

When Growth Outruns Delivery

A practical guide to finding the
constraints between signed deal
and customer value



Why I Built the Delivery Stress Test

Debbie O'Meara

Founder and Principal, ClearWay Projects



I've spent more than 20 years working in SaaS implementations and Professional Services, building delivery teams, creating implementation processes, and helping organizations adapt as their businesses grow.

I've seen what happens when growth starts to outpace the systems that support it.

Sales is doing its job. New customers are signing. But implementations start taking longer. Teams get stretched. Handoffs get messy. Customers aren't ready when everyone thought they would be. Leadership can see that delivery is getting harder, but not always where the problem starts.

And the answer isn't always more people. Sometimes the problem is the handoff. Sometimes it's customer readiness, capacity, process, ownership, technology, or the way the work itself is structured. Often, several of those things are interacting.

That's why I built the Delivery Stress Test.

My approach is to start with the outcome the business needs to produce, then work backward to determine what has to happen to produce it consistently. From there, the goal is to identify where the current system breaks down, understand why, and determine what needs attention now. And what can wait.

You don't need another generic assessment telling you that implementation is "important." You need to know where growth is putting pressure on delivery, and what to do about it.

That starts with finding the constraint.

ClearWay Projects.

Confident delivery. Clear results.

Delivery Stress Test

Is your delivery engine keeping pace with growth?

Revenue is growing. New customers are signing. So why does delivery suddenly feel harder than it used to?

Kickoffs slip. Implementation queues grow. Go-lives move. Teams spend more time chasing missing information, managing exceptions, and rescuing projects. Nothing is obviously broken. But the gap between signed and value keeps getting wider.

That gap matters.

What do I mean by “first value”?

It is the first point at which a customer can use your product successfully for the business outcome they bought it to achieve. For one company, that may be a live workflow; for another, a completed transaction, a launched program, or a measurable operating result. The important thing is to define it consistently.

The longer it takes a customer to reach value, the more the business carries work, cost, and risk without realizing the outcome of the sale. Time to Value (TTV) stretches. Customer confidence can soften. Teams work harder just to keep up.

Growth tends to expose these problems before they become obvious at the executive level. This stress test is designed to help you find them.

You'll assess four parts of the journey from signed deal to customer value:

1. **Predictability** — Can you predict what happens after the sale?
2. **Readiness** — Are customers actually ready when implementation begins?
3. **Throughput** — Can delivery absorb the pace of demand?
4. **Operating Model** — Do your teams have a repeatable way to move customers to value?

You don't need perfect data to use this assessment. If you don't know an answer, that's useful information too.

Use this with the people closest to the journey. The CEO or executive sponsor can set the context, but the most useful assessment often includes input from Sales, Professional Services or Implementation, Customer Success, Product, and Finance or Operations where relevant.

Part 1: The assessment

This assessment looks at the full signed-to-value journey, not only the work performed by the implementation team. It includes the handoffs, customer responsibilities, internal dependencies, decisions, and waiting time that determine how quickly a new customer reaches value.

Score each statement:

3 = Strong: We know this and can demonstrate it.

2 = Mixed: We have some visibility, but it isn't consistent or reliable.

1 = Weak: We don't know, or it varies significantly from customer to customer.

Note the evidence behind your score where you can. "I think so" is different from "we track it every month."

1. Predictability

Can you predict what happens after the sale?

Growth creates a different kind of delivery problem than simply having too much work. When sales volume increases, leadership needs to know what happens next: how long it takes to get customers started, how long implementation takes, and when customers should reach value.

If those answers depend on individual knowledge, averages that hide a lot of variation, or simply hoping things stay on track, growth can expose the gap quickly.

Assessment	Score	Evidence / Metric
We know our average time from contract signature to kickoff.		
We know our average time from kickoff to go-live.		
We know our average time to first value.		
We can predict, with reasonable confidence, whether a new customer will start on time.		

Predictability score: ___ / 12

What your score tells you

10–12: Predictability is relatively strong

You have a useful baseline for the customer journey and enough visibility to anticipate whether new customers are likely to stay on track.

