CHAPTER SEVEN

10 CRITICAL STEPS FOR CHANGE AGENTS

This Chapter (Chapter Seven: 10 Critical Steps for Change Agents) is freely available as an excerpt from *The Process of Improvement.* You may freely download and distribute this chapter (in various electronic forms) from https://www.TheProcessOfImprovement.com, so long as the material is neither modified nor sold.

For those who have already read the rest of this book, several topics in this chapter will resemble those we have already defined for organizational leaders. That is by design. You will be doing many tasks similar to those undertaken by the leader of a process improvement effort, but your role will be to dig deeper into details and specifics.

The leader of a process improvement effort should focus on communicating the why, engaging various stakeholders and stakeholder communities, and providing a clear source of inspiration and a commitment to ongoing support during the change process. Those items focus on facilitating high-level outcomes.

On the other hand, change agents will need to focus on the nuanced specifics like the data input requirements, the artifacts that need to be collected, the distribution and presentation of decision support information or contribution requirements (i.e., the exact bits of work we need some

individual or team to do at a specific stage), and the analytics required to see if our incremental changes are leading to the desired outcomes.

You are likely reading this because your organization's leadership has seen innovative potential in you and appointed you as a change agent. We already know you want your part to go smashingly! So, what should you do to facilitate implementing changes that make a difference and last? Here are the key steps to consider.

Step 1. Define the key obstacles and bottlenecks within the first process.

Core question: What *specifically* needs to change to move our process closer to the requirements or objective outcomes stated in our project's charter?

If mapping out your process is like playing chess, mapping out the obstacles and bottlenecks is more like 3D chess. So, before we head into the pragmatics of discovery, design, documentation, and deployment, let's first examine what we are heading to find.

We are not only trying to determine the typical flow of information related to work; we are also trying to determine the specific obstacles and bottlenecks, such as:

- **Information Obstacles:** Areas where information is not available, incorrect, unreliable, or where upstream stakeholders or contributors haven't provided the data needed for decision support or next steps.
- **Resource Obstacles:** Areas lacking the people, machines, finances, computing, validating, or finishing resources necessary to complete a step or handle some specific part of the process.
- **Human Bottlenecks:** This may be related to the obstacles above because someone who does not have the correct information available may slow down the process while they try to determine what to do. There are also circumstances where someone lacks the training to quickly accomplish a particular step while consistently producing the desired outcome. While you may find areas where you suspect someone is not doing their job, it would be wise to wait to draw that conclusion until you have determined whether or not

there are information or resource obstacles at play. While laziness may cause human bottlenecks occasionally, it is not uncommon to find a competent person withering in the face of broken processes. An effective process improver can often restore them to maximum productivity by quickly resolving those core issues.

- **Opacity obstacles:** Unclear processes lead to unclear results. The difference between a low-, medium-, or high-performance team often comes down to how quickly those teams can process or manage exceptions. In a rigid process, it may be necessary to clarify for employees when a service request does not fit the system's process. You should inform your users when they are authorized to manage the exception outside of the process. In other cases, handling potential exceptions and events inside the process may be critical.
- **Alignment obstacles:** A simple example of an alignment obstacle might be a utility company's new service installation team setting the objective to turn new clients up within 60 days, while the quality control team that approves these installations has created a new quality assurance process expected to take 90 days. Again, you do not need to solve all these problems at this stage. You need to make sure that you identify them. Issues of this type that impact different teams or departments will most often need to be escalated to the leaders of those departments or higher to be solved. However, that is a better problem to run into early. If you have not identified them in the discovery phase, they will undoubtedly surface when you deploy your new system, making your deployment look like a failure.

It is also critical to ensure that you, as a change agent, are aware of and focused on the objective outcomes of the new process. Your job is not simply to discover how your contributors are doing work today but to study it with that three-dimensional view to determine which forces are pushing towards the new objective and should be enhanced and which forces are pushing against it so you can resolve those hindrances.

Let's dig further into these obstacles to provide some examples of the challenges you might find and the information you must capture in this step to help overcome them. We offer several low-tech discovery and data capture tools and examples at the URL: https://www.TheProcessOfImprovement.com

- **Information obstacles:** Here, detail matters. You must gather and inventory all the needed fields of information. That is an opportunity for aggregation or disaggregation, consolidation, and organization.

  For example, you may discover that you have been storing all customer names in a "name" field. But you have already heard that marketing wishes you had separately stored first names so that they could set up automated email notifications to clients that would address them by their first name. So, you will need to disaggregate the name data. Perhaps you would record that the existing system includes the field called "name," but you would add "first name" and "last name" as required data capture fields and "middle initial" as a suggested field for the new system. You don't need to work through how to get that information out of the current system or whether someone will need to do that manually for new records only. That is something that the data experts can work on for you. You simply

need to determine that this is an information obstacle that could be resolved by this means.

Another example might be that someone must decide about client investments based on various forms. Still, those forms are in a client information directory that is cumbersome and time-consuming to access. You might record that making those decision-support files directly viewable within your new system would help resolve the information obstacle. You don't have to know how to integrate information systems that way; that is up to the developers or the team who configure your solution in a no-code environment. Again, your job is simply to determine whether this information obstacle can be resolved in this way.

- **Resource obstacles:** You may discover significant bottlenecks in your investment client onboarding process because legal team members must review certain disclosures your prospective client has prepared. As someone focused on the customer service portion of this process, you may wonder if it is your job to try to address this. Good news! It isn't. You're not responsible for whether that bottleneck gets resolved, but you are responsible for identifying and documenting it. By raising that issue as a change agent within your team, you ensure that the information will make it onto the radar of people who may be able to change that or will understand that they may need to modify the objective outcomes if they cannot change that other part of the process. There may be a simple opportunity, like training people in another department to review these documents when they are all in order and only sending the complex exceptions to the legal team. That may work in some organizations and under some

regulations, while in others, it will not. At the risk of being redundant, you're not responsible for solving this problem; you are simply responsible for identifying, documenting, and escalating it.

You will gain empathy and trust by listening to the people in the team you are working with about how this obstacle impacts them. It is essential in these scenarios where you cannot guarantee the change will be made that you disclose, "I do not have a magic wand; I only have my pen, but I will use it to its greatest effect."

- **Human bottlenecks:** It is surprisingly common while doing process discovery to realize that part of the problem with the existing system has nothing to do with the existing system. In fact, one of the most common incorrect assumptions made by organizations is that new people who have joined the team have learned how things are done from the people who were already there. Some organizations are excellent at training, but most have more significant gaps than they realize. Effective training is not measured by whether or not that training gets completed, how long it takes, or how frequently it is repeated. Effective training is measured by whether or not the people trained have retained and can apply the information.

  Many organizations provide significant training on how systems work. Still, that training often serves people who have not previously done any of the work nor had any experience with the system. In this case, research suggests that the average trainee will only retain 10% to 20% of the information. When they arrive at their new job, freshly trained, they will encounter exceptions they do not know how to resolve. They will look to a coworker for advice, but that coworker

may also have retained only 20% of their training. So, they will learn about the organization's many organic workarounds instead of the official system. Some of those were healthy workarounds that dealt with a lack of capability in the prior system. Still, many will be workarounds addressing the fact that people do not know what to do within the system. Unfortunately, they will have learned how to do things that create new obstacles or bottlenecks elsewhere.

