# A Small Amount of Six Sigma Doubled My ERP Specs-Setting Skills … and Can Do the Same for You!

Kurt Goldsmith

*Kurt Goldsmith is an IT productivity developer for Enowa Consulting, specializing in the diagnosis and resolution of integration issues between a company's division of labor (end users, managers, executives) and SAP software (R/3, BW, APO, CRM). He also has a lifetime performance record of one win and two third-place finishes from five career starts as a thoroughbred racehorse trainer.*

*(complete bio appears on page 122)*

Setting the right specifications for an ERP implementation, upgrade, enhancement, or optimization project is a very difficult thing to do. Even the best field consultants and project managers can be taken by surprise at the post-cutover outcome:

- I have seen an SAP R/3 project team set off with a goal of reducing quote-to-cash cycle time from 115 days down to 90 days, and then "prove" during pre-cutover system integration testing that all the customization was working fine (no error messages), only to discover after the cutover that the 90 days target depends on something the team has no control over — two of the biggest customers agreeing to a significant change to their billing terms.

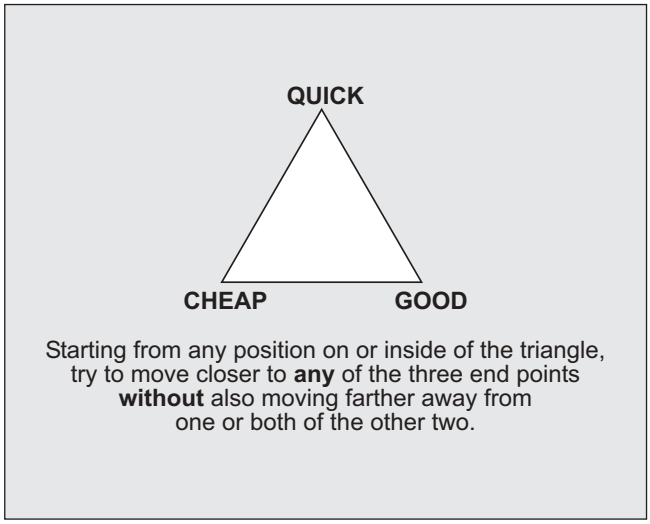
- I have seen a project team given a goal of helping the organization lower costs from a fully absorbed overhead rate of $8.50 per unit sold to $7.50 per unit sold, and then get no resistance to that goal from the user community, only to discover after the cutover that the one department no one was paying attention to — the IT department — needed 300% more budget to handle all the extra servers, user requests, and maintenance introduced by the ERP system.

- I have seen a project goal of better shipping consistency call for 85% of deliveries to arrive on the customer's originally requested date, prompting the inclusion of wonderful amounts of detail and milestone checkpoints in the project plan, only to discover after the cutover that the Sales team and the Shipping team had no measurements in place that could keep each side from blaming the other for late deliveries.

Most sites that implement SAP ERP systems tend to express their requirements for the software (i.e., specifications) in terms of three things that all businesses love to get more of from their employees and factories — faster processing, lower costs, and improved consistency.

However talented you might be today at developing or recognizing "good" specs for each of these three noteworthy targets, I predict that your spec skills will double — that your project implementations will achieve the hoped-for business results *and* avoid unintended consequences — if you follow the lead of Six Sigma enthusiasts and practice expressing these three kinds of specs (faster processing, lower costs, and improved consistency) in terms of their relationship to each other. Why? Because these three popular criteria are almost always tradeoffs of one another, which can be thought of in ways such as what is shown in the **Figure 1** "Quick-Cheap-Good" triangle, or in the **Figure 2** example of relationship-based specs.

*Figure 1 The "Quick-Cheap-Good" Tradeoff Dilemma*




Don't worry. You do not need prior Six Sigma experience to practice the ideas I will show you in this article. Nor am I suggesting that you even adopt any Six Sigma measures. I contend only that targets for processing speed, cost, and consistency should be viewed collectively rather than individually, and that Six Sigma offers great insight for doing this. In my opinion, Six Sigma in the hands of an IT practitioner is kind of like rocket fuel in the engine of an Austin Powers Mini Cooper automobile (see **Figure 3**) — a little takes you a long way, fast!

In this article, I will show you why this is so and how you can start using Six Sigma yourself.

- First, I will introduce you to some helpful Six Sigma concepts; how they can help you view speed, cost, and consistency targets in terms of one another; and how to decide how much

*Figure 2 Simple Example of Relationship-Based ERP Specifications*

| | |
|---|---|
| **Current Status/Needs** | Given a sales volume of 500 units per day, Acme Inc. is currently able to achieve a quote-to-cash (QTC) cycle time of 90 days, with 85% on-time delivery (OTD) to the customer, incurring a fully absorbed overhead rate (OHR) of $7.50 per unit sold. However, if sales volume increases to 650 units per day, it is likely that the only way to maintain the 85% OTD performance is to allow the QTC cycle time to regress from 90 days to 100 days, and the OHR to worsen from $7.50 per unit sold to $8.50 per unit sold. |
| **Desired Status/Specs** | Acme Inc. is willing to spend $1 million and 10 months on an ERP project, and is willing to see total IT monthly spending increase by $50,000 after the cutover, if and only if the technology change will allow Acme to maintain its current QTC / OTD / OHR ratio of 90 days / 85% / $7.50 at *any* sales volume between 500 and 650 units per day. |

variance from a "best-case" speed vs. cost vs. consistency performance target is okay.

- I will introduce you to the Six Sigma concept of CTQ (critical to quality) and one simple way you can apply CTQ to establish relationship-based specs for an SAP project.

- I will show you the Six Sigma FMEA (failure modes and effects analysis) matrix and one way you can use it to get specs agreement to speed vs. cost vs. consistency tradeoffs among a diverse group of people.

- I will acquaint you with the concept of "toll gating," and why its highly formalized nature makes it more helpful for setting collective specs for your ERP projects than the very similar — and more frequently used — technique of establishing milestone reviews.

- Lastly, I offer some sample exercises that will help you get started using these concepts.

Remember, there is no need at all to become a Six Sigma expert in order to gain benefits. In fact, because CTQ, FMEA, and toll gating map so well to methods that are already commonly used by most SAP project teams, you will not even have to tell anyone that what you are doing is in any way part of Six Sigma!

## What Is Six Sigma?

Six Sigma is its own field within the business management industry, complete with external consultants and certificates (e.g., black belt, master black belt). It grew out of the total quality management (TQM) movement when Motorola applied seemingly impossible standards to their manufacturing sites … and succeeded.

It could have remained solely in the manufacturing world if it weren't for the well-publicized successes also achieved by the General Electric group of companies, whose leaders insist that Six Sigma can

*Figure 3* *ERP Specs Development with and Without Six Sigma*

| **Relationship-Based ERP Specs Development Without Six Sigma** | **Relationship-Based ERP Specs Development with a Small Amount of Six Sigma** |
|---|---|
|  |  |

Photos © New Line Cinema

*Figure 4 Four Six Sigma Definitions for Better Writing and Reviewing of SAP System Specs*

| | |
|---|---|
| **Quality** | The consistency goal of *your* choice, which is usually expressed in terms of + and - tolerance. |
| **Distribution** | The shape (usually curved) formed by plotting on a chart the measured outputs or performance of any repetitive process. |
| **(One) Sigma** | A statistical measure of variability. For any given population, this represents the average distance for the whole population away from the mean value, whether measured in time, by size, or by weight. |
| **In Control** | Output performance where the values are distributed equally around both sides of a mean value. The performance is rated as One Sigma, Two Sigma, Three Sigma, Four Sigma, Five Sigma, or Six Sigma depending in part on *your* tolerance choices. For example, if you decided that +/- 4 minutes from a target of 4 minutes (0 to 8 minutes) is acceptable for a customer phone call to be on hold, and if no more than 3.4 calls per million are outside of that tolerance range, then that process is said to be operating at Six Sigma. |

yield positive impacts when applied to the service side and daily management side of business, not just the manufacturing side. Whether true or not, GE's undeniable annual sales and profit growth prompted a whole lot of people to start asking where and how Six Sigma principles could be applied to a whole lot of things, including the SAP projects we're up against (to check out the buzz, do an Internet search using the keywords "Six Sigma" and either "SAP" or "ERP").

