Level up Your Value Delivery

By Gil Broza, 3P Vantage (gil@3PVantage.com)

Introduction

If you're a product development or IT leader, you probably ask yourself:

What can we do to deliver better results?

Effective answers require taking a holistic look at **Product Development** – your company's **value delivery system** that extends from idea to production.

Having studied many such systems in the real world, we've discovered that as they improve, they move through **five distinct levels** of **fitness for purpose**: the higher a system's level, the better it helps the company achieve its vision and objectives. It's the same five whether the systems are product- or project-oriented, traditional or agile. You could say that these levels are characterized by the biggest problem that the system, working as it does, presents to the company.

Level 1	The system is unable to contribute adequately to achieving company objectives.
Level 2	The system contributes to achieving company objectives, but not effectively and efficiently enough.
Level 3	The system seems to work well, but is really only as good as the few people who make high-level decisions.
Level 4	The system is effective and efficient, but it's slower to achieve mid/large outcomes than it needs to be.
Level 5	The entire system works well for the company's needs.

We've further discovered that it takes **only 10 strategies** to move from Level 1 to Level 5, and they're incremental: Based on your system's current fitness level, you need to execute only two or three specific strategies to level up.

All this is part of my new SQUARE model. Using it on the next few pages, you'll discover answers to the question we opened with.

Ready to get started?

Find out your value delivery system's current level

Step 1: Determine the scope of the system

It bears repeating that value delivery (usually called Product Development) is a **system.** It's not only a section of the org chart or a set of processes and artifacts. **It's all the contributors, management, and ways of working involved in conceiving, making, and delivering a technology product that matters.**

As such, it's rarely just Engineering or a team in IT. The people probably even report to different functional managers. However, they're all interdependent and necessary; no part of the system can achieve user outcomes and move business needles without the other parts. For this reason, SQUARE's strategies are less useful if they're applied only to specialized parts such as product management, DevOps, or UX.

Take a moment to make a clear mental picture of the scope of your value delivery system — who and what it includes.

If you need more advice on determining its scope:

- How you define the scope depends on many factors specific to your company. In most tech companies, it's likely to include product management, design, development, and testing for a product line. It would also include delivery, unless that's elsewhere in the company; the same goes for operations. If you develop software products for internal purposes, the system is the equivalent in terms of IT and their Business partners. It might also include vendors' people and contractors.
- In most cases, the system comprises multiple teams and individuals. In some cases (usually small companies or internal software development), the system is a single team, but it's still not only development and testing.
- The value delivery system is not the entire company; it is a distinct subset of it. A lot of people may care about the product (or even suffer if it's poor), but unless they have some influence or authority over it, they are not part of the value delivery system. For example, this is the situation in some companies with respect to marketing and customer support; in others, representatives of these functions have some say over product choices, which makes them part of the system.
- It's often helpful to take the perspective of a stakeholder or someone who benefits from the end product. Unlike you, they don't care about reporting lines and team structure, people's specialties, or their processes; to them it's one organization that "should work."
- If the company makes multiple products, there may be several value delivery systems, which might overlap. Focus on one at a time. Don't necessarily pick one for which you're collecting process metrics; the advice in this book works without them.

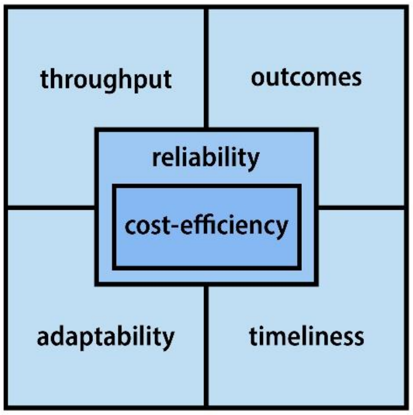
On the next two pages, you'll use SQUARE to calculate your system's raw fitness score and its fitness level. Beyond helping you determine which strategies will improve fitness, the way it models fitness for purpose will provide you with language and insights to discuss with your leadership and stakeholders while sidestepping process, jargon, and silos.

Step 2: Understand how we'll assess fitness

SQUARE defines **six fitness aspects** of a value delivery system, and based on your rating of each aspect, it calculates a raw fitness score and maps it to one of the five fitness levels. It does **not** assume you use Scrum, Kanban, or another common approach, agile or otherwise. And, it does **not** require metrics.

For you to rate these aspects accurately, which you'll do on the next page, it's important that we **agree on the exact meaning of each one**. On first glance they might look familiar, but the specific wording matters.

