Product Team as a Service

The principles of **PTaaS** and the positive outcomes you can expect.

A Thought Leadership Whitepaper

clear object

ClearObject Inc. | 8626 East 116th Street, Suite 300 | Fishers, Indiana 46038

+1 (888) 850-2568 Toll Free | www.clearobject.com

Publish date 2/2020, version 1

"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 (non-bolded) items on the right, we value the items on the left more."

-Manifesto for Agile Software Development

When the Agile Manifesto was published in 2001, it established the principles of a whole new methodology for project management in the software industry. It was a new way of thinking that opposed traditional waterfall approaches, and it proved to be ideal for IT development projects.

All these years later, the Agile Manifesto remains the holy grail of software development — with one exception: It still questions traditional service contracts.

We've written this whitepaper to introduce a new means of creating contracts with customers in a way that fits Agile projects. The technique is one in which we:

- Invert the fixed and estimated parts of the golden triangle
- Introduce checkpoints and exit points in the contract
- Define Agile roles
- Diffuse contract risk

Our reasoning? To promote an understanding that the most efficient way of bringing flexibility to a contract is to base it on a flexible and mutually shared vision of the customer-supplier relationship. Which is where Product Team as a Service, or PTaaS, comes in.

- Why PTaaS?
- What is PTaaS?
- What key elements should appear in a PTaaS contract?
- How ClearObject delivers to the PTaaS model

Introduction

Understanding the problem

The Agile approach for software development was born from a simple observation: In most cases, at the beginning of a development project, the customer didn't know what the final deliverable would be or how it should be built. Finding a way for the customer's needs to evolve along with the project therefore became essential, as did facilitating the interaction between the customer and the development team throughout the development process.

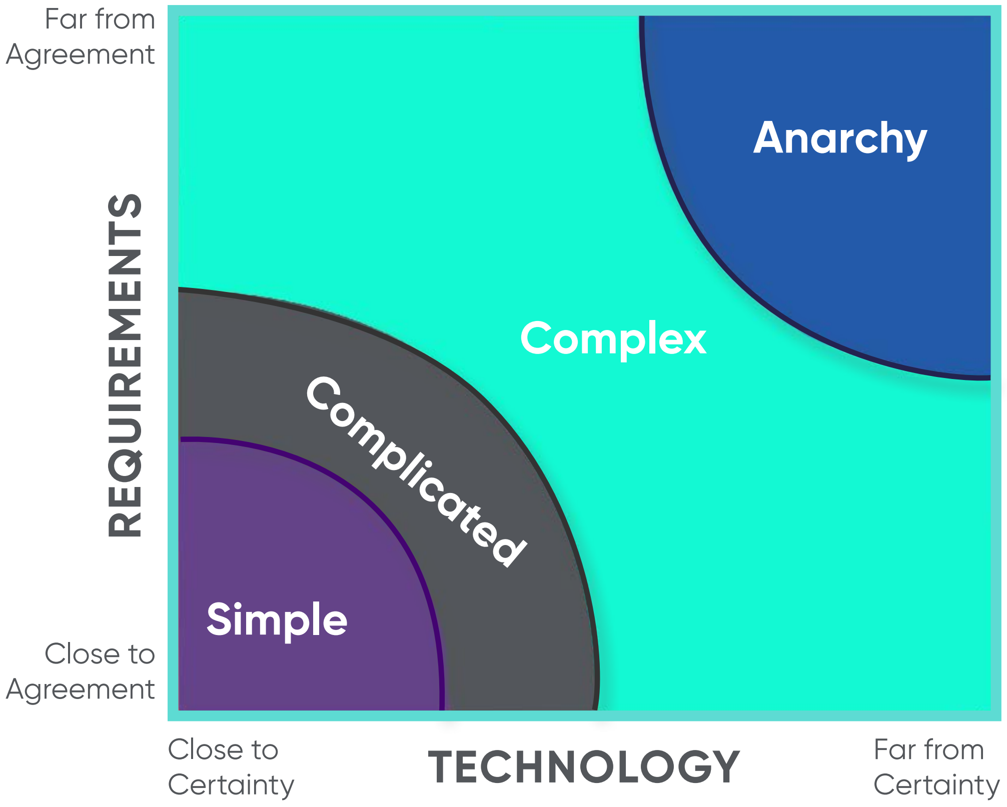
In a typical contract for a software development project, the aim is to ensure that both the supplier and the customer, seen as businesses within a business, agree on four associated details:

- The deliverable (product or service)
- A payment, for example by distributing a portion of the organization's budget
- The project's start date and duration
- Accountabilities

Conversely, the core principles of Agile methods are to prioritize customer satisfaction during the development process. This includes the early and continuous delivery of valuable software — working software delivered frequently (weeks rather than months) and reviewed by the customer for improvement — as well as welcoming customer changes in requirements, even in late development. The result is that neither the customer nor the developer knows exactly what the final deliverable is when the contract is signed.

So how can such a contract be formulated without knowing what will be delivered? Isn't the deliverable the core objective of a software development project contract?