The popular definition for Six Sigma offered in books, classrooms, and Web sites is "3.4 defects per one million chances." Try not to worry too much if you do not understand right away how this definition relates to our ultimate goal of being able to set or recognize "relationship-based" specs.

Instead, just think of Six Sigma as one possible way to reach agreement with other people (coworkers, customers, vendors) on *who* should be doing *what* with *which* resources during any given business day, and then the relevance of Six Sigma to those of us in IT becomes easier to understand. After all, this is oftentimes one of the biggest problems we face on any SAP R/3 or BW or SEM or EBP or portals project — getting everybody to agree on how fast, how low cost, and how consistent the different kinds of system-supported daily work tasks should be, and on who should be responsible after the cutover for performing those tasks!

In **Figure 4**, I've listed what I believe are the definitions of four Six Sigma words that could have a significant, positive impact on your ability to craft or review system specifications for your next SAP project. In the next sections, I'll discuss these definitions in detail.

### *Six Sigma "Quality" vs. the On-Time Delivery Spec*

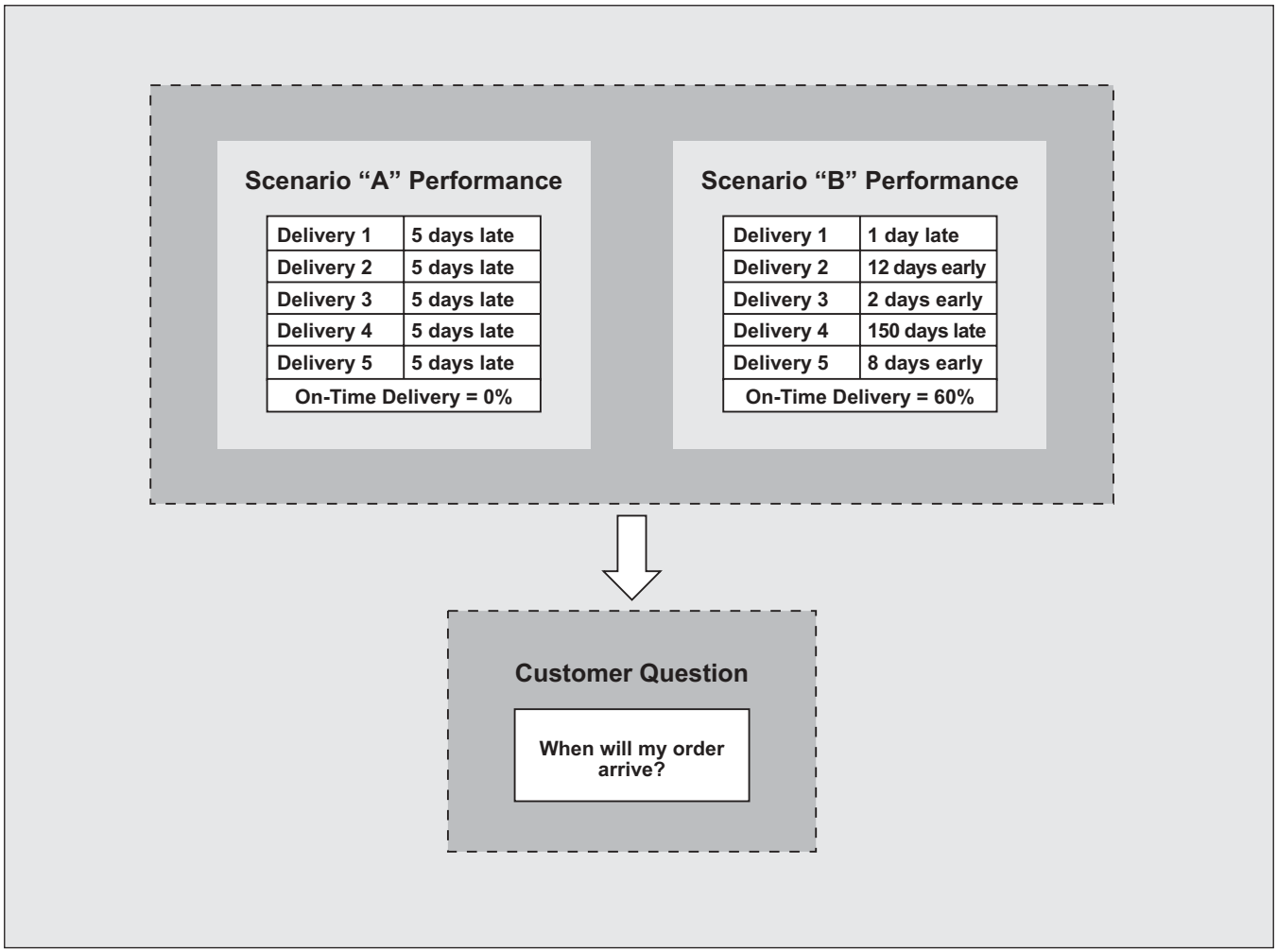
Perhaps the most important term to define is listed in Figure 4 first — quality. To a Six Sigma person, this word means "consistency."

However, Six Sigma offers a slightly unique view of the word "consistency" compared to what we in the SAP world might be used to.

Consistently bad, consistently good, consistently

*Figure 5 Is Slower and Consistent Better Than Faster and Inconsistent?*




medium — it does not matter. To a Six Sigma believer, variation is evil. Variation causes problems. So, even something bad can be good, if it is consistently bad.

In the introduction, I offered an example spec from one of my prior SAP R/3 implementations — 85% on-time delivery performance.

But there is no doubt at all that when we have a statement like that for an ERP project, what we really mean is an *average of* 85%. A company could achieve 85% on-time delivery while being extremely inconsistent (1 day late, 12 days early, 2 days early, 150 days late, 8 days early, etc.).

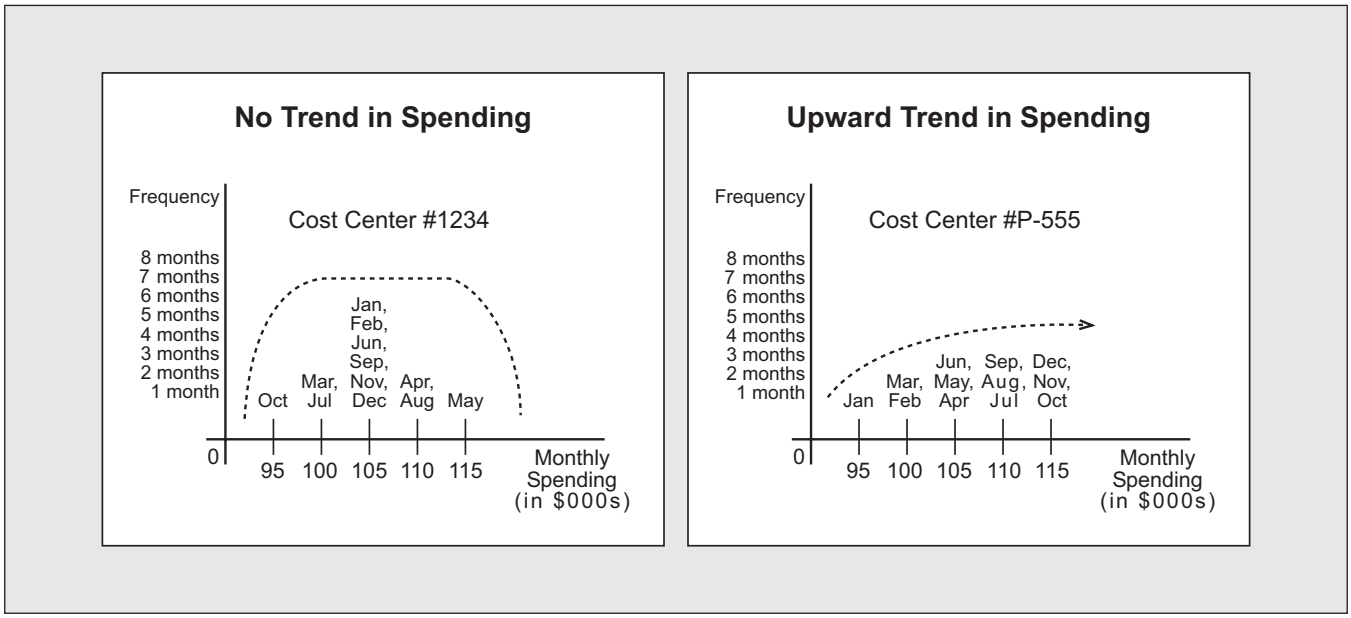
Compare the two scenarios shown in **Figure 5**. The one on the left is for a business with a 0% on-time delivery performance. The one on the right is for a business with a 60% on-time delivery performance. Which one is exhibiting more "quality"? Before you answer, imagine that you work in the customer service department and that a customer has just phoned you to ask when they can expect to receive their order.