- **Throughput:** The extent of usable working product delivered by the system in short spans of time
- **Outcomes:** The system's achievement of valuable user and business outcomes (solving problems, addressing needs, achieving goals, seizing opportunities), both large and small
- **Timeliness:** The system's delivery of results when they're still valuable enough
- **Adaptability:** The adaptability of both the system and its product to important changes, whether due to internal choice or to external conditions, and whether adaptations are net positives or not
- **Consistency:** Under normal conditions, the consistency of the system's throughput, outcomes, timeliness, and adaptation (put another way, the range of fluctuations in the system's results on these four aspects)
- **Cost-efficiency:** The efficiency of achieving the system's current throughput, outcomes, timeliness, adaptation, and consistency for the money spent



Step 3: Calculate your system's current fitness level

Using the following table and proceeding from top to bottom, **rate each aspect of the system as it is today** relative to its **realistic contextual ideal.** That is, relative to the ideal that would make most practical sense for your company.

- **realistic:** You operate in a specific business landscape with constraints, expectations, and obligations. Even if you had ample time and funds, there's a limit to what would be practical.
- **contextual ideal:** Instead of comparing the aspect to an absolute yardstick or industry standard, you're assessing it relative to what you, as a leader, believe would *matter most* to your company's success. That ideal might not be perfect, cheapest, or fastest.

Rate each aspect by asking yourself:

- Is it far enough from its realistic contextual ideal, that it is (or *should be*) a constant point of concern for management and stakeholders? If so, rate it as "**far**."
- Is it near its realistic contextual ideal enough that it's not an issue ("good enough")? Rate it as "**near**."
- Is it between near and far? Rate it as "medium."

Advice about making your ratings accurate:

- When rating an aspect, be careful not to let noticeable outliers bias you. Examples of outliers: the one team that frequently deploys to production (while other teams don't), the two product people who always talk about outcomes (while others focus primarily on populating a backlog), the one team that spends forever in inefficient meetings (while other teams move on).
- While this type of assessment is admittedly subjective, do try to be as *neutral* as possible. That may be harder the more you're invested in the system. Consider this angle: What would an independent outsider, who knows your system well, say?

Aspect	Meaning	Rating
Throughput	The extent of usable working product delivered by the system in short spans of time	
Outcomes	The system's achievement of valuable user and business outcomes (solving problems, addressing needs, achieving goals, seizing opportunities), both large and small	
Timeliness	The system's delivery of results when they're still valuable enough	
Adaptability	The adaptability of the system and its product to important changes, whether due to your choice or to external conditions, and whether adaptations are net positives or not	
Consistency	Under normal conditions, the consistency of the system's throughput, outcomes, timeliness, and adaptation (put another way, the range of fluctuations in the system's results on these four aspects)	
Cost-efficiency	The efficiency of achieving the system's current throughput, outcomes, timeliness, adaptation, and consistency for the money spent	

For best results from the tool, we suggest you double-check your ratings for the aspects. Advice:

- Did you rate each aspect as it's defined above, and independently of the other aspects? For example, the observation "We deliver releases once a year, and that's not frequent enough" might make throughput "far" while consistency is "near" (and the annual release might be quite cost-efficient!)
- Do you believe you were as neutral and critical as you could be?
- Did you consider actual results and behaviours, rather than how people talk about them?

Note: If your analysis reveals that the system optimizes for some aspects at the expense of others, that's useful information – it doesn't mean you've rated them incorrectly. An example of that is when cost-efficiency is poor due to excessive switching of direction and focus (= high adaptability) that results in a lot of abandoned work.

Now, calculate your system's fitness-for-purpose level

1. Convert each rating to a numeric value as follows: far = 1, medium = 2, near = 3.
2. Sum up the numbers to produce the **raw fitness score**, which should be between 6 and 18.
3. Find your system's fitness-for-purpose **level** in the following table:

Level 1	Level 2	Level 3	Level 4	Level 5
6-8	**9-11**	**12-13**	**14-16**	**17-18**

Keep a record of your six ratings and the raw fitness score. However, the improvement strategies we're about to review require only the fitness level. Two points about that:

- The six aspects' ratings are externally visible effects of the system's workings. However, they are not entirely independent of each other, and they cannot be manipulated directly. The level reflects the overall system fitness, so that's the focus of improvement.
- Despite its simplicity, SQUARE's mapping of the aggregated ratings to a 1-5 scale seems to work reliably in the real world.