When you discover human bottlenecks where correct procedures are not being followed, have not been provided, or are being avoided at the expense of downstream exceptions, you must empathize with the user experience at each stage. Ask for significant detail about what might have made it intuitive for them to know what to do next or how to solve what they view as exceptions within the system. In other words, you should have a good library of questions like, "is there any information that someone else could have included in this record to make it easier for you to solve this problem?" Note that this question format externalizes the root cause of the problem. That makes your user feel safe to give productive suggestions about what could have made their job easier.

The data you gather as responses to those questions will often correlate with information or resource obstacles. You may discover that someone upstream had to hurry through something to meet a quota. When gathering specific information needed for a process's later stages, you may find that data collection should have been made mandatory at an earlier point. You may also find out that everything a user needed was provided, making it even more critical to study why they cannot

find or incorporate that information. What you are documenting is a combination of the needed information and resources and any obstacles they had with accessing, analyzing, or leveraging those resources.

- **Opacity obstacles:** "When the leader is clear, everything is clear." -Ken Gosnell. The same can be said for processes. When the process is clear, everything is clear. Managers may have trouble making resource decisions because they cannot see the bottlenecks. Thus, a bank of questions designed to reveal the information managers need can be helpful. Remember the same technique: "What could *someone else* have done differently, and what information could have been provided to you that would help you manage your team and resources more effectively?"

The other side of this equation is that you will be doing discovery work that will often expose where things go off the rails, and people will have to make decisions independently. Wherever possible, it is ideal to gather all the information that could have avoided the problem in the first place and ensure it is at the fingertips of the person who is accomplishing this part of the process. However, there may be exceptions that cannot be easily preconceived. For instance, in a loan approval system that contemplates the ratio of debt to income for a person or company, there will be no effective means for systemic calculation if someone who has recently funded a startup has $10 million in the bank, no debt or credit history, and is simply looking for a $1 million loan to build a new building. Upon quick consideration, one might say, "this is a no-brainer. Loan them the money!" But is that correct? What if the organization has already pledged its assets?

Will that building be worth anything if they don't stay in business?

A traditional underwriting process would address many of these items. However, even if all the needed information conveys, there are still times when someone will need to be able to override the requirement for a specific approval ratio subjectively. Perhaps you could say that any credit analyst should be able to ascertain the creditworthiness of the situation.

While you're discovering and identifying opacity obstacles (where a lack of clarity around problem resolution causes slowdowns), you may decide that your organization wants to contemplate making loans in these exceptional circumstances. However, you want the risk to be tiered. In other words, a credit analyst can approve the exception if the risk is less than $1 million. If the risk is between $1 million and $10 million, you may want an officer of the company to approve the transaction. If the risk exceeds $10 million, you may wish to have a subcommittee of the Board of Directors approve the transaction directly. Suggesting a simple laddered exception management routine like this does not make you responsible for approving those loans; it is simply your proposal to clarify a streamlined path forward where operational opacity emerged.

As another example of an opacity obstacle, an expense report that does not meet predefined criteria can cause significant obstacles and bottlenecks in a large organization. Questions will quickly arise like, "Who should we send this to?" or "Who can approve this amount?" Both questions would seem relatively easy to resolve, but are they?

Should an atypical expense report be escalated to a manager within the finance team who approves other general expenses? Or should it be sent to the manager of the person who submitted the expense report? Is there a specific dollar threshold that should trigger higher escalations? Is getting the person's manager to approve the expense sufficient if the expense does not match one of the year's predefined budget categories?

Over time, the answers to these permutations could become predefined process rules. When a new system goes live, especially when the process is more complicated than an expense report, the system designers likely have not considered all potential exceptions. Therefore, it is incumbent upon the person doing detailed discovery of a process to identify that potential opacity as a bottleneck. That is an opportunity to replace opacity with clarity.

For instance, you could predetermine that any expense not meeting predetermined rules, levels, or categories should be routed to a finance leader who will either be authorized to decide how to handle that particular expense or able to reassign it to someone who is authorized to approve or decline the specific type of exception. As a lightweight first step, you could suggest that any such escalation action should automatically create an exception report that gets auto-routed back to the process improvement team. That would ensure accountability around the exception and that the process improvement team would study the exception later, empowering continuous improvement.

Remember: You do not need to solve all these problems right now. You simply need to identify them.

Step 2. Define process contributors, their roles, and contributions

Core question: Who should help me gather the details needed for my part of this process improvement effort?

Your process or subprocess contributors are the people who do the work you are looking to improve, automate, or streamline. If at all practical or possible, we strongly advise you to engage a subset of these contributors to assist you with your forthcoming detailed discovery work. This part of the guide assumes that you are working on a project for a relatively large team of knowledge workers, engineers, or other intellectual contributors who can assist you with your discovery and subsequent documentation efforts. If that is not the case, and you will be the sole discovery resource, we still believe you will find reviewing our guidance on this step helpful.

Ascertaining who the "contributors" are or "who will need to be involved" might seem easy. Most would contemplate the "who" in terms of the process itself. In other words, let's say our widget or item of work is an "X." It would be typical to think we will need the person who receives an "X" and the person who processes an "X." If there are exceptions, we would probably assume we will also need to include the person who handles/reroutes/decides upon the exception handling for "X."

That's good thinking, but it is not all that we mean. When we say 'define who is involved,' we mean 'who will be involved in the discovery, testing, communication, early adoption, and training for your specific part of the process.' Your efforts leaders have recruited you as a change agent. However, effective change agents know they need to foster high engagement to achieve high adoption, which is the key to success. Your objective is to assemble a list of stakeholders for the part of

the process you are examining. They do not need to be process improvement professionals, but they do need to be professional and knowledgeable regarding the process areas you want them to help you examine.

The people involved in your discovery should include the process owner and key contributors. Your team should consist of people who are logical and systematic thinkers. In Chapter 6, we discussed the Working Genius Assessment. Your discovery team should include people with the "W" (Wonder) profile as they are the people who will be comfortable asking the question, "Why do we do this?" That's important because not everything an organization does needs to be done, and this is the time to ask. Additionally, the answers to "why" questions will often lead to a greater understanding of the artifacts and information that must be gathered along the way.

To ensure you don't end up in a rut of simply *automating* the dumb way you've always done something, look for people with the "I" (Inventor) profile. People with this attribute don't just hear about how a thing happens and imagine it being made repeatable but are comfortable asking, "Is that the best way to do it?" They are also natural fountains of new and better ideas.

Be forewarned that these human idea factories often produce great and bad ideas on the same assembly line. That is why it is also a good idea to include someone with the "D" (Discernment) attribute, as they will naturally help you separate those items as your Inventor produces them.

While you need a room filled with talented people, you must ask them to leave their egos outside. Process experts may be offended by people who constantly ask the question, "Why?" Inventors can get their feathers ruffled when someone with the discernment gift says, "But I think that's going to cause more problems than it will solve." And likewise, that

person with the eye for what works may want to be in charge, but that is not what you brought them for. All in all, you need to help them see themselves as a team that will share credit. It is advisable to set this as a ground rule: "We are going to fight like hungry dogs in this room until we are confident we all agree. Then, we will go out there and win together and take credit together. Everyone agree?"