When Six Sigma folks speak of quality, they are not only referring to consistency in terms of how often a target goal such as on-time delivery is hit, but also in terms of how far away from the target each iteration occurs.

*Figure 6 Example "Cost Center Spending" 12 Periods Report*

| Cost Center #1234: Headquarters Recruiting Department | | | | |
|---|---|---|---|---|
| Category | JAN 2004 | FEB 2004 | … | DEC 2004 |
| Salaries | $92,000 | $93,000 | … | $90,000 |
| Depreciation | $10,000 | $10,000 | … | $11,000 |
| Operating Supplies | $2,500 | $3,000 | … | $4,000 |
| TOTAL | $104,500 | $106,000 | … | $105,000 |

*Figure 7 Cost Center Monthly Spending Shown on a Distribution Chart*




### *Six Sigma "Distribution" vs. an Example ERP Spec for $7.50 per Unit of Overhead*

The typical overhead cost in any business is repetitive. Categories such as salaries, rent, depreciation, and operating supplies tend to have similar values each month. For this reason, a common kind of overhead cost report in any business — the Cost Center Spending report — often has 12 monthly columns to allow the reader to more easily see if a department's monthly spending is trending (up or down) or if it is more or less staying the same. An abbreviated example is shown in **Figure 6**.

For measuring any repetitive process, whether for cost, timing, or physical properties such as weight, Six Sigma also uses a report that tries to make it easy for a participant to see if a trend is occurring or if values are more or less staying the same. It is referred to as a "distribution chart." An example using cost center spending is shown in **Figure 7**.

Notice how easy it becomes to look for trends. The graph on the left shows a random pattern of monthly spending by Cost Center #1234 on either side of what appears to be an average spending amount of $105,000. Meanwhile, the graph on the right

### Understanding Six Sigma the Measurement

The word "Sigma" means a measure of variation. Hey — if a management philosophy exists that believes variation is evil, we certainly need a way to measure that evil! In this case, what we are trying to measure is an average distance away from a hoped-for "zero variation" performance.

In order to explain what is meant by Sigma, let's pretend that in support of a 90-day quote-to-cash cycle time, our company's goal is for every employee in the Accounting department to respond to any emailed request from our Sales department for a sales order or sales delivery credit release in exactly 30 minutes. Not sooner. Not later. Exactly 30 minutes. Of course, in actual practice, this is hard to do. But, it is really easy to measure. Emails are time-stamped by the computer.

If you respond to my emailed request in 25 minutes, then this is 5 minutes away from the target. If you respond to my next emailed request in 33 minutes, then this is 3 minutes away from the target. In total, in two chances, the target was missed by 8 minutes. The Sigma is the average of how far you are missing the target by, which is calculated as 8 minutes divided by 2 chances, which equals 4 minutes. One Sigma in this example is 4 minutes — i.e., the average variation from the hoped-for 30-minute response time is 4 minutes.

Quick — what's Three Sigma in that example?

If you guessed 12 minutes (4 minutes times 3), give yourself a free dinner. If you also guessed that Six Sigma would be 24 minutes, give your high school math teacher a free dinner. And, keep reading. Your assumption is only half correct!

shows that the spending by Cost Center #P-555 has not yet stabilized around an average value, and is instead increasing by somewhat equal increments each fiscal quarter.

Although the way the data is presented looks quite different when comparing Figure 6 and Figure 7, the goal of both is the same — to expose trends in the results of any repetitive process, and to do so in a way that a direct participant can easily interpret the data shown.

### *Six Sigma "Sigma" vs. the ERP Spec of Quote-to-Cash Cycle Time*

Earlier, I offered an example spec of a 90-day quote-to-cash (QTC) cycle time. Then, in the discussion of the Six Sigma vocabulary word "quality," I illustrated that there are situations where it is useful to measure not just whether or not a particular target such as a customer-requested delivery date is being hit, but also how consistently near or far we are coming to that target date. The measurement term known as "Sigma" is one way to allow us to compare how fast something is with how consistent it is.

If you want to know how to calculate Sigma, have a look at the example discussed in the sidebar above.

But even if statistical analysis has never been a friendly topic for you, there is nothing to fear. There is no need to become a Six Sigma expert. Our job as IT project consultants, managers, or executives is to recognize and ensure "good" specs. For this, it is enough if you merely understand *why* we would

want to include a measurement of consistency in combination with a measurement of speed. And, with our fourth and final vocabulary term ("in control"), which is discussed next, we will begin to see how Six Sigma can combine all three of our typical specs criteria — speed, cost, *and* consistency.

### *Six Sigma "In Control" vs. Expressing ERP Specs as a Relationship*

My premise in this article is that you do not need Six Sigma in order to practice expressing the typical ERP project specs of speed, cost, and consistency in terms of a relationship instead of in terms of isolated goals. But, I know full well that one of the biggest hurdles in any SAP project is finding a way to get lots of people to agree on how fast, how low cost, and how consistent each of several jobs should be after the cutover.

