DevOps Institute
ADVANCING THE HUMANS OF DEVOPS

Observability: DevOps' Crystal Ball

MISSION: Bringing Joy to Work

Helen Beal

Herder of Humans

@bealhelen

Helen Beal is a DevOps and Ways of Working coach, Chief Ambassador at DevOps Institute and an ambassador for the Continuous Delivery Foundation. She is the Chair of the Value Stream Management Consortium and provides strategic advisory services to DevOps industry leaders such as Plutora and Moogsoft. She is also an analyst at Accelerated Strategies Group. She hosts the Day-to-Day DevOps webinar series for BrightTalk, speaks regularly on DevOps topics, is a DevOps editor for InfoQ and also writes for a number of other online platforms. She regularly appears in TechBeacon's DevOps Top100 lists and was recognized as the Top DevOps Evangelist 2020 in the DevOps Dozen awards.

What is Observability?

Clue: It's not monitoring.

Observability is a characteristic of systems; that they can be observed. It's closely related to a DevOps tenet: 'telemetry everywhere', meaning that anything we implement is emitting data about its activities. It requires intentional behavior during digital product and platform design and a conducive architecture. It's not monitoring. Monitoring is what we do when we observe our observable systems and the tools category that largely makes this possible.

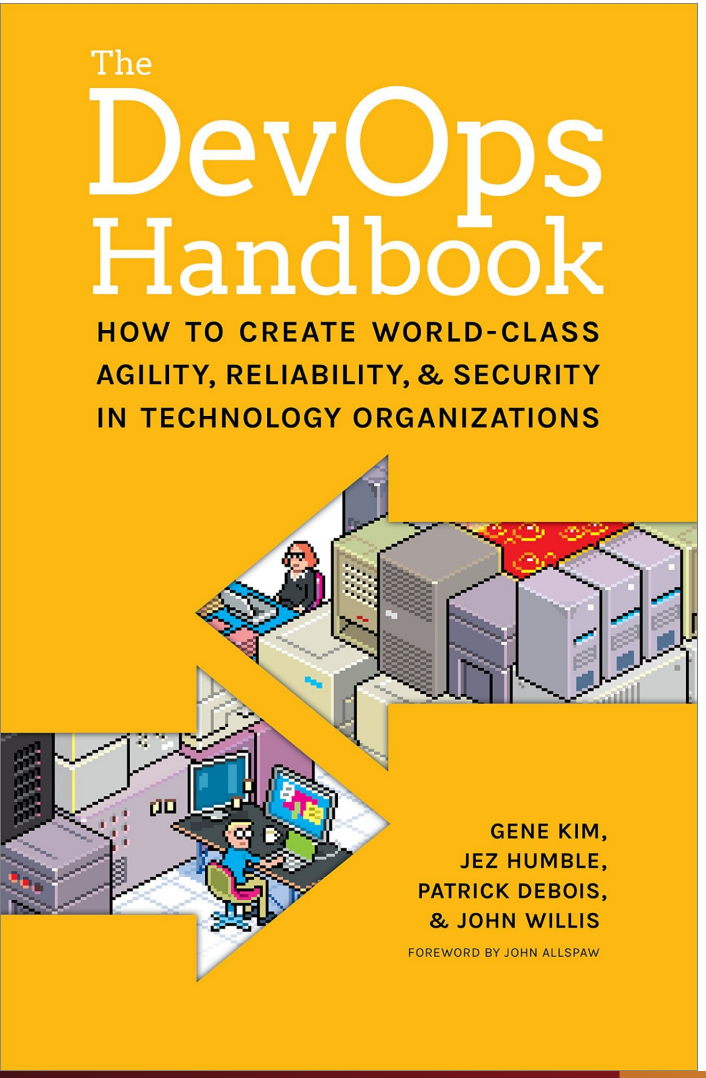
Where has the concept come from?

“On the General Theory of Control Systems’ by Rudolf E. Kálmán in 1960

In control theory, observability is defined as a measure of how well internal states of a system can be inferred from knowledge of its external outputs.

