POLARIS
SOLUTIONS

The Little Book of

Agile Metrics

Contents

AGILE METRICS AT EVERY LEVEL

In the last decade, Agile has evolved from a framework used by development teams to a topic of discussion across entire organizations, including at the board and shareholder levels. Despite all the attention, Agile efforts suffer from false starts and unexplained stalls all too often. A surprising number of these issues sprout from metrics.

Polaris was founded with a mission to share the joy of successful software development projects with everyone. To that end, this book is designed to be used by **teams, Agile leaders, and executives** to ensure success with Agile. The metrics we cover:

☑ Represent the most critical Agile measurements, collected from Polaris' team of experts

☑ Have been field tested in real world projects, including both software development and organizational transformation efforts

☑ Have proven to enhance software development, collaboration, and business and IT transparency

How to Use This Book

Agile methods are frameworks and processes that provide guidance for managing teams and developing software. They aim to deliver value to users quickly, with feedback loops to help ensure delivery of the right product at the right time.

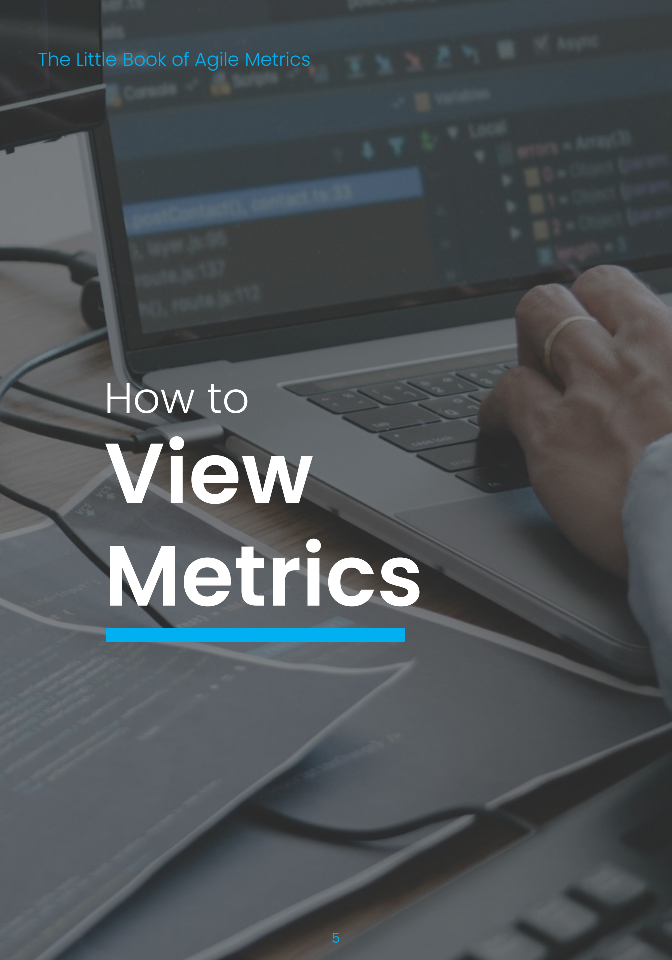
Examples
SCRUM, EXTREME PROGRAMING, CRYSTAL CLEAR, FEATURE-DRIVEN DEVELOPMENT

Agile metrics serve as a compass to keep teams on track and enable data-driven conversations about ROI, value, and Agile maturity. *A metric is* a quantification that can be used to narrow in on something we could change, determine if we are adding value, and/or test hypotheses. *A metric is not* a flawless and unbiased representation of team productivity, software quality, or delivered value.

Examples
THROUGHPUT, CYCLE TIME, WIP LIMIT, IMPEDIMENT AGE

This book covers the most critical Agile metrics that can guide your software development and delivery practices.

QUICK TIP: AVOID GOING RIGHT TO OVERWHELMING YOUR TEAM WITH DASHBOARDS FULL OF METRICS.



How to View Metrics

DON'T Jump to Conclusions

During World War I, British Army officials became concerned about head injuries and decided to exchange cloth headgear for metal helmets.[1]

After implementation, there was actually an increase in soldiers hospitalized with head injuries.

Looking at this metric in a vacuum, we might jump to the conclusion that metal helmets resulted in more head injuries. In reality, the helmets allowed more soldiers to survive head injuries.

What does this have to do with Agile metrics?
We live in a complex world and often use biases to interpret it. We must be aware of these biases when viewing metrics.

Next, we'll discuss two out of hundreds of known biases and psychological phenomena that can cloud a clear view of metrics...

