

# MANAGING IT AS A PRODUCT—NOT A PROJECT



According to Mik Kersten, author of [Project to Product](#), an historical pattern occurs during technology revolutions.

This pattern holds great significance for digital transformation during our current revolution, the “Age of Software”. Kersten makes the case that, as organizations undergo [digital transformation](#), they must treat their digital initiatives as products—not projects.



Organizations that master the means of software production will dominate the rest of this age. This means deploying software-based solutions at scale as a product, not as a cost center. Those continuing to use project and cost-center management will become irrelevant.

This article will cover an overview of Kersten's *Project to Product* argument, how it makes the case for managing software as a product, and how IT professionals can change management systems to meet that challenge.

## The three periods of a technology revolution

Per Kersten's book, the world usually resides in one of the following three periods of a technology revolution.

- **The Installation Period.** A [new technology wave](#) combines with new financial and innovation ecosystems to launch the technology. A “Cambrian explosion” of startups disrupt entire industries, provoking a frenzied migration into the new technology. Financial capital drives the Installation and fuels the disruption. The Installation Period ignites one or more economic bubbles and ends with a crash.
- **The Turning Point.** After the crash occurs, a long period of economic prosperity ends. The current economic context changes. Businesses must quickly adapt to new economic conditions or disappear. Societal problems speed the technology's adoption, as the world progresses to the next period.
- **The Deployment Period.** Here, the focus shifts to production. Organizations mastering the new means of production earn larger shares of economic activity and control the new infrastructure. Society, business, government, and economies are reshaped to fit the new age. Organizations that cannot adapt die. The Deployment Period continues until the next technology revolution arrives, and the cycle begins again.

The previous technological revolution was the “Age of Mass Production”, lasting from about 1908 to 1974. Advances in manufacturing, transportation, energy, synthetic materials, and infrastructure drove that age.

## At the Turning Point in the Age of Software

We are currently living in the Age of Software, which began in 1971. Advances in microprocessors, communications, the internet, and software drive this age.

The Age of Software's Installation Period ran through the first decades of the new millennium, where the world rushed to adopt new hardware, software, and communications technologies. Kersten recently asserted that [2020 has catalyzed the Turning Point in the Age of Software](#). The Turning Point can last many years until we move into the Deployment Period.

Survival in the Age of Software depends on an organization's ability to deliver value by creating digital experiences and delivering software products. Digital technology becomes the organizational core. It cannot be siloed away from the rest of the business, as is common with traditional IT.

Companies in all industries that master new means of production will displace companies that are slower to adapt. The big tech giants—companies like Amazon, Facebook, Microsoft, Netflix—are taking over more of our economic capacity and markets. They treat software as a product, not as a project. Companies that can deploy software-based solutions as a product *at scale* will survive. Those that cannot die or become irrelevant.

Many of us talk about treating software as a product not as a project, but what does that mean and what does it take to make it a reality?

# The problem with IT cost center & project management

IT is often treated as a cost center, while digital initiatives—such as Web and mobile applications, fintech, or [IoT applications](#) that ride on the [IT infrastructure](#)—are treated as profit centers. Under cost center management, IT success is defined by delivering the project on-time, on budget, and enabling cost reductions. This is known as the cost center trap, where the need for cost reductions overwhelms the need to deliver increased velocity, efficiency, and productivity in your digital initiatives. IT and business objectives become disconnected.

Traditional [IT project management](#)—typified by Gantt charts and productivity software such as Microsoft Project—also causes value creation issues during the Turning Point. Project management is heavily related to management techniques refined in the Age of Mass Production. Its defining features are:

- Milestones for on-time delivery.
- Waterfall deliverables, where project objectives are delivered in a linear sequence, and the project flow resembles a waterfall.
- Project management focuses on the [Project Management Office \(PMO\)](#). This can create a black-box environment with complex requirements and low transparency.
- Projects that are siloed in IT, separated from the business.
- Fixed capital and operating budgets. It is difficult to “go back to the well” once a budget is approved. This encourages the project sponsor to ask for everything they may need up front.
- Risks are defined and planned for up front when a project charter is created.
- Fixed scheduling covering software launches instead of product lifecycles. Lifecycle support, maintenance, and end of life are usually covered under separate teams and budgets.
- Interchangeable resources that are assigned at beginning of project. Project team members are assigned multiple projects and their involvement may end with deployment. Business resources may not be assigned to the project.
- Success is measured as being on time, on budget, and reducing costs.