7–9: Predictability is uneven

You probably know some of the numbers, but variation or gaps in the process make it difficult to predict what will happen with individual customers.

4–6: Predictability is a constraint

Leadership may know that delivery is taking too long, but not have enough visibility into *why* or where a new customer is likely to get stuck.

Dig deeper

Don't stop with the averages. Look at your last 10 implementations.

- How much did the total delivery time vary?
- Where did the biggest differences occur?
- How much time was active delivery versus waiting?
- Could you have predicted the delays before they happened?

That last question matters. If you couldn't predict which customers would be delayed—or why—the problem may not be speed. It may be predictability.

What to do next

Establish a baseline for the customer journey:

Contract signed → Handoff → Customer ready → Kickoff → Implementation → Go-live → First value

Then identify where variation enters the process and what conditions consistently create delays.

The goal isn't simply to make the average implementation faster. **It's to make the outcome more predictable as volume grows.**

2. Readiness

Is the customer actually ready to begin?

A signed contract does not mean a customer is ready for implementation. Customers may still need to provide data, make decisions, involve the right people, complete technical work, or resolve internal dependencies before the implementation can move forward.

If those requirements aren't clear—or aren't checked before kickoff—the implementation team often ends up absorbing the delay.

Assessment	Score	Evidence / Metric
Sales and delivery follow a clean, consistent handoff process for new deals.		
Customer readiness is confirmed before kickoff is scheduled.		
Required inputs are consistently complete before implementation begins.		

Readiness score: ___ / 9

What your score tells you

8–9: Customers are generally ready

The organization has a defined approach to preparing customers for implementation, and delivery isn't routinely discovering missing requirements after kickoff.

6–7: Readiness is uneven

Some customers arrive prepared, while others require significant work before implementation can really begin. The process may depend too much on Sales, the implementation manager, or the individual customer.

3–5: Readiness is a constraint

Customers are likely entering implementation before the conditions for success are in place. Delivery may be spending time chasing information, coordinating resources, or solving problems that should have been addressed earlier.

Dig deeper

Look at your last 10 implementations.

- What did each customer need to provide before implementation could begin?
- Which of those requirements were actually confirmed before kickoff?
- How often did your team discover missing information or dependencies after kickoff?
- When a customer wasn't ready, who absorbed the impact?

Pay particular attention to work that gets labeled an "implementation delay" even though the implementation team couldn't actually proceed.

If your team regularly waits for customer inputs, decisions, technical access, or internal stakeholders, the constraint may be readiness rather than delivery capacity.

What to do next

Define what "ready for implementation" actually means. Identify the customer inputs, decisions, resources, technical requirements, and internal ownership needed before kickoff. Establish who confirms them.

The objective isn't to make customers jump through more hoops. **It's to start implementation when the conditions exist for it to move.**

3. Throughput

Can your delivery system handle more work without simply working harder?

A delivery team can be fully staffed and still struggle to move customers through implementation.

Work gets stuck. Projects wait for internal resources. Teams switch between priorities. Rework consumes capacity. Go-lives slip. And when sales increases, the instinct is often to add people before understanding what is actually limiting throughput.

The question isn't simply how much work your team can do. It's **how much work can move through the system reliably.**

Assessment	Score	Evidence / Metric
Delivery capacity matches the current sales pace.		

Go-live dates are missed rarely and for known reasons.		
Implementation delays are tracked and reviewed.		

Throughput score: ___ / 9

What your score tells you

8–9: Throughput is relatively strong

Work is moving through the delivery system at a pace that generally matches demand, and leadership has visibility into significant delays.

6–7: Throughput is uneven

The team can handle the current workload, but bottlenecks, delays, or capacity pressure appear when volume or complexity increases.

3–5: Throughput is a constraint

Work is accumulating or moving inconsistently. The team may be at capacity, but you may not yet know whether the problem is staffing, workflow, dependencies, rework, prioritization, or something upstream.

Dig deeper

Look at your current implementations in progress.