[1] *Blast wave protection in combat helmet design, Ohio State University, 2018*

How to View Metrics

Goodhart's Law

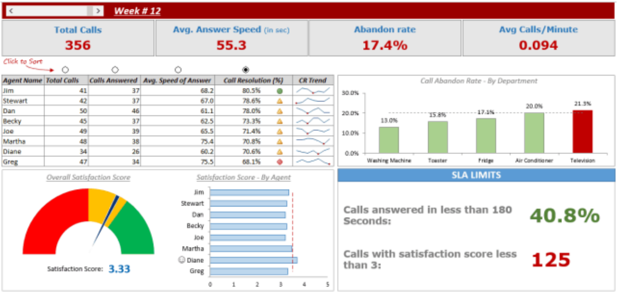
> "When a measure becomes a target, it ceases to be a good measure."
>
> –Marilyn Strathern[2]

Goodhart's Law[2] suggests when we optimize for a specific objective, people will work to obtain that objective regardless of the consequences.

Example A Call Center sets a new KPI for low call times, and reps are coached to bring their call time numbers down. After a month, one rep significantly decreases his call time.

When pressed on how did it, he says he hung up on every third caller within 30 seconds.

The lesson? Be careful what you measure, because you just might get it. What metrics are you using that could encourage your team to "hang-up prematurely" on customers?



Week # 12

Total Calls	Avg. Answer Speed (in sec)	Abandon rate	Avg Calls/Minute
356	55.3	17.4%	0.094

Click to Sort ○ ○ ○ ●

Agent Name	Total Calls	Calls Answered	Avg. Speed of Answer	Call Resolution (%)	CR Trend
Jim	41	37	68.2	80.5% ●	
Stewart	42	37	67.0	78.6% ▲	
Dan	50	46	61.1	78.0% ▲	
Becky	45	37	62.5	73.3% ▲	
Joe	49	39	65.5	71.4% ▲	
Martha	48	38	75.4	70.8% ▲	
Diane	34	26	60.2	70.6% ▲	
Greg	47	34	75.5	68.1% ◆	

SLA LIMITS

Calls answered in less than 180 Seconds: 40.8%

Calls with satisfaction score less than 3: 125

[2] Goodhart's Law Rules the Modern World, Bloomberg, 2021

How to View Metrics

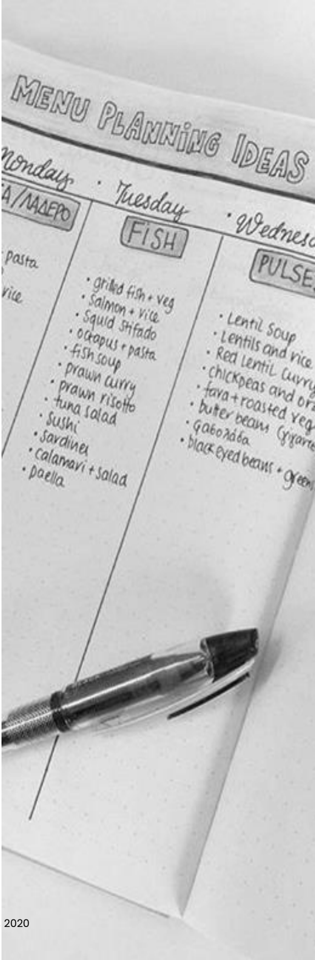
Hawthorne Effect

The Hawthorne effect[3] occurs when people change behavior because they're aware that of being observed.

Example People asked to record what they consume each day are likely to lose weight, regardless of diet. Knowing that someone is paying attention makes them more aware of what they eat, often leading to better choices.

The Hawthorne effect requires us to be careful not to draw false cause-and-effect conclusions based on targets and single measurements.

Instead, we should ask questions and look at a combination of metrics.



[3] *The Hawthorne Effect and Behavioral Studies*, Verywell Mind, 2020

10 Questions to Ask Before Choosing Metrics

To avoid bias and false conclusions, answer the following questions before settling on a metric.

1. Who is the metric for?
2. What is the goal of the metric? Is it achieving its goal?
3. What information can be misleading or give an illusion of progress?
4. What complimentary metric can we use to balance this metric out?
5. How can you game and/or manipulate the metric?
6. What information about a team's progress is useful to see visually?
7. Are you using the metric for the intended purpose, or do you need to revisit?
8. When will you stop using the metric (e.g. when a goal has been reached)?
9. Do you have any metrics that help drive change or motivation?
10. Are you tracking any metrics for the wrong reasons?

How to View Metrics

Metric Viewing Principles

Needs to be consumed by the team

- There isn't an easy way to measure productivity.
- Delivering valuable software is the primary measure of success.
- Metrics are a catalyst for conversations.

Needs to be surrounded by conversations

- In isolation, spreadsheets and presentations don't tell the entire story.

You get what you measure

- If we're only interested in showing how hard we've tried to meet a goal, traditional metrics will serve us well.
- Unfortunately, this creates "perverse incentives" that detract from the larger goal of delivering working product quickly.