In some organizations or processes it may turn out that you are the only one doing discovery, and some of the following remarks on how to prepare your people to do discovery may seem not to apply. You can use these best practices if you are the only one mapping out your process or part of the process. We will work from the assumption that you will empower others to do at least part of your discovery.

Once your discovery team is designated, informed, fully committed, and officially commissioned, turn them loose and let them get started. *They* should ask questions that lead to a highly detailed understanding of what must be captured as the work within a process flows through your organization and what must be displayed (or made ready to analyze) to ensure the right decisions are made. Open-ended questions are best. Binary questions are the worst. In discovery, something that could be answered with a yes or no should be considered an immature question until a thorough understanding of the process has been ascertained.

The next team you want to assemble will be the testing team for your part of the new system or process. These may be a subset of the people you put together for the discovery team. Still, the best testers are detail-oriented people who don't mind repetition and don't mind investing a lot of time in what might seem like uninteresting work to your people who have the wonder or inventor attributes (from the working genius profile). People who are good testers will need to be

bright, will need to know that their work matters towards producing an excellent outcome, and will need to have their expectations set correctly. You are not engaging them to test a great new solution; you are engaging them to help prevent a terrible one from being delivered. And while we are on the topic of Working Genius profile types, the "E" (Enablement) attribute will lend itself well to your selection of testers, and a "T" (Tenacity) profile will be a great find to help you manage the testing.

A great testing team will slog through a lot of mud and help you work the kinks out of early concepts. If you don't have a great testing team, then engage your end users to become your testing team. If those end users don't believe they signed up for testing, they will be unhappy and likely tell people that the experience is rough. The people they tell will translate that to mean, "the new system is bad." You want to avoid that at all costs, so remember that getting your detail-oriented and highly invested team of testers together is critical to your success. Tell them how much their great work will contribute to the future of this new and better process. Why? Because it's true. A well-tested solution comes out of the gates strong, and that will make a world of difference when you get to the critical phase of end-user adoption.

Step 3. Provision Real-Time communications

Core question: How will your team quickly communicate and resolve critical issues to keep your contributors and stakeholders engaged?

Few things can negatively impact a team's engagement as effectively as slow responses or poor communications. I worked with a highly successful vice president who often said, "A rapid response is the best response to any problem." His point was not to suggest that a quick phone call would resolve a big problem but would quickly terminate the potential perception that a big problem was being left unaddressed.

Over the years, his frequent quip has proven exceedingly true. There will be problems along the way in any change effort. For example, key stakeholders won't remember critical things after your meeting has ended. If they don't know how to tell you immediately, they may never tell you until that critical detail harms your final deployment. Likewise, if you gather a team to assist with testing proposed changes, and they report a problem but perceive that the problem report goes into a black hole, they will likely disengage. The result? You will not be getting the high-quality input that you need.

Your stakeholders roughly fit into two groups. On the one hand, there are "insiders" who are part of your efforts and will be "in the know" and, therefore, aware of changes as they occur. On the other hand, there is your "audience." Your audience stakeholders will be affected by your new process or solution but will not help to "drive the train." Your communication with and to these groups will be different.

The proven need for real-time collaboration for your "inner team" is why we want to ensure you build instant or near-instant communications into your process improvement effort.

- **Collaboration Software:** If you do not already have a team or work management system that includes collaboration, a quick Internet search for "collaboration software" should yield many options. Your team probably already uses an application in that category (Slack, Teams, etc.). Perhaps you need to set up a private channel within your company's collaboration application. If that is not an option for you, and there are no approved applications you can use for real-time communications, set up an email distribution group at the very least. The point is to ensure that you have created a relatively quick (and, if possible, real-time) means for your process improvement team members to collaborate and ensure that everyone is on the same page.
- **Team-wide inbound:** You also need to ensure that your extended stakeholders (those not on your team or in your inner circle) have an easy-to-use means for reporting or escalating problems, questions, concerns, or late discoveries.
    - That could be an email address (team1@...) or a separate slack channel for your team (not your internal "back channel"), etc.
    - Set clear expectations for how quickly your team will respond to new communications in that "inbox" and communicate those to your team. You should also decide who is responsible for doing that and who will be the alternate if that person is unavailable.
    - Treat all inbound communications as critical until they have been triaged and responded to. Remember, "a rapid response is the best response to any problem."

- **Audience-wide outbound:**
    - **Critical:** Critical issues will impact a large group or make all or some vital part of a system unusable. Critical issues must be addressed with urgency. Contact information should be gathered for all key stakeholders so critical events or changes (outages, impacting errors in a new system, etc.) can be immediately communicated to the appropriate parties. When these critical issues arise, ask someone to pick up the phone and call the key stakeholders. Ask those stakeholders to make others aware of the problem and the active effort to resolve it. Then, ALSO ensure that someone quickly follows up with an email or other real-time communication to the remaining stakeholders.
    - **Non-Critical:** Considering who your changes could impact in advance is instrumental. If possible, gather the email addresses of those audience stakeholders and set up an email forwarder (or some other means of broad communication) to inform them of any non-emergency or limited-impact issues.
    - **Updates:** It's time to decide on the specifics of your communication plan. In other words, how often will you update your audience stakeholders? It may not be one size fits all. Perhaps you have a department that will be impacted by some of the changes you will make at milestone three. You may want to keep them on all updates until you reach milestone three. You may have a leadership team that has given you a budget, and perhaps they don't have the time

for the weekly updates you've decided to give to your other stakeholders. Maybe once a month is good for them. You will determine whether they get a monthly version of the same report or whether you want someone to generate a summary version for them.

You should consider and document your project's communication plan details beforehand. It would be best to let your audience know why these reports will be meaningful to them and why they should read them. If you have leaders or stakeholders who will not or do not have the time to read those updates, then perhaps you should determine who on your team will make time to stop by or call stakeholders and update them when each of these reports comes out. That is often a critical step or misstep.

An example of a quick periodic update follows:

Today (15 Jan 2026), the team working on the new solution for process X met and made the following decisions:

1. Insufficient data is available to determine how many cases get escalated per week. Therefore, we will allow cases to be escalated manually, run reports on this, and revise the solution 30 days after go-live. We decided to make this easy for the end users by simply giving

them a check box to click when they want a case escalated and a text box next to that to enter the reason for escalation. We will include text on the screen to let them know that the audit trail will record who made the escalation and when, along with the information they entered in the text box.

2. The team decided that by deferring the decision and solution development regarding the aforementioned escalation problem, the project could move the go-live date forward to February 5th.
3. If you have any concerns regarding these changes or reasons the changes should be reconsidered, please be sure to communicate those to *sally@corpdomain.com* by Thursday, 18 January.
4. The testing team has received a draft prototype of the solution this morning, and another report will be delivered next Monday morning to include the early results from that testing.
5. There are two open issues:
   a. issue number one description
   b. issue number two description
6. This process improvement project is currently on time with no risk of being behind deadline or over budget.

