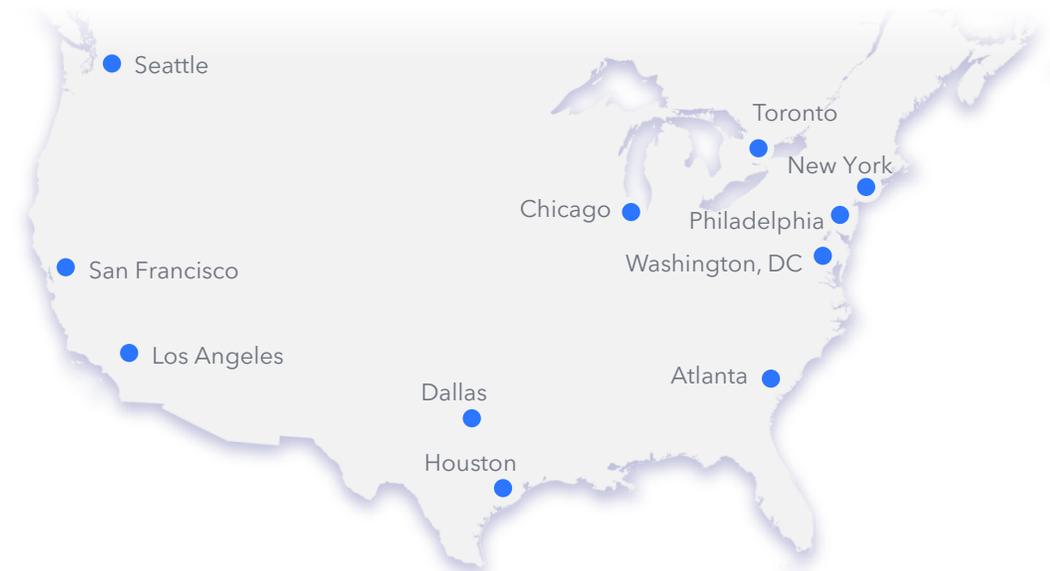
94% repeat/referral  
business



### Business Transformation

Valued by over 400  
clients

## Our Locations



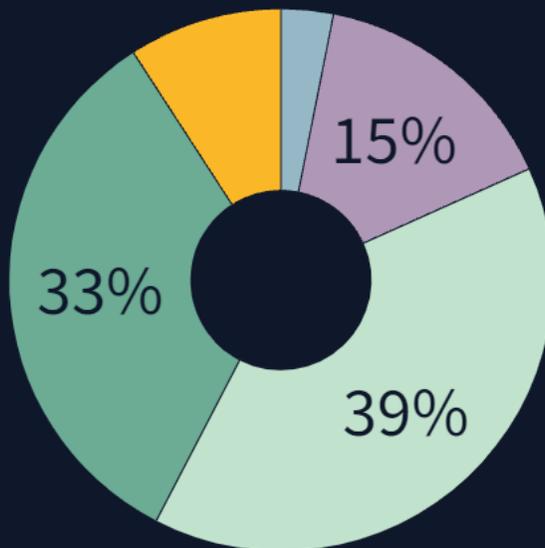
Respond at [Pollev.com/pariveda101](https://pollev.com/pariveda101)

Text **PARIVEDA101** to **22333** once to join, then **A, B, C, D, or E**

# What best describes your experience with Agile software development?

- None: I've never heard of Agil... **A**    Novice: I'm familiar with Agil... **B**    Beginner: I have some experien... **C**    Competent: I've been practic... **D**  
Proficient: I'm an Agile pro; ... **E**

Competent: I've been practicing Agile for a while and I'm very comfortable with common tools and practices



Novice: I'm familiar with Agile terms and principles, but have almost no hands-on experience

Beginner: I have some experience on Agile teams, but would still consider myself a beginner

## What is one thing that you want out of this session? One question that you'd like addressed?

### Top

15	Future of BAs in the Agile world
9	BA skills on the Agile team.
7	Understand what people are doing with agile. I'd also like to hear about use of requirements management tools
5	Any insights on Best Practices or trends evolving
4	I want more information/options as to how to practice Agile in real world scenarios.
4	value and pitfalls of Agile framework (perhaps compared / contrasted to more traditional methods -- e.g., waterfall)
4	How to respond to statements of "there's no BA in agile"
3	Role of BA's in Agile
3	How do you transition from a non-Agile BA to an Agile one?
3	How, as a BA, I can best fit into the agile product team
3	Examples/cases of real world complex situations in BA life and potential solutions
3	the role of the BA
3	The BA role in a DevOps agile team

### New

0	Overview of the BA role in Agile Environment
1	Important considerations when implementing AGile where org and teams are new to Agile
3	Role of BA's in Agile
0	coming off of acing the cbap exam so I would like to know how bass tie into agile.
1	How does an Agile approach to requirements apply to large scale projects with a single launch date, given all of the interdependencies in processes and solution approaches?
2	How projects fit into the Agile world when Projects are either waterfall or Agile but the teams that need to do dev work are Agile.
3	How do you transition from a non-Agile BA to an Agile one?
1	practical aspects
0	best ways to adapt quickly
3	How, as a BA, I can best fit into the agile product team
2	best practices
1	I'm interested in your experience with reinforcing to clients the value of the BA role in Agile practice.
1	Is "hybrid" Agile - Waterfall a valid method?

# If you've never heard about Agile

Agile is the ability to create and respond to change

- It is a way of dealing with, and ultimately succeeding in, an **uncertain and turbulent environment**
- Agile is a **mindset** informed by the Agile Manifesto's values and principles, which provide guidance on how to:
  - **create and respond to change**
  - **deal with uncertainty**
- Agile methodologies are the **conventions that a team chooses to follow** in a way that follows agile values and principles

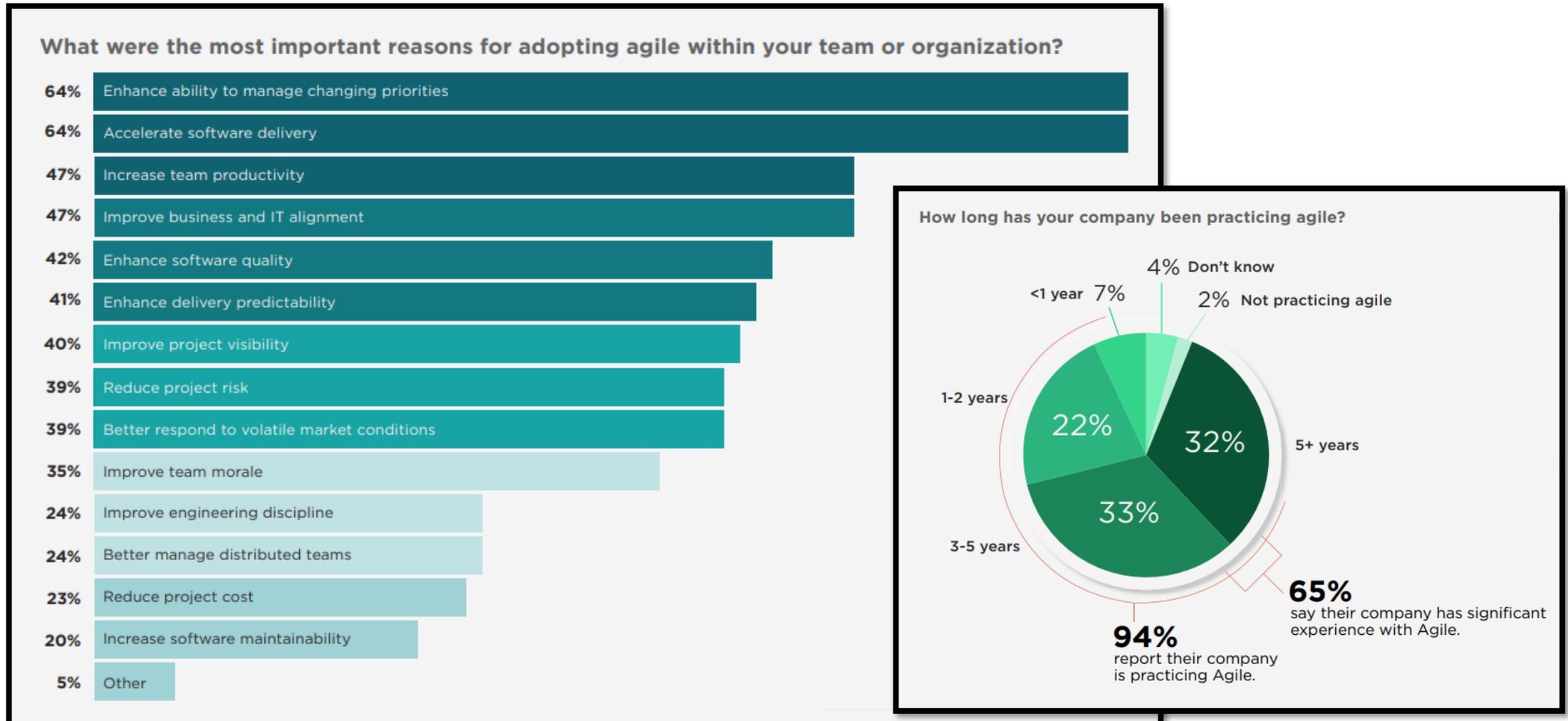
## MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT



