APPTIO

The Complete Guide to Building a Successful Agile Team

Contents

1 Introduction 3

2 Overview of Roles 5

- Product Owner PO
- Scrum Master SM
- Agile Developers

3 Building an Agile Team Isn’t Easy... 22

4 About Apptio Targetprocess 23

Introduction

In a digital world where market demands and customer expectations evolve and change rapidly, organizations often need to prioritize their IT investments and innovations to get their products to the market faster. A methodology is required which grants them the support to work towards customer satisfaction and efficiently deliver business value.

The Agile methodology gives companies frameworks that help them achieve:

- Speed and efficiency while delivering their competitive product to the market
- Organized project management opportunities and roadmaps
- Quality products and customer satisfaction
- Quick customer feedback

The initial step for an organization considering an Agile adoption or a group undergoing a full transformation is to have a clear understanding of their priorities, motivations, and possible challenges. **It begins with people.**

All successful Agile projects start with the team at the center. This means a clear identification and clarification of roles within your Agile teams. Role division is important to understand how a team should perform holistically and at all levels of delivery. These roles often change and evolve when scaling agile across enterprise.

To create a strong foundation, an in-depth examination of basic Agile roles is necessary. Moreover, understanding and carrying out the responsibilities of each role is important to ensure the success of Agile transformation and guarantee the establishment of Agile culture. This means placing people at the center.

Henry Ford once said:

> **“Coming together is a beginning**
> **Keeping together is progress**
> **Working together is success”**

This guide outlines the key roles in the Agile team, why they are critical as part of the team, and how they impact success.

Overview of Roles

Agile teams are made up of a variety of roles ranging from developers to quality assurance engineers to systems architects, and so on. Agile teams are often characterized as self-organizing—they receive challenges from team leaders and, within the boundaries and requirements provided, they develop a solution that fulfills the needs of the organization.

Self-organization does not mean that there is a lack of leadership. Quite the opposite. Despite the autonomy of Agile teams, there are clear roles for Agile team leadership.

For example, according to Scrum, three specific roles make up a team:

- Product Owner
- Scrum Master
- Agile Developer

Product Owner (PO)

Product Owner

Voice of the customer

Product Owner(s) (PO) are the key stakeholder of an agile team. They are the bridge between the development team, scrum master and other stakeholders.

Criteria

- Communicate the needs of both business stakeholders and users
- Drive the vision of the product development
- Align strategy with vision, budget, and work development

Responsibilities

- Owns the product backlog and the goal and makes them visible
- Optimises the value of work
- Ensures the developers understand what is required

Common Job Titles

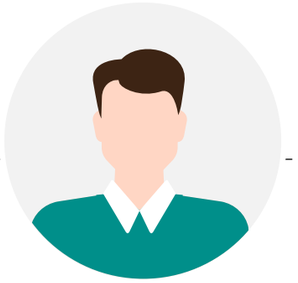
- Chief Product Officer
- VP of Product Management
- Group Product Owner/Manager

Product Owner

The product owner is the “voice of the customer” — or better — your key stakeholder. An organization relies on a product owner to help the business crystalize its vision and then knowingly, and confidently, convey that vision to the team. As a result, success relies on their effective engagement. A lack of someone in this role, or worse, someone with inadequate capabilities can spell disaster for any product.

There are many desired competencies for a product owner, but for them to be true to the Agile philosophy, they should largely share the concepts above and continue to build on them.

Let’s focus on the following three characteristics:



Product Owner

The Key Stakeholder:

In purer Agile implementations, the PO comes out of the business organization and is assigned to the team to act as the “voice” of the customer.

At many mid-to-large-sized companies, the business may not have the capacity to assign individual product owners to each team. In those situations, you may see the concept of a proxy business product owner, often from within the ranks of IT. Regardless of where they originate, the role is the same.

The PO remains responsible for the business content of the product development backlog; and even in cases where architectural features and stories are written, they remain the key owner. The PO must understand the purpose, value, and intent of every requirement committed to by the team.

