

The Improvement Kata: Annual Improvement Planning Meets Agile

Kathy Iberle
Iberle Consulting Group, Inc.
www.kiberle.com

Abstract

Many IT and high-tech organizations create yearly plans for advancement and improvements. All too often, new information quickly invalidates the plan, necessitating costly re-planning. Team-level Agile retrospectives avoid this “big design up front” problem, but retrospectives don’t easily handle improvements that require coordinated effort across multiple teams.

The Improvement Kata solves this dilemma by providing an *agile* method for planning and executing large-scale improvements. Large-scale changes are systematically broken down into small pieces, and each of those pieces is treated as an experiment rather than a fixed plan. This enables the organization to navigate efficiently through the many surprises of process improvement, while implementing “the simplest thing that could possibly work”.

This talk will discuss the fundamentals of the Improvement Kata and share some experiences with applying the Improvement Kata in development and IT organizations.

Biography

Kathy Iberle is the Principal Consultant at Iberle Consulting Group, Inc., where she helps clients improve their productivity and quality. Her extensive background in process improvement and quality methods enable her to blend classic quality control with the best of current Lean thinking.

Kathy started at Hewlett-Packard, where over the years she worked as a developer, quality engineer, test manager, and process improvement expert in software, firmware, and IT. Kathy’s deep knowledge of Lean and Agile theory combined with a flair for pragmatic applications has enabled her to successfully extend Agile development methods into areas where most people think “Agile just can’t work”. She has published regularly since 1997 on quality and Lean topics. Today, she continues to explore new methods while helping her clients implement more effective processes.

Kathy has an M.S. in Computer Science from the University of Washington and an excessive collection of degrees in Chemistry from the University of Washington and the University of Michigan.

Copyright Iberle Consulting Group, Inc. 2015

Acknowledgements

Adam Light of SoTech Advisors (<http://www.sotechadvisors.com/>) introduced me to the Improvement Kata and was a key player in much of the work described in this paper. His wise advice also improved this paper considerably.

1 Introduction

Effective process change presents a difficult challenge. Far too often, we see the problems but don't arrive at successful solutions. Common causes of failure include:

- Starting so many change efforts that we can't finish most of them.
- "Analysis Paralysis". We can't decide where to start, so we keep analyzing instead of starting.
- Jumping to conclusions. We implement our favorite solution without knowing whether it is applicable to this particular problem.
- Insufficient support from the "powers that be".

However, the biggest obstacle to process change usually is the vast number of things that we *don't* know. We don't know how much our problems are costing the organization, we don't know exactly why they are occurring, and we don't know precisely what changes will resolve them. This makes it impossible to create a successful long-term plan. Each time we make a change, we discover more about our process. We uncover more problems and find better ways to solve the problem than those in our current plan.

Does this sound familiar? It's the same state of affairs which drove the invention of agile development methods – we can't know everything up front. Process improvement typically involves even more uncertainty than software development. There's little past data to help estimate how long a process change will take, and changes to processes often have large unexpected side effects.

The *Improvement Kata* is a process improvement framework which allows us to navigate this uncertain territory by taking one small step at a time. Just like agile methods, we make one small process change, evaluate the result, and adapt our plans in response to our learnings. The Improvement Kata steers the project using data from your real-life performance, rather than an industry-standard checklist.

In this paper, I will explore how the Improvement Kata works and share my recent experiences implementing the Improvement Kata

2 The Improvement Kata

2.1 Origins of the Improvement Kata

Kata is a Japanese word which means a small, structured practice routine or protocol. A Kata is practiced by the student, with feedback from a teacher, until the student has mastered the fundamentals and the pattern becomes second nature. A violinist practicing scales and a baseball player in batting practice are both following a *Kata*.

The Improvement Kata is based on the process improvement practices used at Toyota for manufacturing and related processes. The practices did not have a name at Toyota – they were simply "the way we do things here". American researchers eventually realized that these "kata" were a key part of implementing Lean practices within Toyota and named the practices "the Improvement Kata". Today, a leading researcher in this area is Mike Rother, the author of *Toyota Kata: Managing People for Improvement, Adaptiveness, and Superior Results* (2009), and a continually updated *Improvement Kata Handbook*: http://www-personal.umich.edu/~mrother/Materials_to_Download.html.

Most of the published work to date on the Improvement Kata involves, relatively predictable systems such as those in manufacturing. In manufacturing, *batches* of work are roughly homogeneous in size, arrive at a fairly predictable rate, and go through a fairly standard process. Health care also involves many predictable workflows such as appointment scheduling and lab tests. By comparison, software development and other forms of product development are highly variable. Batches of work arrive unpredictably, are of widely differing size and difficulty, and their sizes cannot be reliably known up-front.