The noteworthy elements of the above update are:

- updates on any decisions recently made
- updates on the current status
- updates on the next steps
- a running list of open issues or concerns
- a general statement on the health or risk level of the project (Color coding or links to supporting data and dashboards are helpful if you can support that in your environment)

- **Communications Tone:** If your portion of a project has suddenly shifted to risky, overdue, or possibly going over budget, your stakeholders responsible for the outcomes may feel fear, sadness, or even fury when they read the update. Make sure the people on your communications team are thoughtful communicators. Taking the extra time to detail why a project has become "at risk" and including information about everything the team is proactively doing to get it back on track will help significantly. In addition, and as mentioned above, this is one of those times when verbal communication is vital.

  If you know a stakeholder is likely to be personally or negatively impacted by a communication about going out, make sure someone on the communication team talks to them personally. Let the email come out as a follow-up to that conversation.

  Good news can also have a negative impact. What!? Really. Suppose you plan to be on vacation the week before a new system will go live. If someone suddenly sends an email telling you that it will go live the week you are on vacation, you will imagine your

team running around with their arms waving in the air while your boss remembers that you are taking it easy while they catch the flack. Remember, any change stakeholders haven't already bought into is potentially negative. That's why our sample above included a point of contact for the reader of the update to reach out to with concerns or issues. It's also why we chose the words, "If you have any concerns regarding these changes or reasons the changes should be reconsidered, please be sure to communicate those to…".

Step 4. Define the high-level current and future states for your target process

Core question: What problem are you trying to solve? How will things improve once fixed? How will those improvements help the contributors and stakeholders you will be engaging?

The leader who asked you to read this chapter (or excerpt from *The Process of Improvement*, if you have downloaded just the excerpt) has probably read the entire book. If so, they have probably also communicated what part of a larger process improvement effort you will address. Hopefully, they have clearly articulated the overall problem, its negative or costly impact on your organization, why it needs to be changed now, and what part they ask you to help address. We also expect they have articulated the future state and how that will help you and your organization.

We are now asking you to do roughly the same thing. Consider the portion of the overall problem you will engage with, then consider how you would answer the core question(s) at the beginning of this step. Anyone impacted by your work must receive a clear statement as to why their inconvenience is worthwhile in light of the future benefit their team will experience and why it will improve their work or lives.

Putting this into a PowerPoint slide or equivalent would be a good idea. Type it on a document and paste it on the wall near your phone. Read it frequently. Memorize it. Be ready to recite it on demand. It would be best to consider this a pre-requisite calling card for all your stakeholder engagements.

Here is a basic example:

- Current State (As-Is):
    - One of our largest competitors has promised investors they can be onboarded and receive dividends within 10 days
    - We pay a slightly higher dividend, but online forums have informed prospective investors that our onboarding process typically takes 60 days
    - Savvy investors realize they would need to stay with us for two years to make up for the roughly two months of lost dividends
    - Therefore, we are signing up fewer savvy investors
    - If this continues, the company could have to cut headcount
    - Our department is currently seen as one of the bottlenecks to being competitive
    - Our team is being assembled to address the specific complexities of regulatory requirements while moving at this accelerated speed
- Future State (To-Be):
    - Tiger teams are working in seven departments simultaneously to enable the SLA of onboarding new investors in three business days
    - Marketing states that our higher dividends and faster onboarding will make us market leaders
    - The company will grow by 20% or more next year due to this competitive advantage
    - Management expects to approve doubling annual bonuses as a result of this strategic win and growth

You may want a more in-depth version for your detail-oriented stakeholders, such as:

- Current State (As-Is):
    - One of our largest competitors has promised investors they can be onboarded and receive dividends within 10 days.
        - We have done independent research and validated that they almost always meet or exceed this commitment.
    - We pay a slightly higher dividend, but online forums have informed prospective investors that our onboarding process typically takes 60 days.
        - We believe our higher dividends would attract more high-quality and savvy investors if we could at least match this onboarding time.
    - Savvy investors realize they would need to stay with us for two years to compensate for the roughly two months of lost dividends.
        - To address this challenge via raising dividends alone, we would have to increase dividends by 10-15% or institute a much higher dividend rate for the first year.
        - Additional costs would likely necessitate reducing staff levels by 8-10%.
        - As a result of our inability to address the near-term expectations of our best prospects, we are signing up fewer savvy investors.

        - Sales in the last quarter have declined by more than 18%, and the prior quarter was down by 11%.
        - Market research revealed the concerns 60 days ago, which prompted this process improvement effort.
    - If this continues, the company could have to cut headcount
        - Our CFO believes we will be able to avoid this as long as our project succeeds
        - If we fail, the company might have to consider outsourcing our team's work to provide the savings needed to raise dividends.
    - Our department is currently viewed as one of the bottlenecks to being competitive.
        - Therefore, we would likely be one of the first departments to experience cuts or find our jobs getting outsourced.
        - That is not the desire of our management team.
    - Our team is being assembled to address the specific complexities of regulatory requirements while moving at this accelerated speed.
        - We know we can do this, but not without your help.
- Future State (To-Be):
    - Tiger teams are working in seven departments simultaneously to enable the SLA of onboarding new investors in three business days.

        - Most teams are already ahead of schedule, and excitement is building that we can turn the corner and surprise our competitors with a market-leading solution to this challenge.
    - Marketing states that our higher dividends and faster onboarding will make us market leaders.
        - Our team will be a critical part of making this happen, and your name has already been in the air as someone who can help us make this happen.
    - Due to this competitive advantage, the company will grow by 20% or more next year.
        - That turns this threat into an opportunity.
        - It will also warrant an increased headcount budget to help handle the additional work, allowing us to bring on more great talent.
    - Management expects to approve doubling annual bonuses due to this strategic win and growth.

If you need to add graphics, charts, graphs, or the competitor's logo to help identify your target, do it. The goal is to connect people to the higher cause instead of just connecting them to more work or incoming work requests. Inspired people work. Overworked people just expire.

Step 5. Gather the facts

Core question: What data and artifacts will we (or our new system) need to gather to support subsequent decisions, recordkeeping or regulatory requirements, and metrics or key performance indicators?

Quick considerations:

- **End users** typically offer the most nuanced detail about what can go wrong with individual work transactions as they flow across a company's process.
- **Those who manage the workflow and contributions of others** can typically offer meaningful insights and perspectives into how errors, exceptions, and unexpected nuances get handled.
    - It is not uncommon for the perspectives of these first two groups to differ. It does not mean that one is correct and the other is wrong. They simply bring you different perspectives on the same data, but they must be reconciled early in your process.
- **Working with management to determine the key metrics and milestones** that must be recorded or observed **is critical**. Building recordation of those metrics and milestones into your process improvement plan will ensure you can leverage management's big-picture expertise in real-time to help you iteratively improve the process once it goes live.
- **Plan more time than you think you need for discovery**, and don't assume that everyone will be on the same page.
- It is vital to **gather as much information as possible** before designing systems.

- Incorporating agility or agile systems that can quickly change is essential because new information will be discovered later. The ultimate test of your discovery will be the deployment of your solution. If your launch state is brittle or hard to change, it may fail when just one or two details are wrong.