Drives the Vision:

One of the key responsibilities of a PO is to drive the vision of the product’s development. The team aligns with the vision and trusts that the PO has a careful eye on not only what the business is asking for at the project’s start, but also the changes that are bound to arise during the product’s evolution.

A good PO understands the industries and markets that their business operates in, and when needed, can be relied upon to voice the details of the requirements and any changes to vision or direction that may occur.

A Good Communicator:

A PO cannot be effective unless they are a good communicator. Conveying and translating business concepts is a key ingredient needed for authoring effective and functional user stories, not to mention, prioritizing the important aspects of all product backlog items (PBIs). This awareness enables the PO to act as a collaborative agent in planning and coordinating all product releases. Of course, communicating a point is one thing, but having a true skill in bartering and negotiating with the team, the scrum master, and even other stakeholders is an important trait of a product owner.

The following 5 tips will enable a Product Owner drive downstream benefit and help the entire agile team operate in a true Agile spirit:

1 Keep the big picture in mind:

Backlogs might contain great ideas, but the team still needs to stay focused on the vision that is driving the product. When teams lose track of that, they start letting milestones slip and features become incomplete.

2 Write clear and crisp user stories:

A PO must know the why and who of every user story. This perspective enables teams to develop product features closer to what you want rather than guessing how the functionality might have to integrate with other stories and features, or who will be the end-user.

The product owner is normally the only role that can accept completed stories. That said, in many environments, special technical stories are accepted by architects and/or QA managers. Nevertheless, the PO should maintain control of the stories backlog: business, technical, and/or QA driven. It only makes sense that if a technical story is introduced into the backlog, the PO keeps a firm understanding of the why and reason behind the effort.

3 Embrace change:

The PO is a change agent on the team. Market knowledge, constant research, anticipating changes and understanding how these changes might affect the product, help the PO manage his/her team effectively.

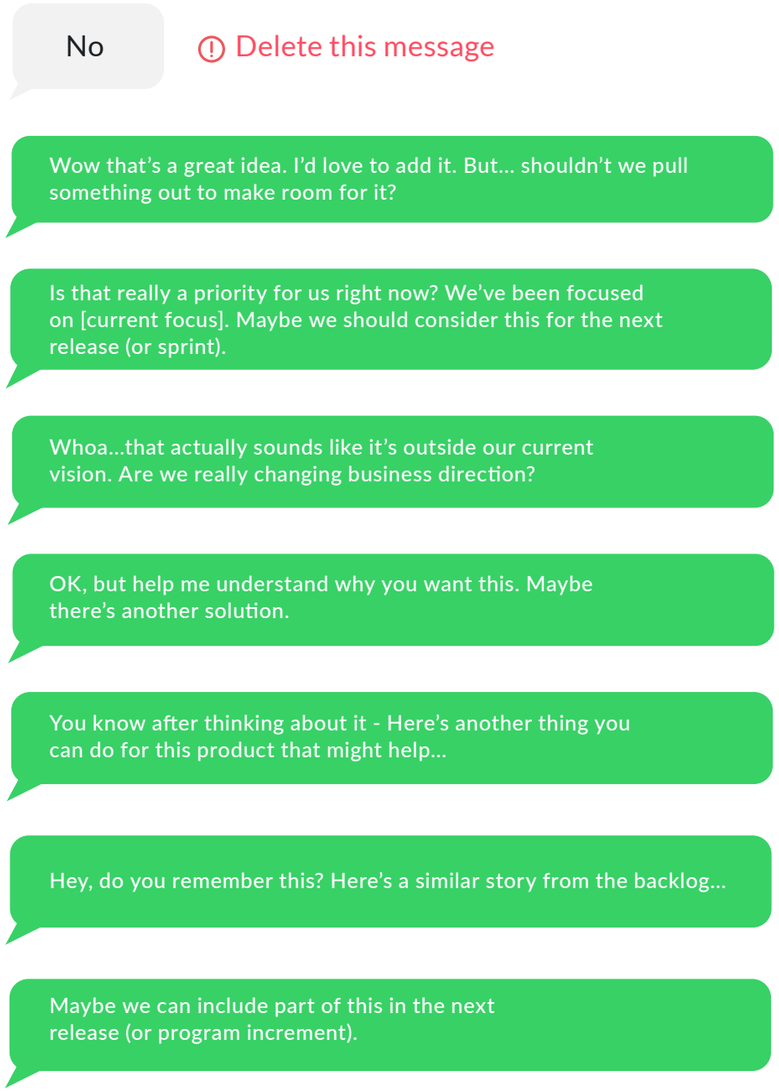
4 Split when splitting makes sense:

A great PO knows how to take a big idea with high-value, high-learning opportunities and break it into a core subset of essential user stories that can be quickly released and marketed. This often means taking a big user story and splitting it into several smaller stories.

5 Learn to say no:

The PO is trusted by the business leaders to make the right decisions about the direction of the product. The PO after accepting the responsibility needs to learn the art of saying “no” when it benefits the product.

Here are a few ways to say “no” when someone wants to add something to your backlog.



Note, while the PO tends to operate at the team level, in scaled Agile environments (like SAFe) you may also find product managers who operate at the program level and maintain a keen eye on the big product picture.

The **Product Manager**, similar to the product owner, is a customer-facing role owning the vision and roadmap(s) of a program — usually at a large or enterprise company. The PM concentrates on defining the strategy through market research, maintaining a solid understanding of company objectives, and building a reasonable plan based on the current state of the product.

This is critical for any Agile Release Train (ART) — a self-organizing group of Agile teams (typically 5–12) that plan, commit, and complete work together. The product manager drives the Program Increment (PI) planning process through to prioritized features and stories. At the same time, the product manager works with the product owners on the Agile Release Train (ART) to establish a plan of prioritized work for the teams.

Scrum Master (SM)

Scrum Master

A Leader Who Serves

Scrum Master(s) (SM) are the guides who help the team members complete their tasks by removing potential blocks. They drive progress and productivity of the team.

Criteria

- Accountable for creating a nurturing environment to motivate and coach the team
- Promote self-governance and team empowerment

Responsibilities

- Manages and organizes the Scrum framework
- Teaches, coaches, and mentors and facilitates events
- Principle accountability: maximizing the delivery capability of the scrum team