If the fitness level seems low to you, **take heart.** Getting to a great fitness-for-purpose is a **long** journey even when the stars align. Take a moment to acknowledge the successes and positives in your journey so far. As you look to improve the fitness, the following will show you where to concentrate your efforts.

Strategies for leveling up

Risks to the system's continuing performance

First, a note about risks. Systems generally face multiple risks to their continuing performance and fitness. At each level, the greatest risk is different, and though the risks from lower levels have likely been mitigated, they *have not been eliminated*.

Level 1	Level 2	Level 3	Level 4	Level 5
Loss of the few people who keep the system operational	Getting knocked off-course easily due to both interference from the outside and high levels of unplanned work	Chaos & breakdown due to big changes	Breakdown of process and good habits due to loss of stakeholder trust and/or system leaders' patience	Foundational changes to the system due to shifts in senior leadership

Executing the strategies

You need to execute all of your level's strategies past a certain threshold to get to the next level. However, no strategy is ever one-and-done:

- The strategies are incremental. The lower-level ones should be "baked in," and you might go deeper with some.
- If you let lower-level strategies slip, the ones at your current level won't deliver maximum benefit, **and** it will be much more difficult to level up sustainably. Your system's fitness might even drop a level.
- Every strategy takes minutes to describe and months to implement. Expect to implement it using an iterative, test-and-learn approach.

A necessary foundation for moving through all the levels is intentional leadership that actively builds and protects an explicit culture and way of working across the system.

Most systems advance one level at a time. Advancing two at a time requires expert guidance, appetite and tolerance for change, and supportive leadership for applying both levels' strategies simultaneously.

How to read the following

1. Start with the strategies for your system's level.
2. Then read them for all lower levels, starting with Level 1.
3. Familiarize yourself with the higher-level strategies, but you needn't worry about executing them until your system reaches those levels. Some of them are not relevant or practical yet; the others go deeper on matters that you implement just enough to make your current-level strategies viable and practical.

If your system is at Level 1

At this level, the system is unable to contribute adequately to achieving company objectives. The team works, sometimes very hard, and they have some successes. But on the whole, their deliverables fall short of what the business needs.

Often, the main culprit appears to be the team's process, but that's a symptom, not the cause. The cause is usually at the portfolio level: too many big-ticket items (projects, initiatives, etc.) are in flight or about to be. Various stakeholders, trying to accomplish their objectives, push work into the system, and all of it fights for attention. As a result, few items are acceptably completed in a timely fashion, the cost of managing work (and escalations!) balloons, and the cycle continues.

This is usually a tense situation, because the system's people – competent and dedicated as they are – feel that they can't succeed. **The highest risk to the system** is that the key people who keep it operational may decide to leave.

Strategies for moving to Level 2

Manage the project portfolio with greater strategic control over committed and in-progress items. Limit, choose strategically, and properly charter (frame) all portfolio items. (A portfolio item may be a feature, feature set, "epic," technical initiative, experiment, maintenance project – something big enough to make a demonstrable difference to the customers/users and the organization.) Self-imposed limits on the number of items will clear some of the logjam and allow Product Development to get more work to a deliverable state more often. Coupling the limits with strategic prioritization of the exact items to work on will also result in better deliverables. This strategy requires real accountability from the decision-makers, since their choices – and how frequently their choices change – have a large effect on effort, focus, and complexity, and therefore on company results. As well, this strategy often requires the influence and involvement of very senior people, because it's hard to say "no" or "not yet" to important work, especially when it's tied to managers' objectives.

Design the system's team structure and processes based on what matters most for achieving the vision and objectives. Start by getting leaders on the same page about what the system needs to be like if it's highly fit for its purpose. To put this in the terms you encountered earlier: Be explicit, and achieve alignment and consensus, about the realistic contextual ideal for each fitness aspect. From there, work together to answer, intentionally and explicitly, these questions: What is vital for the system to deliver the right product to the right customer at the right time, and why do we choose the answers we do? Use the answers to determine the way of working. For example, if frequent delivery and easy adaptation are vital for product success, you'll probably create a structure based on independent teams using methods such as Scrum or Kanban. Alternatively, if the imperative is to make early commitments and get them right the first time, you might opt for a more centralized, project-oriented structure. Notice what we're saying here: *first* know your objectives and most important fitness aspects and *then* choose the operational model, instead of adopting a favourite/familiar/popular model and trying to make it deliver on the objectives.