As my colleagues and I have applied the Improvement Kata within IT departments, we've discovered the need for some modification of the original methods published by Mike Rother. As I explain the Improvement Kata, I'll point out some of the differences.

2.2 Steps in the Improvement Kata

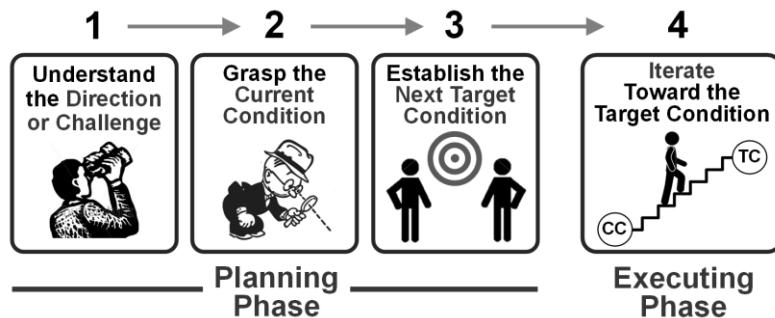


Figure 1: Basic Steps of the Improvement Kata. Rother 2015 A

1) **Understand the Direction.** Where do we want to go? The Improvement Kata starts with a long-term vision for the organization. Hakan Forss, who has been using the Improvement Kata in software development, gives some examples of typical visions for software development: (Forss 2014)

- Zero defects in production
- 100 percent value added
- Highest value first, on demand

Starting with the vision, leadership selects a single Challenge or “stretch goal”. This focuses the energy spent by the organization on process improvement, rather than spending the energy on a random selection of whatever seems urgent at the moment. The Improvement Kata doesn't specify a particular timeframe for the Challenge, but it is a good fit with the annual improvement planning or annual “scorecard” planning performed in many large corporations.

The goal or Challenge specifies what and when, but not how. It must be stated clearly enough so that we can tell whether it is achieved or not. The classic example of a well-stated goal is John F. Kennedy's 1961 address to Congress: **“I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the Moon and returning him safely to the Earth.”** [Kennedy 1961]¹.

2) **Grasp the Current Condition.** What is our starting point? This step is an assessment of the overall process and how it differs from the goal. This isn't meant to be a lengthy, detailed analysis, but rather simply captures what is currently known, focusing on aspects of the situation which are relevant to the future goal.

So far, the Improvement Kata is very similar to traditional annual planning for process improvements. There's a vision, a yearly goal, and an assessment of the current situation. Now, however, we “go agile”.

3) **Establish the Next Target Condition.** Where can we go in the next few weeks? We define a target – a better future state which can be achieved in a *short timeframe*. We don't plan the whole year, or even the whole quarter. Instead, we choose one aspect of the Challenge which can be affected right away, and we set a measurable short-term goal or “Target Condition” for that aspect with an “achieve-by” date. Once the achieve-by date is reached, we stop, assess the Current Condition again, and set a new Target

¹ Kennedy's vision wasn't explicitly stated, but it appeared to involve a future state where the U.S. citizenry and military felt comfortably superior to the Soviet Union.

Condition based on what we now know. As in an agile sprint, we take a small bite, deal with it, and then choose the next bite.

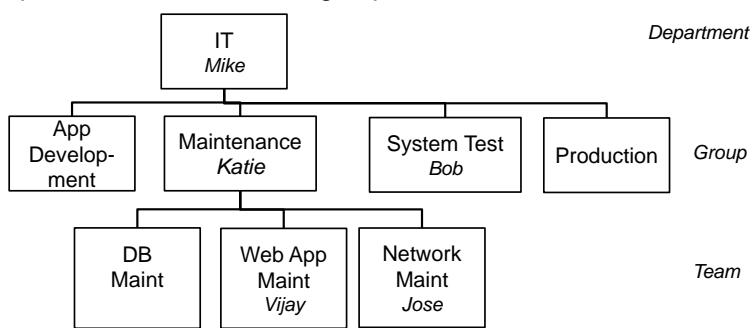
4) **Iterate Toward the Target Condition.** How will we get from our Current Condition to our Target Condition? We don't know exactly how to get from where we are to where we want to be, so we take a small experimental step every day through the unknown territory between the Current Condition and the Target Condition. After each step, we reconsider the next step. This constant course correction minimizes the time spent on ineffective paths. Now let's look at each step in more detail.