Common Job Titles

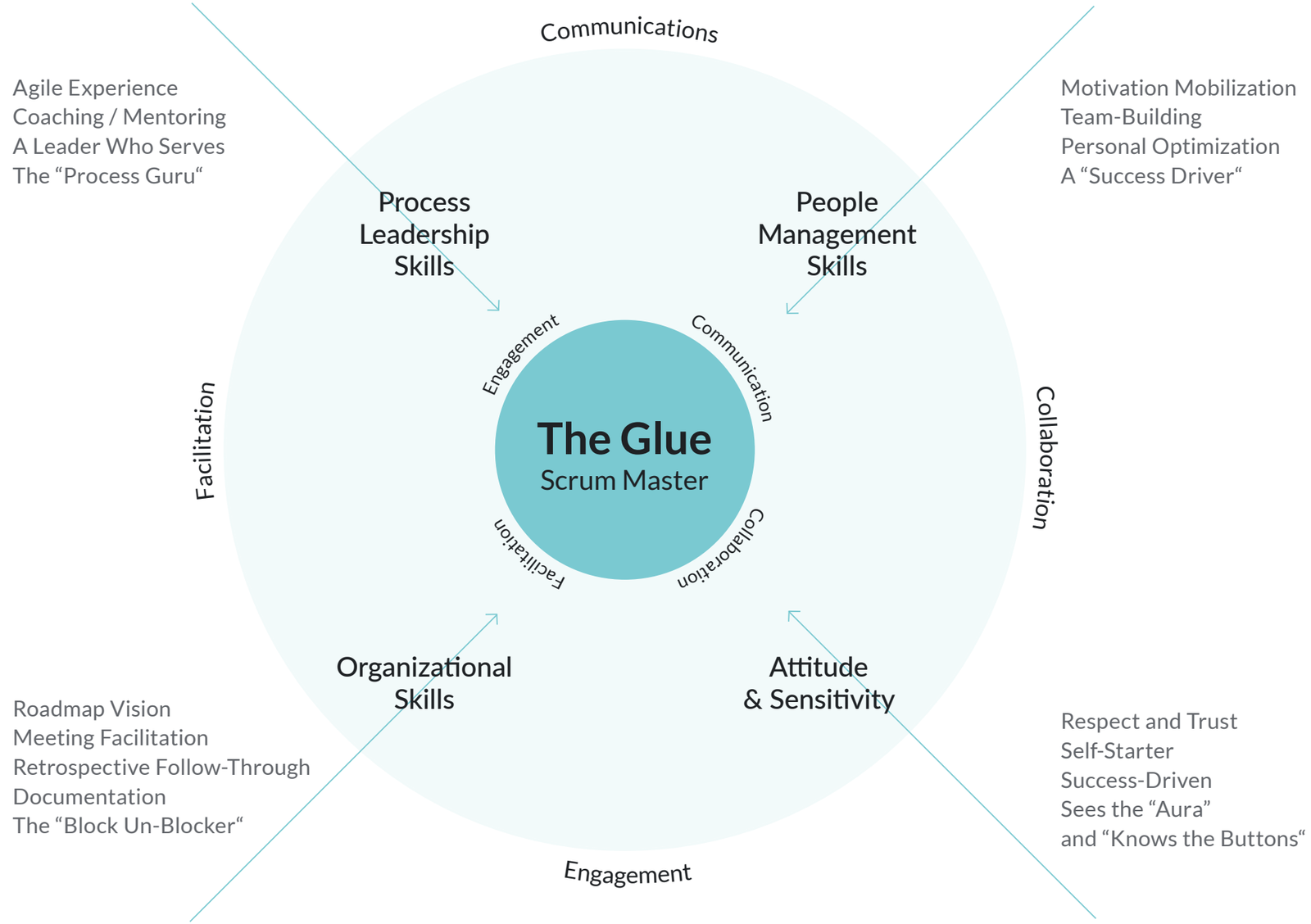
- Chief Scrum Master
- Agile Leader
- Release Train Engineers (Agile Release Train environment)

Scrum Master

A Leader Who Serves *is someone who inspires action in others by sharing power, putting the needs of others first, and helping people develop and perform as highly as possible.*

In many ways, this is how a scrum master coaches and guides a scrum team. A good scrum master is the “glue” that holds a team together.

For example, a scrum master might help team members complete their work by removing impediments that are potential blocks for progress. The bulk of the SM’s time is spent communicating with team members, helping the team self-organize, and providing the right level of governance in the application of Agile practices. They also work closely with the product owner to drive progress and improve productivity.


Communications
Agile Experience
Coaching / Mentoring
A Leader Who Serves
The “Process Guru“
Process Leadership Skills
People Management Skills
Motivation Mobilization
Team-Building
Personal Optimization
A “Success Driver“
Facilitation
Engagement
Communication
The Glue
Scrum Master
Collaboration
Facilitation
Collaboration
Organizational Skills
Attitude & Sensitivity
Roadmap Vision
Meeting Facilitation
Retrospective Follow-Through
Documentation
The “Block Un-Blocker“
Respect and Trust
Self-Starter
Success-Driven
Sees the “Aura”
and “Knows the Buttons“
Engagement

A successful scrum master must exhibit four main qualities:

- **Be a process guru:** A scrum master leverages Agile best practices and personal experience to lead and coach the team in the Agile process.
- **Motivate the team:** A scrum master is an inspirational team builder and acts as the team's rudder on the "boat towards success."
- **Unblock the blocks:** A scrum master follows through on retrospective outcomes, facilitates meetings, intervenes when necessary, and runs interference when distractions appear. They don't allow disruptions to distract their team. They are the "un-blocker".
- **Keep an eye on the big picture:** A scrum master is a proactive and enthusiastic team leader who has an overarching sense of what matters most. They promote self-governance and team empowerment to earn and maintain the team's respect and trust.