Should be part of an investigation/experiment

Each metric tells a story that can be used to solve a problem:

- What problem are we experiencing?
- What experiment can we run to try and improve it?

Agile Metrics

at Every Level

Team-Level Metrics

- Velocity
- Say:Do Ratio
- Production Cycle Time
- Work in Progress (WIP)
- Story Churn
- Sprint Goal Success Rate
- Work vs. Wait Queues
- Throughput
- Escaped Defects
- Deployment Time

Team-Level Metrics

Velocity

WHY

Velocity tells us when something can be delivered. It represents the predictability of the team's estimates and ability to complete work from sprint to sprint**.** Stability in this metric drives trust.

WHAT

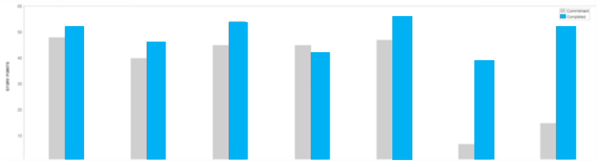
A Velocity chart shows the number of story points completed in a sprint. **Aim for flat Velocity, or +/- 10 percent.**

Each team's relative story points are unique and cannot be compared. Never compare two teams' Velocities.

ASK

- Has the team's makeup changed?
- Are we in a seasonal period?
- Is the team working at a sustainable pace?
- Is the team swamping on issues?
- Is the backlog of work unstable?
- Have there been escalations, causing iteration priorities to shift?

Velocity Chart



Team-Level Metrics

Say:Do Ratio

WHY

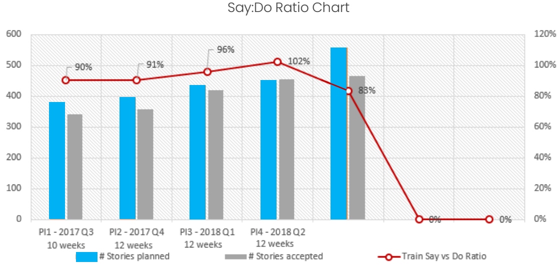
The Say:Do Ratio helps new teams understand Agile maturity and existing teams understand the health of the backlog. It also aids in understanding the relationship between the team and the Product Owner.

WHAT

A percentage of the number of user stories committed to in a Sprint compared to actual number of stories finished gives you your Say:Do Ratio. Committed stories are the specific stories planned during Sprint Planning.

ASK

- Has the team’s makeup changed?
- Are we in a seasonal period? – There are times when the seasonal nature of the work might cause higher volatility
- Is our backlog of work unstable?
- Have there been escalations, causing iteration priorities to shift?



Team-Level Metrics

Production Cycle Time

WHY

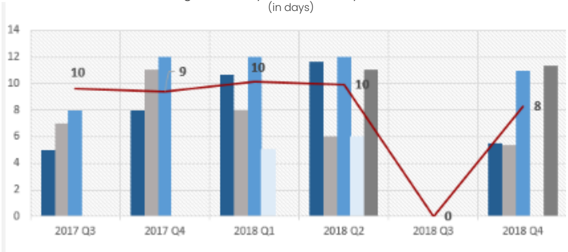
This metric helps teams understand the impact of changes. Decreased Production Cycle Time leads to decreased Business Cycle Time. Large wait times can increase both Business and Production Cycle Time.

WHAT

Production Cycle Time is the measure of how many days it takes to process a request, from when development starts until the team's definition of done (DoD). **Steady or decreasing Production Cycle time is preferred.**

ASK

- Is everyone testing, code reviewing, etc.?
- Is the team swarming?
- Does the team need to limit work in progress?
- Does the team have dependencies that have not been resolved?
- Are there handcuffs or impediments getting in the team's way?



Avg. User Story Production Cycle Time
(in days)

Team-Level Metrics

Work in Progress (WIP)

WHY

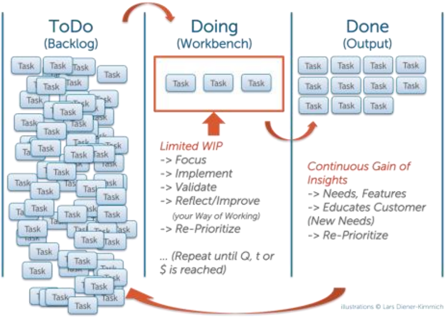
Tracking and limiting WIP ensures the entire team takes ownership of a project and produces high quality code. Limiting makes it easier to identify inefficiency in a team's workflow.

WHAT

WIP limits restrict the number of stories in each status of a workflow. When the upper limit for a status has been reached, the team stops and works together to clear the bottleneck.