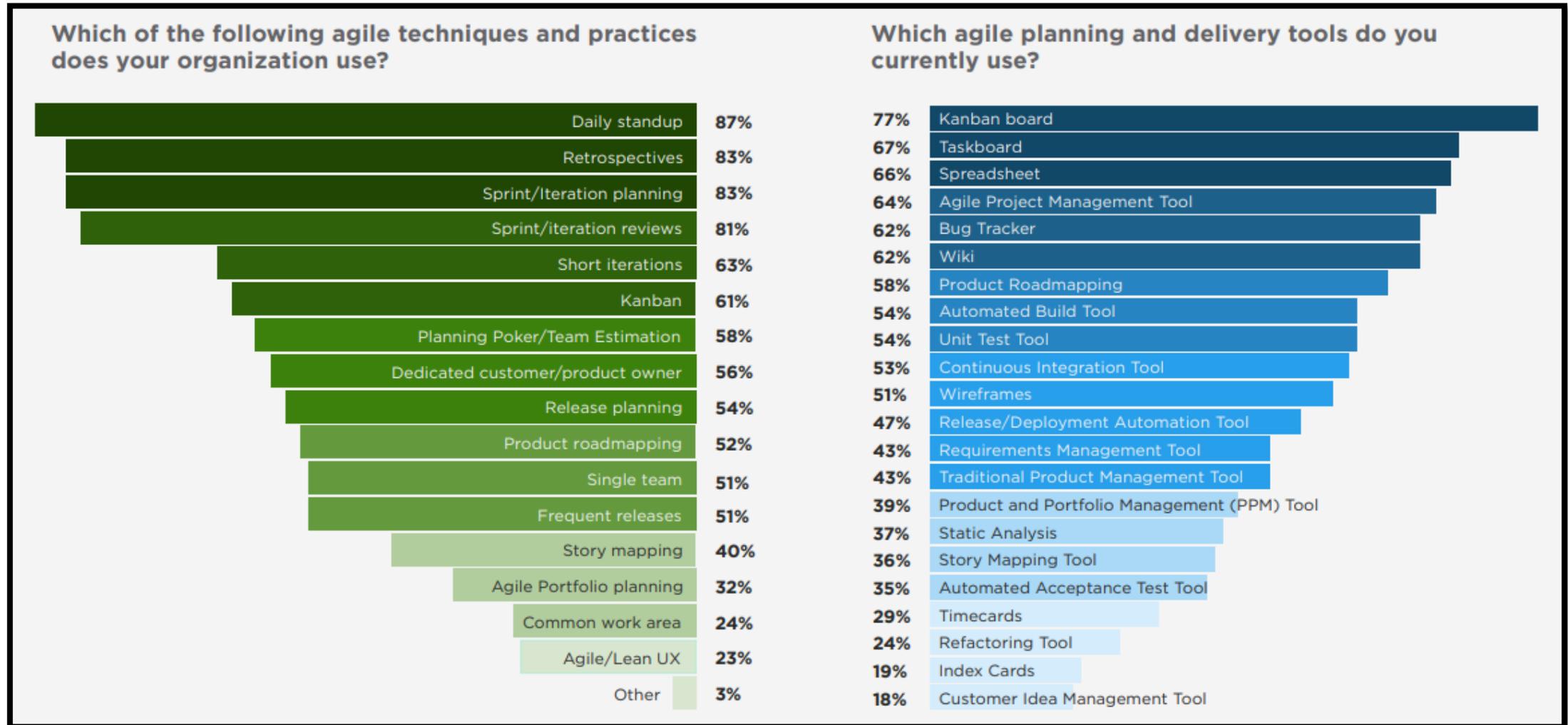
Items on the right ARE important

Items on the left are MORE important

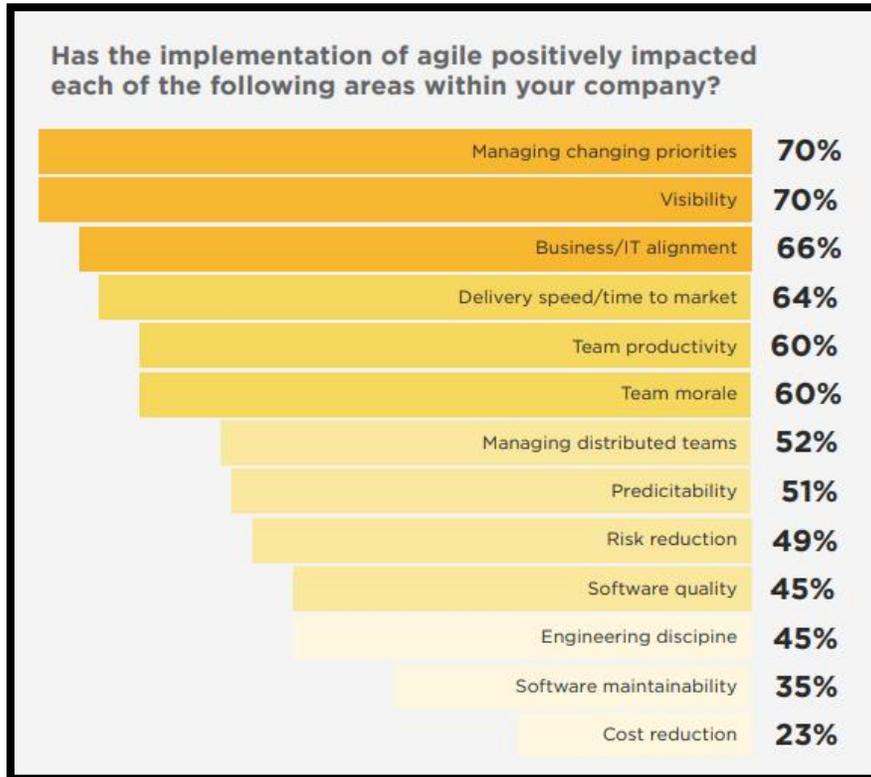
# Agile Today: More companies are using agile than ever, and the reasons have remained consistent for years