Formulating a project contract within PTaaS

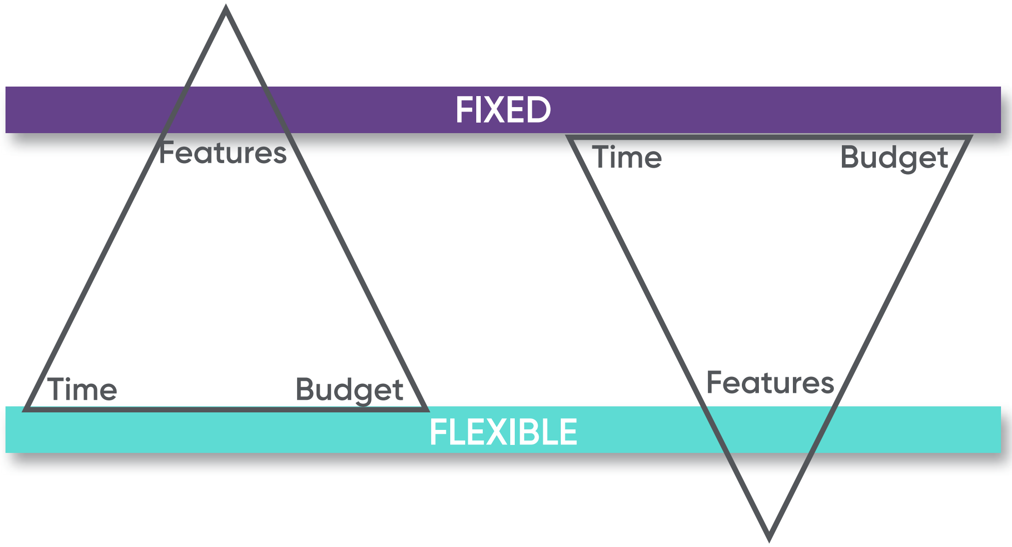


The Agile approach finally takes hold

The Agile approach for software development has gained significantly in popularity the last few years. In a 2016 survey of 601 software developers and IT professionals (https://techbeacon.com/survey-agile-new-norm -Jeremiah, J.), two-thirds of the survey's respondents described their organization as being either "pure Agile or leaning towards Agile." Measured against the prior traditional waterfall approach for software development, this survey was one of the more recognized signals that Agile Software Development was finally becoming the new standard — 15 years after the Manifesto for Agile Software Development was first issued.

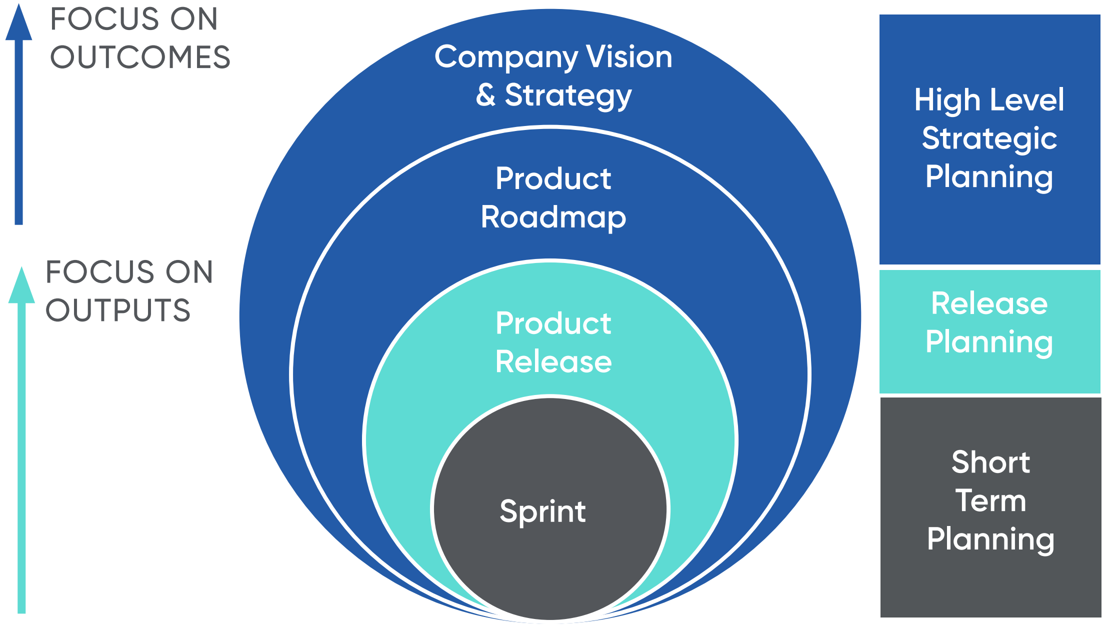
As Agile projects have proven to increase software quality and customer satisfaction, development teams have steadily applied the approach to external as well as internal projects.

PTaaS inverts the fixed & estimated parts of the Golden Triangle



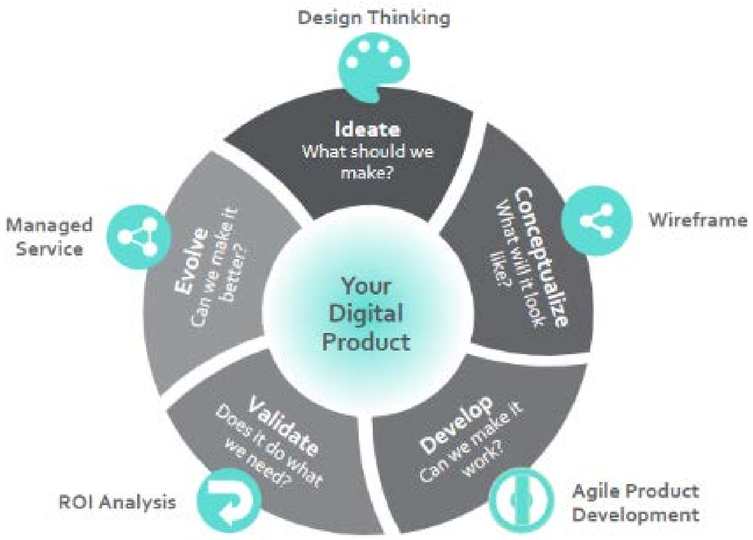
Product Thinking

Understanding a new way of thinking.