Key performance indicators (KPIs) are often inserted into processes as an afterthought. If you want to look like a pro when doing your discovery work, follow Stephen Covey's famous advice, "Begin with the end in mind." If you take the time to determine what key performance indicators your organization already values, that will be a great starting point. You should also ask your various stakeholders what information they wish they had about the process. You may need to prod them to imagine enabling currently impossible things. For instance, perhaps they wish they knew the difference between how long something sat in a queue and how long it took the person to do it. That might give them insight into the backlog. Perhaps today, they can only see how long something takes as an aggregate of both the waiting time and production time. If they could separate those two items, they could gain greater clarity over whether they had slow production or needed more people to do the work.

Don't worry about figuring out how to address the analytics side of this problem yet. If you have the right technology or technologists, this will simply be a subsequent effort to facilitate the recordation, organization, and storage of various data elements for later analysis. Keep it simple at your level and ensure you record data points like the date and time that something enters a queue, the date and time that it gets assigned to someone, the date and time that they start working on it, and the date and times they say they are done.

Assuming you've got a modern system or tools, you should be able to organize and analyze that data at a later point quickly. Your job right now is to make sure you've identified and documented the data and artifacts required to build the dashboards you are probably already starting to imagine.

Capture and Display

The individual contributors within a process have a pretty good sense of the data they need to make decisions. Therefore, they can tell you pretty clearly what they now have that they also must have in their new system and what they wish had been provided for them upstream.

You can think of these as the elements you need to "capture and display." You must ask your various stakeholders, "What data must be captured before an item gets to you so you can do your job quickly and effectively?" We also ask, "How do you need that data presented to you?" We are seeking both summary answers (e.g., "When showing me a list of requests, I need to see X and Y at least.") and detailed information, such as, "When I drill into a specific request, I Must Have, Should Have, Could Have, and Won't Need or have A, B, or C."

This line of questioning is known as the **MoSCoW** model:

Must Have
o
Should Have
Could Have
o
Won't Have

We must ask these same questions again for each type of work or request that will come to our team. We must share this data with people doing upstream discovery, just as people or teams doing downstream discovery should share the same data with us, so we are aware of what we must capture to meet downstream productivity requirements as well.

Here are a few more specific examples:

- Suppose a contributor in our team completes a quality assurance step on every 10th record. Do we need them to attach something, enter data to provide context for their findings, or prove they completed the quality assurance control?
- Will an auditor require information that validates that a client-initiated a request even if it has been entered into our system by an internal employee?

Here are a few of the secondary considerations we need to keep in mind when we determine what we need to capture and display in our part of the overall process:

- Who will we capture this data from?
- To whom do we need to display it, and when will they need it?

High-level workflow diagrams are practical artifacts to help clarify the capture requirements harvested from your discovery sessions. However, nothing, including the best of fact-gathering questions, beats empirical data. If you can build a lightweight system quickly and run it for a while, you will give yourself an edge over everyone who gathers data through subjective means because you will have collected your data objectively. In other words, if you can quickly set up a form

or simple recording mechanism for your process contributors to use to track when a thing actually happens (i.e., "We just got another request without the proper signatures") versus designing solutions for perceived problems, volumes or severities, you will be able to validate that you are investing your time and resources wisely.

We've seen it time and time again. Objective data is king, while the very best of subjective data remains just that: subjective.

Step 6. Broaden the Horizon

Core question: How will we address the transactions that don't go as expected?

The discovery completed in step five was primarily oriented around "the happy path." The happy path is a phrase that defines a typical record or transaction traversing a process or organization in an expected fashion. If only life worked that way, we would all be happy, too. That is why it is called the happy path. In reality, we know that a significant portion of any organization's work includes some level of exceptions. In other words, we will receive requests that do not fit our ordinary bounds. We need to be able to deal with those exceptions, but it is nearly impossible to map them all out ahead of time. These kinds of exception pathways can be simple things like someone requesting a service we do not offer. The outcome of that should be fairly obvious. However, what if it's an excellent customer that our company cannot afford to lose asking for something we could do but typically don't?

Exceptions can also be embedded far more deeply into a process. Perhaps in onboarding a new client, we have designed a new process assuming that our new anti-money laundering check will yield a binary answer of yes (they are laundering money, and we cannot open their account) or no (they are not laundering this money). That would be sufficient if that were the end of our process, but what if the person designing that process was unaware that the infrequent "yes" outcome required a report to the authorities? We need to bake in the agility so that someone can handle that exception without breaking our process. We also need, where and whenever possible, to figure out how to work that exception into the process.

At this early stage of discovery, we must extract as many of the "known potential exceptions" from our contributors as possible. Existing transactional records will probably show us many exception types and potential outcomes we must replicate. However, as people are excellent at creating workarounds, we mustn't assume that all exceptions will appear in the traditional system of record. These exceptions may end up found in the miscellaneous notes included in our system of record, and they may also end up recorded in various ancillary mechanisms, such as some manager's spreadsheets. That latter example can be more problematic to find. Therefore, asking questions designed to surface these hidden workarounds is critical.

These organic workarounds will not be at the top of most people's minds. However, if you ask numerous questions about exceptions, you will often find many not typically handled within the system. Keep asking questions until you find out whether they are handled in an ad hoc form or recorded in some "work around system" designed for handling these exception cases. That is where we will often find the limitations of the system or process we are trying to fix, resulting in workarounds. They frequently include emails and an associated (potentially volatile) archive, sticky notes that end up in a file folder, or spreadsheets.

In addition to expanding our discovery, in step six, to include exceptions, we need to consider and record critical or noteworthy events. We've mentioned some of these in step five but want to provide more details about a few examples. For instance:

- **Start and stop events:** This can be as simple for a short transaction as recording when a request was received and completed. For a long-running, multi-departmental, or

multidisciplined transaction or process, we may want to record the start and stop times for various events, such as the entry to a team's queue. We may want to record something like the date and exact time a request entered the queue for legal review. We may also want to record when the legal team supervisor assigned the new contract to a paralegal or legal resource. We may wish to record the time that the specific person begins reviewing the contract, and we may want to record other specifics like when they had submitted it back to a stakeholder with questions, when those questions were answered, or when the document was approved or declined. Recording these times will allow us to gather empirical data that will empower us to improve the process over time and based on facts.

- **Intermediate events:** We should take some time to contemplate intermediate events such as a route back, approval of a request, rework, clarification step, etc. In a complex process, we may be unable to ascertain all these possibilities. That is where agility is key. Creating "relief valves" is critical so that an end user does not become stuck in a transaction that requires rework—i.e., building a general utility for assigning something back to someone else with notes that clarify the reason for the exception or intermediate event. You can detail the specifics of an intermediate event, think of it as an individual process, and ask the same questions we've already discussed. What information will this person need to complete their portion of the job? When, where, and from whom will we have gathered it? Etc.
- **Inter-related tasks:** In a manufacturing environment, an order that is received may be for something wholly

produced internally by machining from raw materials to finished goods. However, in certain circumstances, external items may need to be procured to complete a final assembly for the customer. That order may roughly follow the same path as an internally produced order. However, that order will fall behind in assembly if someone has not already externally sourced the required (sub)components. That would be an interrelated or parallel task. These kinds of tasks can often be discovered by asking, "Is there anything else we ever need to do to complete one of these orders that doesn't happen in your department?"