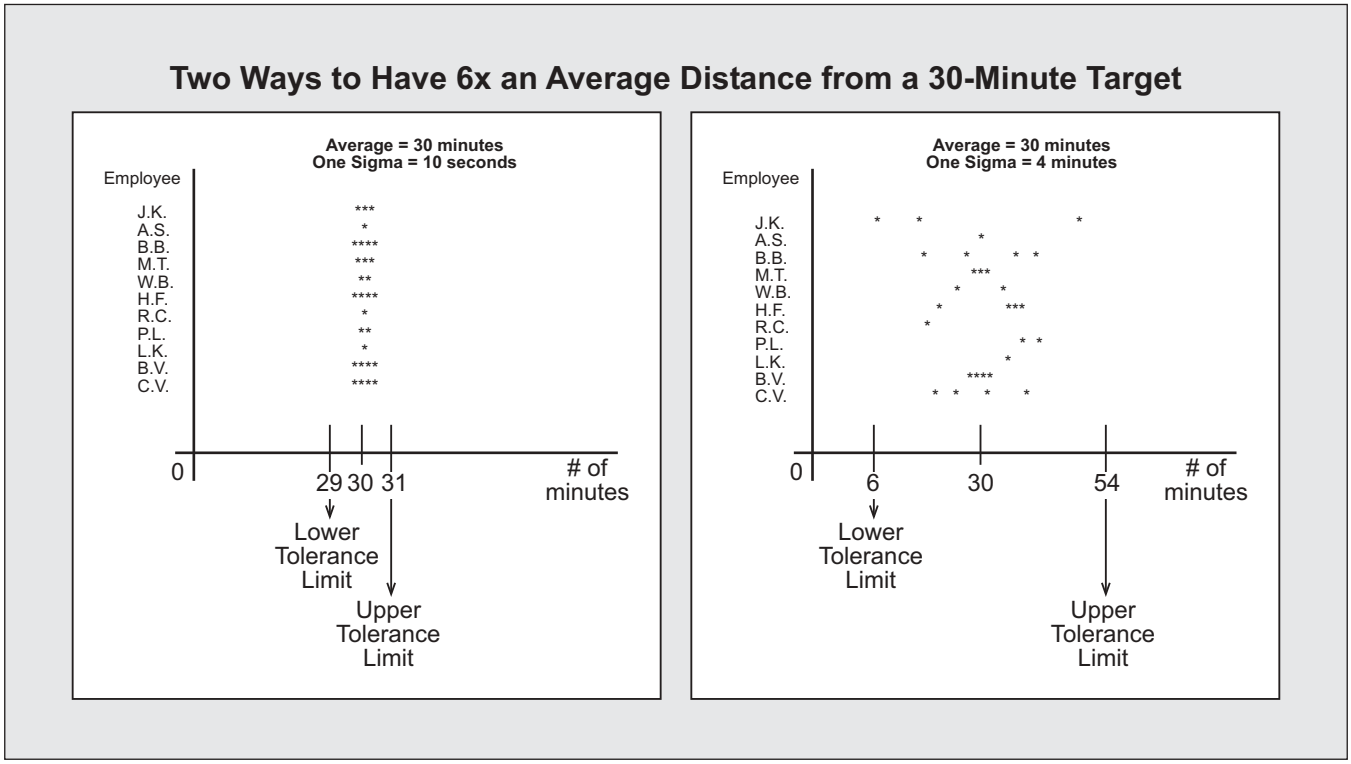
The Six Sigma concept of "in control" aims directly at that problem of creating agreement.

Ready? Here it is. *The affected participants* get to decide how much variation in any direction away from a best-case target in *any* form of a hand-off (customer-to-employee, employee-to-employee, employee-to-vendor) is acceptable. Performance then becomes measured and evaluated as "good" or "bad" based merely on whether or not the participants stay within that variation range at least 70 times out of 100 (really bad), 93 times out of 100 (bad), 99.4 times out of 100 (good), or 99.97 times out of 100 (really good).

An unavoidable element to any such discussion will be cost. This is because no person who manages a department will agree to a speed vs. consistency target without first knowing how much in the way of non-discretionary spending (e.g., general corporate costs allocated to their department) and discretionary spending (e.g., headcount) he or she is being given to accomplish the goal.

To visually reinforce this concept, have a look at what's shown in **Figure 8**, which presents an

*Figure 8 Your Choices Guide the "In Control" Six Sigma Reading*



imaginary set of data regarding how long it is taking for the Credit department to respond to sales rep requests for credit releases on specific sales orders. Notice how both examples can be said to be "in control" up to Six Sigma in that none of the data points are occurring beyond the upper or lower tolerance boundaries. The only difference is that in the second example, we can assume that the participants feel fine about having + or - variations of up to 24 minutes. In the first example, we can assume that those participants felt that it was important to have much more predictability than + or - 24 minutes in that particular handoff task.

I'm sure some of you are wondering what the harm might be in receiving an email response earlier than the target. Remember, this is a simplistic example designed for teaching purposes. Even so, imagine a salesperson sending a request for a credit release for Order #1 at 9:00 am, another for Order #75 at 9:05 am, and another for Order #122 at 9:20 am. The salesperson might access his/her email inbox at 9:29 am and see a response. But, a response to which request? Under a Six Sigma concept, that same recipient would have 99.97% confidence that the responses would be time-stamped 9:30, 9:35, and 9:50 in that exact sequence. And, as I mentioned, this is highly simplified. Imagine real life, where a department might have a dozen people each sending a dozen requests per hour! Any introduction of uncertainty — even early arrivals — is potentially going to slow down human decision-making.

So, which of the two in Figure 8 is correct? Both are!

In other words, the practice of Six Sigma is not merely taking calculations of speed vs. consistency. It is also the decision-making on how much variation is okay for your particular company, and for your particular situation.

The challenge, of course, is to achieve less and less variation around a best-case performance target *without* spending more and more money to get there. This reality of expense forces the participants to look for, discuss, and prioritize the kinds of variations that come into, or that exist within, the business.

Therefore, not only do all three kinds of typical specs — speed, cost, and consistency — get reflected by the Six Sigma concept of an "in control" process, but also the participants generally are asked to decide on the speed, cost, and consistency "control" definitions. This kind of shared decision-making tends to foster a spirit of agreement.

## Reaching Agreement on Speed vs. Cost vs. Consistency Specs

The issues that participants in an SAP project team need to agree upon when setting specs for the cutover are:

- Which business sequences or cycles to focus on
- What consistency tolerances to take
- Who decides

Can a project team even agree on which handoffs within the business do and do not exist?

Here, the formalized Six Sigma implementation methods known as CTQ (critical to quality), FMEA (failure modes and effects analysis), and toll gating can help us.

### *Handoffs, Consistency Tolerances, and CTQ*

The first Six Sigma method that I want to touch on is CTQ. (Keep in mind that Six Sigma practitioners view the word "quality" to mean "consistency".) CTQs establish guidelines for a successful handoff — what needs to be delivered to whom and when, and how much deviation from the targets that are set by the recipients of a handoff is acceptable.

*Figure 9 An Example of Identifying Process Handoffs with the Help of a Flowchart*