ASK

- Has the team's or teams' makeup changed?
- Does the team have a continuous improvement mindset?
- Has the work complexity changed?
- Are quality issues hampering the ability to deliver stories?



illustrations © Lars Diener-Kimmich

Story Churn

WHY

Teams use this metric to help improve their ability to focus and achieve the goals originally set forth in Sprint Planning.

WHAT

Story Churn, also known as the “Chaos Metric”, is the number of stories/defects added to or removed from an iteration over time. If the number is high, it’s a strong indicator the team is frequently context switching. **The goal should be a low, steady downward trend.**

ASK

- If churn is zero, is the team taking enough risks? Are they gaming the metric?
- Is the team constantly putting out fires? Are they responding to reprioritized requests?
- Is there a reciprocal commit in place between the Product Owner and the team?
- Are quality issues resulting in rework?

Sprint Goal Success Rate

WHY

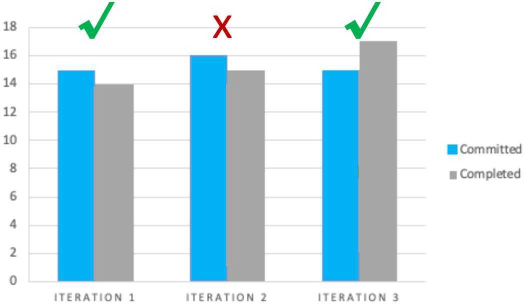
This metrics acknowledges that teams perform “better” when working towards a common goal. A Sprint Goal can also help break down objectives into smaller, easier-to-deliver outcomes.

WHAT

Sprint Goal Success Rate measures the percentage of time a team successfully reaches its sprint goal in a given period.

ASK

- Has the team’s or teams’ makeup changed?
- Does the team have a continuous improvement mindset?
- Has the work complexity changed?
- Are quality issues hampering the ability to deliver stories?
- Have there been escalations, causing Sprint priorities to shift?



Team-Level Metrics

Work vs. Wait Queues

WHY

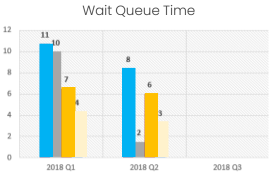
We measure Work vs. Wait times to identify bottlenecks and areas of improvement. This metric clarifies the time it takes to develop a request according to your team's DoD.

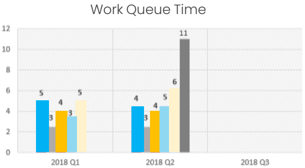
WHAT

Work time measures the time a team member actively works on a story, bug, spike, or tech debt in an iteration. Wait time measures the time a team member waits on something or someone before continuing work. **It's ideal to see a decrease in wait time and an increase in work time.**

ASK

- Has the team's or teams' makeup changed?
- Does the team have a continuous improvement mindset?
- Has the work complexity changed?
- Are quality issues hampering the ability to deliver stories?
- Are the teams swamping around bottlenecks and/or blockers?
- Is there a process bottleneck getting in the way?





Throughput

WHY

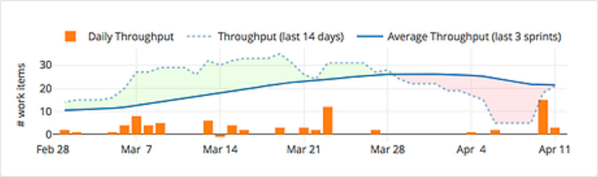
Measuring throughput as a trend over time can provide insight into the rhythm of the team's ability to deliver. It can also help to identify bottlenecks and areas of improvement.

WHAT

Throughput is the measure of how many days it takes to process a request, from when development starts until the team's DoD.

ASK

- Has the team's or teams' makeup changed?
- Are we in a seasonal period?
- Does the team have a continuous improvement mindset?
- Has the work complexity changed?
- Are quality issues hampering the ability to deliver stories?
- Are the teams swamping around bottlenecks and/or blockers?
- Have there been escalations, causing iteration priorities to shift?
- Is the backlog of work unstable?



Source: www.mazzlo.com

Escaped Defects

WHY

When we are shipping code ship fast, we want to have confidence it is defect free.

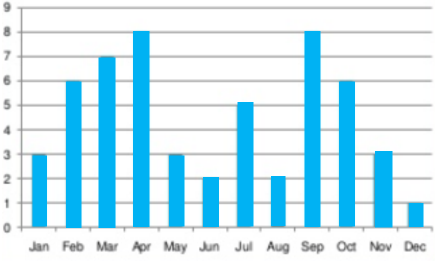
WHAT

Escaped Defects measures the number of defects that are found after they have gone to production. **Escaped defects should be zero or close to zero.**

ASK

- Are we working at a sustainable pace?
- How can we catch bugs as they are created?
- Are we testing at the right level? Unit? Functional Regression?
- Does the team have a process improvement mindset?