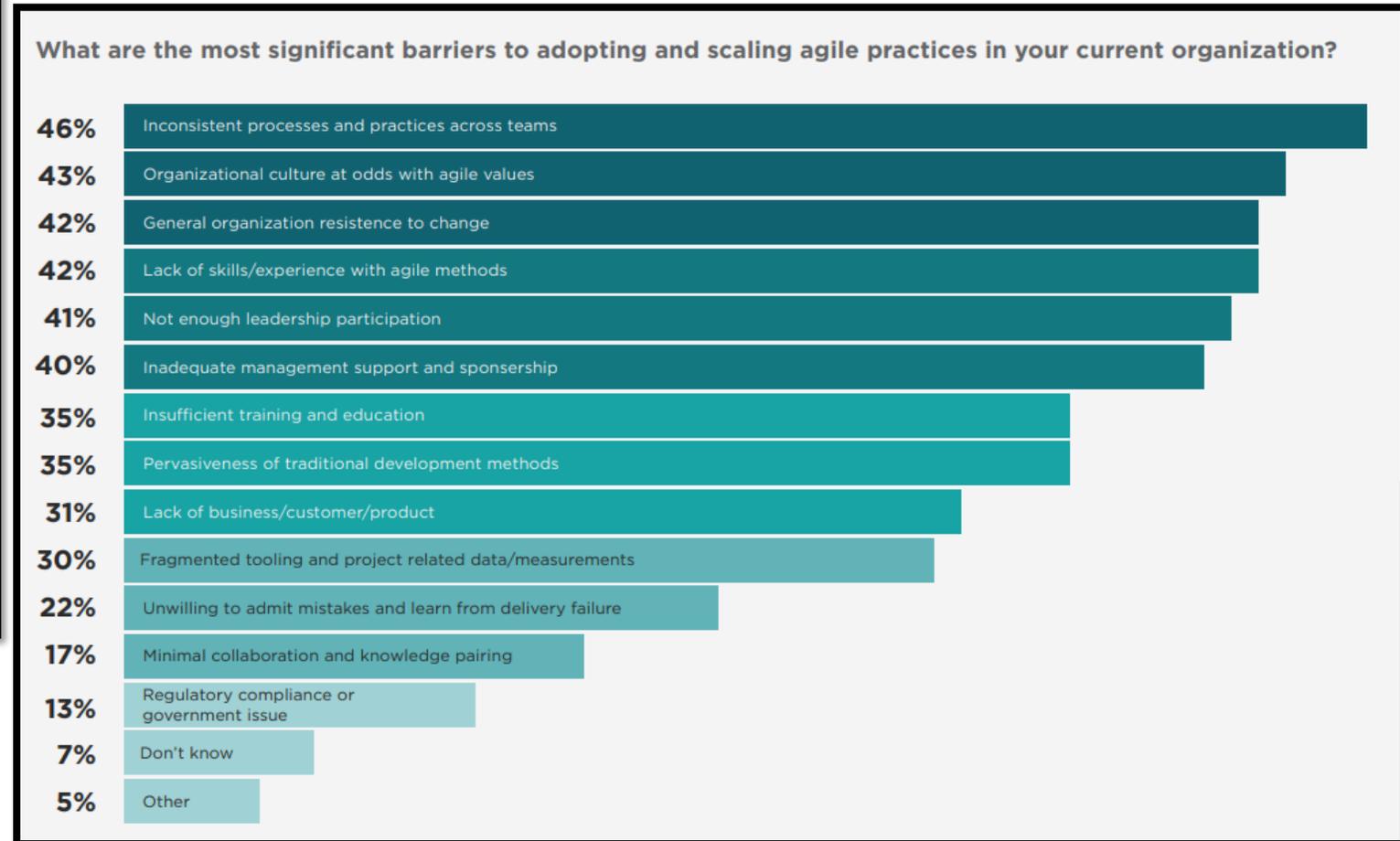
# Agile Today: Techniques and tools have coalesced around emerging “best practices”, but there is broad variability



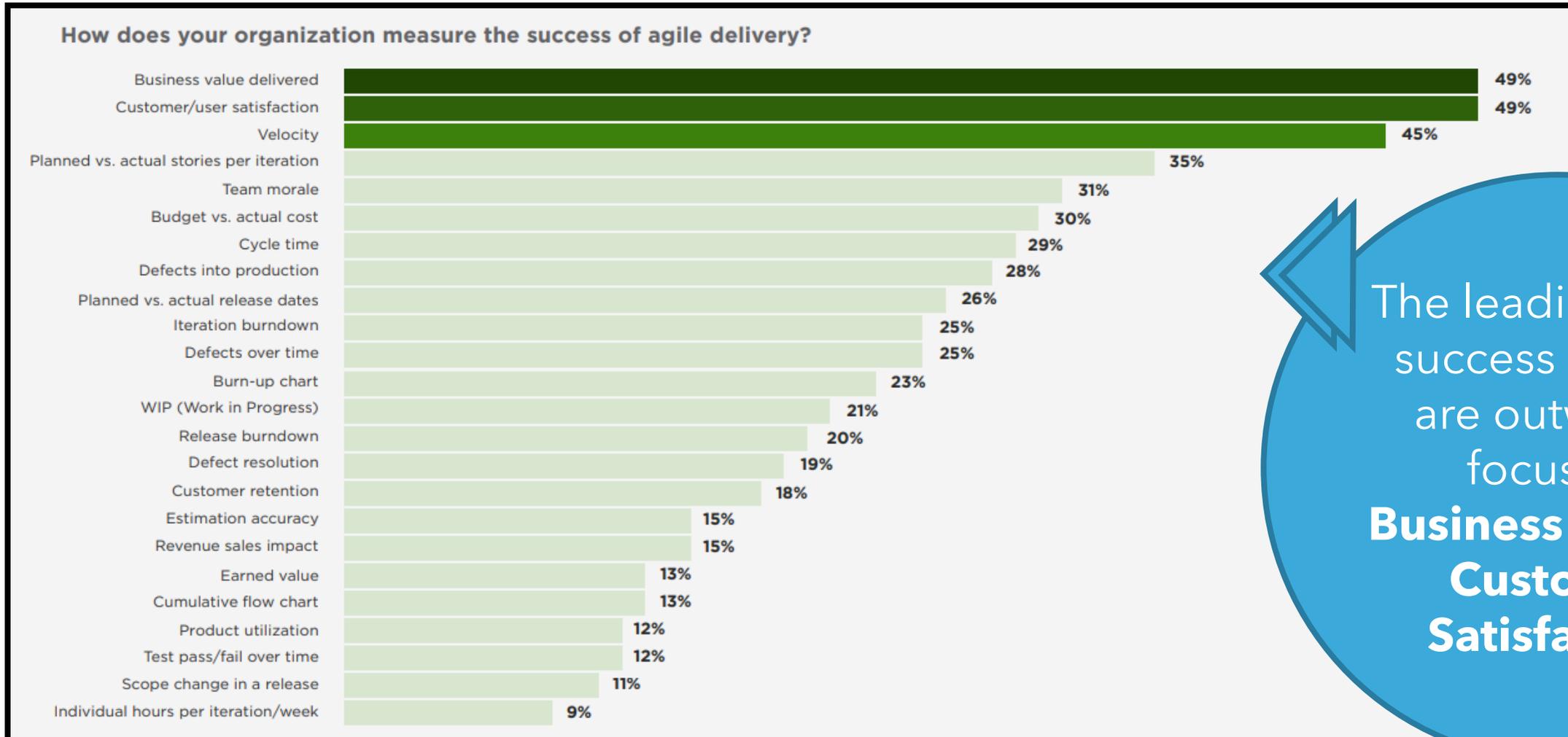
# Agile Today: Positive impacts tend to be more subjective; The top challenges are organizational



State of Agile report, 2021.