## The three epiphanies

According to Kersten, a large portion of organizational spending on IT and software delivery creates little to no product value. There is no standard agreement for measuring how value flows to the business through software delivery. Only costs are measured, which leads to Kersten's three epiphanies concerning how organizations try—and fail—to apply management concepts from previous technological revolutions to the Age of Software, (quoted exactly from *Project to Product*:

***“Epiphany 1: Product declines and waste increases as software scales due to disconnects between the architecture and the value stream.***

***Epiphany 2: Disconnected software value streams are the bottleneck to software productivity at scale. These value stream disconnects are caused by the misapplication of the project management model.***

***Epiphany 3: Software value streams are not linear manufacturing processes but complex collaboration networks that need to be aligned to products.”***

# Moving to product-oriented management

Product-oriented management connects software delivery to the businesses' *value streams*. A value stream is much more than an IT project. It covers all the end-to-end activities that deliver customer value for a product or service, including software. A value stream may span different teams, experts, specialists, processes, and tools.

The defining features of product-oriented management include:

- Timeframes that include the lifecycle of the product, which can be multiple years, including maintenance, upgrades, health of the product, and end of life.
- Funding product value streams that are based on business needs. Products are the units of investment.
- Success is measured by whether a product value stream meets its business objective, including revenue generation.
- Risks are spread out over time. This creates options to terminate the product, if assumptions were incorrect, or to pivot to a different approach as opportunities arise.
- Groups of stable cross-functional teams are typically assigned to one product value stream.
- Managed by roadmaps and hypothesis testing. Focus is on feature and business-value delivery, which drives an Agile orientation.
- A focus on directly mapping activities to business outcomes, enabling more transparency. Bottlenecks in the value stream are visible when looking at the end-to-end solution, allowing for quicker identification and resolution.

The tech giants and startups have mastered product-oriented management. But, according to Kersten, [modern software delivery practices](#) are not fully translating to business value because enterprises are still managing IT through projects and costs centers rather than as part of the business's value stream.

The Flow Framework: IT product-oriented management

There are many relevant frameworks that IT is using to account for the new challenge of thriving in the Age of Software's Turning Point, including:

- [The Scaled Agile Framework \(SAFe\)](#)
- [DevOps](#)
- [Lean management principles](#)

And the latest ITIL® iteration, [ITIL 4](#), focuses on co-creating business value, rather than just delivering IT services. In the transition from project to product management, we are concerned with **Software Value Flow**: The activities involved in producing business value along a software value stream.

Kersten's argument is that we need a new framework to help organizations determine the **outcomes** of their [Agile and DevOps](#) practices and to provide metrics to improve those practices. To that end, he developed the Flow Framework:

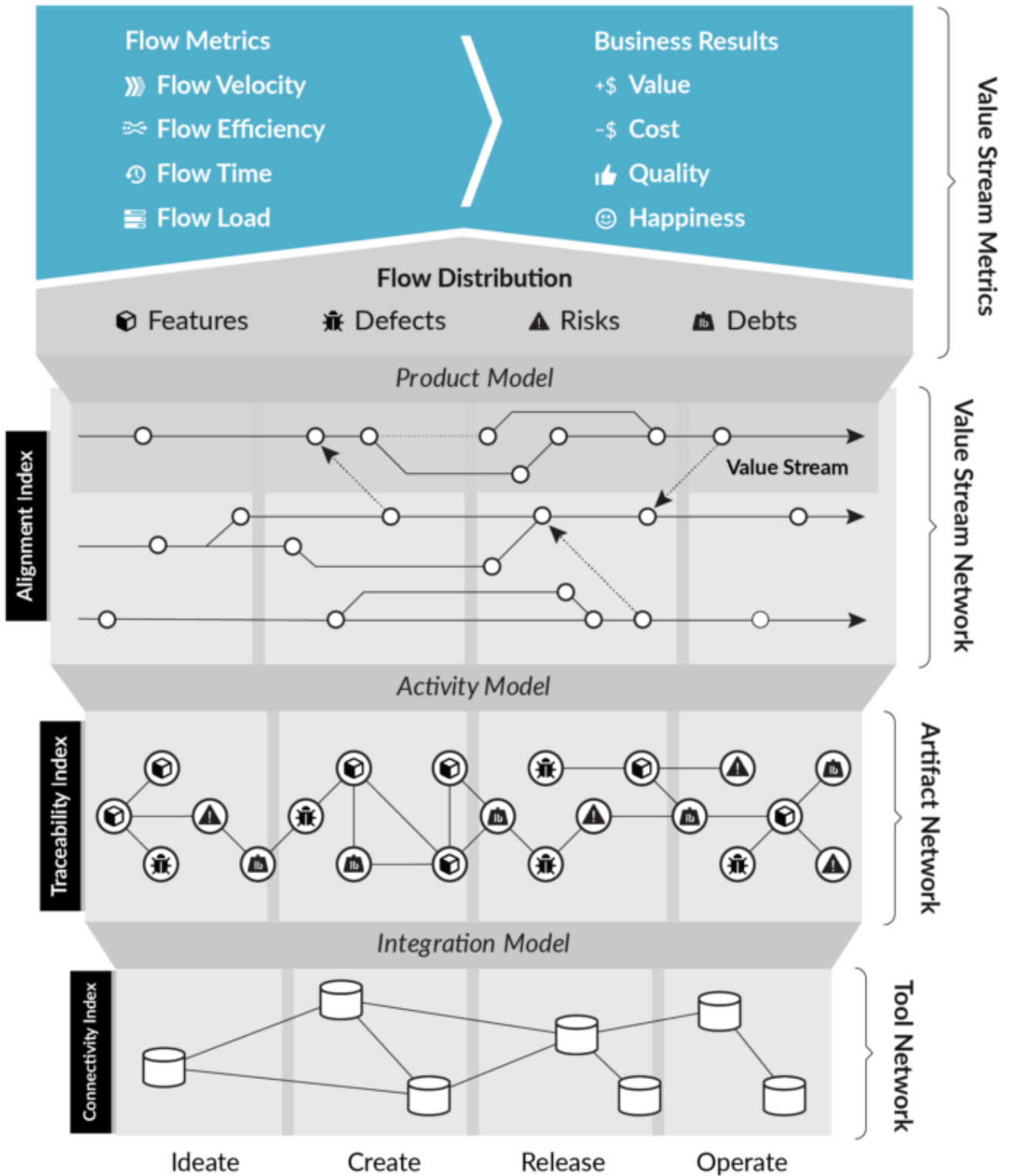
*"...a mechanism for aligning all delivery activities in your organization around your software products, tracking the business results of those activities in order to create a feedback loop."*