Telemetry Everywhere

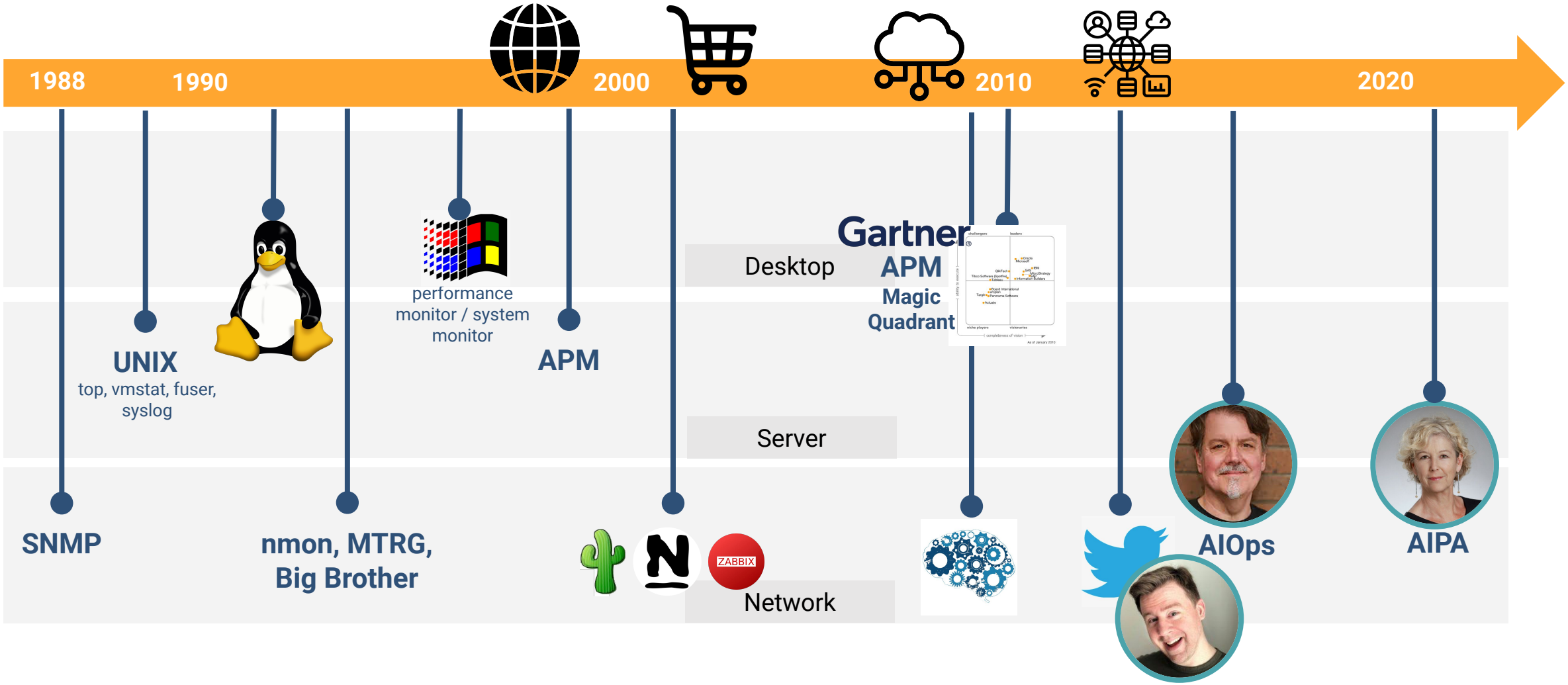
Is it the same as observability?



"We need to design our systems so that they are continually creating telemetry, widely."

"Telemetry is what enables us to assemble our best understanding of reality and detect when our understanding of reality is incorrect."