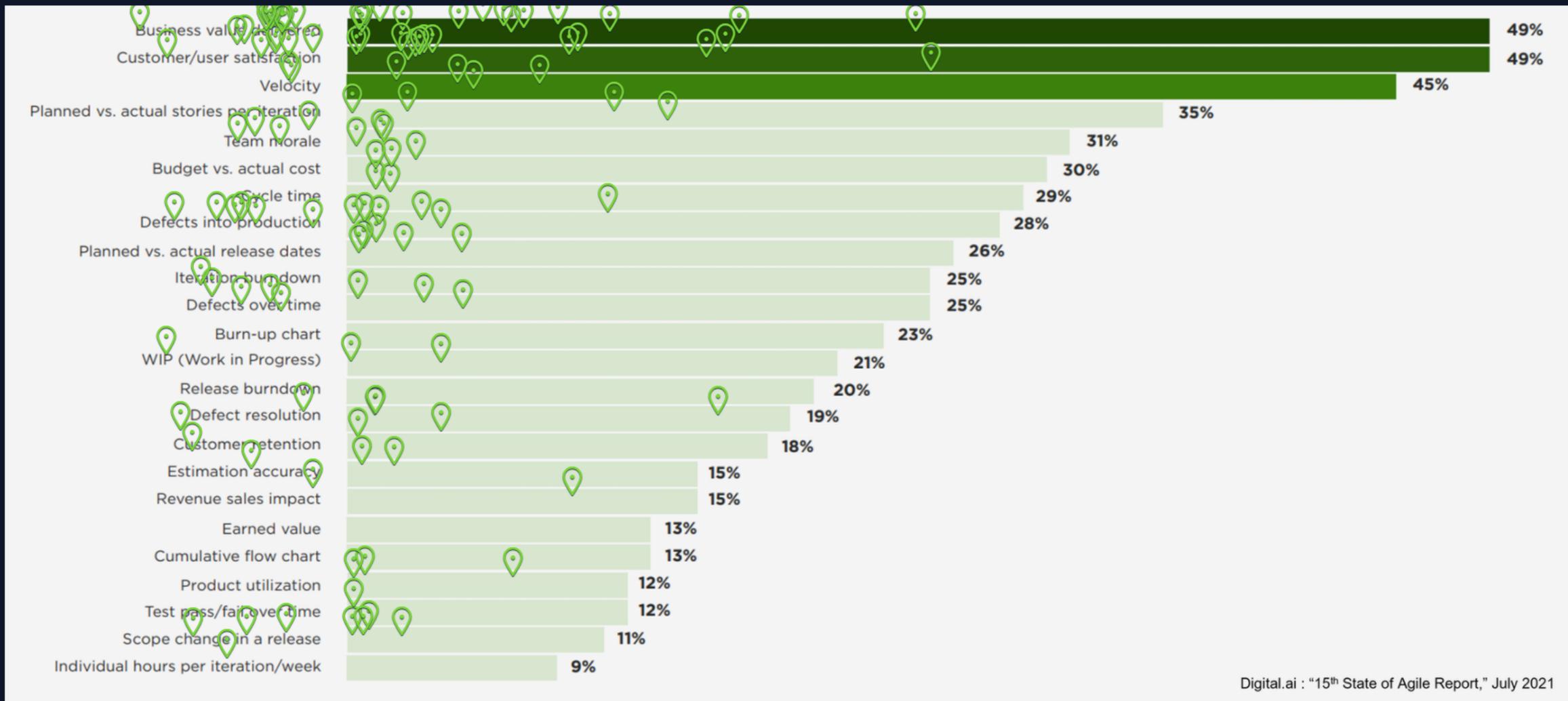


# Agile Today: Companies evaluate success of Agile projects and teams in many different ways



The leading Agile success metrics are outwardly focused:  
**Business Value & Customer Satisfaction**

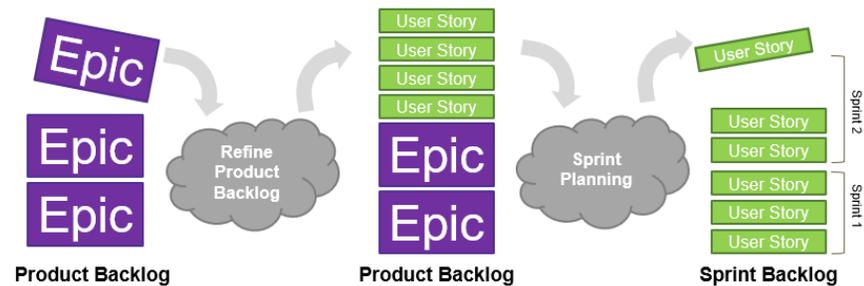
# As a Business Analyst, which success measures are you best positioned to influence? (pick as many as you'd like)



# Textbook Definition: Agile principles, terms and process

**Theme** - highest level grouping of requirements. Optionally, requirements can be further grouped into Epics and Features

**User Story** - format for expressing desired scope from the perspective of an end user



**Definition of Done** - a set of criteria which need to be met before a user story can be ready for sign off by the product owner

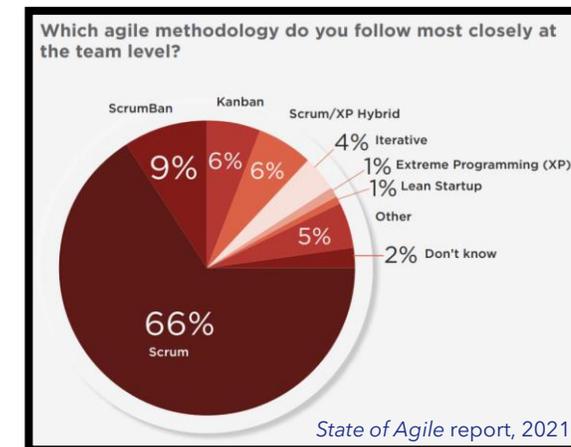
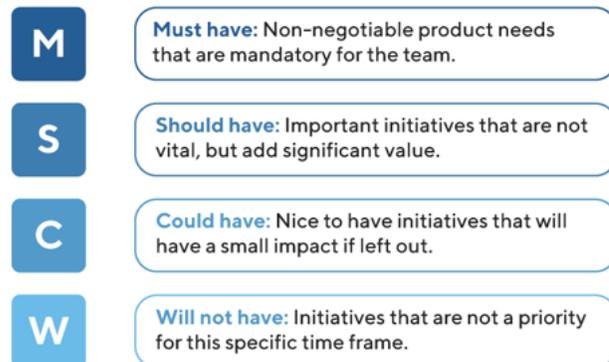
**Definition of Ready** - a set of criteria which need to be met before a user story can be taken into sprint planning