- How many are actively being worked versus waiting?
- Where are projects spending the most time?
- How much capacity is consumed by rework, escalations, or work outside the normal process?
- What happens to delivery when several customers need the same resource at the same time?
- If Sales brought in 25% more customers next quarter, what would break first?

Then ask what is causing that constraint. Is it implementation capacity? Customer readiness? Technical resources? Product dependencies? Rework? Scheduling? Unclear ownership? Something else?

The goal is to identify what actually limits the amount of work your delivery system can move, rather than assuming the answer is simply more people.

What to do next

Map demand, work in progress, capacity, and constraints. Identify where work accumulates, what prevents it from moving, and which constraints are structural versus temporary.

The objective isn't to maximize utilization. **It's to create enough flow that growth doesn't automatically create delivery bottlenecks.**

4. Operating Model

Is the way your teams work together designed to produce the outcome you need?

As companies grow, the work between Sales, Professional Services, Customer Success, Product, and other teams becomes more complex.

The problem isn't necessarily that any one team is doing something wrong. The problem can be the way the work is divided, handed off, prioritized, or escalated across the organization. A process that worked when the company had fewer customers, simpler implementations, or a smaller team may no longer work at its current scale.

Assessment	Score	Evidence / Metric
We have a defined implementation process that can be repeated across customers.		
Professional Services, Sales, Customer Success, and Product understand their roles in getting customers to value quickly.		
Ownership is clear when an implementation gets stuck or moves off plan.		

Operating model score: ___ / 9

What your score tells you

8–9: The operating model is relatively strong

The major teams and responsibilities involved in implementation are reasonably well defined, and the organization has a repeatable way of moving customers toward value.

6–7: The operating model is uneven

The basic model works, but it may depend too much on individual people, informal relationships, or workarounds when something falls outside the expected path.

3–5: The operating model is a constraint

The way work is structured may be creating friction that looks like a delivery problem. Teams may be working hard and still stepping on each other, waiting for decisions, duplicating work, or leaving important decisions without a clear owner.

Dig deeper

Map what actually happens from sale through first value.

For each stage, identify:

- Who owns the outcome?
- Who does the work?
- Who makes the decisions?
- What gets handed from one team to another?
- What happens when something doesn't go according to plan?

Then look for the places where work crosses organizational boundaries.

- Where does Sales need delivery to make an exception?
- Where does delivery depend on Product?
- Where does Customer Success receive a customer that isn't ready for the next stage?
- Where does everyone assume someone else owns the problem?

A strong operating model doesn't mean every situation has a rigid process. It means people know who owns the outcome, how decisions get made, and how the work moves when reality doesn't match the plan.

What to do next

Map the customer journey across organizational boundaries and make ownership explicit.

Identify where responsibilities, decisions, or handoffs are unclear. Then determine which changes are needed in the process, roles, decision rights, or cross-functional operating rhythm.

The objective isn't to create more process. **It's to make the way the organization works together support the customer outcome it needs to produce.**

Part 2: What does your score tell you?

Your overall score

Add your scores from all four categories:

Overall score: ___ / 39

Your overall score gives you a sense of how well your current delivery system is positioned to support growth.

It is not a diagnosis by itself. Your category scores tell you where to look; the exercises in the next section help you determine what is actually happening there.

33–39: Delivery is keeping pace

Your delivery system appears to have a solid foundation across predictability, readiness, throughput, and operating model.

That doesn't mean there are no problems. It means you have fewer obvious signs that delivery is currently limiting growth.

Look at your category scores anyway. A relatively low score in one area may point to a constraint that isn't causing significant pain yet, but could become one as volume or complexity increases.

25–32: Friction is showing

Some parts of the delivery system are working well while others are creating delay, inconsistency, or rework. This is often where growth starts exposing weaknesses that were manageable at a smaller scale.

Use your category scores to identify where to investigate first. Then look at the actual customer journey to determine whether that weakness is creating a meaningful constraint.