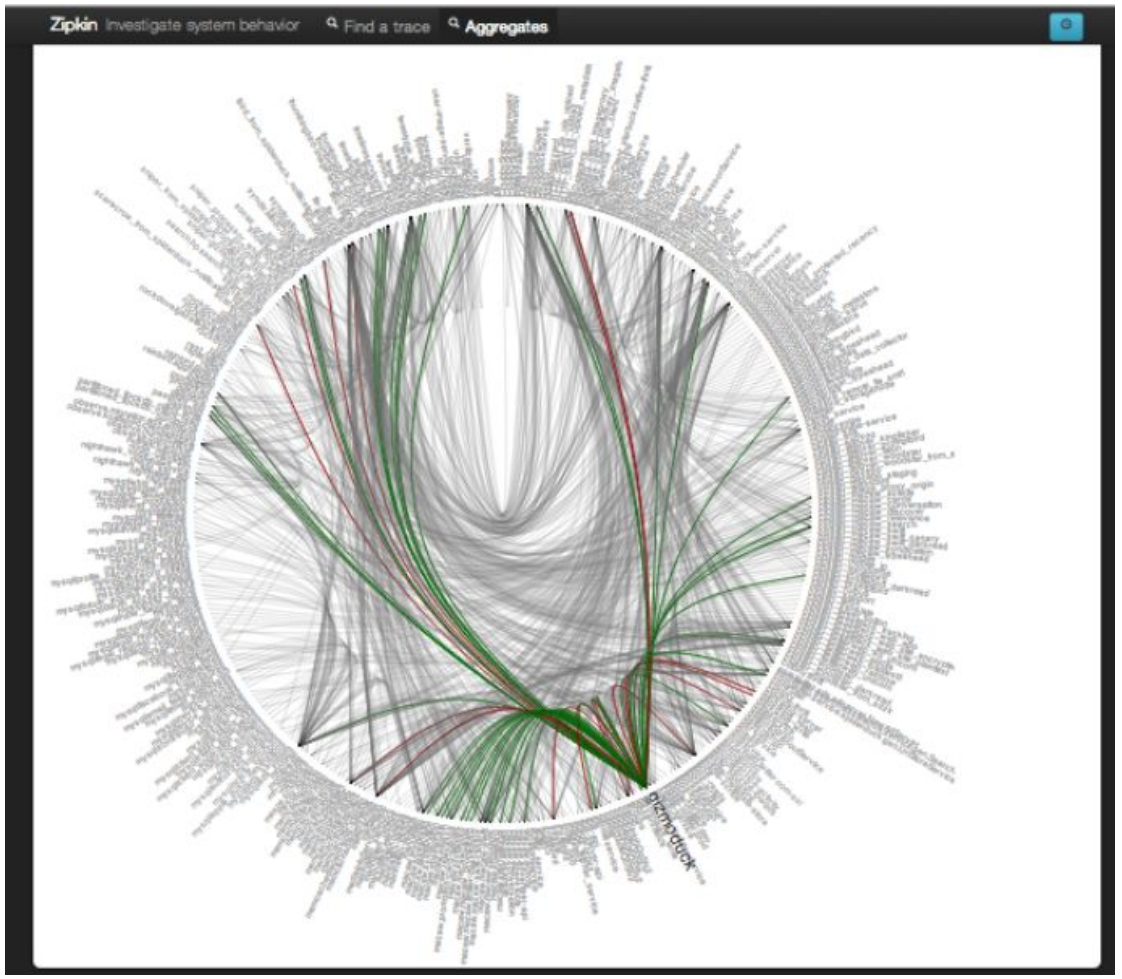
Evolution of Monitoring to Observability