If your system is at Level 2

A Level 2 system addresses the needs of management, customers, and stakeholders, but not effectively and efficiently enough. As a result, some of those needs produce interference, unplanned work, scope creep, frequent priority changes, escalations, and aborted work. This kind of disruption is **the highest risk to the system's continued fitness**: even though the structure and process ought to produce some reliability, they are constantly under threat. Often, this triggers process patches and restructurings that satisfy a few powerful people at the expense of overall system fitness, risking a drop to Level 1.

Strategies for moving to Level 3

Establish clear and appropriate decision-making across the system. For every type of product-affecting decision, it must be clear and accepted which individuals or groups make it and how. That is a necessary, though insufficient, condition of both effectiveness (producing impactful deliverables) and efficiency (doing so with minimal waste). In fact, your next Level 2 strategy – stabilization – will not bear fruit if decision-making is hazy, inconsistent, or not fully defined. Base the choice for each decision on the system's intended values and principles as determined in Level 1 (and perhaps improved since then). For example, your functional and project managers should make very different kinds of decisions if you operate in an Agile model than if you operate in a more centralized, maximum-utilization model**.**

Stabilize the system. Thinking of the system as receiving variable demand on one side and producing supply on the other side, create an acceptable and sustainable balance between the two. Be wary of common tactics that may help some, but when executed regularly they compromise various fitness aspects: detailing more requirements upfront, preparing accurate estimates, maximizing people's utilization, working longer hours, and adding people. Instead, visualize the work, define explicit intake and completion standards, cultivate learner safety, break work down, finish what's started with minimal delay, reduce bottlenecks, constrain intake using time-boxes and/or WIP limits, reduce unplanned work and proactively manage what's left of it, keep spare capacity, enable people to contribute outside narrow specialties, and manage high-variability delays. Collect process data, but use it only for system improvement; be careful not to let data collection, analysis, and subsequent actions compromise psychological safety.

If your system is at Level 3

The typical Level 3 system seems "okay," contributing rather reliably to the achievement of company objectives. Usually, only one or two of the six fitness aspects is of ongoing concern; however, one or two others are close enough to their realistic contextual ideal – they are good enough. If the operational model is Agile, the teams churn out working product on a rather even keel; if the operational model is more project- or date-driven, teams generally hit their dates. Look closely, however, and you'll notice why the system is only at Level 3: it is really only as good as the few people who make high-level decisions about the work. Typically, those folks are product leads, technical architects, and middle management.

The overweighting of those few people in decisions about the work, and the teams' unwavering cadence of delivery, lead to team disengagement. Team members feel like cogs in a machine – fully loaded “resources” that check tasks off, disconnected from the mission and/or their customers. The machine, however, is optimized a certain way, and has low capacity to adapt to big changes. **The biggest risk to the system’s fitness is chaos and breakdown due to big changes.**

These phenomena may create a vicious cycle: the less team members engage (the more they “just do the work”), the more the decision-makers operate in isolation. Such systems are likely to move to Level 2 if enough key people leave.

Strategies for moving to Level 4

Increase contributor safety, real teamwork, and cross-system collaboration. Usually, when a few people are the main decision-makers, team members don’t feel psychologically safe to question ideas and assumptions, offer different perspectives, or pursue experiments. Two unvoiced assumptions are shared across the system: The planned work equals the right work, and the potential improvement gains from collaboration aren’t worth the time. Consequently, the most important thing to do is to maximize the production of deliverables and have individuals focus on their specialties (which usually means that teams are really only workgroups). Help the system shift away from this by increasing contributors’ safety, creating real teamwork, and enhancing collaboration across the system. This is also an important enabler for the next two strategies.

Defer more commitments and shorten release lengths while controlling costs. Consider your decision-makers’ planning approach: there are varying levels of commitment and detail at different time horizons, for example 12-24-month roadmaps, 3-6-month releases, and 2-week touchpoints. In this strategy, help them commit to less, and plan in less detail, in the longer horizons/cycles, and to reduce release length. Reduce both toward your company’s realistic ideal for throughput, outcomes, timeliness, and adaptability while controlling the cost of change (keeping future changes affordable) and transaction costs (the overhead of planning, coordinating, releasing, etc.). Performing this strategy will result in risk reduction, earlier value to customers, easier pivoting, less wasted work, and greater resilience to big changes. These changes in commitment and planning usually work synergistically with the next strategy at Level 3, that of engaging teams more and connecting them better to the meaning of their work.

Have teams contribute more meaningfully and efficiently to planning. Aim to improve outcomes by including multiple points of view and sparking collaboration across the system. Such outcomes include the identification of problems and solutions, allocation of tasks and responsibilities, avoiding unnecessary work, engagement, redundancy, and more. Involve contributors in multiple ways: as teams or individuals, synchronously or offline, opt-in or opt-out, etc. This comes at a cost, though: all the time people spend planning and thinking about the work is less time spent doing the work. Make sure it’s worth the actual outcomes and that it’s used efficiently, or the strategy will backfire.