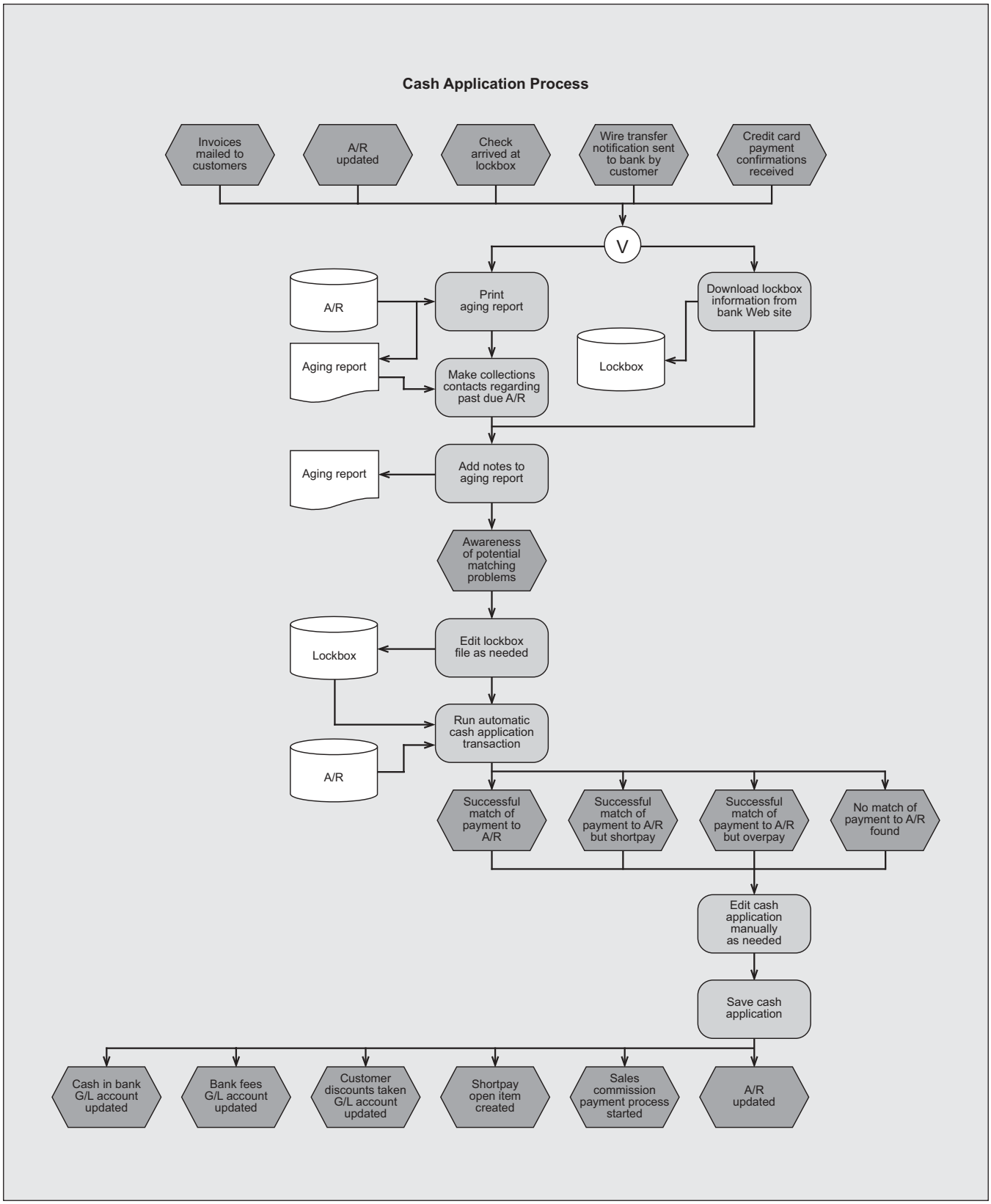
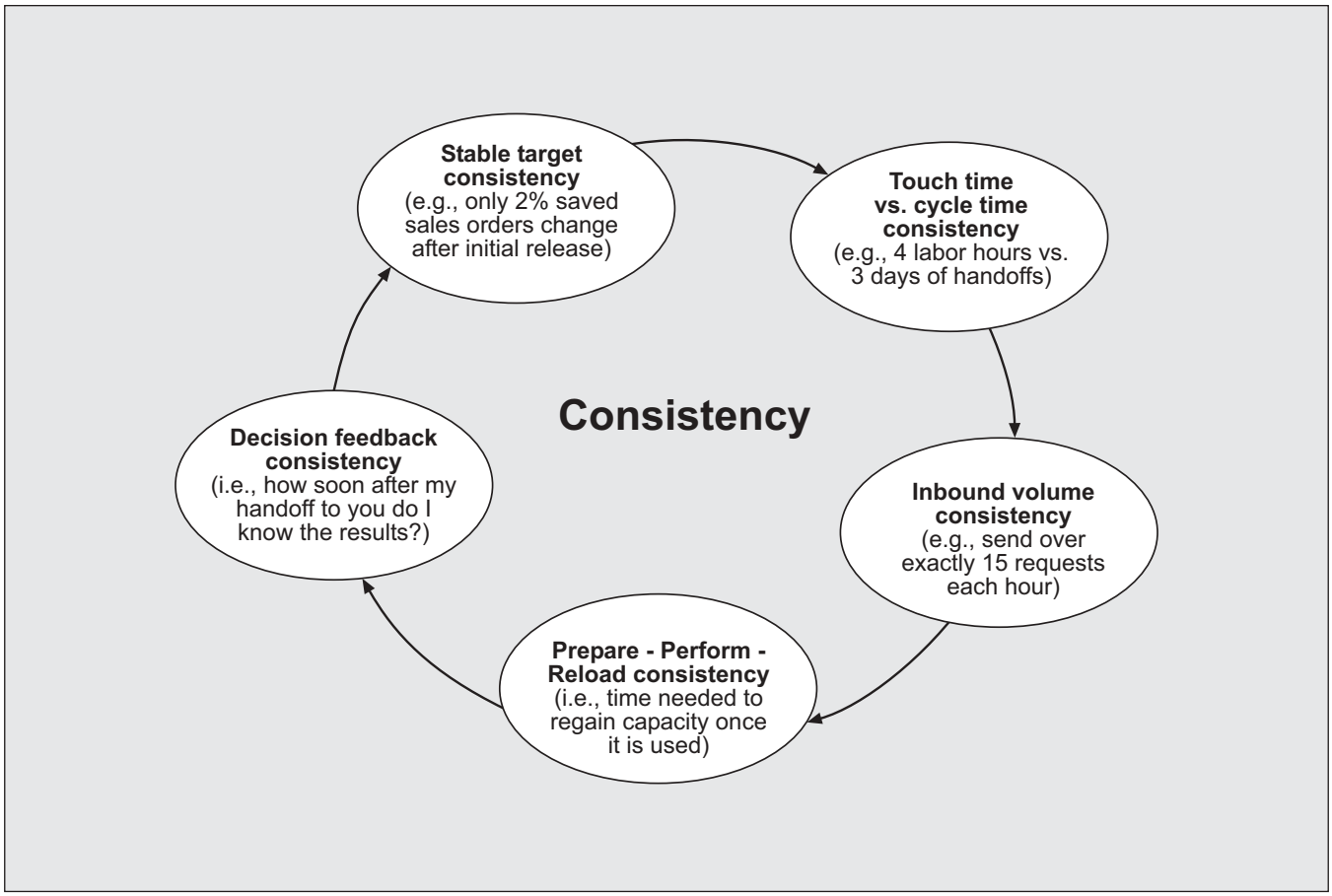


*Figure 10 Example Views of "Consistency" in Person-to-Person Handoffs*




To prioritize specific process handoffs over others, you can use a common tool that most people already know how to create, use, and view: a flowchart. **Figure 9** shows a simplified example of this, which is sometimes also referred to as a process map or (if you use Visio software) a Visio diagram.

Lessons from Six Sigma teach us to use the flowchart for reaching agreement among the team members as to which work steps or handoffs exist, *plus* agreement on which of those are repetitive enough to measure against a consistency target, or in the vernacular of Six Sigma, a CTQ.

One more nuance worth mentioning is that a flowchart is a great visual tool for helping a group of people decide on what particular kind of consistency targets they should aim for. There are several ways to measure consistency in any handoff. **Figure 10** shows a number of different views that I have heard on the word "consistency" in terms of the step-to-step handoffs shown in a flowchart.

Of course, time-clock consistency (e.g., 30-minute response to an emailed request for a sales order credit release) is often chosen. But more choices exist, and I am sure that you can think of additional versions beyond the ones in Figure 10. These targets are something the team will need to reach agreement on.

The people expressing the views shown in Figure 10 could be customers, vendors, and employees, or they could just be three different kinds of

*Figure 11 The Failure Modes and Effects Analysis (FMEA) Method*

| FMEA Formula | Perspectives to Consider |
|---|---|
| Things That Can Go Wrong with an Input (1 to 10 importance)<br>x<br>Likelihood It Will Go Wrong (1 to 10 how likely)<br>x<br>Negative Impact on Output If It Goes Wrong (1 to 10 weight)<br>=<br>Maximum 1,000 Points (10x10x10) per Item | ☑ Customer<br>☑ Vendor<br>☑ User<br>☑ Project Team Member<br>☑ Business Unit Owner |
| **Example of FMEA for ERP End-User Training: Importance of What Could Go Wrong** | |
| ☑ Users might not be motivated until after the cutover = 7<br>☑ Training content might be too narrow or too broad = 5<br>☑ Documents could become obsolete two weeks after the cutover = 10<br>☑ Training materials lost and/or training client crashes the day before the class = 9<br>☑ Instructor forgets to discuss the "how-to" on fixing data entry errors = 8 | |

employees, depending on the kind of flowchart your team is discussing.

As an example, consider the item "touch time vs. cycle time consistency" and compare that to how credit memo requests in your company eventually become credit memos.