18–24: Delivery is under pressure

Your scores suggest that the delivery system is already showing meaningful signs of strain. You may be seeing longer implementation cycles, slipping dates, growing work in progress, increasing rework, or greater dependence on individual effort.

Don't assume the answer is more capacity. Use your category scores and the exercises that follow to determine where the pressure is coming from and what is causing it.

13–17: Delivery may be constraining growth

Your scores indicate significant pressure across the delivery system.

At this level, adding more sales without addressing underlying constraints can increase the amount of work entering a system that is already struggling to absorb it. The answer may eventually include additional capacity, but first determine whether the current model is creating avoidable work, poor handoffs, customer delays, bottlenecks, or other constraints.

The goal isn't to fix everything at once. It's to find the constraint that matters most.

Part 3: Find the leak

Your overall score gives you a sense of how much pressure your delivery system may be under. Your category scores tell you where to look.

Now let's look at what is actually happening.

Find where time is going.

Take one recent customer—ideally one that took longer than expected—and map its journey from contract signature through first value. For each interval, record how many days elapsed between the two milestones, what was happening during that time, and whether the work was moving forward or waiting on something or someone.

Milestone interval	Elapsed time	What happened during this time?	Who/what were we waiting on?
Contract signed → handoff			
Handoff → customer ready			
Customer ready → kickoff			

Kickoff → configuration			
Configuration → confirmation/testing			
Confirmation → go-live			
Go-live → first value			

Now look across the intervals. Where did the most time accumulate? Where was work actually moving, and where was it waiting? What were you waiting on most often?

Look for patterns

Review the reasons for waiting and circle the ones that appear more than once:

- Customer not ready
- Missing information
- Internal approval
- Sales clarification
- Scope uncertainty
- Product dependency
- Technical dependency
- Resource availability
- Customer scheduling
- Rework
- Unclear ownership
- Something else

Then ask: **Which issue accounts for the most lost time?**

And one level deeper: **Why does this keep happening?**

The biggest delay isn't necessarily the root cause. For example, if customer scheduling accounts for a large amount of elapsed time, the underlying issue might be that scheduling isn't addressed until after kickoff. If missing information appears repeatedly, the issue might be customer readiness, the Sales handoff, or unclear requirements.

You're looking for the underlying constraint, not simply the longest interval.

What did you learn?

Consider:

- In what part of the customer journey do you lose the most time?
- What is the most common reason work stops or slows down?
- What does the underlying issue appear to be?
- What would be the biggest impact from solving that issue?

You aren't trying to explain every day of an implementation. You're trying to identify where time is accumulating and what is causing it. That gives you a much more useful starting point than simply knowing that implementation takes "too long."

Part 4: Stress-test your growth

Your delivery model may be working today.

The more important question is whether it can support where the business is going.

The 25% Test

Imagine Sales brings in 25% more implementation work next quarter.

Don't try to calculate whether you have enough capacity using a simple customers-per-implementation-manager ratio. Implementation work isn't evenly distributed. A customer may require relatively little effort during one phase and consume significant team capacity during another. The same is true of technical resources, Product, testing, and other dependencies.

Instead, look at where work is concentrated today.

Where does the work pile up?

Think about your current implementations.

- Which phase typically requires the most team capacity?
- Which roles or resources are most involved at that point?
- How many customers are typically in that phase at the same time?

- What happens when several customers reach that phase together?

Now apply the 25% test:

If 25% more customers entered the system, where would the additional work show up first?

- Customer readiness
- Kickoff capacity
- Implementation manager capacity
- Technical resources
- Product availability
- Configuration
- Testing / confirmation
- Go-live capacity
- Customer Success handoff
- Another point in the process

Look beyond Professional Services. The constraint may not sit inside the implementation team. Ask:

- Would Sales need to change what or when it promises?
- Would customers need to arrive more prepared?
- Would Product or Engineering become a bottleneck?
- Would technical resources need to be scheduled differently?
- Would several implementations compete for the same internal resource?
- Would the team spend more time on exceptions, escalations, or rework?
- Would work become harder to manage because too many projects hit the same milestone at once?

What did you learn?