Escaped Defects Over Time



Team-Level Metrics

Deployment Time

WHY

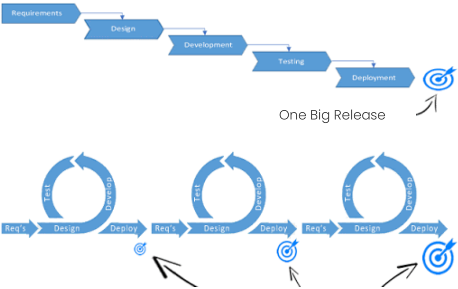
Faster deployment increases time to market, resulting in the end user obtaining value sooner. Smaller, faster batch sizes also help improve overall quality through lowered variability.

WHAT

Deployment Time simply measures the time it takes to deploy.

ASK

- What processes are taking the longest to deploy?
- If there was a sudden decline, did we miss a step? What does our quality look like?



One Big Release

Small, Frequent Releases

Agile Leadership Metrics

- Technical Debt
- Value Delivered
- Cumulative Flow Diagram (CFD)
- Team Morale
- Skill Versatility
- Team Member vs. Company Turnover
- Epic or Release Burndown
- Percentage of Automated Test Coverage
- Impediment Age
- MTTR
- Agile Release Train Predictability
- Cyclomatic Complexity
- Running Tested Features (RTF)
- Release Cycle Time
- Process Improvement Efforts

Technical Debt

WHY

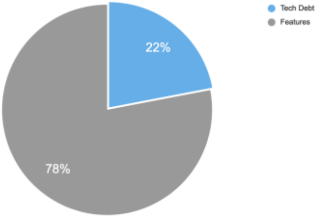
Unaddressed Technical Debt compounds on itself. Products or systems that carry a lot of Tech Debt lead to unhealthy code bases, unhealthy backlogs, bad products, and unhappy employees and customers.

WHAT

Tech Debt reflects the implied cost of additional rework caused by choosing an easy solution now, instead of using a better approach that would take longer. This metric measures the total number of Tech Debt items accomplished at the end of each iteration.

ASK

- Is the team balancing feature development with Technical Debt?
- Are defects and failures related to compliance, security, and performance increasing?
- Are decisions being made that defer the need for necessary redesigns and/or needed maintenance of code?



Value Delivered

WHY

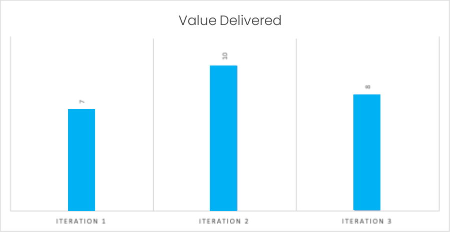
We all want to deliver valuable software to our customers, quickly. This measurement helps organizations be more well-rounded in their analysis, versus simply measuring flow. It doesn't matter if teams over deliver, if customers don't want what the product, service, or new features.

WHAT

Value Delivered measures the value, at a feature/epic level, that a team delivers to the end user/customer. Note: value can come in many different forms.

ASK

- If we are not delivering as much value as expected, could we be spending a lot of time on Prod Support?
  - If so, should we complete more tech debt to increase our Value Delivered?
- If the number isn't where we anticipated, are we working on lower valued features?



Cumulative Flow Diagram (CFD)

WHY

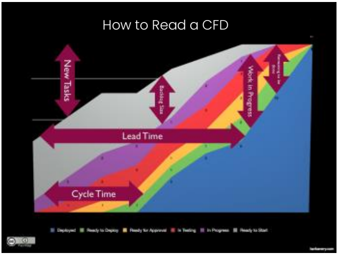
A Cumulative Flow Diagram (CFD) is an important part of increased value delivery. Consistently increasing WIP will lead to more meetings and missed deadlines.

WHAT

CFDs allow you to see your team's flow at a glance. **An ideal CFD consistently trends upwards, with a steady number of WIP items.**

ASK

- Cliffs - Has there been significant scope change?
- Flatlining - Do we wait to move our items along the Kanban board until the end of an iteration?
- Flatlining - Is there something blocking the team's flow?
- Increased WIP - Is the team starting work before completing current work?



Agile Leadership Metrics

Team Morale

> "[Team] Morale is the enthusiasm and persistence with which a member of a team engages in the prescribed activities of that group."
>
> – Frederick J. Manning[4]

WHY

Morale is not as subjective as a happiness metric and less susceptible to changes in mood. Teams with high morale are more likely to support each other, have grit in the face of adversity, and be proud of their work. Ultimately, high morale results in higher quality work being delivered quicker, plus a more pleasant work environment.