**Story points** - Relative measure of a user story estimate based on size and complexity

**Velocity** - Rate of work done by the team. This rate is a key component used to plan out an iteration and release

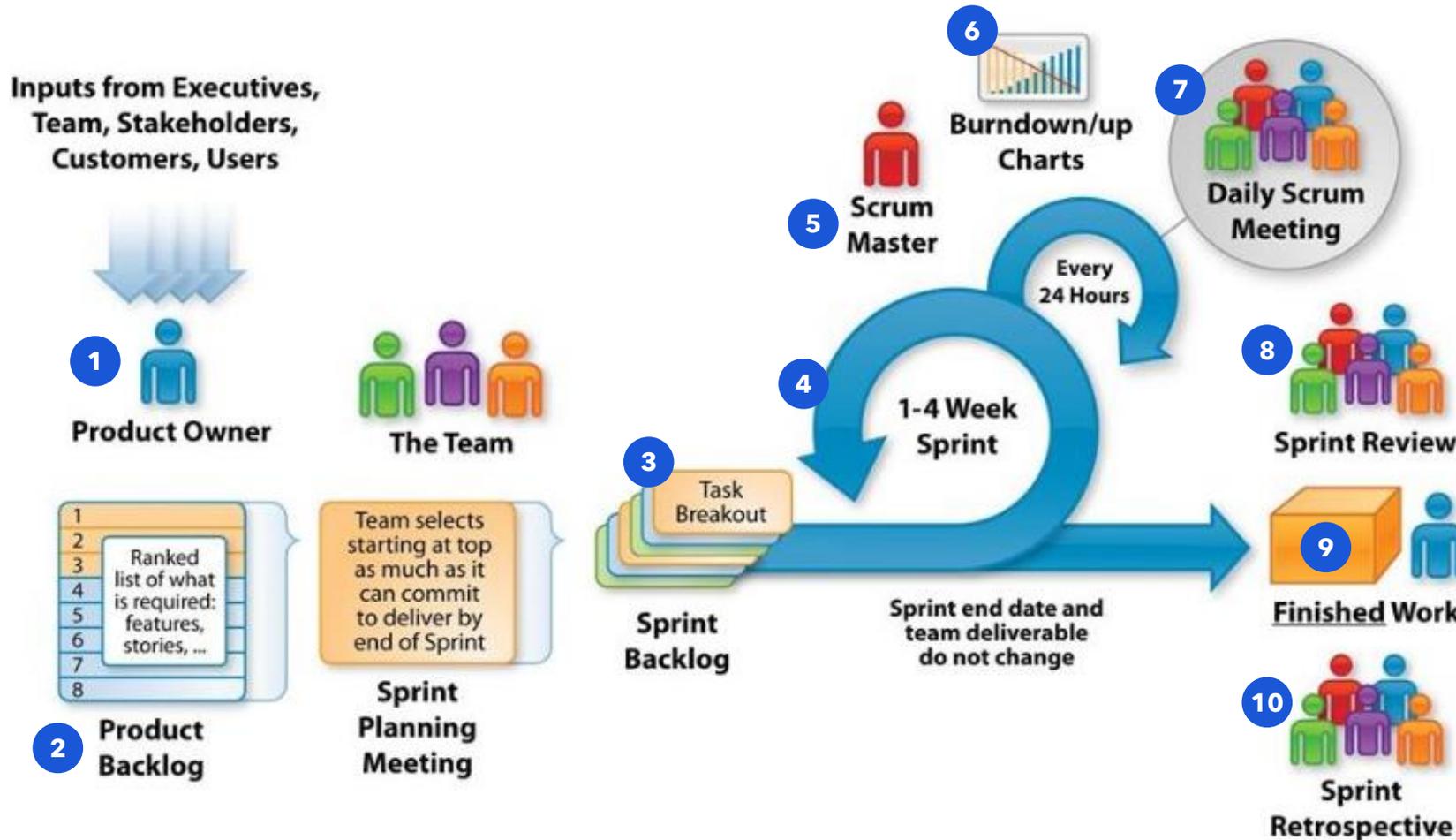
**Scrum, Kanban, Extreme Programming, Adaptive Software Development, Lean Software Development, SAFe, LeSS** - Agile methodologies that vary in their terminology, scaling, release timelines, organization of work and roles, while still guided by core agile principles

**MoSCoW** - This prioritization technique is used to reach a common understanding on the importance of each requirement



# The Agile - Scrum Framework

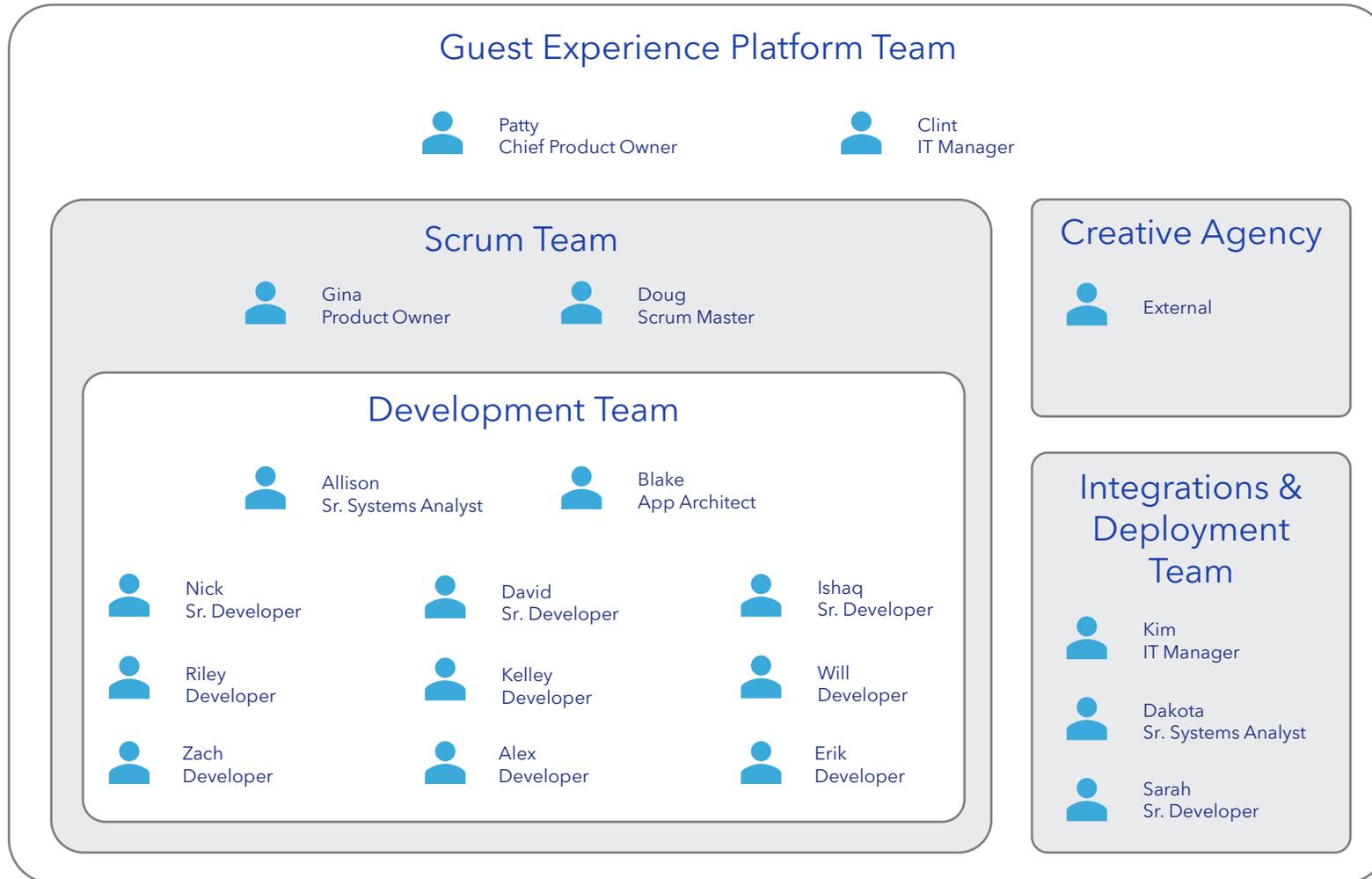
Scrum outlines the team process for a **Release**, a set of iterations in which software will be delivered



- 1 **Product Owner** - Represents the voice of the customer and is accountable for prioritizing user stories that deliver the highest value
- 2 **Product Backlog** - A prioritized inventory of yet-to-be-worked-on user stories
- 3 **Task** - Individual unit of work that is defined when breaking down a user story
- 4 **Sprint** - A structured time duration where a potentially shippable piece of software can be developed and measured for progress
- 5 **Scrum Master** - Facilitator for the team, responsible for planning and execution of the product delivery with the team
- 6 **Burndown/up Charts** - Health report of the team's progress during each iteration
- 7 **Daily Scrum Meeting** - Recurring forum for communicating user story status to the team
- 8 **Sprint Review** - Demo at the end of each iteration and overview of the team's accomplishments
- 9 **Finished Work** - post-review, work can continue through phrases of additional validation, testing, and final delivery
- 10 **Retrospective** - The team to inspects itself -- what went well and what needs improvement -- and creates an action plan to be enacted during the next sprint

<https://www.neonrain.com/agile-scrum-web-development/>

# Real-life Agile: Example 1 - The Mature Agile Team



This large 13-person Scrum team has been executing Agile for 5+ years, going through several iterations to scale over time

Many roles (Scrum Master, Product Owner, Architect) are formally owned by individuals, but the team often rotates other responsibilities (DevOps lead, Release Coordinator, etc.)