The new way of product thinking encompasses all phases of the software development journey, as shown here. Yet where the thought process begins for a new product — especially in the mold of today's connected digital products for IoT and data analytics — depends on the customer's current situation. For example, if the customer has already chosen a solution to build and knows why they're building it, the thought process for development could reasonably start at a Wireframe stage to conceptualize the final solution (graphic below). Or if the customer has chosen a solution to build but hasn't fully thought through why they're building it, the better starting point would be Design Thinking to ideate what the final deliverable should be.

ClearObject's belief is to ensure that, as we're working with you to build your solution, we can tie success directly to why the solution is being built. This helps us provide more value and metrics for the solution we deliver. As importantly, you avoid building a solution that has no discernable positive outcome, or value.



ClearObject Digital Product Process

IoT Analytics Product Design, Development, Validation & Evolution

The Principles of PTaaS

Product Team as a Service to drive positive outcomes

As a customer, trusting the firm and development team you work with to develop your new product is vital. The principles of a PTaaS contract create a shared confidence that we're building the solution you envision. Under the framework of a PTaaS contract, this can mean starting with smaller, mutually manageable engagements such as a design sprint and discovery sessions to set the precise course for development. PTaaS also helps both sides identify further opportunities for your solution, with the flexibility to extend your PTaaS contract should you decide to build accompanying products.

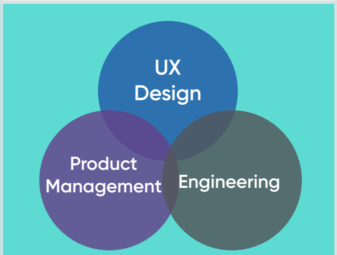
Overall, our PTaaS offering is based on the following principles.

Getting Started is more important than Being Right.
Rather than meticulously planning every detail of your solution and strategizing on the best way forward, take a smaller step to get started. You'll learn quickly and avoid endless push-back.

Scope Doesn't Creep, Understanding Grows.
When it comes to being innovative, you're likely trying to solve a problem using a new solution. But by the definition of "new," you won't know what you don't know. Again, rather than trying to plan an entire solution path, just get started and learn as you go. This is the best way to grow your understanding of your solution's final outcome.

Your product will not be perfect for everyone.
No matter how much time you spend, the product you create will never please every user 100% of the time. Instead, set up your product to learn from users and continue to develop and improve it as time goes on with iterations to the product.

As the adage goes: Build the Right Product. Build the Product Right. And build the Right Product Fast.

[YOUR LOGO HERE]	You focus on the "WHAT"	**Strategy + Vision** *You tell us what you want to achieve (Outcomes)*	**Domain SME** *You help identify risks & verify assumptions*
clear object PTaaS Product Team as a Service	 We focus on the "HOW"	Product Delivery Team: (dependent on solution) ○ Product Owner ○ Development ○ Full Stack ○ Back End ○ UI/UX ○ Embedded ○ Machine Learning ○ Scrum Master	Technology SME • Cloud • IoT • Data Analytics • Machine Learning • Product Thinking • Design Thinking • Managed Services

Contracts

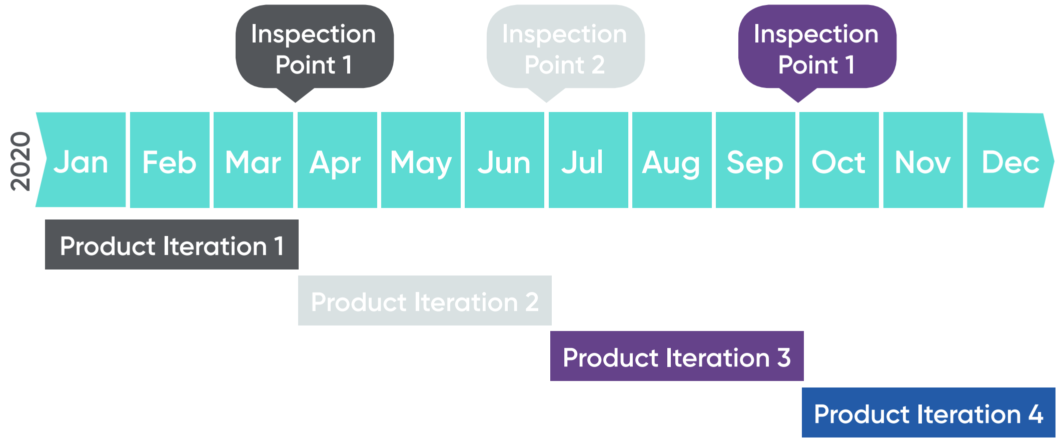
"How can we establish a contract without knowing what will be delivered?"