3 How Does the Improvement Kata Work?

3.1 The Situation

The Improvement Kata is easier to understand with an example, so we'll illustrate by walking through an imaginary annual improvement effort. This example takes place in a company of a few thousand employees. This company has both internal web applications to support business operations and customer-facing web applications. Both sets of web applications are handled by the same busy IT department, managed by Mike. Mike's department consists of four groups:

- App Development: creates new applications
- Maintenance: cares for applications in production
- System Test: serves both development and maintenance
- Production: manages networks, deployments, and so forth.



Each group consists of several teams.

This IT department has been struggling with patch failures. These patches fix defects in production systems, introduce minor new features, and install upgrades to deal with routine network and security issues. Sadly, about 20% of the patches fail in production.

Mike's boss, the VP of development, suggests that the IT department's annual improvement plan really should include getting on top of this problem.

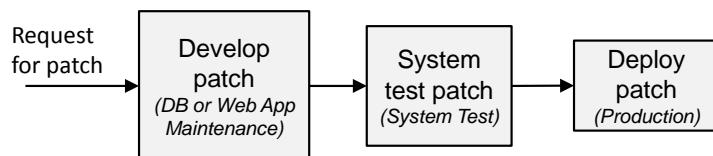
3.2 Planning: Set the Challenge

Mike and his managers agree that their vision is "Zero defects in patches to production systems." In their annual planning, they decide to aim for a 75% reduction in failures by the end of the year. Since 20% of patches currently fail, here's their challenge:

At the end of this calendar year, only 5% of patches fail in production.

Mike and his management team then look at the overall *value stream* for patches – the end-to-end delivery of value to the customer, beginning with the initial idea and ending with final deployment.

They sketch the value stream:



Mike sees that the end-to-end process isn't owned entirely by any one group in his department. Rather, it crosses several groups. This means Mike can't simply delegate the entire problem to the Maintenance group, because it might not be entirely solvable within that group. He has to keep an eye on the possible interactions between groups and teams. This "systems view" prevents unintentionally making the process worse by pushing work from one department into another.

3.3 Planning: Grasp the Current Condition

The next step is to understand what the situation is *now* with respect to the challenge. Rother's methods for measuring the Current Condition and Target Condition (Rother 2015 B) are not directly applicable to software development. We instead start with Rother's fundamental question "How is this process performing over time?" (Rother 2015 B, p. 75) and apply basic Lean principles:

- Identify the process and make it visible. We use a simple process map.
- Measure the *outcome* or bottom line of the process. How fast and well can it deliver value to the end users of the process' output?
- Look for the batches of work and measure them. User stories, anyone?
- Use data and measurements, not gut feel, opinions, or arbitrary assessment against an external standard.

We use metrics similar to what Forss recommends (Forss 2014 p 47-48).

- Overall *throughput* (# user stories completed per sprint)
- Average *Cycle Time* (end-to-end time to develop and deploy a story)
- Average Work in Process (# of user stories started but not finished at a specific point in time)
- # defects per story remaining after release.

Mike's management team currently measures the patch cycle time (average time to deliver a patch from request to deployment) and the failure rate, which are both outcome metrics. They use the value-stream map to capture the *Current Operating Pattern* - how the process works now.

The management team captures all this in a *Learner's Storyboard* (Figure 2). The *Learner* is the person who is responsible for learning about the process and changing it – in this case, Mike. The *Learner's Storyboard* is an Improvement Kata tool that is used at every level from top management to the people closest to the work. The Learner's Storyboard embodies several principles of Lean management artifacts:

- Limited space, to force a concise summary of the issue.
- Standard layout, so everyone knows where to look for different pieces of information
- Real data gathered by observing the current process.
- Visual and physical – usually a large sheet of paper on a wall.

Focus Process: Patch creation process.	Challenge: By end of this calendar year, only 5% of patches fail in production.		
Target Condition Achieve by:	Current Condition: <ul style="list-style-type: none"> • 20% of patches fail • Avg cycle time 10 days • Current operating pattern: 	PDCA Cycles Record Learner: Mike Coach:	

Figure 2 – Mike's Learner's Storyboard for the entire value stream

3.4 Planning: Set the Target Condition

The first steps of the Improvement Kata strongly resemble the first steps of more traditional annual planning methods. But instead of creating a detailed year-long plan, the Improvement Kata employs a cadence of short time boxes, just like Scrum sprints. Mike's company uses a three-week cadence, so Mike's next step is to define a Target Condition which he believes can be met in three weeks. Like the Challenge, a Target Condition is a description of the state you want to achieve, *not* a plan to get there.