To operate at scale and support requests and requirements from multiple dependencies (EA, InfoSec, Legal, etc.) the team adopts 2 practices:

1. Teaming on assignments each sprint so that 2-3 people share responsibility for related scope
2. Dividing triage and analysis of requests and requirements to look ahead and plan for what's coming next

# Mature Agile Team Detail: Thorough planning standards ensure requirements and dependencies are covered each sprint

## PROGRAM

🔄 Iterate ongoing to define and prepare releases until fully transitioned to DevOps mode

### Readiness Criteria–PROGRAM INCREMENT :

- 1.1.1  Release-level feature roadmap prepared and published M
- 1.1.2  Cross-program dependencies (e.g., architectural, design comps) tagged to roadmap and backlogged with a minimum  $R_{N-1}$  dependencies tagged M
- 1.1.3  Roadmap reviewed and refined in collaboration with major stakeholder groups: Product owners, App Leads, Design Leads, Dev Leads, Enterprise Arch. M

## RELEASE PREPARATION

🔄 Every 1..N Sprints

### Readiness Criteria–PSI

- 2.1.1  Prior release retrospective completed M
- 2.1.2  Backlog groomed to minimum  $R_{N+2}$  capacity M
- 2.1.2.1 \* Design comps completed / reviewed for PSI features M
- 2.1.2.2  Effort estimated (in story points) M
- 2.1.2.3  Value and risk assessed M
- 2.1.2.4  Functional dependencies identified M
- 2.1.2.5  Non-Functional dependencies identified M
- 2.1.2.6 \* Analytics tagging requirements identified M
- 2.1.2.7 \* Legal review completed for all PSI features M
- 2.1.2.8  Architectural runway confirmed in place / operational M
- 2.1.2.9  Stories prioritized M
- 2.1.2.10  Functional acceptance criteria defined per story M
- 2.1.2.11  Non-Functional acceptance criteria confirmed by story M
- 2.1.3  Potentially Shippable Increments (PSI) defined M
- 2.1.3.1  Sprint(s) sized based on forecasted capacity M
- 2.1.3.2  "Must haves" for PSI planned to conservative-case capacity (allowing buffer for hot fix / prod ops) M
- 2.1.3.3  Account for longer running features (toggling tactics) M

## RELEASE MANAGEMENT

🔄 Iterate ongoing across all releases

### On-Going Release Management Activities (across teams):

- 3.1.1  Cross-workstream dependencies managed M
- 3.1.2  Risks identified, assigned, mitigated and reported M
- 3.1.3  Demand and capacity allocation forecasted, tracked and reported across sprints for M
- 3.1.3.1  New feature / functionality M
- 3.1.3.2  Planned maintenance M
- 3.1.3.3  Hot fixes (production issues with SLA within sprint) M
- 3.1.4  Release-level burn-ups tracked / reported across teams M
- 3.1.5  Aggregate velocity tracked, calibrated and reported M

SPRINT(S) EXECUTED (next page)

### Definition of Done–PSI

- 3.2.1  Deployment Planned (e.g., feature toggling, etc) M
- 3.2.2  Cutover Planned M
- 3.2.3  Rollback Planned M
- 3.2.4  Communications Planned M
- 3.2.5  Sprint demo session conducted across teams M
- 3.2.5  Go / No-Go Decision per sprint / workstream M

## RELEASE TO PRODUCTION (OPTIONAL)

🔄 Every 1..N Sprints

### Readiness Criteria–RELEASE

- 2.2.1  Deployment plan executed M
- 2.2.2  Knowledge transfer / training for prod ops completed M
- 2.2.3  Communications plan executed M
- 2.2.4  Manual penetration tests passed (for major releases as per InfoSec) M
- 2.2.5  Manual E2E testing passed M

### Definition of Done–RELEASE

- 2.3.1  Cutover plan executed M
- 2.3.2  Release-level retrospective completed (Scrum-of-Scrums level) M

### Ongoing Program Activities:

- 1.2.1  Processes assessed for consistent application to established / agreed practice (i.e., release level / scrum level "contract") M
- 1.2.2  Teams coached on consistent application of established practices M
- 1.2.3  Process improvement opportunities identified, prioritized and implemented M
- 1.2.4  Event, Incident, and Problem Management (including hot fix exception process where severity warrants) A/M\*

### Ongoing Testing Activities:

- 1.3.1 \* Ad hoc penetration tests executed M
- 1.3.2  Exploratory testing conducted M

# Mature Agile Team Detail: Thorough planning standards ensure requirements and dependencies are covered each sprint (cont.)

## SPRINT PLANNING

Once per Sprint (every 2-3 weeks)

### Readiness Criteria - SPRINT

4.1.1	☐	Sprint capacity forecasted for each	M
4.1.1.1	☐	New feature development	M
4.1.1.2	☐	Planned maintenance	M
4.1.1.3	☐	Hotfixes	M
4.1.2	☐	QA capacity assessed	M
4.1.3	☐	Story decomposition to 2-3 days max	M
4.1.4	☐	Sprint task and dependency planning	M
4.1.5	☐	Sprint backlog locked and loaded	M

## SPRINT EXECUTION

Daily

### Ongoing Activities

5.1	☐	Daily Stand-up conducted	M
5.2	☐	Tasks recalibrated / assigned	M
5.3	☐	Progress tracked / burn-downs updated / velocity tracked	M
5.4	☐	Ongoing cross-functional conversations / clarifications (product owner, design, architecture, etc.)	M

### STORY DEVELOPMENT & TEST PLANNING

#### Readiness Criteria-STORY (DEV Environment)

6.1.1	☐	Designs Completed:	M
6.1.1.1	☐	Functional designs	M
6.1.1.2	☐	Technical designs	M
6.1.2	☐	Tests Authored:	M
6.1.2.1	☐	Test Plan for story	M
6.1.2.2	☐	Functional test cases	M
6.1.2.3	☐	Automated unit tests	M
6.1.2.4	☐	Component tests	M
6.1.2.5	☐	App-specific integration tests	M
6.1.2.6	☐	UI/Service Smoke/ETE	M
6.1.2.7	☐	UI page performance	M
6.1.2.8	☐	Short burst load tests	M
6.1.2.9	☐	Standard load tests	M
6.1.2.10	☐	UAT	M
6.1.3	☐	Coding completed	M
6.1.4	☐	Tests executed	A/M
6.1.4.1	☐	Automated unit tests	A
6.1.4.2	☐	Automated component tests	A
6.1.4.3	☐*	Static analysis (Lint/SonarQube)	A
6.1.4.4	☐	Veracode sandbox static scan	A/M
6.1.5	☐	Dev team code review	M
6.1.6	☐	Promote to DEV	A

### STORY VERIFICATION (IN CYCLE)

#### Readiness Criteria-STORY (TEST Environment):

6.2.1	☐	Automated App-specific integration tests passed	A
6.2.2	☐	Promote to TEST	M

#### Readiness Criteria-STORY (STAGE Environment):

6.3.1	☐	Automated UI/Service Smoke Tests passed	A
6.3.2	☐	Promote to STAGE	M

#### Readiness Criteria-STORY (INT Environment):

6.4.1	☐*	UI page performance tests passes	A
6.4.2	☐*	Short burst load test passed	A
6.4.3	☐	Promote to INT	M

#### Definition of Done-STORY

6.5.1	☐	Manual Smoke tests passed	M
6.5.2	☐	Functional QA Test passed to defined defect standards	M
6.5.3	☐	Cross-Browser tests passed	M
6.5.4	☐	Automated regression tests author	M
6.5.5	☐	Documentation completed	M
6.5.6	☐*	UAT passes	M
6.5.7	☐*	EA Code Review	M
6.5.8	☐	Record story points as "DONE"	M