This is a fair question. And a common one. In line with PTaaS and the Agile Software Development approach, ClearObject has found Incremental Delivery contracts to be the most effective type of agreement for customers and their development projects.

Incremental Delivery contracts
Incremental Delivery contracts are structured with regular inspection points in which the customer can decide to continue or discontinue the development of their product. In stopping development – "We approve the model… it's good to go." – the customer can push the product into production and save the remaining balance of the contract. This style of contract works quite naturally for Agile teams because they simply work in an iterative fashion until the point of inspection.

The following chart is an example. Consider a year-long contract that specifies inspection points each quarter; an Agile team working with 2-week sprints would work for six sprints while demoing their work to the customer at the end of each 2-week sprint. At the end of the 6th sprint, the customer then decides to either continue their product development, or not. This avoids contract lock-in and provides the customer an off ramp in case they decide to move in a different direction than PTaaS or their current development project. The customer can adjust the intervals between inspection points as they see fit.

Incremental Delivery Contract



Delivery

How does ClearObject deliver PTaaS?

Think of mapping a road trip from a starting point to an intended destination. The starting point is certain, although the mapped route to the destination might not be – the driver can't be sure until they get to where they want to go. That is, they must test the outcome (destination) against the forecast (the map).

The delivery of PTaaS works the same way, in accord with the following three elements of forecasting:

1. PTaaS provides a statement about your development project's future outcome or event,
2. It provides an associated statement about the level of project uncertainty, although not guaranteed,
3. And it describes a way of eventually testing the actual outcome of your project against its forecast.

Common Questions

We expect you to have questions

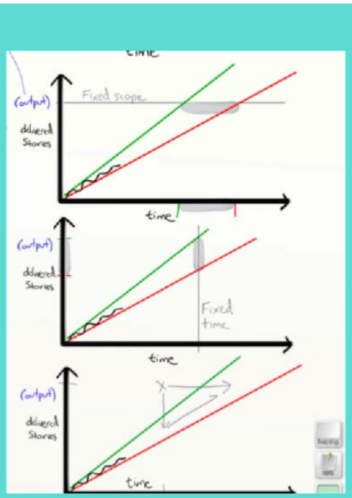
We expect you to have questions, as well as concerns. (We'd be concerned if you didn't!) For today's digital products, ClearObject's software development projects have let us know customers primarily want to:

- Understand when a feature or project might be done, including how progress is tracked — the #1 question (see the graphic that follows this list).
- Better understand the size of a feature or project with minimal effort.
- Accurately determine if their project's scope is too extensive, "What can I get by or do without when…" The answer in this case often lies in seeing options and making trade-off decisions earlier in the development process.

#1 Question - When will this be done?

Typical Method
How long = size/velocity
= 300 stories / 15 stories per Sprint
= 20 Sprints
= 40 weeks @ 2-week Sprints

Assumptions
- You know all the stories ahead of time
  - No Scope Creep
  - Stories split to appropriate size
- You know exactly how much you can get done each sprint
  - No change in performance
  - No vacations
- No bugs will be introduced
- Project will start at 100% efficiency



Source:
www.youtube.com/watch?v=502ILHjX9EE

Suggested Method = Forecast
How long = size/velocity
= 300 to 600 stories
/ 5 to 15 stories per Sprint
* 1 to 3 split rate

= 52 Sprints with 90% confidence on delivery
= 44 Sprints with 50% confidence on delivery

Assumptions
- Stories will be split with a given range
- There will be a range of velocity for delivery
- Stories remaining to deliver

Tools to help answer common questions

For these and other common questions from customers, ClearObject utilizes a set of automated tools to help answer them: Story Count Forecaster and Throughput Forecaster. Both applications are the brainchild of Troy Maggenis at Focused Objective, who developed these and other tools for Agile forecasting and data. (Go to https://www.focusedobjective.com and check out their page for Free Tools/Stuff.) We like to use these tools when we meet with you to discuss PTaaS and answer your specific questions.

The rest of this section is a close-up view of the Story Count Forecaster and Throughput Forecaster, including results readouts for two commonly asked questions.