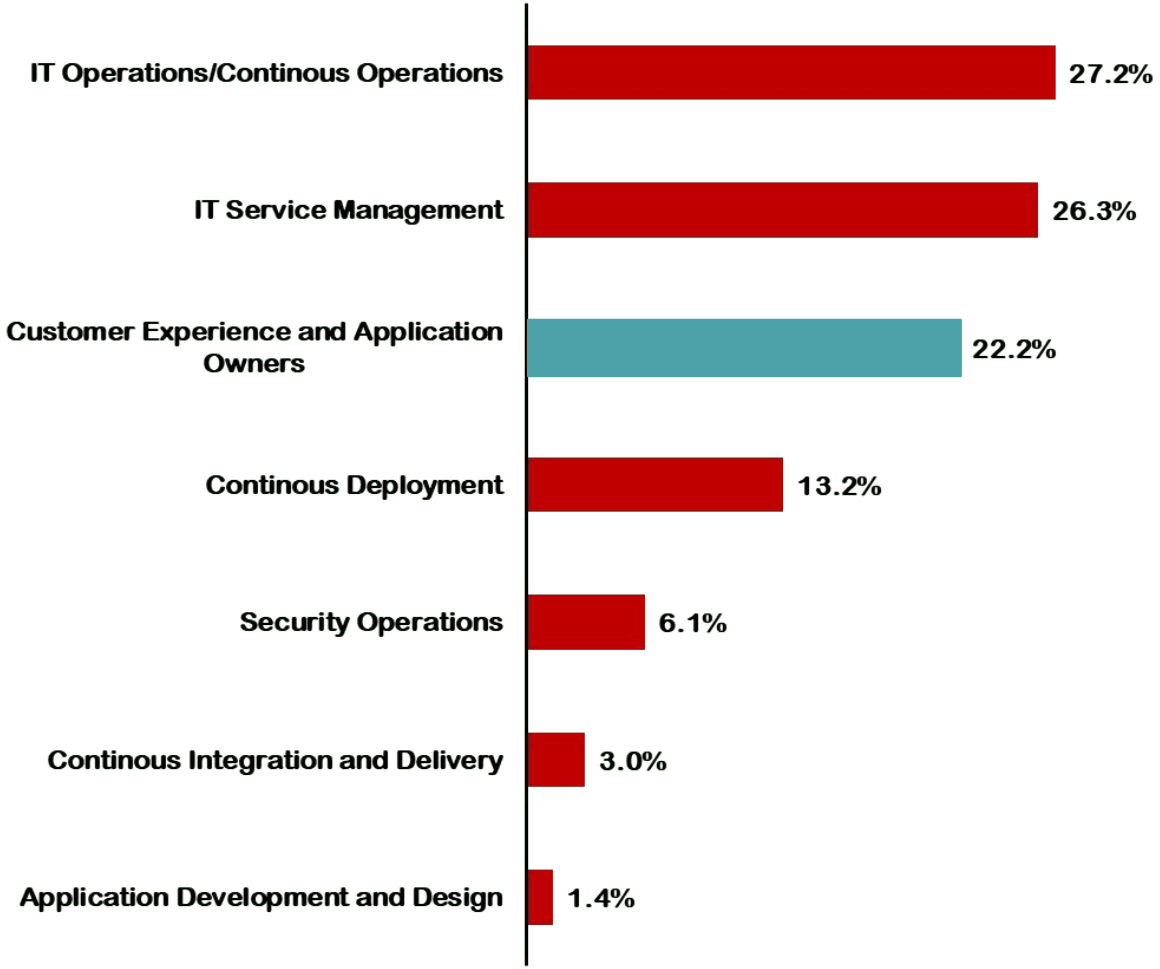
Observability at Twitter

@gphat 2013

"Thousands of service instances with millions of data points require high performance visualizations and automation for intelligently surfacing interesting or anomalous signals to the user. We seek to continually improve the stability and efficiency of our stack while giving users more flexible ways of interacting with the entire corpus of data that Observability manages."



AI Predictive Analytics



From the Research in Action AIPA Vendor Selection Matrix 2021

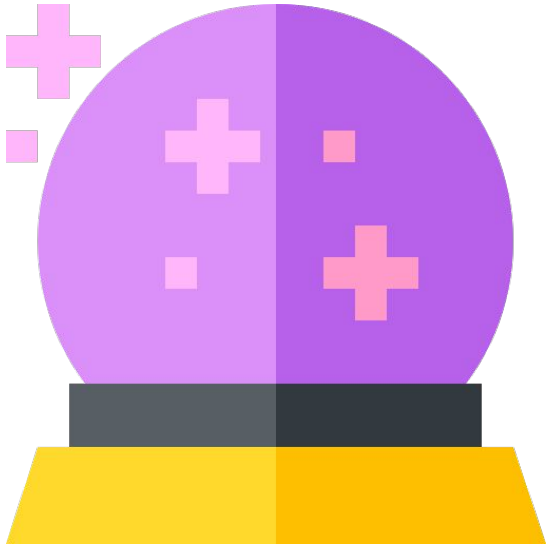
"The future lies in leveraging AI's power to predict across application development, IT operations, and service management which is why Research In Action has decided to rename the AIOps research into AI Predictive Analytics." *Eveline Oehrlich*

The Crystal Ball of Observability

Reality - now

REACT

Problems



Reality - future

PREDICT

Value

Advantages of Observability

Leaders are...

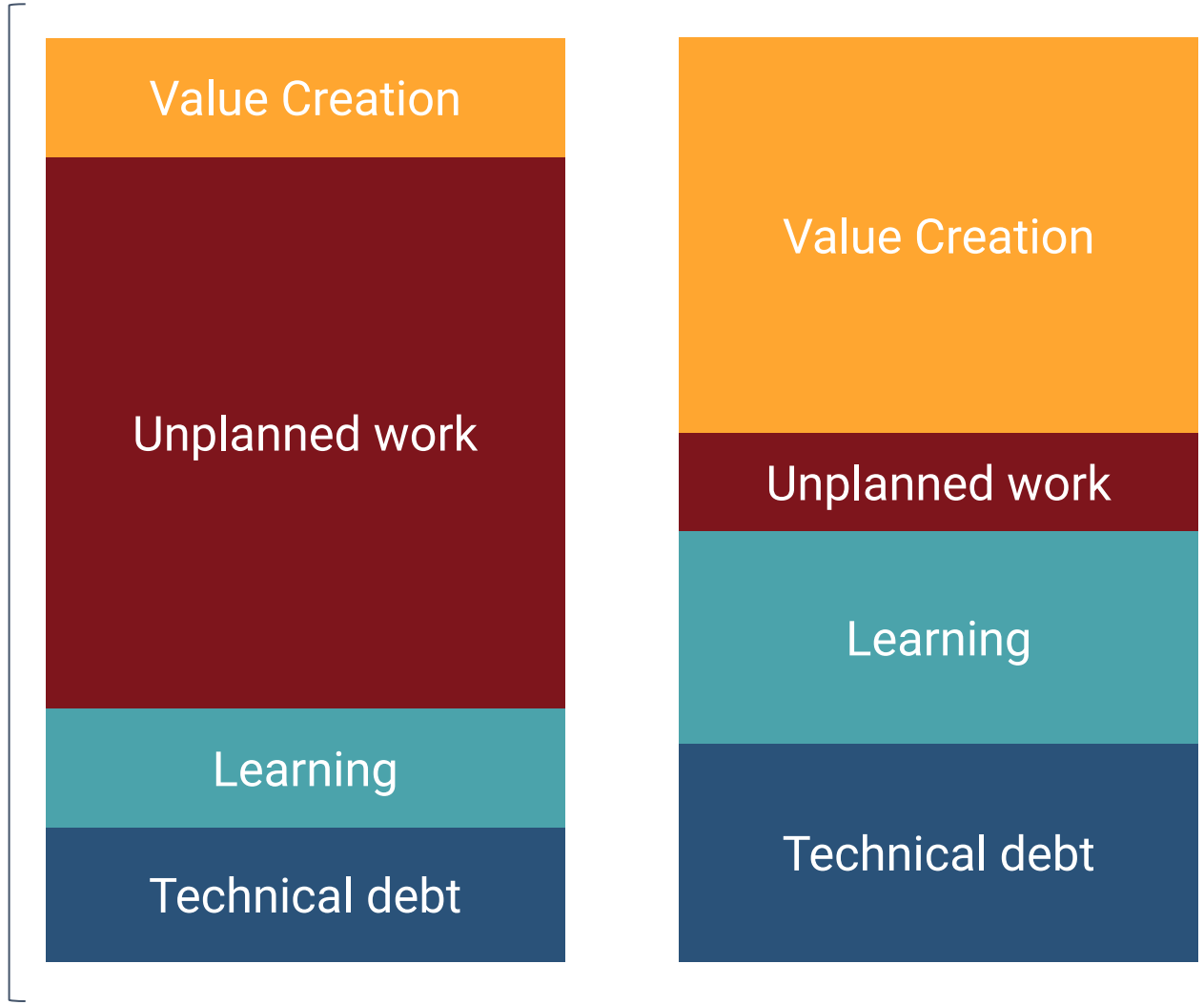
- **2.9 times** as likely to enjoy better visibility into application performance
- Almost **twice as likely** to have better visibility into public cloud infrastructure
- **2.3 times** as likely to experience better visibility into security posture
- **Twice as likely** to benefit from better visibility into on-premises infrastructure
- **2.4 times** likelier to have a tighter grasp on applications, down to the code level
- **2.6 times** likelier to have a fuller view of containers (including orchestration)
- **6.1 times** likelier to have accelerated root cause identification (43% of leaders versus 7% of beginners)