- **Alerts:** When transactions fall outside the bounds of "normal," it is advisable to ensure that the system will automatically alert the appropriate individuals or teams. That requires two parts of discovery:
    - **What are "the norms?"** In other words, we need to determine whether this is a constant, variable, contractual, or other form of commitment. For instance, in a manufacturing environment, every order may include a need date when the customer wants to receive it. In a financial services environment, and onboarding process may include an internal SLA (Service Level Agreement or Commitment) regarding how many days it should take (or less) onboarding new client. Assuming that every shipped good will take an average of three days to arrive at a client facility, we could take the need date for that manufacturing order and subtract three days. But if we need that order to be at least 90% complete the prior business day to have any chance of meeting that expectation, then

we may want to set up an alert for any order that is not at least 90% complete four business days before the need date.

  - **Who needs to know?** Creating alerts is something you should handle carefully and sparingly. First, alerts should only go to people who can directly impact the outcomes. Second, alerts must be judiciously limited to ensure they do not become noise. Determining who needs to know should be as simple as figuring out who can impact a change if something is off course. Just keep in mind that you want to limit these alerts to things worth derailing the daily plans of the person with that influence. You certainly want to save them from losing a critical client, but you don't want to text them every 15 minutes during a critical board meeting about things often resolved by the time they can break free and check on them. For that reason, consider the philosophy of incremental escalation. In other words, as an example, wherever possible and practical, start by alerting someone at a lower level one business day earlier.

- **High-level escalations:** Remember to coordinate with the larger process team regarding who needs to be aware of errors, exceptions, escalations, or other problems that might be arising from the part of the process that you are doing your discovery work on. That is best to coordinate after most (if not all) of your department's process-specific discovery has been completed. That way, you will have an inventory of your internal escalations, and you will be able to ask

the overall process champion whether or not those bits of information may need escalation to a higher level and when and how. For instance, you may want a Chief Operations Officer to have a dashboard to show the average SLA success rate, with a drill-in capability to see the exceptions if they are of interest at the time. If 99% of the records are running on time, the COO may have no interest in drilling in. However, if that number falls into the 80s, the COO may be particularly interested in seeing the specific records and analyzing for patterns. Making that information available at the dashboard level will be another way of surfacing the data you have already gathered for your part of the process.

Step 7. Contemplate the Milestones

Core question: How will we measure success?

We've already discussed some of the specific things we will need to record to make milestones, performance, and success more visible, but let's take it a level deeper:

- We need to determine what stages matter
- We need to ascertain how long we believe various parts of the work or process *should* take
- We need to ensure that we have recorded the data necessary to surface our KPI (Key Performance Indicators)
- We need to create (or create mockups of) our alerts and dashboards and present them to our key stakeholders and audience to ensure that we capture and display the information they need.

Determining what matters, or more specifically, what stages matter. We may realize that there are two or three "resistance points" with in a process or sub process. We may recognize that they require and deserve the recordation of additional information to help make exceptions or bottlenecks in these areas visible. For instance, in the course of gaining regulatory approval for building a new multinational natural gas pipeline, we may realize that all land agreements and environmental commitments require an average of forty-five days for legal approval. We may also recognize that drafting the first version of one of these documents or commitments may require thirty days. Suppose we also believe that it takes six months of conversations with a prospective landowner to get to the point where drafting a prospective agreement is appropriate and two more weeks after they've seen the agreement

to get the documents signed. In that case, we can add these together and develop several potential "pinch points."

First, we must engage our landowners or regulatory agencies at least nine months before we hope to make a go/no-go decision. But that nine months would assume that everything goes through each stage in the amount of time expected. Therefore, it would be too late to realize we don't have a land agreement one week before making a go/no-go decision. Instead, we need to look at each stage in light of the bigger picture. If it has only taken us four months to reach a verbal agreement with a ranch owner, and it only takes us two weeks to develop the contract, we will now be ten weeks ahead of schedule. However, if it took us seven months to reach a verbal agreement, we would be four weeks behind schedule, and we must pay critical attention to these particular records as they traverse the other parts of the process.

Defining the risk level and critical timelines for various items moving through a process is vital. To surface these elements, we need to record more than just the time a record enters a particular stage of work. Myopically assuming that a 42-day approval for a land agreement meant that we were ahead of schedule would be missing a critical part of the story if we had not received the authorization to develop that land agreement until the seventh month. While we are three days ahead of the typical legal turnaround, we are running out of time to make up the other 27 days by which we are behind schedule. Therefore, when we record the entry and exit into and out of each stage, it is important to consider the risk level (a factor of size, financial considerations, need date, and other relevant data points) and highlight those items for special handling. That might require us to build workflow considerations like, "Escalate all 'at risk' transactions to 'high priority,' and move them to the front of the queue."

We might only need to flag a transaction somehow so it will show up on a management dashboard.

The point here is to build upon the elements we have previously been recording, such as the date and time a record enters or exits any specific stage or sub-portion of a process by adding meaning and context. One part of that exercise is the recordation of "risk level" (or whatever you might decide to call it), and the other part is to ensure that that information is appropriately surfaced (through dashboards, alerts, notifications, or routing into a special handling queue, etc.).

Identifying these kinds of needs will come from asking questions like:

- What problems have resulted when these transactions have not gone as expected?
- Who will need to know if something is running behind?
- Who can escalate transactions that have a higher level of urgency, criticality, or risk?
- What regulations are we subject to where these transactions are concerned?
- Who waits for us to complete our work downstream?
    - Are our downstream deadlines affected by what we do?
    - Can downstream deadlines be changed if we fall behind?
        - If downstream deadlines cannot change, where else can time be made up?

These are just examples of the questions that can help you discover "the bigger picture." We hope that some of this information will come to you along with your project charter. In other words, your management team will likely know the

risks they are trying to address and be aware of the information they would like you to capture and present to them for later analysis. However, it is best that you never assume they have either completed that step or been able to provide you with a comprehensive list. Your value will always increase when you discover a previously unseen risk.

What does normal look like? We want to ask copious questions about the expected outcomes our process or subprocess should produce. We want to interrogate the stakeholders who have:

- Requested our work
- Will be impacted by our work
- Will receive our work (both in terms of the next recipient and the final recipient)
- Or who are responsible for our work

We want to know what each group believes is "normal" regarding delivery time, update practices and periodicity, and overall outcomes. We are primarily concerned with the delivery and stage times in this step. Examples of the timeline elements we are trying to capture include:

- How quickly do our requestors expect a response to their new request?
- How quickly do our internal contributors expect to be notified of new requests, and by what means or media?
- How many stages, teams, or departments will the work traverse?
    - Are any stages critical, or could they block other tasks, processes, or overall projects or programs if they fall behind?

- Which, if any, of the stages or milestones will warrant notifications for the stakeholders?
    - Should these notifications be in real-time or summarized across time?
- What internal resources need critical performance indicator data for any or all of this process or sub process?

These are just examples of the questions we can ask to determine what normal looks like and how we can record the data to show that we are succeeding with or exceeding expectations.