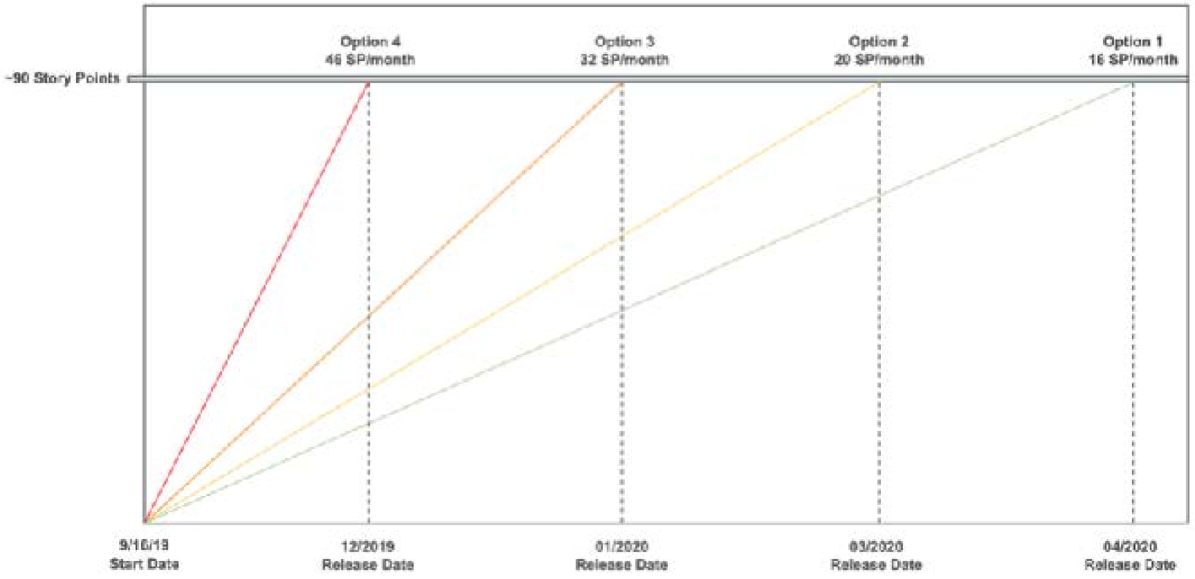
STORY COUNT FORCASTER

Question: How can I estimate the size of a feature or project without analyzing every piece of work?
Theory: The "size" patterns of randomly sampled features; will persist through all other features. Analyze a few and compute for the many.

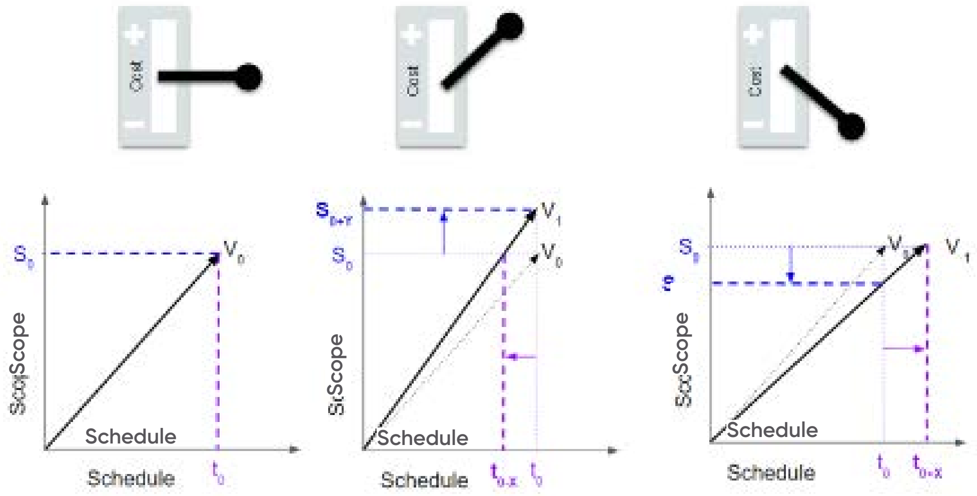
THROUGHPUT FORCASTER

Question: How can I estimate the amount of time it will take to deliver a feature or project?
Theory: Using a range estimate or actual team delivery rate, calculate how many time periods to complete delivery.

Story Count Forcast Results:
Estimating the size of a feature or product



Throughput Forcast Results:
Estimating the time requried to deliver a feature or product



Forecasted release schedule (t_0) given once scope (S_0) and velocity (V_0) established.

Increasing costs increases velocity (V_1), thus either delivering more scope (S_{0+Y}) in the same time(t_0), or delivering the same scope (S_0) earlier (t_{0-X}).

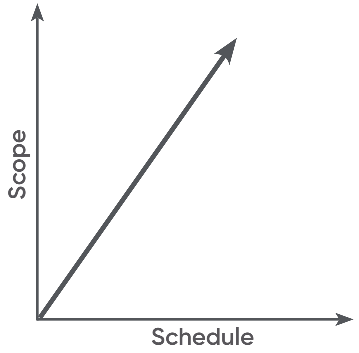
Decreasing costs decreases velocity (V_{-1}), thus either delivering less scope (S_{0-Y}) in the same time(t_0), or delivering the same scope (S_0) later (t_{0+X}).

If cost is an issue
- Decrease velocity (costs) and delay release
- Reduce scope and keep same release date

If schedule is an issue
- Increase velocity (costs) to deliver more at time of release
- Reduce scope and keep same release date

If scope is an issue
- Increase scope and delay release
- Increase velocity (costs) and keep same release date

If understanding of scope is an issue
- Match velocity with understanding – Lean



Delivering PTaaS

- Product Vision
- Business & Product Objectives
- Opportunities / Opportunity Solution Tree
- Solutions
- Story Maps
- Backlog Refinement
- Version Reports
- Velocity Reports

Product Vision

A Product Vision clarifies why you are bringing a product to market, and what its success will mean to the world and the organization. This vision should inspire purpose and clarify what the product is offering and what problem it is addressing. For example:

For (target user)
Who (want to do ABC)
The (product)
Is (describe what the product is)
That (does XYZ)