What is meant by "touch time" is the amount of hours a person is actually processing, reviewing, or otherwise getting involved with each request. At your company, perhaps people in three or four different departments have to get involved. Each handoff from one person to the next person will eat up additional hours of "dead" time where the request is merely sitting in somebody's inbox waiting for attention. So, the overall time needed to get a request approved might be 3 days (72 hours), even if the actual work requirement (i.e., the "touch time") is just 4 hours.

The ratio in this example would be 4:72. If other requests had a ratio of 2:35, this would almost be the same proportion. The process is consistent in regards to the touch time to cycle time ratio. However, if some credit requests have a ratio of 3:5, and some are 4:72, and some are 1:230, then we can say that the process is not showing consistency when measured by touch time to cycle time.

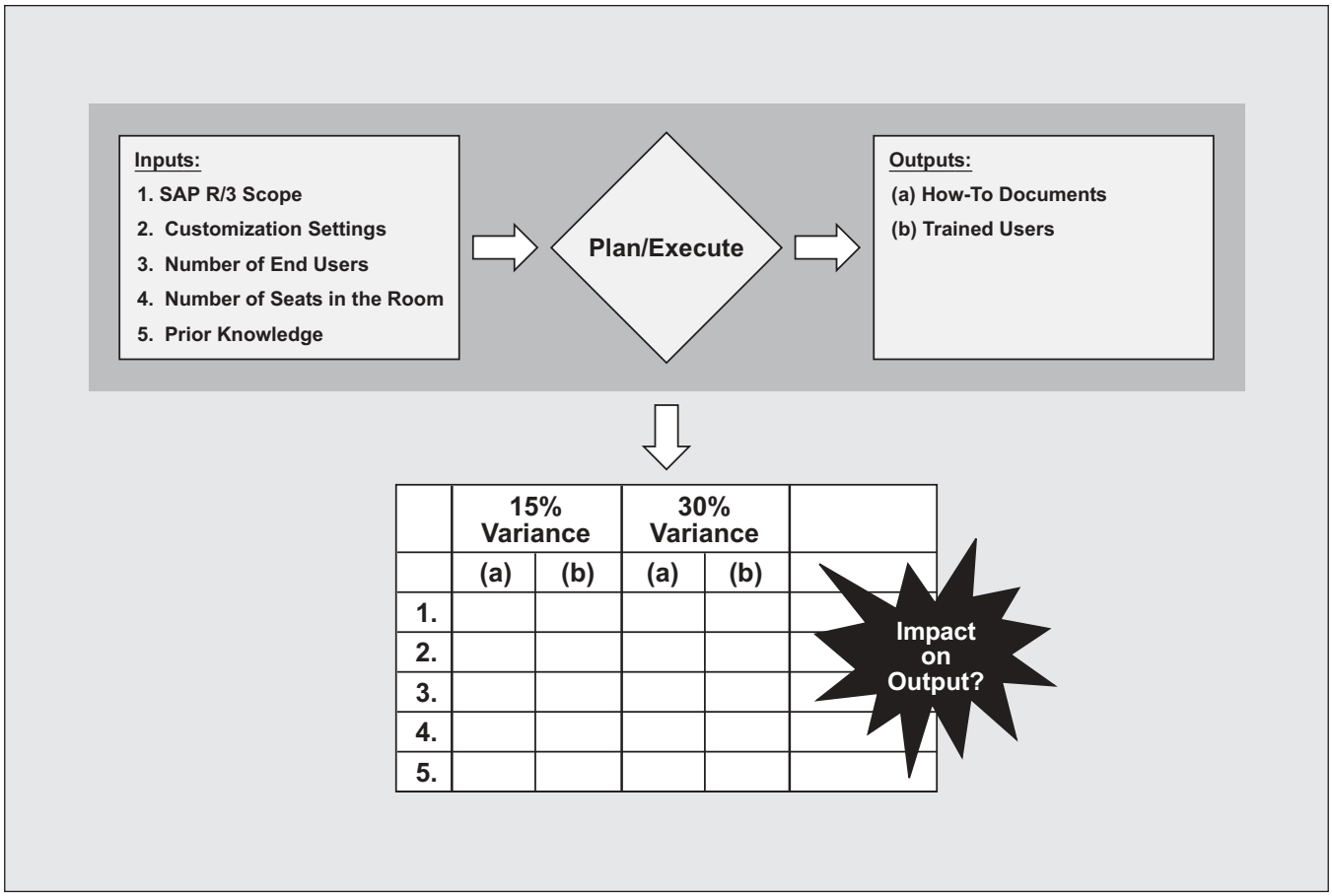
Is that amount of ratio inconsistency bad? Is the touch time to cycle time ratio a useful way to measure a particular set of handoffs? Does a different kind of consistency definition — something other than touch time to cycle time — fit better? That is not for me to decide. That is for the employees, vendors, and customers at each company to decide.

### *Agreeing on Priorities with Help from FMEA*

The acronym FMEA stands for failure modes (of inputs) and effects (on outputs) analysis. Basically,

*Figure 12* *An Example FMEA Grid: End-User Training*




you identify, rank, and prioritize every conceivable breakdown or failure of a particular spot in a process and how that failure will affect downstream activities.

The method is shown in **Figure 11**, and includes a simple IT project-related example for the first of the three steps in the formula, using the topic of end-user training. But you have probably seen an FMEA grid before. You may have called it an option comparison chart, or something similar.

The idea with FMEA is that you can use it as a way for the different kinds of ERP project participants to share their knowledge and perspectives and, hopefully, reach agreements on which handoffs to focus on and the consistency targets (and acceptable deviations from those targets) that will be set for those handoffs.

As suggested in Figure 11, the participants can include customers, vendors, end users, project team members, and business unit owners. Or, if you do not have any customers, vendors, etc., available for your discussions, someone in your group can at least act as the voice of that perspective.

You use FMEA to try and agree on a priority list of "Inputs > Outputs" pairings that the team will focus on, and to agree on what kind of consistency the "performance" will be measured against.

An example is shown in **Figure 12**. Notice how a matrix can be arranged to easily represent the team's answers to questions such as: "How much more of an impact will there be on each of our two hoped-for end-user training outputs (how-to documents and

trained end users) if the ERP customization settings in place right now change prior to the start of the end-user training classes by 15% vs. by 30%?"

Although I wanted to go with an example in Figure 12 that I knew all of you would be able to relate to, I want to be clear that I am not advocating Six Sigma as a method for ERP project management. I view it as a method to get "good" specs *for* an ERP project. It should be easy to apply the lessons from my end-user training example to day-to-day business tasks. One example might be evaluating the impact on two or three hoped-for manufacturing outputs against a situation where customer sales order information (i.e., an input to that process) is changing each week by 15% vs. by 30%.

If all goes well, the FMEA will act as the device that brings all the combined knowledge of the team together. You are looking for not just a list of inputs, of course. You are looking to expose and agree on the most important relationships — i.e., to predict which input(s), if slight variation occurs, yield big differences (good or bad) on the required output(s). Those relationships then become the most important ones for the ERP project to preserve or to improve.

### *This Way to the Toll Gate*

The two former GE project managers I spoke with for this article both mentioned the "toll gate" method of Six Sigma as particularly useful to an ERP project.

The definition of the toll gate might not surprise you. The benefit you get probably will.

As for a definition, you have probably referred to toll gating in your prior projects as "milestone reviews." We can think about the ASAP (AcceleratedSAP) methodology as an example. It has five distinct phases — Project Preparation, Blueprinting, Realization, Final Preparation/Testing, and Cutover. If you included toll gating with an ASAP-based project, this would mean that each phase could not begin until after a formal review of the preceding phase took place.

According to one of the former managers I spoke with, GE projects often involve not only multiple phases, but also multiple teams. A business case must be done by each process team at each toll gate that says: "This is where our productivity is going to come from." It is in writing. A review board can allow the project to go forward or can reject the business case. If allowed to go forward, this is looked at a year later to see how accurately the team was able to predict and/or control its implementation results.