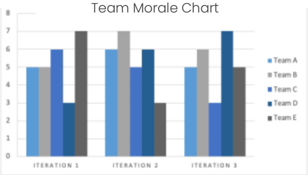
WHAT

To gauge Team Morale, measure before a retrospective. Average the scores and divide by number of prompts. Prompts you could use include:

- In my team, I feel fit and strong
- I am proud of the work that I do for my team
- I am enthusiastic about the work that I do for my team
- I find the work that I do for my team of meaning and purpose

ASK

- Does the organization have a culture of openness, transparency, and support of growth?



[4] Frederick J. Manning. "Morale, Cohesion, and Esprit de Corps." In Handbook of Military Psychology, 1991

Skill Versatility

WHY

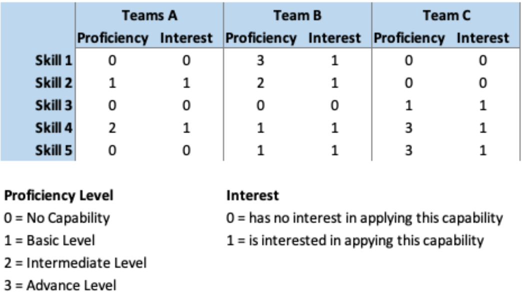
Agility leans towards small, cross-functional teams with an emphasis on skills over titles, as this structure helps reduce a hands-off approach. The Skill Versatility metric helps in understanding the Agile maturity of an organization.

WHAT

Skill Versatility measures the growth of cross-functionality within a team or teams.

ASK

- What are our most critical applications and are they resilient?
- Where do we have a gap in demand vs. knowledge vs. desire?
- How do we work to close these gaps?



	Teams A		Team B		Team C	
	Proficiency	Interest	Proficiency	Interest	Proficiency	Interest
Skill 1	0	0	3	1	0	0
Skill 2	1	1	2	1	0	0
Skill 3	0	0	0	0	1	1
Skill 4	2	1	1	1	3	1
Skill 5	0	0	1	1	3	1

Proficiency Level

0 = No Capability

1 = Basic Level

2 = Intermediate Level

3 = Advance Level

Interest

0 = has no interest in applying this capability

1 = is interested in appying this capability

Team Member vs. Company Turnover

WHY

People are a company's most expensive and valuable asset. Teams working in an Agile fashion tend to have high job satisfaction and are less likely to leave the organization, and this metric provides direct insight into that.

WHAT

Team Member vs. Company Turnover measures the attrition of members on a team relative to attrition of the company overall.

ASK

- Are we addressing raised impediments?
- Do the teams have the ability to solve impediments?
- Have we had a significant change within the past 6 months to the way teams are working?

Epic or Release Burndown

WHY

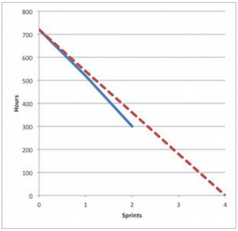
This metric helps predict how many sprints it will take to complete an epic or release based on past sprints and changes during the sprints. It helps the business understand if a release is on track or if adjustments need to be made.

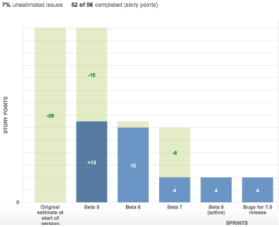
WHAT

Epic or Release Burndown shows the progress of an epic or release. This metric is similar to a Sprint Burndown, but focuses on the bigger picture. The actual number is based off team estimates and capacity.

ASK

- Have there been escalations, causing iteration priorities to shift?
- Does the team have a DoD and/or definition of ready (DoR)?
- Has the team's makeup changed?
- Are we in a seasonal period?
- Is the backlog of work unstable?





Percentage of Automated Test Coverage

WHY

Percentage of Automated Test Coverage helps businesses understand the progress of automated test coverage and Agile maturity. Automated testing can bring huge gains in terms of speed and efficiency.

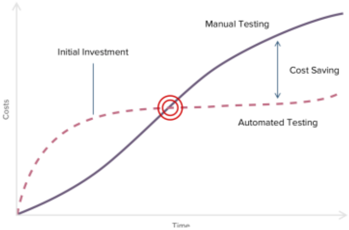
WHAT

This metric shows the percentage of automated test coverage compared to manual test coverage. Note: it does not indicate anything about the effectiveness of the testing; it only measures its dimension.

$$\frac{\textbf{\# of automated test coverage}}{\textbf{total test coverage}}$$

ASK

- Are we still able to catch defects before they are released to production?