Gathering this data early in the process will allow your designs to incorporate the recordation of the information necessary to quickly surface an atypically positive amount of information to your key stakeholders. Remember, you are building an agile system capable of providing real-time insights that will show you and others how to improve the process iteratively. It is essential not to invest time into capturing details for things that are not important. That is why we preceded this topic by leading you to determine what matters. Once you know something matters, it is hard to gather "too much data."

Organizational blindness is a pervasive challenge in numerous industries, and curing it will make you look like a hero.

Step 8. Plan for the plan to change

Core question: If at first you don't succeed, how quickly can you recover?

In process improvement and systems implementation and deployment it is difficult to get an A+ with the first draft. The first draft you give to your testing team is rough. The first draft you give to your early adopters is less rough. By the time they've helped you polish it, you feel like it's wonderful and ready to go. However, by the time you roll it out to your mainstream end-users it is unlikely that you will receive the warm reception expected. Don't let that slow you down at all!

If you expect a rollout of new technology to go smoothly you probably have an unrealistic expectation. That isn't to say that simple changes, or abstracted changes can't be implemented with minimal impact, it simply to say that the closer you are to core operations the more likely you are to cause instability when you implement a significant change. Instability is typically not well received by mainstream users. However, mainstream users have something in common with all of your prior user groups. They respond well when they are responded to well, and quickly. Likewise, their trust will grow if their feedback can be quickly incorporated into corrective actions that resolve their pain and allow them to quickly get back on track towards experiencing the intended benefits of your new enhancement.

Because it is safe for us to assume that a new deployment will cause some level of instability, it is critical for us to prepare for that in multiple ways:

- Ensure sufficient human and other resources are ready to deliver rapid responses when it is time for "go live."

- Be sure there are clear mechanisms for end-users to report any impacting issues they encounter.
- Take care that your mainstream end-user community has been briefed about the coming change, understands why it was necessary, understands what the expected benefits will be (including to them personally), and above all that they know exactly how, to whom, and specifically by what means to report any of the aforementioned impacting issues.
- Last but not least, in fact probably foremost, I sure that you have selected or built an agile system that can quickly incorporate the feedback you will most certainly receive when you deploy your new system or process.
    - o If your new process is simply a trained procedure, then agility may be as simple as assuring that the first draft of a process manual is delivered via three ring notebooks where pages can be replaced when they are updated.
    - o If your new process is rooted in a technical system, then you must ensure ahead of time that you will be able to rapidly make changes based on feedback. Examples might include:
        - A user running into an unexpected condition (a valid record or transaction that does not fall into preconceived boundaries) must be able to execute a work around within a reasonable amount of time. This may require you to design a workaround in the solution, or accommodate a hybrid solution such as someone temporarily attaching a spreadsheet to a record until a more

sanitized solution can be designed and integrated. You may recoil at the idea of a spreadsheet being attached to a record in your new system, but believe me that is far better than people being convinced that your new system cannot handle the exception in deciding that the spreadsheet should live somewhere outside of your system. People will create workarounds, except that and do everything you can to accommodate them being with in your new system.

- One critical consideration: Determine ahead of time who can make changes to the system and at what levels. Many modern systems will allow you to delegate administrative control or configuration capabilities across various components within a larger process or workflow. If you can allow people who are close to an organization's operations to make changes without impacting other parts of the organizations flow, that will facilitate rapid corrections. Systems that provide audit trails for the configuration of the system will provide exceptional safety and accountability. There are risks in distributing and delegating the ability to change a rapidly evolving system, but in our experience, when correctly mitigated, these risks are significantly smaller and less costly then failed adoptions.

To summarize this section, it is critical to insist on agility being part of the design of any new system whether that's system or process be simply policy-based were technically enforced. Your organization will have to invest time and other costs to achieve that upfront, but the dividends that you will harvest when it is time to deploy your new process or solution will more than make up for the investment. Adoption is the difference between successful deployments and unsuccessful deployments. Agility is the key to adoption. No deployment is perfect. But a rapid response will assuage the concerns of a frustrated user, and a rapid response that includes rapid results and corrections (based on their feedback) will gain trust. Trust is a critical component to adoption, and adoption is a critical component to success. As the success of your project will reflect on you, get ahead of the curve and make sure you build agility into your plan at the start.

Step 9. If at first you seem to succeed, keep going anyway

Core question: Is a grand launch the end of your effort or the start of something bigger?

You've done excellent discovery work. You've considered all the key performance indicators your management team could want. You built a system so agile that all the feedback you've received has enabled you to incorporate that feedback into beneficial changes quickly. And the result? You launched your new system based on a thorough redesign process more than two weeks ago, and despite the first few days being a handful, your user community is now raving about your great new system.

So, is this the end of the road? Do we write down all of the details in case we ever need them on a resume, and go and ask our manager for the next project? Or, is this an opportunity to figure out how to significantly impact our organization more than we had expected?

Process improvement is an art and a science. If you have succeeded at both, you have already proven yourself to be a rare individual. Simply ascertaining the details of a complex process can be a lot of work. It requires significant brainpower. Designing a new and better state in the future involves a lot of creativity and thoughtfulness. But getting it to work requires getting people on board and addressing their various inputs and interests. Accomplishing all of that is a big deal. But it is still just the start.

Once you have accomplished something like this, you could write it all down and call it a success, which would be true. However, it is just a milestone along the way for those "in the know." If you take this success and continue iterating it in a shorter fashion, your success will increase and become

harder for competitors to imitate or overtake. Because you have gathered empirical data about your operations performance, you have empowered your organization to study that performance data for yourself, looking for bottlenecks and opportunities that could never have been the scene without access to all of this data. There may be cases where you find you have done such a great job up front that there is little left to improve, but that will be rare. In most cases, once you have 30, 60, or 90 days' worth of empirical data, you will find all sorts of new opportunities for improvement if you take the time to look.

You are not responsible for the overall process or processes of the entire organization. Still, you have become an expert and champion for improvement of the portion of the process you engaged. It is likely that you now know that process as intimately as anyone else in the organization. You know what data you have gathered and have come to understand the meaning behind it. You could assume that the managers for whom you have created these dashboards, notifications, and alerts will take that data and run with it. Perhaps they will call you if they need help, but that would be an assumption. Or, you could be proactive.

Consider the opportunity at hand. While your success may get you pulled into other process improvement efforts, there will likely be pauses. What if you took some of that time and proactively reviewed the state of your prior efforts? What if you were to find additional opportunities for incremental enhancement proactively? Then, you would be moving to the next level of expertise in process improvement. That is where you go from being able to execute a one-time change to where you can make change an ongoing process. If trust is already high, why not leverage it? If people believe your questions will lead to new and better tools, outcomes, or recognition

for them, why wouldn't they invest more time when you ask for it?

What we have described is the core of iterative process improvement. It is also called continuous improvement. It's the same thing we do in everyday life. We buy a house because it is the perfect place to live. A few years later, we will probably change something like the deck, the garden, or the kitchen. Why? We now have a better understanding of this new environment, and we have learned where and how we could optimize things. While many organizations aspire to continuous improvement, few achieve it. It is not easily orchestrated from the top down, but it is easily affected from the bottom up.