I've listed the somewhat uninteresting highlights behind toll gating in **Figure 13**. It's more or less what you would expect to hear from a slogan — heavy on the generalities, light on the specifics.

Kind of boring.

However, the much more relevant aspect of toll gating to those of us in ERP is in *heading off two unpleasant issues that we know are going to come our way*.

First, all ERP projects face the dreaded 11th hour scope creep. I'm sure you know what I mean. As the cutover date nears, all of a sudden come the complaints from the user community that "this" or "that" business requirement hasn't been addressed yet, and that terrible things will happen if functionality to address it does not get added to the project scope. However, because a toll gate is such a formal procedure, and because it solicits opinions from a variety of people, it becomes difficult for someone to claim that an overlooked business requirement a few weeks or a month before a cutover is important enough to warrant a scope change.

Second, every business problem under the sun will get blamed on the ERP project after the cutover. In some cases, the blame is deserved — the implementation truly did get something wrong. But in a number of other cases, the negative item has a

*Figure 13* *Toll Gating Steps Summarized*

- ☑ Launch a project team to make a particular goal happen.
- ☑ But, at each phase's end, evaluate whether all the requirements for that phase are truly done before going on.
- ☑ In terms of phases at GE, the DMAIC (decide, measure, analyze, improve, control) approach is typically used if the goal is to fine-tune the consistency of an existing process, for example from Two Sigma to Four Sigma performance. The DMADV (decide, measure, analyze, design, verify) method would be more appropriate if the goal is to implement a brand-new process. Since both have five phases, each would end up with four toll gate reviews.
- ☑ Toll gating is an attempt to ensure that the team performs the right work sequences at the right time.

cause that's not related to the software at all. The toll gating documentation from pre-cutover will help a lot in determining categories for these issues (project team screw-ups vs. totally new issues vs. business issues).

In this sense, the toll gating aspect of Six Sigma does begin to flirt with ERP project management. But, that is something of an unavoidable consequence. For the most part, my goal for discussing the toll gate concept was to introduce it as a simple (i.e., don't need a Six Sigma expert) way to keep the focus on the prioritized project specifications that will come from CTQ and FMEA.

## When to Practice Your Specs Writing or Reviews

In this article, I noted that most SAP projects tend to express their requirements in terms of three criteria: faster processing, lower cost, and improved consistency. My opening argument was that the results from ERP projects become easier to predict when we express these three commonly used criteria as "relationship-based" specs, since each tends to be a tradeoff for the other two.

I also noted that our ability to get a group of people to agree on a set of "before" vs. "after" measurements becomes an important factor in how successful our ERP efforts will end up.

For both of these tasks — expressing the criteria as a relationship, and getting diverse people to agree on the criteria — you do not need Six Sigma. But, my claim was that a small amount of Six Sigma knowledge in your hands makes these tasks much easier, and that you were probably already quite familiar with the three Six Sigma methods I wanted to discuss (CTQ, FMEA, and toll gating). You simply knew them under a different name.

***Question:*** *When is the right time to practice your specs writing or reviews?*

***Answer:*** *Any time. Today, tomorrow, next week — all times!*

Think about what happens at most organizations that implement SAP R/3. The cutover is only the beginning. After that, we can expect at least some of the following projects:

- Upgrades
- Rollouts to other sites
- Functionality enhancements
- Introduction of complementary software such as SAP BW, SAP EBP, or SAP CRM
- Organizational structure or data-naming changes prompted by new business decisions
- System consolidations brought on by business mergers or acquisitions

My point is not that it is easy to predict what new ERP requirement is coming to the IT department and when; it is easy to know that something is coming. And, it is probably coming sooner rather than later. It pays to be ready. If you believe as I do that ERP projects are easier to accomplish when you begin with agreement on which speed, cost, and consistency relationships must be preserved or improved, then the best way to be prepared for the future is to start getting those agreements today.

How can you take the first step?

Mostly, this will be a personal decision, and it will be based on the unique circumstances you are currently in. However, after I close this article with a summary of what I wrote about, I will provide you with some specs-practicing jumpstart exercises to help you on your way.

## What You've Learned

I used the first portion of the article to introduce two Six Sigma concepts via four definitions.

The first of these was the idea that speed, cost, and consistency targets can be viewed collectively rather than individually. This becomes important because in some cases, the consistency leg of that "Quick-Cheap-Good" triangle (so to speak) can sometimes have a greater value to overall business performance than speed or cost.

The second of these involved allowing the direct participants to decide for themselves how much inconsistency (i.e., variance) away from a "best-case" speed vs. cost vs. consistency performance target is or is not okay.

In the second portion of the article, I identified specific ways that Six Sigma and ERP specs writing and reviews fit together. I wrote about CTQ (critical to quality), FMEA (failure modes and effects analysis), and toll gating.

I noted that what makes CTQ particularly nice for us is that its main work step, the process flow map, is something that most people who participate in ERP projects are already familiar with.

The FMEA, more commonly known to IT project teams as an option comparison chart, might require a bit more effort in the learning curve process. But, it also offers a unique benefit toward the goal of achieving agreement among a diverse group of people. In this case, the benefit comes from the FMEA's ability to present a wide variety of perspectives in a simple chart or matrix.

The final Six Sigma method that I mentioned, toll gating, which is sometimes referred to in ERP projects as milestone reviewing, might easily be confused at first with the typical and not particularly useful slogans so often associated with project management teams. I noted, however, that the formal nature of each toll gate directly addresses two unpleasant issues that we know in advance almost any ERP project will run into — the "11th hour scope creep" issue, and the post-cutover "blame the software for everything" issue.

So now what? In the next section, I will leave you with some exercises you can apply to your own business tasks to put Six Sigma to work for you.

## Where to Go from Here: Jumpstart Exercises for Practicing Specs

You can perform similar exercises to those that follow with any kind of business task that is repetitive and involves handoffs among three or more departments. My example will focus on a quote-to-cash cycle time for a "Sales from On-Hand Stock" scenario. It is a simplified example for using the three Six Sigma methods identified in this article (CTQ, FMEA, and toll gating) as a way to arrive at an agreement on ERP specs, and is not meant to be comprehensive.

### *Exercise 1: Getting Specs Agreement via Critical to Quality (CTQ)*

The main visual aide for getting agreement via Six Sigma's CTQ on which handoffs, and which kinds of consistency metrics related to those handoffs, deserve priority attention on an ERP project is the process flow diagram. The following exercise consists of five steps:

1a. List the individual steps that occur in each and every iteration:

- Quotation.
- Sales order.
- Pull the ordered quantity from stock.
- Ship to customer.
- Invoice the customer.
- Receive payment.

1b. List the parameters that any speed/cost/consistency goal should be valid for:

- Between 15 and 25 requests for quotations each day.
- Between 130 and 175 not-yet-shipped "open" sales orders on any given day.
- Between 300 and 400 "sold-to" customers with accounts receivable balances.