Impediment Age

WHY

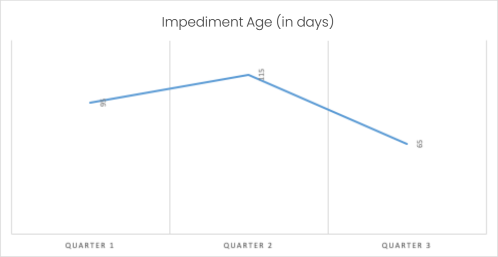
Impediments or waste slow down progress and can increase Lead (Business) and Production Cycle Time and frustrate employees. Note: some impediments are a slow burn.

WHAT

An impediment is anything that slows the teams down, also known as waste, and Impediment Age measures how long it takes to remove an impediment.

ASK

- Do we have a culture that focuses on continuous improvement?
- Are we involving the right people/teams to solve impediments?



Mean Time to Recovery (MTTR)

The average speed of a system's or product's recovery process:

$$\frac{\textbf{down time in a period}}{\textbf{number of incidents}}$$

Mean Time to Repair (MTTR)

The average time it takes to repair a system or product:

$$\frac{\textbf{sum of time spent on repairs}}{\textbf{number of repairs}}$$

Mean Time to Resolve (MTTR)

The average time to fully resolve a failure of a system or product:

$$\frac{\textbf{down time in a period}}{\textbf{number of incidents}}$$

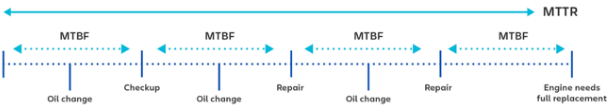
Mean Time to Respond (MTTR)

The average time to recover from a product or system failure from the time the incident was first reported:

$$\frac{\textbf{full resolution time}}{\textbf{number of incidents}}$$

ASK

- Is there an issue with the alerts system?
- Is something preventing the team from fixing the system quicker?
- What is the response time for fixing the request?



https://www.atlassian.com/incident-management/kpis/common-metrics

Agile Leadership Metrics

Agile Release Train Predictability

WHY

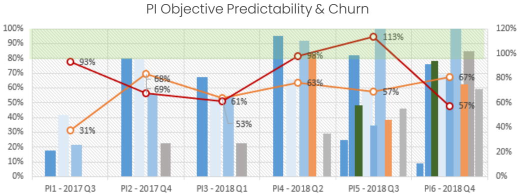
This metric helps build trust with stakeholders by showing how teams are completing what they committed to completing. It can be used to help eliminate a culture of escalation.

WHAT

Agile Release Train (ART) Predictability compares planned business value vs. actual business value attained during a Program Increment for an ART. **The goal is 80% to 100% predictability.**

ASK

- Have there been escalations, causing iteration priorities to shift?
- Has the work complexity changed?
- Are quality issues hampering the ability to deliver stories?
- Has the team's makeup changed?
- If we are not delivering as much value as expected, could we be spending a lot of time on Prod Support?



Agile Leadership Metrics

Cyclomatic Complexity (CYC)

WHY

CYC helps teams understand software quality, and can be used to:

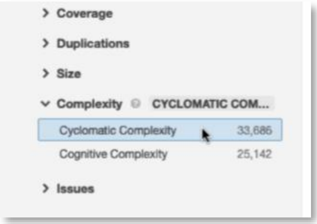
- Limit code complexity
- Determine the number of test cases required
- Help understand complexity of code for maintainability

WHAT

To arrive at this metric, simply count the number of decisions in the source code.

ASK

- What are our most critical systems? How do can we decrease their complexity?
- What enabling architecture can we put in place to ease complexity?



Cyclomatic Complexity 33,686

Cognitive Complexity 25,142

Running Tested Features (RTF)

WHY

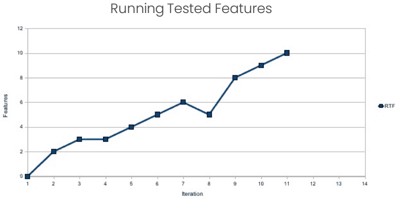
RTF gauges the general health of a product's development and ultimately helps the organization focus on outcome instead of process. It helps teams balance speed and quality.

WHAT

RTF measures how many features are passing all their acceptance tests.

ASK

- Were their changes to scope of work items?
- Is there an opportunity to adjust our acceptance criteria?
- Is the team working at a sustainable pace?



Release Cycle Time

WHY

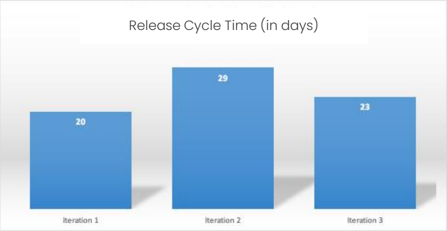
Small batch releases with short Release Cycle Time reduce risk by injecting less variability into an environment. Additionally, smaller releases can help improve predictability of when new features can be released to production.