Earlier in *The Process of Improvement*, we included a brief write-up regarding the phrase "going to the Gemba." Suffice it to say that phrase is about going to where the deed is done to get the best information about how work is completed. If you are embedded within a core process contributing team, and you have already been part of a successful process improvement effort, consider yourself as someone who can positively impact the performance of that organization by looking for the opportunity to repeat the process as often as possible.

The delta between good organizations and great organizations is in the details. While two organizations may be able to on board their clients in three business days, the one who can delight their customers while doing that will still win even though their outcomes are equal from a performance perspective. What's the difference between simply satisfying a customer's expectation and delighting a new customer? It can often be as simple as making something just seem easy. Perhaps it is giving them more access to information about how the process is going in real-time. These are often things

that are improved iteratively by studying the data that you gathered during your process improvement effort.

Whether this is the first process improvement effort you have ever participated in or the 30th, if your project has succeeded wildly, you are in an exciting position of opportunity. Therefore, we would highly encourage you not to consider this the end of the process of improving yourself in this regard but just the beginning.

Step 10. Game Day – Going Live in style!

Core question: How can you avoid the typical go-live crises that other organizations experience?

It's part of the modern human experience: You call your [Bank | Insurance Company | Cell Phone Provider | Benefits Company…], and a frustrated customer service person apologizes, "I'm so sorry, but we just got a new system, and I am [waiting for the system to load, unable to find your account, not able to get it to take this request you're giving me, etc.]. The long and short of it is they can't do their job because somebody has just delivered them a "new and improved" way of doing things.

As someone who has put a lot of work into your Discovery, Design, Documentation, and now Deployment (by the way, we call that 4-D process improvement), you want your project and all the work you put into it to be an obvious success. Experience has shown that very few people expect that story we can all relate to above. Most are surprised by it. They believe they have checked every detail so often that nothing can go wrong. The more confident they are in statements like that, the more certain they are to live out the old proverb, "Pride cometh before the fall."

So, what differentiates between a smooth rollout, a not-so-smooth rollout, and an unmitigated disaster? There are three factors:

- Testing
- Staged Adoption and feedback loops
- The provisioning of ample resources by the team that expects problems.

Problems are inevitable. Surprises wouldn't be called surprises if you already expected them. So, the best way to ensure you are not surprised by the certainty of problems is to anticipate their probability. Hopefully, you have already done the testing, staged adoption, and set up feedback loops based on the guidance we have given you in the prior nine steps. So, what's left? It's time to provision the ample resources you need when it's time to go live.

You could have 100 people ready to help the five people who were going to be affected by your new system. That is one part of being well prepared, but it won't do any good unless the five people receiving the new system know how to get a hold of you and your ample team. Therefore, please don't overlook the critical importance of distributing your cell phone number and other real-time means of communication. You want to distribute this to those affected by your rollout.

You may think launching the new system is the end, but it is not. You are heading into the final phases of refinement. Soon, your new solution will go live, and that is when you will receive the most significant amount of feedback you have received at any point so far. Hopefully, it's largely or entirely positive. But anything that your brilliant early adopters were able to figure out that this larger group is struggling with will come to light very quickly.

Because you have your back-channel communication set up, you can leverage that to promptly update your team on problems and opportunities as they arise. If you figure out a better way to explain something to your new and larger user base, share that information quickly so everyone else can use the same language. You don't want five people explaining different ways to do things. The scale of complexity that could result from confusing or contradictory messaging is more than you want to deal with. Unify the messaging as quickly as you

can, and ensure that your team escalates anything that causes a critical impact as soon as possible.

Because the people above you have also committed to this project being a success, you can reach them quickly if they are needed. Don't hesitate to do that. Everyone is going to have challenges on go-live day. Don't try to be the person who doesn't ask for any help. Ask for all the help you need and more. The measurement of your success will not be in the eyes of the supervisor that you did or did not call. The true measurement of your success will be in the eyes (and subsequent words) of those impacted by the rollout of your new process or system. Those may be the users, the managers waiting to see things flow through the system, or the clients you hope are not waiting on the phone while a frustrated employee mutters under their breath about not being able to save a record.

It would be wise to model the behavior you want from the people who helped you test all this because they are now your best resources for helping the larger user community. If the masses have issues when it's ' Go Time, ' you will want those insiders to serve as your extended support team. It would be best to have them believe that this is their system (and remind them that is true because you have made it their system) and that their success and reputations are also connected to the new system's success. Make them feel like insiders. Get their managers to clear their schedules on "game day,' and setup a war room. Order pizza. Make coffee. If you don't need much of their time that day, excellent! But if it turns out you do, you want them on standby, ready to rescue that person on the phone with a customer.

You want that employee on the phone with a customer to say, "Do you mind hanging on for just a second? We've just rolled out a new system today, and I'm having trouble saving

this record. I've got someone standing right down at the end of my row of cubes who can probably help me get it done. I will just need to place you on hold for a few moments."

Because your contributor has a very positive and well-supported go-live experience (thanks to your thoughtful "go-live planning"), their customer, who is ultimately your customer, will have a positive experience as well.

You may have reacted to the idea that you should share your cell phone with all of your end-users. But think it through. Wouldn't you rather get a call in the evening (when someone is running into a newly discovered problem) than find out the next day that people could not do their work, making the project (*their* project) look like a failure?

By letting people know that you or someone on your core team of process change champions are available 24/7 until this project is a success, you are, in our humble experience, actually reducing the risk that your users will frequently call you off-hours. When people encounter difficulties, they will be less likely to express angry eruptions. If someone is trying to get something done in your new system and runs into trouble, but know that they have been encouraged to call you at any time, they may contact you late at night. More likely, though, they will realize it's unnecessary, and they can email you and follow up in the morning. However, suppose what they're working on is critical, and they cannot reach anyone. In that case, they are far more likely to get infuriated and send a lengthy email about the challenges this problem is causing. They may send it to you and copy numerous others you wish they hadn't copied. You can make your own decisions on this. Still, I'd prefer to be interrupted for 10 or 15 minutes in the evening to resolve an issue than to lose a significant portion of the following morning explaining it and doing damage control.

Having to do a bit of damage control may be inevitable – after all, you are changing things, but keep in mind that each time you have to enter damage control mode, you put yourself in a situation of rebuilding trust. If you find yourself in that mode too often, you may find yourself in a situation like trying to climb a mountain during an avalanche. If that happens, you may be expending a tremendous amount of energy with minimal results. It is far better, if possible, to avoid ever getting to that point by ensuring people can reach you at any time.

When new users know they can efficiently and promptly escalate items directly to someone who can resolve them or help them keep going, they won't escalate those problems to everyone else. That's why you will also have to predispose yourself to the idea that you will receive a few angry phone calls during your go-live from people you may have thought were "on your team." Be ready to weather a few of those calls or escalated response cycles. There is a good chance that those same folks who are angry in the moment will turn out to be your most forceful advocates across time. Why? Because you are building trust by demonstrating a commitment to respond quickly and take appropriate action. Intuitively, we all know how rare and valuable that is, and people react to it, so commit to being accessible and taking action and then expect dividends.