Mike envisions a state in which one group of failures has stopped happening. To choose a group, he asks his managers "Why are the patches failing?". It quickly becomes apparent that there's no consensus as to what constitutes the biggest cause of failures, and no reliable data is available. The management team needs more information. Specifically, they need a profile of failures versus causes so they understand which causes create the most failures. Their first Target Condition is "A known profile of failures vs. causes". This is added to the Learner's Storyboard (Figure 3).

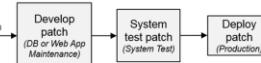
Focus Process: Patch creation process.	Challenge: By end of this calendar year, only 5% of patches fail in production.	
Target Condition <ul style="list-style-type: none">• 20% of patches fail• Avg cycle time 10 days• Known profile of failures vs. causes Achieve by: Now + 3 weeks	Current Condition: <ul style="list-style-type: none">• 20% of patches fail• Avg cycle time 10 days• Current operating pattern: 	PDCA Cycles Record Learner: Mike Coach:

Figure 3 – Mike's Learner's Storyboard with first Target Condition

In real life, Learners often encounter this situation. They haven't been thinking about or measuring the entire value stream, so they don't have the data to understand their system. The first Target Condition often must be "We can see how our entire value stream performs." This implies adding measurements, but we don't prescribe a specific set. Instead, we let the measurements develop out of the problem at hand by adding one measurement at a time and checking whether that provides enough information for now. In other words, do the simplest thing that could possibly work!

3.5 Planning: Identify the Obstacles

Mike and his managers spend a quarter of an hour brainstorming the *obstacles* to the Target Condition. Why isn't there already a known profile? They quickly come up with a list of reasons.

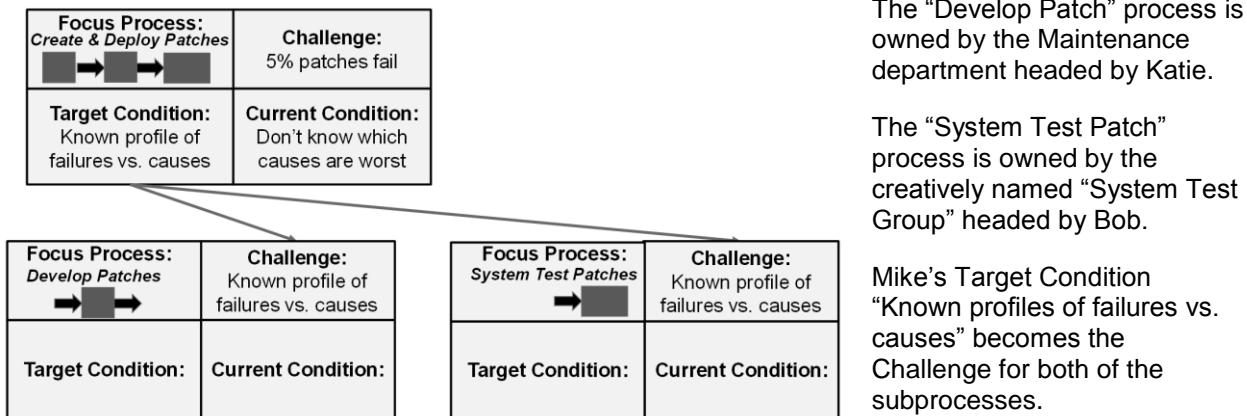
- Nobody is responsible for doing root cause analysis – it's not part of the regular process.
- We don't have an up-to-date list of patch failures to analyze.
- We're not sure how to do a root cause analysis.

Mike reminds the managers that the Target Condition is a profile of causes good enough to drive their improvement efforts, not a great root cause analysis process.

3.6 Decomposing the Problem: Where the Rubber Hits the Road

This is all very nice, but the problem isn't going to get fixed in management meetings. Processes get fixed at the *process level*, where people do the hands-on work. The managers need to decompose the problem into process-level pieces so individual Learners can tackle each piece. The decomposition is done in a fractal manner, where the Target Condition at the manager's level becomes the Challenge at the process level. As shown below, the problem is broken down by along the boundaries in the process map, which

are not necessarily the same as the boundaries in the organizational chart. A process may be co-owned by two departments, or languish in the “white space” between departments.



In order to get a hands-on view of the patch development process, Katie asks her technical lead, Mark, to act as the Learner. Katie and Mark work together to prepare the Learner’s Storyboard shown below. First they copy the Target Condition in Mike’s Storyboard into the Challenge of their Storyboard. They then describe the Current Condition, considering obstacles relevant to their group.