CALMS and Observability

Culture	Automation	Lean	Measurement	Sharing
Visibility and transparency builds trust Data-driven not opinion-driven conversations Fast feedback on experiments A tool that supports team autonomy: “We build it, we own it”	Accelerated root cause(s) analysis and insights Pre-emptive warning and forecasting operating behavior Automated service assurance Data discovery, crunch & insights	Accelerates flow (MTTx) Removes handoffs and delays between teams Observability across the end-to-end value stream Focus on customer experience	Real data that measures progress and improvements operations, SRE, SLOs and error budgets Actionable insights based on streaming data Telemetry everywhere	Provides a shared platform for collaborative analysis Builds a knowledge base so local discoveries become global improvements ChatOps

The Cost of Unplanned Work

What the team spends their time doing



Without observability

With observability

The Three Pillars

LOGS

An event log is an immutable, timestamped record of discrete events that happened over time

Easy to generate and instrument.

Can cause performance issues.

METRICS

Numeric representation of data measured over intervals of time.

Well-suited to dashboards and aggregation.

Historically poor dimensionality.

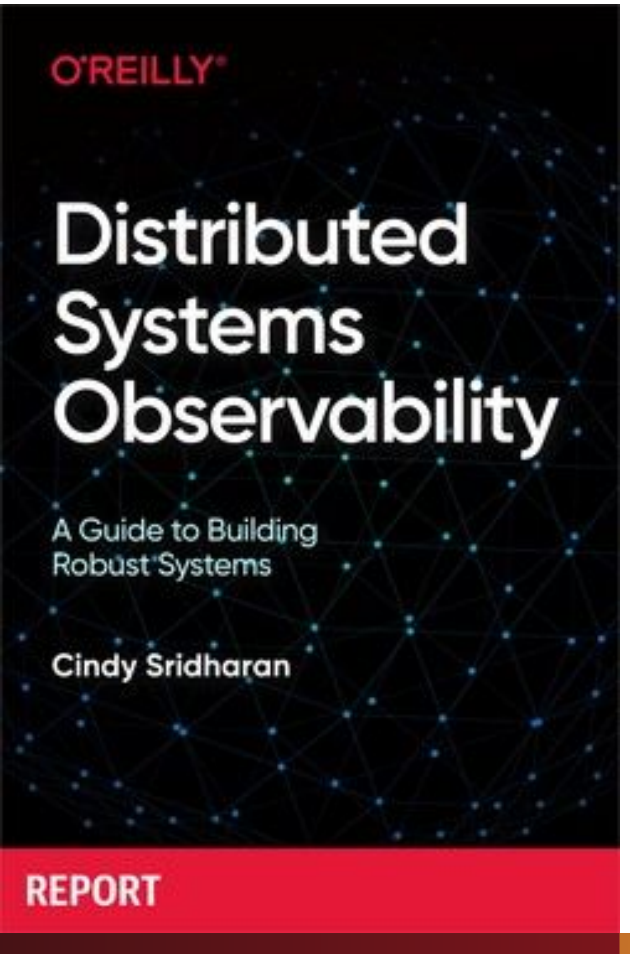
TRACES

A representation of a series of causally related distributed events that encode the end-to-end request flow through a distributed system.