## SPRINT CLOSE-OUT

### Definition of Done-SPRINT

4.2.1	☐	Story tagged for deployment	M
4.2.2	☐	Sprint verified	M
4.2.2.1	☐*	Veracode Static Scan	A/M
4.2.2.2	☐*	Veracode Dyn. Scan	A/M
4.2.2.3	☐	Automated QA Regression Tests	A
4.2.2.4	☐*	Standard Load Tests	A
4.2.2.5	☐*	Sustained Load Tests	A
4.2.2.6	☐	Sprint Demos	M
4.2.3	☐	Sprint Retrospective done	M

Automation is crucial in enabling iterative delivery

# Mature Agile Team Detail: A consistent cadence of meetings and ceremonies enables planning and collaboration

Meeting	Owner / Scheduler	Regular Attendees	As Needed Attendees	Frequency	Suggested Duration
Program Review	PMO	Product Owners, App Leads, Workstream Leads, QA, D&IS, Sponsor		Weekly	1 hour
Backlog Grooming	App Lead	Product Owner(s), Workstream Lead, Design, PMO Rep	QA, Developers, Analytics, EA	Regularly per sprint; will vary per team	1-2 hours each week
Sprint Task Planning	Workstream Lead	Developers, QA, App Lead	Design, Analytics, EA, Product Owner	On or before Day 1 of sprint per team	2 hours (based on a 2 week sprint)
Daily Stand-up	Workstream Lead	Developers, QA, App Lead, Product Owner	Design, Analytics, EA	Daily per team	15-20 minutes
Scrum of Scrums	D&IS	App Leads, Workstream Leads, QA, EA, Analytics, Other teams, PMO	Developers	Daily	< 30 minutes
Code Analysis / Performance Results Review	Workstream Lead	Developers, EA	App Lead, PMO Rep	Weekly Tech Sync meeting or as needed	1 hour per Tech Sync weekly, plus additional time as needed
EA Code Review	Workstream Lead	Developers, EA		As needed	As needed
UAT	QA	Product Owner(s)	App Lead, Workstream Lead, Developers	Regularly per sprint or as needed; approach may vary per team	1-3 hours per team each sprint
Veracode Results Review	PMO	Workstream Lead, InfoSec	App Lead, EA	Once per sprint if needed (only when issues are reported)	As needed
Cutover Plan Review	PMO	App Leads, Workstream Leads, QA	Developers, EA, Analytics, Other teams		
Sprint Review & Demo	Product Owner	Product Owners, App Leads, Workstream Leads, Developers, QA, Design, EA, Analytics, PMO	InfoSec, Other teams	Day 9 of sprint	1 hour per team each sprint
Sprint Retrospective	App Lead	Workstream Lead, Developers, QA		Day 9 or 10 of sprint; will vary by team	1 hour per team each sprint
Release Retrospective	PMO	App Leads, Product Owners, Workstream Leads, QA, Design, EA, Analytics	Other teams	TBD (Following release )	1 hour each release

# Real-life Agile: Example 2 - The beginner agile team



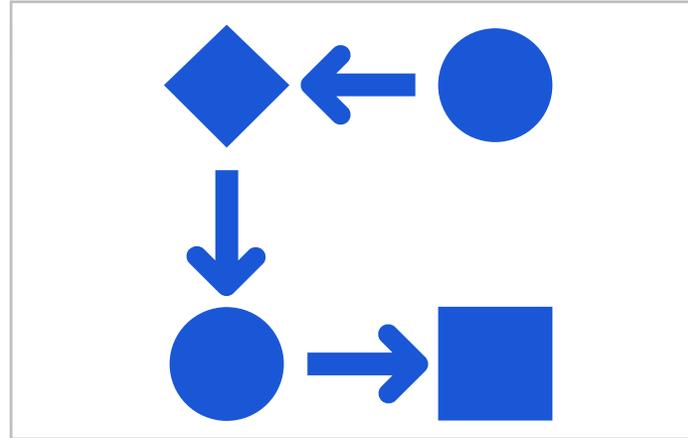
## The Team

Beginning in agile can start with a single team.

Roles (Scrum Master, Product Owner, Tech Lead) may not be covered by an individual, but rather the responsibilities shared across the team or combined.

The team must come up with their criteria sets for when a user story is ready to be worked and when it is ready to be signed off. They review these Definition of Ready and Definition of Done often in the early adoption of agile, so they can establish a consistent pattern for delivery quality.

The team meets to pull together their collective assignments and begin shaping a product backlog. They involve stakeholders to confirm overall vision and assign priority to the items they've been working. They define a process to intake new work so that it is properly sized and prioritized before any one works on it.



## The Techniques

Agile ceremonies are scheduled for the first time to bring the team together for important planning, collaboration, and feedback activities.

At first, these meetings can feel awkward as the team experiences a new level of transparency, and with it, vulnerability. It also puts new demands on team members' time to do work.

As the team begins to follow the patterns of the ceremonies, they can collectively focus that the value of the meeting is being achieved. Over time, the vulnerability and awkwardness make way to a strong sense of camaraderie.

The ceremonies coalesce conversations that were previously happening as one-offs that team members had to chase down themselves, now streamlining the process across the team.



## The Tools

Team members should begin transitioning the tracking of their work into a central location. This could be a shared spreadsheet, or a more sophisticated platform like Jira.

The important principle is that all work being worked by the team is visible and accurate. This requires team members to update status regularly. Though it requires additional overhead, the insight it provides translates into better strategic guidance that product owners can coordinate with stakeholders.

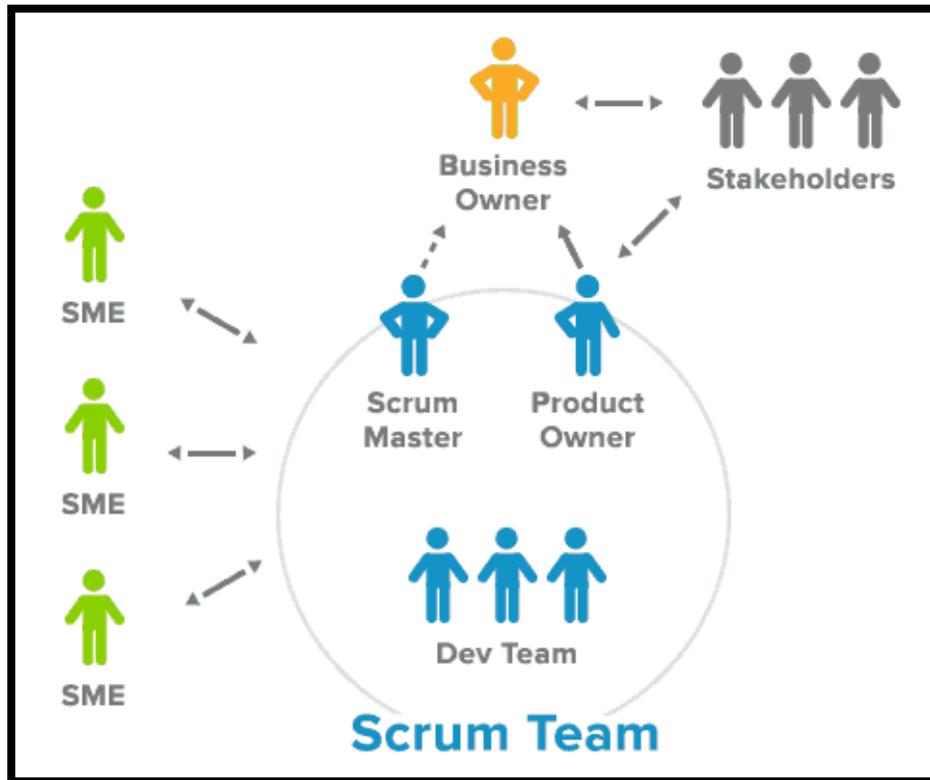
Reporting and charts give the team and leaders insight into how execution looks compared to original plans. The real power in these visuals is the conversations they drive to help the team continuously improve and work towards predictable delivery within reasonable limits.