Where would additional demand create pressure first?

What resource or dependency is most likely to become constrained?

Why would this become a constraint?

What would you need to change to absorb more demand without simply working harder?

If you struggled to answer these questions

That's useful information too.

You may not have a capacity problem yet. You may have a capacity visibility problem: the organization doesn't have enough understanding of how work moves through the system to know where additional demand will create pressure. You can't reliably solve a constraint you can't see.

The goal isn't to show that you need more people. It's to understand how additional demand would move through your delivery system—and where the system would struggle to absorb it.

A capacity problem may require more people. A flow problem may require a different process. A readiness problem may need to be solved before implementation starts. A cross-functional constraint may require a change outside Professional Services altogether.

Part 5: What Kind of Problem Do You Actually Have?

Your scores gave you a place to look. The customer journey showed you where time is going. The 25% test showed you where additional demand could put pressure on the system.

Now put those pieces together.

Look for the connection

Go back to your category scores and the findings from Parts 3 and 4.

- Which category had your lowest score?
- What did you find when you mapped the customer journey?
- What did the 25% test tell you about where additional demand would create pressure?

Now ask:

Do these findings point to the same underlying issue?

If they do, you may have found an important constraint.

If they don't, that's useful too. Your lowest score may identify a weakness, but not necessarily the issue currently limiting growth.

What kind of problem are you seeing?

Predictability:

You don't have enough visibility into how long the customer journey takes or why outcomes vary.

Readiness:

Customers are entering implementation without the conditions, information, decisions, or resources needed to move forward.

Throughput:

Work is accumulating because capacity, dependencies, rework, prioritization, or workflow limits how much can move through the system.

Operating Model:

The way work is divided across teams creates friction, unclear ownership, inconsistent decisions, or unnecessary handoffs.

More than one:

The issues are interacting. For example, inadequate readiness may create rework that consumes capacity, while unclear ownership makes the resulting delays harder to resolve.

Your conclusion

Consider:

- The biggest constraint you see today
- The evidence that led you there
- What you can expect if growth continues without addressing it
- The first thing you need to understand or change

One final check

Before deciding that you need more people, more technology, or a new process, ask:

What is the underlying constraint—and what evidence do we have that we've identified the right one?

That's the question that turns an implementation problem into a business problem you can actually solve.

Part 6: What This Looks Like in Practice

The same delivery problem can look very different from the outside. Here are four examples of what happens when you look beneath the headline.

Scenario 1: "Implementation is slow."

A SaaS company reports an average 65-day implementation.

Leadership assumes the implementation team needs to move faster.

A closer look shows:

- 10 days between contract and receipt of required customer information
- 8 days waiting for an internal decision
- 7 days between sales handoff and an actual kickoff
- 40 days of implementation work

The implementation team may not be the primary constraint.

The 65-day customer journey includes more than the time spent actively implementing.

The useful question isn't: "How do we make implementation faster?"

It's: **"What accounts for the difference between the work required and the time it takes the customer to reach value?"**

Before changing the implementation process, understand where the elapsed time is coming from.

Scenario 2: "We need another implementation manager."

Sales is growing quickly and the implementation queue is increasing.

The easy answer is another implementation manager. But the current team is spending substantial time:

- Chasing missing customer inputs
- Reworking configurations
- Resolving scope questions that should have been settled earlier
- Waiting for Product
- Managing projects that were scheduled before customers were ready

Adding capacity may help. But if the underlying system remains unchanged, the company may simply create a larger version of the same problem.

The useful question isn't simply: "How many more people do we need?"

It's: **"What is consuming our current capacity, and which of those demands would increase as sales volume grows?"**

If avoidable work is consuming capacity, fixing that may matter as much as adding capacity.

Scenario 3: "Our average TTV is fine."

The company's average time to first value is 58 days. Leadership considers that acceptable.

But when the data is segmented, similar customers range from 32 to 115 days.

The average is hiding substantial variation. Customers aren't experiencing a predictable journey.

The useful question isn't: "Is 58 days good or bad?"