Myriad use cases.

Very challenging to retrofit.

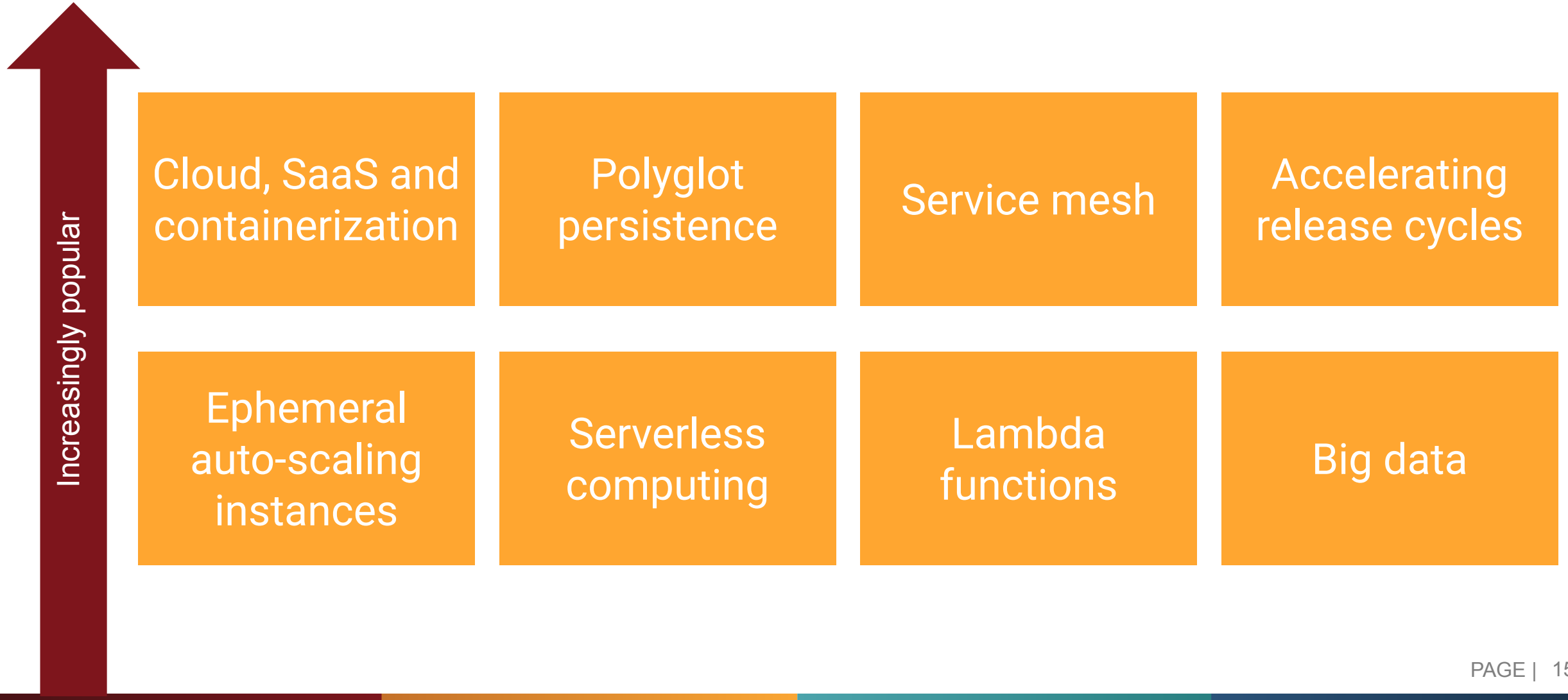
Hidden Assumptions of Metrics



- Your application is monolithic in nature
- There is one stateful data store (“the database”)
- Many low-level systems metrics are available and relevant (e.g., resident memory, CPU load average)
- The application runs on VMs or bare metal, giving you full access to system metrics
- You have a fairly static set of hosts to monitor
- Engineers examine systems for problems only after problems occur
- Dashboards and telemetry exist to serve the needs of operations engineers
- Monitoring examines “black-box” applications that are inaccessible
- Monitoring solely serves the purposes of operations
- The focus of monitoring is uptime and failure prevention
- Examination of correlation occurs across a limited (or small) number of dimensions