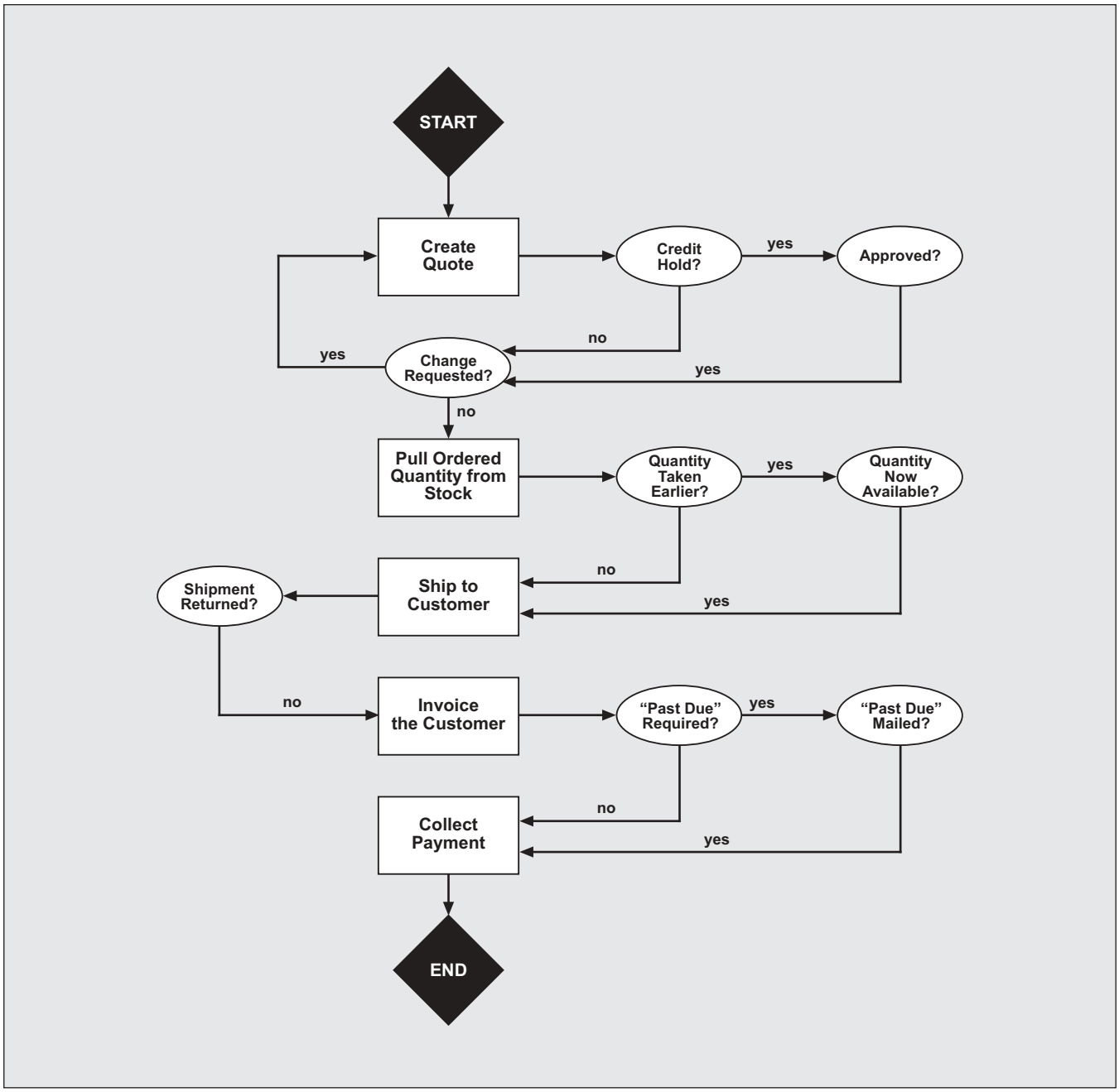
1c. Identify five kinds of exceptions that affect only some iterations:

- Some sales orders automatically get placed on credit hold.
- Some customers request quantity, price, or delivery changes prior to shipment.
- Sometimes the quantity meant for Sales Order A gets pulled early for Sales Order B.
- Some shipments get returned by the customer (e.g., damaged, wrong product).
- Some invoices do not get paid until a past-due reminder is sent.

1d. Using symbols of your choice, create a first draft of a process flow diagram:

- Include in your diagram the items from Step 1a, Step 1b, and Step 1c.
- The Step 1a items will represent the start-to-finish flow.
- The Step 1b items will be the handoff volume ranges.
- The Step 1c items will represent yes vs. no decision points.

*Figure 14 Jumpstart Exercise Example Process Flow Diagram*




An example process flow diagram is shown in **Figure 14**.

1e. Use your flow diagram to create specs:

- Get feedback from at least two of the direct participants on the accuracy of your diagram, and make changes to it as necessary.
- Reach agreement with these participants on a consistency goal for the entire cycle (e.g., 90 days) *and* on at least one of the exceptions

(e.g., 0% of all sales orders can have changes made to quantity, price, or delivery dates once the originally requested due date is 5 days or less away).

- More importantly, these two consistency goals then become specifications for any ERP project to preserve or to improve.

### *Exercise 2: Getting Specs Agreement via FMEA*

The failure modes (of inputs) and effects (on outputs) analysis method includes two different visuals designed to include a variety of opinions on a single sheet of paper. There is the "points ranking" version, and there is the "variation range" version. You should again include at least two of the direct participants while performing this exercise, which consists of the following steps:

2a. Create the "points ranking" version of the FMEA grid:

- Refer to the example shown back in Figure 11.

- Your five exceptions from Step 1c are the "things that can go wrong."

- Rank each of the five on a scale of 1 to 10 for all three questions (i.e., importance vs. likelihood vs. negative impact on the quote-to-cash cycle time). Which one of the five has the most points? The handoff step on your process flow diagram (from Step 1d) that this exception sits between becomes your target handoff.

2b. Create the "variation range" version of the FMEA grid:

- Refer to the example shown back in Figure 12.

- List two kinds of outputs related to your "target handoff" from Step 2a. As an example, if the handoff was the one between the Sales Order and the Quantity Pull from Stock process flow tasks, then two possible outputs to focus on would be: (i) Ordered Quantity Picked, and (ii) "All Ready" Signal to Shipping Dept.

- List three kinds of inputs. Three example inputs would be: (i) How Well Stock Replenishment from Purchasing Stays in Synch with Sales Order Demand, (ii) How Well a Safety Stock Target of 50 Units Stays Maintained, and (iii) How Many Requests from the Sales Department Get Made to Have Quantity Filled for a "Hot" Job Earlier than Scheduled.

- In the grid, estimate the negative impact on the two outputs from a 15% variance vs. a 30% variance by any of the three inputs. You can do the same kind of analysis with other percentages such as 2% vs. 5%. The objective is to reach agreement with the direct participants on how much variation + or - is acceptable. This + and - range then becomes the target for the participants to stay within.

  More importantly, this range then becomes a specification for any ERP project to preserve or to improve.

### *Exercise 3: Getting Specs Agreement via the Toll Gate*

In my article, I mentioned that the toll gate is something similar to a project's phase review. But, more importantly, I also mentioned that the toll gate documentation should be looked at after the ERP project cutover so you can see how well the team was able to predict the project's results. For this reason, each toll gate should include at least two metrics that define what measurement the prediction(s) will be compared against. Follow these steps for this exercise:

3a. Form your predictions:

- Review your parameters from Step 1b.
- Review your speed vs. consistency goals from Step 1e.
- Review your +/- variation range from Step 2b.
- Estimate the amount of department spending required to achieve those three.

3b. Provide spending feedback via a distribution chart:

- Refer to the example diagrams shown back in Figure 7.
- After the project cutover, begin to chart the spending trends for the departments directly involved in the quote-to-cash handoff you identified in Step 2a.

3c. Provide speed vs. consistency feedback via a distribution chart

- Refer to the example diagram shown back in Figure 8.
- Make your own diagram, and use the +/- variation range from Step 2b as the lower and upper "tolerance" limits for your diagram.
- After the project cutover, begin to track the actual performance for the three inputs into the quote-to-cash handoff, as well as the two outputs out of the handoff that you identified in Step 2b.
- Map the actual readings onto your chart.
- Calculate the variation that is actually occurring (i.e., the Sigma).

*Kurt Goldsmith is an IT productivity developer for Enowa Consulting, specializing in the diagnosis and resolution of integration issues between a company's division of labor (end users, managers, executives) and SAP software (R/3, BW, APO, CRM). He also has a lifetime performance record of one win and two third-place finishes from five career starts as a thoroughbred racehorse trainer. If you have questions or advice on either racehorses or SAP software's relationship with organizational building blocks, you may reach Kurt by email at kurt.goldsmith@enowa-consulting.com.*