# Common Questions: What would you like to know about overcoming real-world challenges of Agile?

Response	Net Votes
How do you capture detailed requirements in a concise user story? (field lengths, response times, control placement, etc.)	11
What documentation do you use in agile?	6
How is scope managed when the agile process is meant to be so flexible?	4
If agile is meant to be so responsive, what value is there in planning?	2
BA role, since what i understand BA is the backbone in executing agile. BA mind set, approaches to deal with issues, risks, etc	1
What about external teams? (InfoSec, Legal, etc.)	1
Do you really only plan for one sprint at a time?	1
How much support/buy-in from leadership is needed for my team to run agile?	1
How, as BA, can I influence the management to provide the priorities for the upcoming release a particular N weeks in advance? Have you face similar situations? What helped your team? (input: item has been raised several times during the retrospective; not all planned features developed;	0
How do you get the business stakeholders to be on board with Agile Methodology?	0
What COTS tool do you use to gather and manage requirements? Traceability, dependencies..	0
How do you get the buy-in from all teams fro implementing Agile across the org?	0
Can you really get anything "done" in two weeks?	0
What about bugs? When does QA happen?	0
Are agile projects cheaper than traditional waterfall or other delivery approaches?	0
How fast can a project get started when using agile?	0

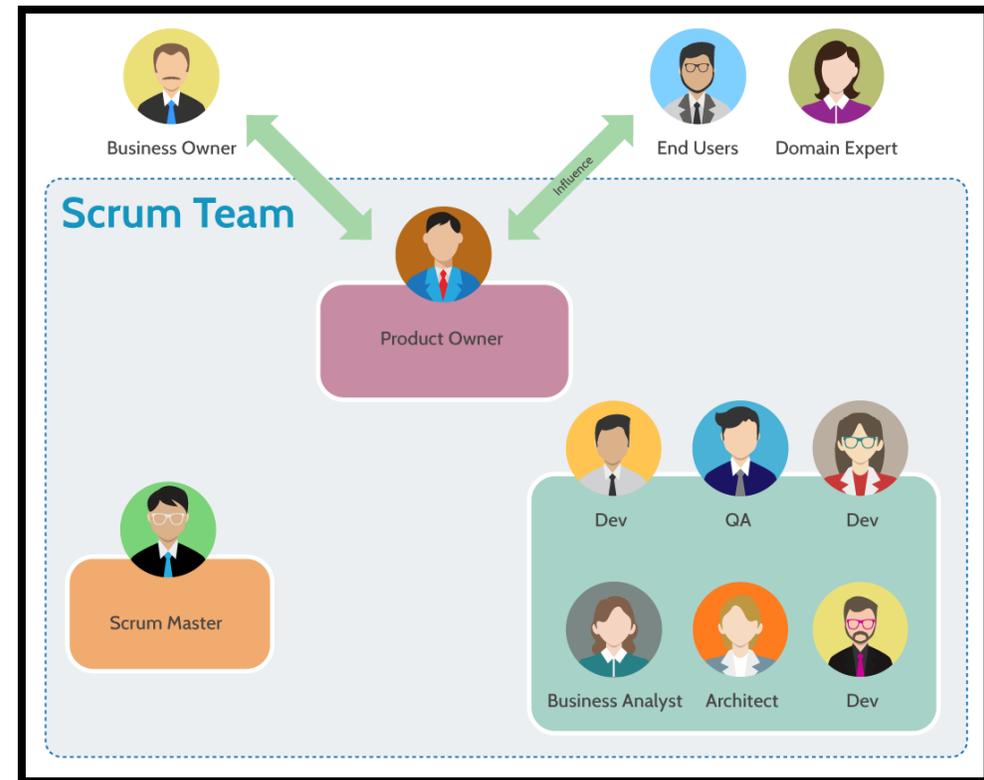
# Now that we understand Agile in practice...where do the Business Analysts fit in?

Agile teams often have minimal roles:



<https://oldane-graham.medium.com/the-ideal-team-for-agile-development-4bb1b52fa9c2>

But specialist roles often emerge:



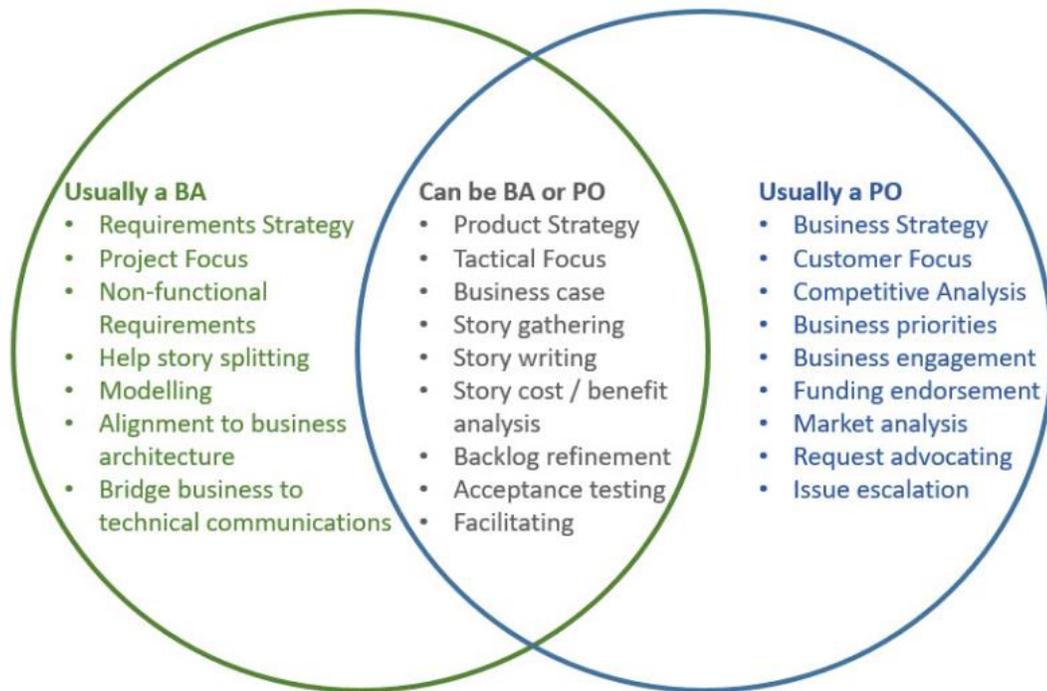
<https://www.tutorialscampus.com/agile/scrum-roles.htm>

**Teams should not hesitate to structure work around traditional roles if this approach helps them**

# Agile has introduced the Product Owner role, which sits between traditional Business Analysts and Project Sponsors

We view these roles as **complementary**, with **flexibility** to share or reassign responsibilities as needed

Common BA and PO Roles and Focus



## Emergence of the PO role has spurred a few trends:

1. Flexible BA's - With PO's often taking on some traditional BA responsibility, BA's tend to flex to support teams in other ways (e.g. wireframing, test planning & execution, change management)
2. BA's becoming PO's (or Scrum Masters) - BA's are often well positioned to take on other Agile roles and it is not uncommon for individuals to move between roles (or create new roles in a flexible Agile environment)
3. Fewer resource managers - Roles that are accountable for deploying and overseeing BA's are less common; This may create autonomy for BA's, but could also make accountable unclear

<https://www.101ways.com/the-business-analyst-and-the-product-owner/>

**QUESTIONS?**

# What would you be interested in exploring in a follow-up session on Business Analysis in Agile?

Top

- 13 — Hands-on Agile practice workshop (user story workshop, story estimation, task breakdown, daily stand-up, etc.)
- 8 — Planning and feedback (Going from strategy to product roadmap to ready scope; How to effectively collect and incorporate feedback)
- 3 — How to get my organization started with Agile
- 3 — Collaborating with non-Agile teams
- 1 — Frameworks and approaches for scaling Agile
- 1 — Hybrid environments - waterfall and agile.

**THANK YOU!**

**LEARN MORE**

[parivedasolutions.com](https://parivedasolutions.com)