The Progressive Platforms

From monoliths to microservices - APIs rule



Cardinality Matters

High-cardinality data is the most useful for debugging

LOW	HIGH
Database column has lots of duplicate values in a data set	Database column has a large percentage of completely unique values

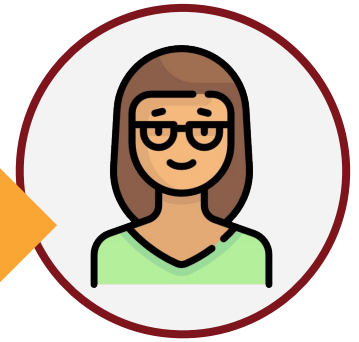
User ID	012345
First Name	Helen
Last Name	Beal
Gender	Female
Species	Human

Highest possible cardinality

Lowest possible cardinality

ITOps Persona

How Observability Helps IT Operations Evolve (AIOps)



More time for value experimentation

Step 5 — Add more automation for self-learning systems

Step 4 — Use chaos engineering for antifragility

Step 3 — Pay down technical debt for increased stability

Step 2 — Automate toil using AI insights

Step 1 — Reduce MTTR through noise reduction

Icons by Freepik and Phatplus from FlatIcon

The Developer Persona

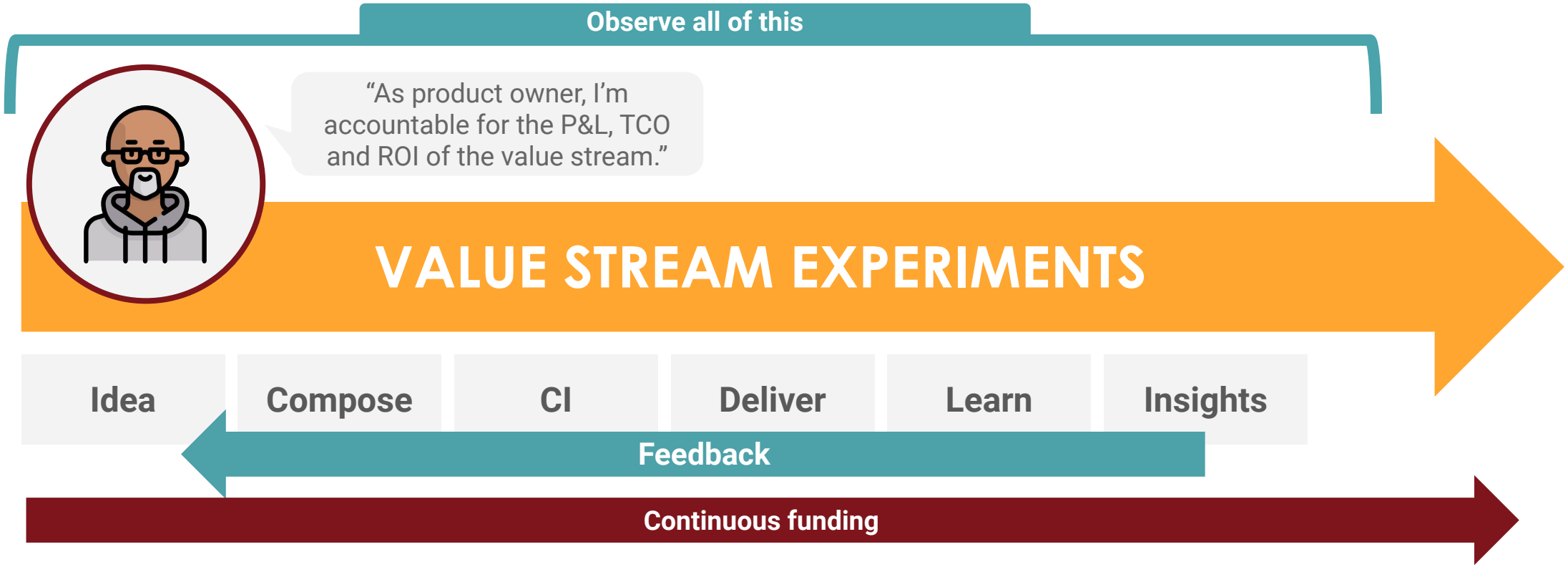
Observability Driven Development: X-Driven Development