Let's break down these attributes in real life. Consider the following common activities for a scrum master:

Facilitate Meetings	▪ Assist the product owner in preparation for iteration reviews, iteration planning sessions, etc. ▪ Be an effective moderator by keeping meetings focused and productive. ▪ Facilitate effective retrospectives, planning, and closing sessions.

Manage Team Dynamics	▪ Be a confident Agile coach. Keep the team on course and identify and enforce rigidity of rules. ▪ Help the team make decisions by asking probing questions. Don't answer for them but assist in finding a solution. ▪ Foster self-organization principles. Maintain open channels of communication. Help reconcile or clarify objectives with the team and the product owner.
Always Be Learning	▪ Continue personal learning regarding Agile (visit user groups, attend conferences, read books, etc.). ▪ Provide constructive feedback to the team. ▪ Participate in Product Planning. ▪ Coach the product owner and team to write better user stories, product visions and to split as needed. ▪ Assist prioritization of product backlog items. Drive practices that effectively prioritize features and stories (WSJF – weighted shortest job first). ▪ Understand the product to stay familiar with the team's work.

Keep the Big Picture in Mind	▪ Be the key contact and promote continuous collaboration with external teams and stakeholders. ▪ Be an Agile change agent and help drive Agile adoption within the organization. ▪ Coach the team in Agile fundamentals and provide internal learning opportunities within the company.
Lead by Example	▪ Remind the team of their commitments, arrangements, and contracts. ▪ Help the team stay focused by acting as a buffer from external distractions and sharing issues and observations. ▪ Encourage the team and product owner to implement a suitable definition of done and a definition of ready.

With a clear understanding of the scrum master's role on a single team, let's expand on how we scale it within larger organizations. Similar to the Product Manager role, which scales the product owner, in SAFe we have the release train engineer (RTE) that scales our scrum master role.

In an environment where Agile Release Trains exist, an additional role is created for the **release train engineer** (RTE). This "chief" scrum master is responsible for facilitating the Agile Release Train processes, program execution, escalating impediments, managing risk, and driving program-level improvements. Additionally, RTEs are responsible for facilitating program events such as release planning, inspect and adapt, and the weekly scrum of scrums.

RTE responsibilities include:

- Establish the annual calendars for sprints, program increments, and all other Agile and SAFe artifacts
- Maintain proper communication with all the technical and non-technical teams involved
- Partner with other scrum masters to coordinate program increment (PI) execution — this includes tracking team progress and publishing accurate team metrics and progress report

Most often, RTEs come out of the program management office (PMO) or Agile management office (AMO) of a company. Although this seems like a natural progression for an existing program or project manager (or scrum master), the person needs to go through a meaningful mindset transition. The servant-leadership model encourages a philosophy that implies a comprehensive view of the quality of work and communication among the employees.

As you can see, the RTE plays an incredibly important role in helping ARTs achieve their goals. A resilient and persevering champion — one with a driven spirit for getting the job done through teamwork and association rather than through ego, heroics, or hysterics!

Finally, let's look at those people who do the actual development work — The Agile Developers.

Agile Developers

Agile Developer

Agile Developer(s) are people who build and complete the products or the design. They ensure the completion of user stories.

Criteria

- Accountable for delivering quality product
- Understand the scope of product to deliver work at the end of each sprint

Responsibilities

- Cross-functional, self-organising
- Responsible for software quality
- Principle accountability: delivering a valuable, quality product

Common Job Titles

- Chief Engineering Officer
- UI Designer
- Product Tester/QA Expert

Agile Developer

The developers are the people responsible for ensuring the completion of user stories. This cross-functional team consists of software engineers, architects, analysts, QA experts, testers, UI designers, etc. Typically, where the product owner is responsible for what is being built, the team is focused on how to build it.