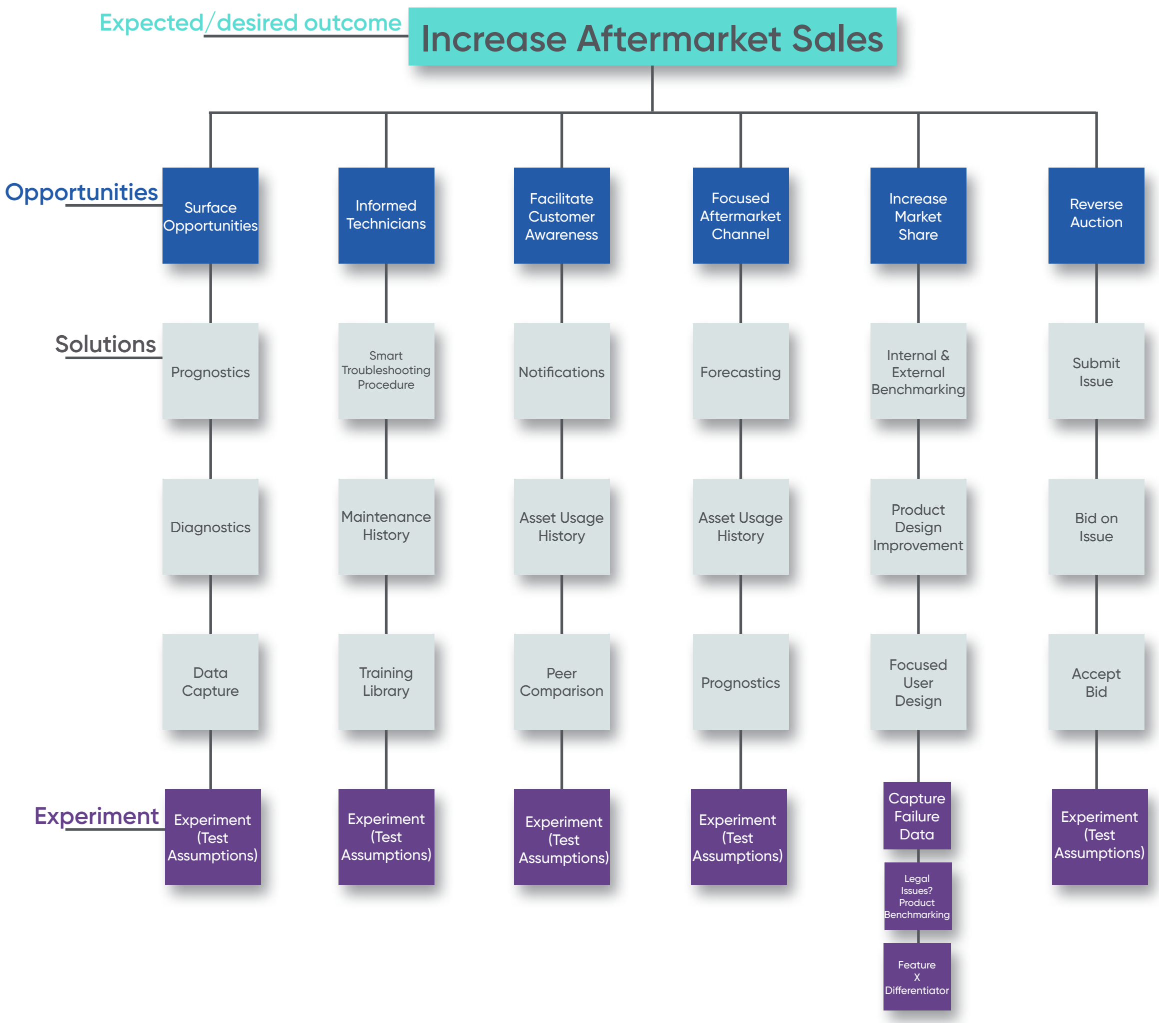
Business Objectives & Key Results

Identifying business objectives that tie to your Product Vision is critical to making that vision a reality. Without specific objectives along the way, that vision will be challenging to implement — there will be too many different product, technology and business directions to get there.

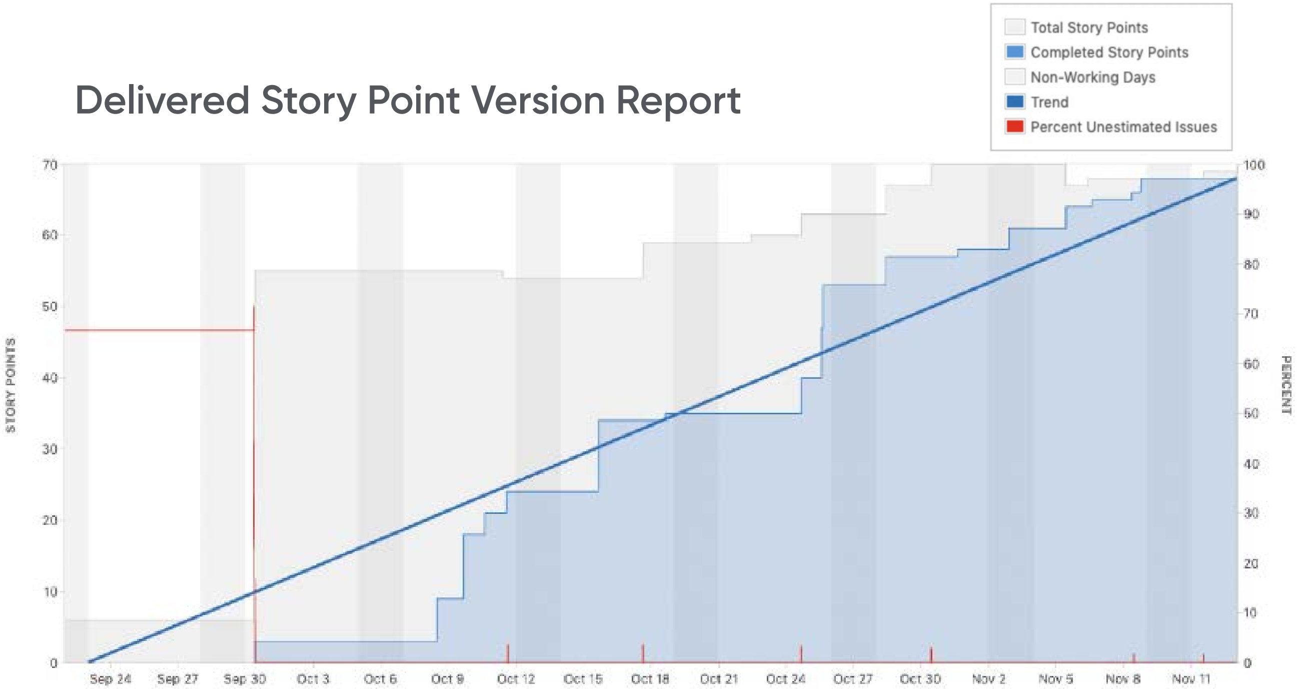
Objectives & key results (OKRs) are a great way to pair your business objectives with success criteria. The premise of the OKR framework is that objectives are specific qualitative goals, and key results are quantitative measures of progress toward achieving these objectives. "Need customer support in order to create Key Result measure."