If your system is at Level 4

The typical Level 4 system is both appropriately adaptive and a reliable producer of deliverables. It’s generally seen as a good place to work. While on a tactical level there seems to be healthy progress, a higher-level view reveals why the system is at Level 4: it struggles to pull off “big work” of strategic impact to customers and the company (such as integrating an acquisition or replatforming). It succeeds eventually, but it’s slower than the company needs it to be. Therefore, **the biggest risk** to the system is a breakdown of process and good habits due to loss of stakeholder trust and/or system leaders’ patience.

Strategies for moving to Level 5

Give teams a greater say in defining large outcomes and how they’ll achieve them. Maximize teams’ empowerment as much as possible and beneficial (continue to steer the system, but make the boundaries more expansive). Enable teams to experiment with their own ideas for better ways of working that would benefit the whole system; this might include reteaming or starting up targeted workgroups and guilds. Tap into their collective brainpower when identifying and sequencing large outcomes, and give them more say over how to achieve them. That will counteract the tendency to concentrate “large-outcome thinking” in the hands of a select few, who then present final-looking but too-big requirements to the product teams. The extra brain power and thought diversity will open up options for better solutions, while the increased participation will encourage the teams to put more heart and energy into work.

Increase the ratio of decisions that result in positive outcomes. When making product, engineering, or process decisions, tie them to *intended* user and business *outcomes*: “What problem / need / goal / opportunity are we addressing? What will change for our users and for us as a result of this work?”. Once executed, observe the decisions’ *actual outcomes* and study the gap between intended and actual. Your system probably does that to some extent on the tactical level; the greater effect will come from consistent application of this thinking to strategic and longer-term choices. Improve the inputs to decisions by understanding your users better, thinking through scenarios, and collecting data and feedback. Improve decision-making both on the people front (practice humility, increase collaboration, and manage incentives) and on the process front (interleave discovery and delivery, test and learn).

Keep the cost of technical changes affordable. To some extent, the system needs its product to be efficiently adaptable, both during the work and after delivery. Previous levels’ strategies dealt with containing the costs of process, people, and product content changes. At Level 4, you also need to keep the likeliest technical changes affordable. Remember, changing a technical element is not only *not* free, it may cost more than it took to build it the first time (which may have been totally appropriate based on what was known then). Control the cost of such changes by augmenting your way of working with Technical Agility principles such as rapid feedback, small and safe steps, and clean code. To make that possible, though, three parties – development, product/business, and management – need to make an intentional, explicit, and mutual commitment to Technical Agility: a social contract, as it were.

If your system is at Level 5

A Level 5 system is uncommon: all fitness aspects are close enough to their realistic contextual ideal that there’s rarely any issue. Getting there takes years of deliberate leadership that builds a self-operating system on top of a strong culture. And that, right there, sometimes creates a problem: the system’s culture experiences friction with the culture of the larger organization. We usually see this in value delivery systems that are based on modern values and principles – usually Agile, Lean, empowered teams, servant leadership – while the larger organization still defaults to hierarchical control and predictability.

The tension is usually felt at the edges of the system, where it interfaces with the rest of the organization. The people at the edges (typically, VPs and product leads) buffer the system from the tension, managing it well through trusting relationships with their colleagues outside the system. **The biggest risk** to this status quo is a change at the top. A new senior leader – at the helm of the system or just above it – may try to reshape it based on their own world-view and experiences in other contexts. We're familiar with several cases where the cultural disruption was powerful enough to send the system back to a lower level.

Strategies for remaining at Level 5

Continue building trusting relationships with stakeholders and senior management. This work – which never stops – is one part of your "insurance" against disruptions to the system. Don't assume that results speak for themselves.

Export the culture to the rest of the company. Other system leaders can learn a lot from your successes. Your helpfulness will improve the relationships you have with them, and if they shift their ways of working to better resemble yours, you'll reduce friction; that's the second part of your "insurance." Remember, you know two fundamental truths that are not common knowledge: optimal system fitness requires customized ways of working, and culture makes those ways of working possible.

Would you like to partner with us to accelerate the changes?

We can help you reduce the time and risk to effective and reliable implementation of the strategies. Schedule a confidential chat with Gil Broza at 3PVantage.com/talk or email gil@3PVantage.com.