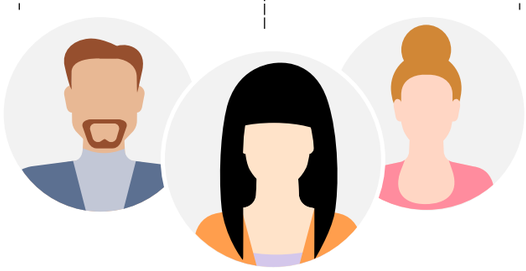
Developers must have the autonomy to determine how and when to complete their work. It's not unusual for them to discover that it has more work to do than originally assumed during the first couple of days in a sprint. And it is in cases like this that ongoing communication between the developers, product owner, and scrum master must remain intact.

Are continuously committed to finding ways to improve.

Every Agile developer must constantly reflect on what is or is not working to drive total quality improvements. For this reason, we should constantly promote the value of effective retrospective facilitation. The truth is that few improvement suggestions come from above or outside the team. For real improvement look within.

Are responsible for managing how the work is done.

It's easy for people with long years of technical experience, who are now in the scrum master or product owner roles, to think they should lead, dictate, or direct development. However, this must be avoided. Advice, opinions, suggestions are good, but never devalue a developer's authority to define, build, and test their work.



Developers

Are responsible for value and continuously seek to improve the quality of their deliverables.

A self-organized team of developers clearly understands that its success is not derived solely from completing tasks according to development standards. Rather, the developers know that stories must be fully developed, tested, and accepted by the product owner to be release-ready. Customers want working features in production, not half-baked stories sitting in the backlog. Therefore, discovering ways to improve the development pipeline is integral to our success and improves the customer perception.

Estimate the size and complexity of the work.

Another bad habit is when the scrum master, product owner, or other external technical lead disclaims or re-sizes the developer's estimation of the work. Realistically, we will never achieve 100% precision in our estimates, but the estimate of the one doing the work is always more accurate than that of an outsider. There's no problem with questioning or probing a bit to refine estimates, but overall, the developer's estimates must hold.

Determine the technical design in their area of expertise within architectural guidelines.

The developers are comprised of either local experts or people who have direct lines into the architectural leadership of the organization. Many teams may even have architectural representation embedded. Part of the Agile movement recognizes how inputs from the developers must interact with the architectural standards or guidelines of the enterprise. This is a healthy dynamic that provides recommendations to improve the product's architectural runway.

Commit to the work they can accomplish in an iteration.

After every sprint planning session, the developers commit to the work. This happens after they review the estimates, create and assign tasks, and set a sprint load agreement with the product owner. The scrum master or product owner does not commit on the team's behalf. The commitment has two parts as it acts as a contract between the developers and the business. The parts are: Team members will do all in their power to fully deliver upon their commitments. If issues or blocks arise that jeopardize those commitments, the business will do all in its power to work with the team towards an agreed-upon solution. This may mean helping the team re-scope or re-prioritize the work to maintain forward progress and continuous value delivery.

Building an Agile Team Isn’t Easy...

The roles and responsibilities listed here cannot be successfully performed unless a team learns to collaborate properly and adequately. This happens when open channels of communication are available and used, sound vision and product context are provided by the product owner, and effective process leadership is facilitated by the scrum master.

There isn’t an “easy” button that you can press to guarantee the creation of a great Agile team. It takes patience, diligence, and quite a bit of determination. Great Agile transformation leaders recognize that change is always difficult, and it takes time to create a self-organizing team that is empowered to confidently assume ownership of projects. With the right distribution of duties across roles and team members, we create an environment that proves our worth to the business. It demonstrates that we understand our actions serve a purpose and that all outcomes impact not only the current project but the company’s direction as a whole.

About Apptio

Apptio gives you the power of trusted, actionable insights to connect your technology investment decisions to drive better business outcomes. Whether you need help analyzing the true cost of the cloud, optimizing your technology spend, or communicating IT's value to the business, Apptio can help.

Our product, Apptio Targetprocess empowers businesses to adopt and scale agile across the enterprise. Connect teams, products, and portfolios to business objectives using SAFe, LeSS, or your own framework to achieve enterprise agility.

Start a Free Trial

@ 2022 Apptio, Inc. All rights reserved. Trademarks and logos are the property of their respective owners. A2954 V2205-1