Objectives generally fall under at least one of these three areas: Sustainable Value, Growth, Profit

Opportunity Solution Tree

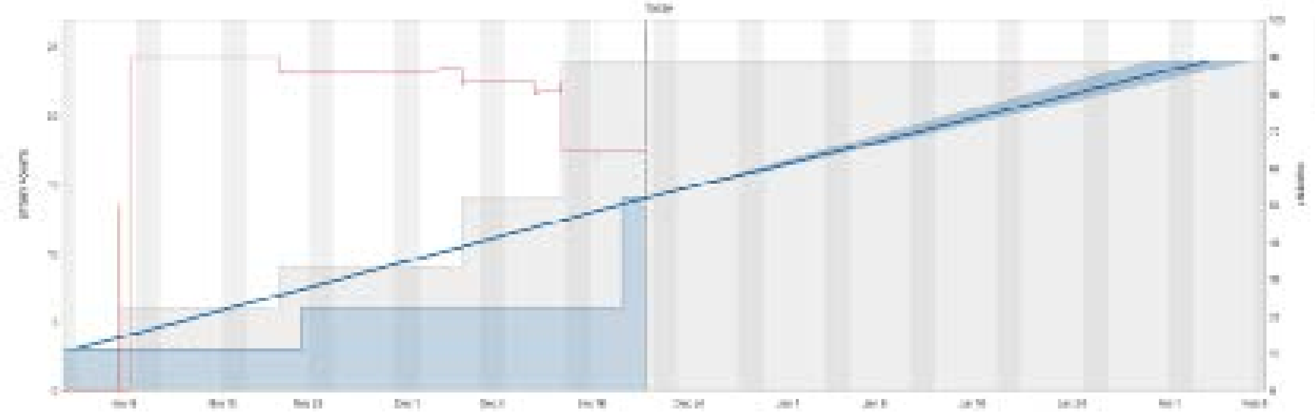


Delivered Story Point Version Report



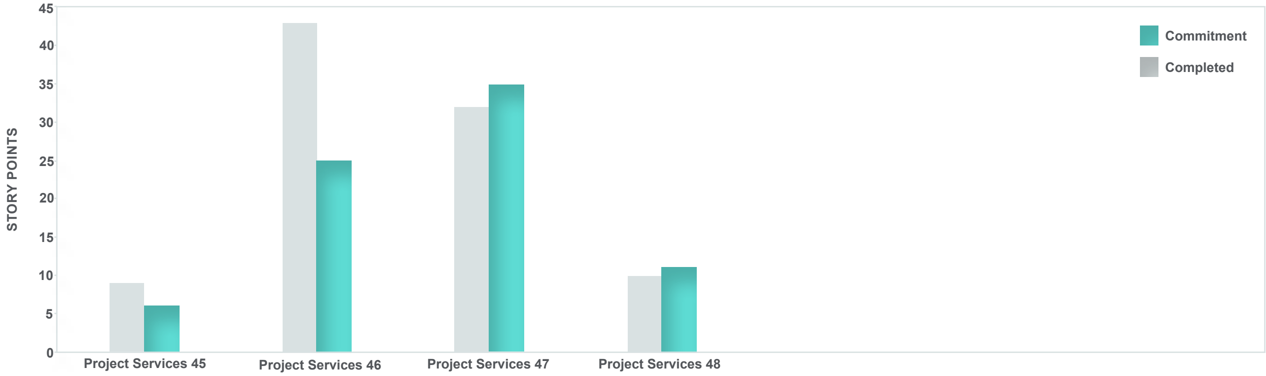
Source: ClearObject

In Progress Story Point Version Report



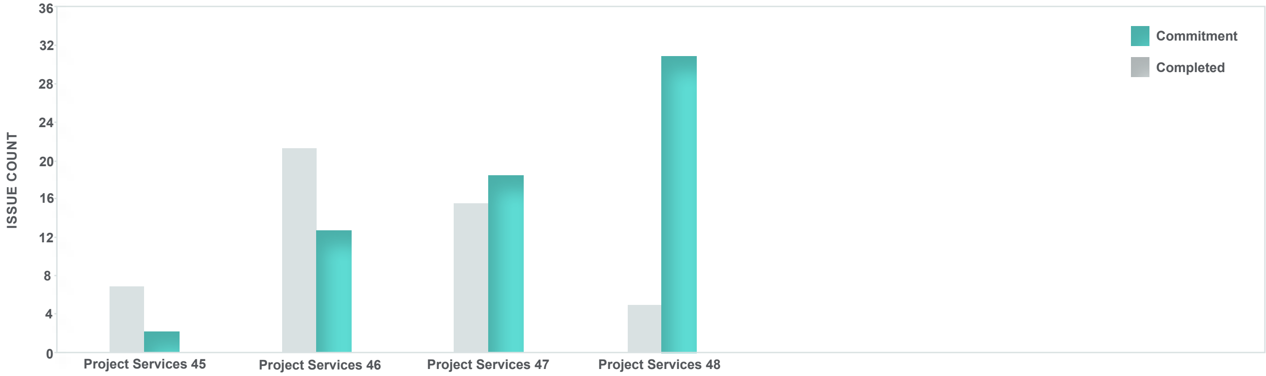
Source: ClearObject

Velocity Chart: Story Points



Source: ClearObject

Velocity Chart: Issue Count



Source: ClearObject

Are you ready for PTaaS?

Let's find out

○ Is innovation in your organization's M.O.?

- ✓ Digital transformation is everywhere… and requires new levels of sophistication.
- ✓ Data and analytics and software and the cloud are driving everything. Technologies like IoT and AI are simply picking up the pace.
- ✓ With the world becoming a connected platform, digital products and connectedness continue to transform how businesses and entire industries work.
- ✓ Innovation must constantly accelerate, without compromising product quality.

Is your organization prepared to think differently, view software development in a whole new way, and learn through ideation and trial & error as well as success?

○ Are you Agile?

- ✓ Agile Software Development has changed how software and product development projects are managed. Will waterfall approaches soon be a thing of the past?
- ✓ The principles of Agile put product users front and center, and put technology products under a long-term lens.
- ✓ Agile is a mindset. To be successful, the entire organization must adopt it.

Does your business put individuals and interactions over processes and tools… working software over comprehensive documentation… customer collaboration over contract negotiation… and responding to change over following a plan?

○ Do you see contracts only in black & white?

- ✓ The process of developing technology is never straightforward. Creating new software and digital products is a winding road, and flexibility is key – contract included.
- ✓ The new product you envision might not even be fully realized until it's finally delivered.
- ✓ The project contract itself must therefore promote a fluid, iterative approach to the development process.

In line with PTaaS and Agile Software Development, Incremental Delivery contracts are the most effective for product development projects. Such contracts put customers in control of development inspection points, timelines, decisions, and costs.

○ Are you willing to rethink your strategy?

- ✓ For any organization, transformation is rarely easy. Especially digital transformation.