It's: **"Why does one comparable customer reach value in 32 days while another takes 115?"**

The answer may reveal differences in customer readiness, implementation complexity, process, resources, or other factors.

The average tells you what happened.

The variation may tell you why.

Scenario 4: "Everyone is doing their part."

Sales closes the deal. Professional Services starts implementation. Product handles technical questions. Customer Success takes over after go-live. Each team can point to the work it completed. And yet the customer experiences delays at every handoff.

No single team has clear ownership of the complete journey from signed deal to value.

The problem may not be any individual team's performance. It may be the way work moves across the organization.

The useful question is: **"Who owns the customer outcome when the work crosses organizational boundaries?"**

The solution isn't necessarily better performance from any one team. It may be clearer ownership, better handoffs, different decision rights, or a different way of coordinating the work.

Part 7: What to Do First

Don't try to fix everything. Start with the constraint that is creating the most customer delay, consuming the most capacity, or creating the greatest business risk.

Your assessment should give you a starting point.

If Predictability is the issue:

Focus on understanding **where variation enters the customer journey**.

Look at the difference between customers who move through the process smoothly and those who take significantly longer.

The goal isn't simply to reduce the average. It's to understand what makes outcomes unpredictable.

If Readiness is the issue:

Focus on **what needs to be true before implementation begins**.

Look at the customers who started smoothly versus those who spent time stalled, chasing information, or resolving prerequisites after kickoff.

The goal is to determine what should happen before delivery takes ownership.

If Throughput is the issue:

Focus on **what is consuming the system's ability to move work forward**.

Look at where work accumulates, what resources or dependencies constrain movement, and how much effort is going into rework, exceptions, and coordination.

The goal is to understand the constraint before deciding that more capacity is the answer.

If Operating Model is the issue:

Focus on **where work breaks down between teams**.

Look at the points where ownership changes, decisions are required, exceptions arise, or one team is waiting for another.

The goal is to determine whether the way the organization works together is creating problems that individual teams cannot solve on their own.

If more than one area is under pressure:

Don't try to solve all of them at once. Ask: **Which constraint is creating the most downstream impact?**

Fixing that one may change what you see elsewhere in the system.

The objective at this stage isn't to redesign your entire delivery operation. It's to identify what needs attention first, why it matters, and what you need to understand before deciding how to fix it.

Part 8: Share This with Your Leadership Team

You don't need another meeting about whether "implementation is a little busy." Ask the actual questions:

1. How long does it take from contract signature to kickoff?
2. How long does it take from kickoff to go-live?
3. Where does the customer journey take longer than expected?
4. What appears to be causing that additional time?
5. If sales increased 25% next quarter, where would the system come under pressure first?
6. What evidence do we have that we've identified the underlying constraint?

If the answers are estimates, opinions, or different depending on who you ask, that's useful information. It may mean the problem is visibility and predictability, not execution.

More importantly, don't jump directly from the problem to a solution. Before deciding to hire, change the process, add technology, or push the team harder, make sure you understand what is actually limiting the customer journey and the business.

The Takeaway

Growth doesn't necessarily create delivery problems out of nowhere. It can expose places where the delivery system was already relying on workarounds, individual effort, or assumptions that no longer hold. A company can have strong sales, good software, talented implementation people, and happy customers, and still have a delivery system that isn't ready for the next stage of growth.

The question isn't simply: "Are we delivering?" It's: "Can we keep delivering predictably as demand increases?"

If your stress test revealed a constraint, start there. **Find it. Understand it. Then decide what needs to change.**

That's a much better starting point than simply adding people or pushing the existing system harder.



ClearWay Projects helps growing B2B SaaS companies identify the delivery constraints that get in the way of growth.

Need help taking the next steps? In a focused 90-day engagement, we help leaders identify and prioritize the changes that matter most, design the needed changes, and enable your team to carry them forward. The result is an operating model designed to make delivery more predictable.

**The first step is not to fix everything.
It is to find the constraint that matters most.**

clearwayprojects.com

contact@clearwayprojects.com