DevOps INSTITUTE

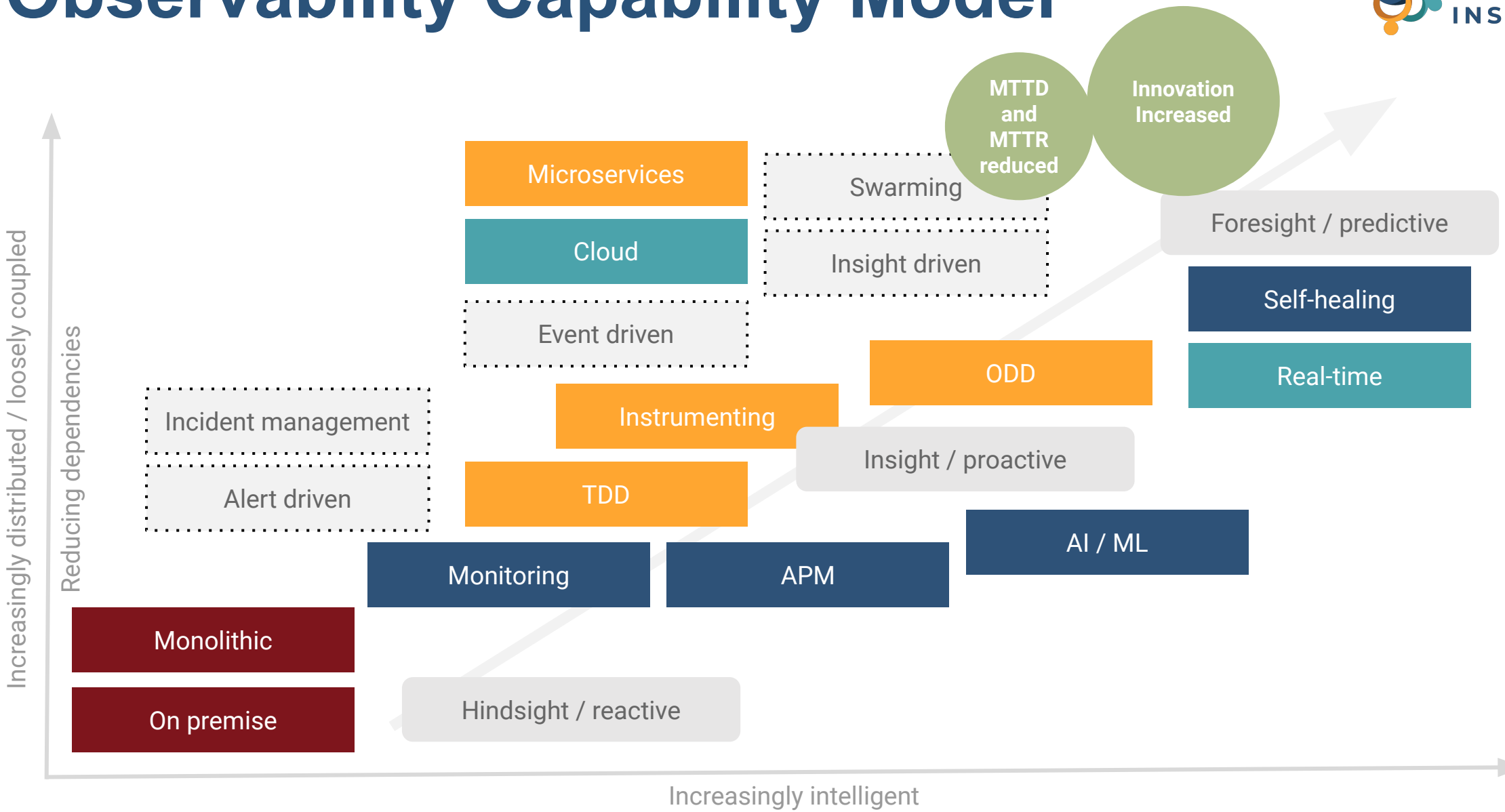
Test-Driven	Behavior-Driven	Hypothesis-Driven	Impact-Driven	Observability-Driven
TDD	BDD	HDD	IDD	ODD
A software development process relying on software requirements being converted to test cases before software is fully developed, and tracking all software development by repeatedly testing the software against all test cases. This is as opposed to software being developed first and test cases created later.	An agile software development process that encourages collaboration among developers, quality assurance testers, and customer representatives in a software project. It encourages teams to use conversation and concrete examples to formalize a shared understanding of how the application should behave.	Hypothesis-driven development is a prototype methodology that allows product designers to develop, test, and rebuild a product until it's acceptable by the users. It is an iterative measure that explores assumptions defined during the project and attempts to validate it with users' feedbacks.	**EMERGING** Takes small steps towards achieving both impact and vision. Impact Driven Development balances the development of a vision with creating real impact for users. It makes sense that the first phase of your product development should involve some users.	**EMERGING** Adds another layer to software development by encouraging the development team to think about the application availability and uptime throughout their development process and similar to unit-testing development, wrap their code with more verbose logging, metrics and KPIs

Observability and Funding

The value stream or product owner is a mini-CEO



Observability Capability Model



DevOps Institute

ADVANCING THE HUMANS OF DEVOPS

THANK YOU