- ✓ Look closely at the role technology plays within your business. Does it dictate strategy? If not, it's time to reevaluate.
- ✓ Who steers such decisions?

Product Team as a Service is a change in thinking. Think innovative high-value digital solutions, products that customers really want, and services that differentiate your business. Think of the future of software product development, which is already here.

Contact us

New digital product idea? Contact us. For your team and ours, PTaaS is a creative environment in which to develop your idea together.

www.clearobject.com/contact

About ClearObject

ClearObject is a digital transformation leader in Internet of Things (IoT) Engineering and Analytics. As IBM Watson IoT and Google Cloud Business Partners, we deliver global embedded software development environments for our customers, and design and deliver unique data analytics digital products that help them recognize the value of their data. Our objective is clear: help the world's best companies build intelligence into their products and gain intelligence from them. The future is clear. Do you see it?

About the Authors

Derek Bleyle is Product Manager, IoT and Data Analytics at ClearObject, where he was also previously a Product Owner. Including his stints as a project manager and service engineer for Belcan Engineering and as a service delivery program manager and service engineer for Rolls-Royce — a "pre-eminent engineering company focused on innovation and world-class power and propulsion systems" — he has worked in digital product development for going on 12 years. Derek holds a BS in Economics & Mechanical Engineering from Butler University and an MS in Engineering Technology from Purdue University. He is a Certified Scrum Product Owner (CSPO) and is additionally certified in the Fundamentals of Engineering by the NCEES.

PTaaS, Agile Software Development, Opportunity Solution Trees and working with ClearObject's customers for timely and successful project outcomes are just some of Derek's areas of expertise in the world of IoT and intelligent data analytics.

Matt Markeiwicz is a Product Owner for IoT and Data Analytics at ClearObject, where he has worked since 2018. He was a Project Manager at ClearObject prior to joining the company's Product team. In helping deliver cloud-based solutions utilizing machine learning and IoT technologies, Matt works as a liaison between customers and ClearObject's development group to ideate, define, break down, and prioritize features in the products being developed. Matt is graduate of the University of Dayton with a degree in Entrepreneurship, and is also a Certified Scrum Product Owner (CSPO).

clear object

www.clearobject.com