WHAT

Release Cycle Time measures how many days it takes to release to production.

ASK

- What processes are taking the longest as we release?
- If there was a sudden decline, did we miss a step? What does our quality look like?
- Have there been escalations, causing priorities to shift?
- Are quality issues hampering our ability to release?



Process Improvement Efforts

WHY

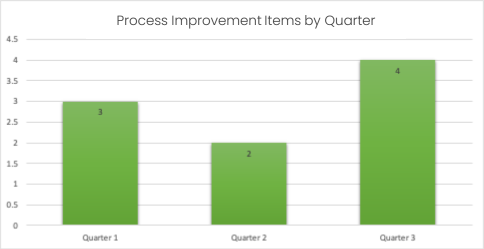
As we learned earlier in this book, you get what you measure. Therefore, if we measure process improvement, it can help our team work to hold space to grow a process improvement mindset.

WHAT

Process Improvement Efforts is a measurement of the process improvement items your team has worked on or is working on.

ASK

- Are we allowing our team to prioritize process improvement items?
- Do we have a culture that focuses on continuous improvement?



Executive-Level Metrics

- Net Promoter Score (NPS)
- Business Cycle Time
- Organization Impediment Age

Executive-Level Metrics

Net Promoter Score (NPS)

WHY

Net Promoter Score helps an organization understand customer satisfaction. Teams can deliver a high-quality product quickly, but if the customer isn’t happy it may not really matter.

WHAT

This score measures how products and services meet or surpass customer expectations. It is impacted by all three aspects of software quality: functional (what the software does), structural (whether the software meets standards), and process (how the code was built).

ASK

- Is there an opportunity to adjust our acceptance criteria?
- Are we inviting end users to our demos and adjusting according to their feedback?
- Have we objectively observed our end users in the wild?

$$\textbf{Net Promoter Score} = \frac{\textbf{\# Promoters}}{\textbf{Total Responders}} - \frac{\textbf{\# Detractors}}{\textbf{Total Responders}}$$

Executive-Level Metrics

Business Cycle Time

WHY

Business Cycle Time can help teams find areas of improvement, assesses how they are operating, and understand how changes are impacting the organization. This is a key metric for understanding the health of your work and your teams.

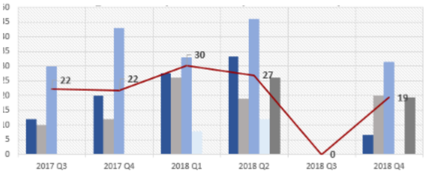
WHAT

This metric measures the time from when a request was entered into your work tracking tool, to when all work on the item is completed and the request has been considered done by the team's DoD. **A steady or decreasing Business Cycle Time is preferred.**

ASK

- Has the team's or teams' makeup changed?
- Does the team have a continuous improvement mindset?
- Has the work complexity changed?
- Are quality issues hampering the ability to deliver stories?

Avg. User Story Business Cycle Time in Days



Organization Impediment Age

WHY

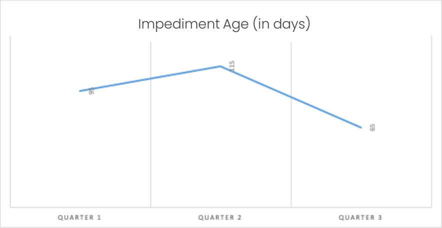
Impediments or waste slow down progress and can increase Lead (Business) and Production Cycle Time and frustrate employees. Note: some impediments are a slow burn.

WHAT

An impediment is anything that slows the teams down, also known as waste, and Organization Impediment Age measures how long it takes to remove an impediment at the organizational level.

ASK

- Do we have a culture that focuses on continuous improvement?
- Are we involving the right people/teams to solve the impediment?



Ensuring Success

Software was already a critical component for most modern companies, and 2020 further enabled "IT to transition from supporting the business to being the business."[5] We hope these metrics help you understand the impact of a development project or transformation on team morale, productivity, delivered value, user satisfaction, and more.

No matter where you're at in your Agile transformation, Polaris can help. If you're wondering where to start, stalled out, or struggling to measure success, **reach out** for a free Agile Enablement discovery session.

About Polaris

We provide modern software solutions to help companies across industries bring out the best in their teams, increase productivity, and achieve bigger wins, faster. Our offerings include:

- Application Modernization
- Agile Enablement
- Azure Cloud Solutions
- DevOps Enablement

Learn more at **polarissolutions.com**

[5] *Gartner Forecasts Worldwide IT Spending to Grow*, 2021

Authors

Angela Dugan
People Lead

Sara Caldwell
Lead Agile Guide

POLARIS SOLUTIONS

www.polarissolutions.com