The Flow Framework provides a means of scaling the flow, feedback, and continual learning practices organizations are already implementing to the entire end-to-end business process for

software delivery. While we won't cover the mechanics of the Flow Framework in this article, the framework uses the following networks, models, indexes, and metrics to model and manage software flow:

# Flow Framework™

v1.0



Copyright © 2018 Tasktop Technologies Incorporated.

The Flow Framework™ is a framework created by Mik Kersten, CEO of Tasktop Technologies Incorporated ("Tasktop"). This diagram is licensed under the Attribution No Derivatives Creative Commons License, accessible at <https://creativecommons.org/licenses/by-nd/4.0/legalcode>.



## **Tools Network, Integration Model, and Connectivity Index**

- The tools organizations use in each value stream
- How integrated the tools are inside each stream
- What artifacts (units of work or delivery items such as tickets, defects, releases, etc.) are produced
- How connected the value stream tools and artifact containers are

## **Artifact Network, Activity Model, and Traceability Index**

- The flow of artifacts through each value stream instance
- Whether each artifact is connected to all other artifacts it is related to
- What percentage of artifacts are visible to the Value Stream Network

## **Value Stream Network, Product Model, and Alignment Index**

- The network formed by an organization's interconnected software value streams
- The activities within each value stream (such as design, development, or support)
- How product value stream activity, flow metrics, and business results are measured
- Whether the delivery organization is aligned to products over projects

## **Value Stream Metrics**

- Measurement of the critical flow items in value streams (Features, defects, risks, and debt)
- Number of flow items completed over a given time period (Flow Velocity)
- What percentage of time flow items were actually worked on, versus the total elapsed time (Flow efficiency)
- Time elapsed from when a flow item becomes active to when it is delivered to a customer (Flow time)
- Number of active or waiting flow items (Flow load)
- Metrics for each product's value, cost, quality, and happiness (Business results)

## **Flow Framework supports value stream performance**

Using the Flow Framework, organizations can monitor and manage the actual performance of their software value streams and make decisions based on what is actually happening. It provides visibility and feedback into how things are running and where the bottlenecks are. It models how software is deployed at scale. For the rapid development, deployment, maintenance, and retirement of digital initiatives, the Flow Framework is a better template than the IT cost center and project management model.

Remember that *Project to Product* and the Flow Framework only provides Kersten's groundwork for transforming dinosaur-based IT cost center and project planning to a product-based management system. There is still a lot of work to be done. Product-based IT management identifies a critical business survival issue and points to a way forward, as we all navigate the Turning Point in the Age of Survival. I'd recommend reviewing Kersten's *Project to Product* arguments and determining if they are valuable for your organization.

## Additional resources

For more on IT management and similar topics, browse:

- [BMC Business of IT Blog](#)
- [BMC Service Management Blog](#)