Katie asks Mark to take a small step by defining a Target Condition that is only one week away. Mark points out that they need a root cause analysis method before they can do the analysis. Developing a method seems like a realistic one-week goal, so they write the Target Condition as a future state “We have a method to classify defects by causes.” Katie asks Mark, “How will we know whether the method is successful?” They agree on two criteria 1) different people using the method on a given failure come to similar conclusions about the cause, and 2) the causes tell us something about how the failures were created. These criteria play the same role that acceptance criteria play for a user story.

Focus Process: Patch development	Challenge: A known profile of patch failures vs. causes.		
Target Condition We have a method to classify defects by causes. <ul style="list-style-type: none"> • Different users get same answers. • Causes suggest how failures were created. Achieve by: Now + 1 wk	Current Condition: <ul style="list-style-type: none"> • We don't have a known distribution. • We don't know how to do the root cause analysis. 	PDCA Cycles Record Learner: Mark Coach: Katie	
	Obstacles Parking Lot No method for doing root cause analysis.		

Figure 4 – Mark’s Learner’s Storyboard

3.7 Plan-Do-Check-Act

Next, Mark attempts to make an actual process change by creating a method for root cause analysis. In the Improvement Kata, the day-to-day activities of process improvement are performed in very small steps called Plan-Do-Check-Act cycles. We experiment towards the Target Condition in small, inexpensive, safe increments.

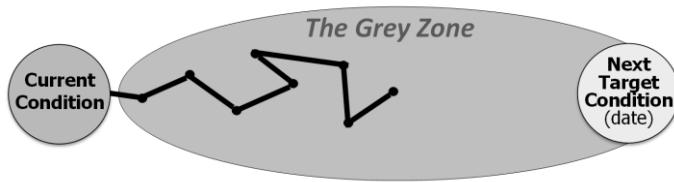


Figure 5 – Iterating towards the Target Condition. Mike Rother 2015 A.

Plan-Do-Check-Act² (PDCA) essentially describes the Scientific Method as applied to process improvement:

- **Plan:** Make a testable prediction (hypothesis)
- **Do:** Test your hypothesis with an experiment. (Go do it and find out what happens.)
- **Check** (or Study): Review the facts and data resulting from the experiment.
- **Act:** Adjust your plan based on what you have learned

Katie asks Mark to start a PDCA sheet in the standard Improvement Kata format. Katie and Mark meet every other day to review the progress. The PDCA sheet, like a lab notebook, captures both the results of the experiments and the learnings. This PDCA sheet shows how Mark progresses in small steps.

PDCA CYCLE RECORD					
Process: Patch development		Challenge: Known distribution of patch failures over causes. Current Target Condition: A method to classify defects by causes. Achieve-By Date: Jan 25.			
Start / End	#	What Will You Do? (Step)	What Do You Expect?	What Actually Happened	What We Learned
1/03	1	Send blanket email to IT asking who knows root cause analysis.	Somebody on some other team in IT has done this before.	Nobody has experience, although Rosa sent me a paper from a conference.	We don't have in-house capability.
1/04	2	Surf the web looking for a reference.	I will find some good descriptions of how to do root cause analysis.	The descriptions are very technical. Many are for manufacturing, not software. A book by Johanna Rothman looks promising.	Root cause analysis is done differently in different fields. In software, root cause analysis is also called defect cause analysis.
1/05	3	Order Corrective Action by Johanna Rothman.	It will have examples of what to do and some categories of causes.	The examples on pp 32-33 seem relevant, although more complicated than we need.	Some people do really detailed analyses of root cause, down to different types of coding logic.
1/10	4	Adapt the checklist on p. 32 to use for our analysis.	I can create a root cause analysis checklist today.	Done. I dropped the irrelevant categories e.g. bad prototypes can't be a root cause because we don't do prototypes for patches.	Not much.

² The PDSA cycle was formulated by Walter Shewhart in the 1930s. (Best 2003)

Excerpt from PNSQC 2015 Proceedings

Copies may not be made or distributed for commercial use

3.8 The Coaching Kata

Katie and Mike know that the Plan-Do-Check-Act cycle doesn't come naturally to most people. Most people can Plan and Do, but many find the Check or Study step challenging. At Toyota, there is a formal Kata for practicing Plan-Do-Check-Act, which Mike Rother refers to as the *Coaching Kata*. Let's see how Katie and Mark practice the Coaching Kata to speed their way through the PDCA cycles.

Each Learner has a mentor or *Coach*, whose role is to help the mentee or Learner use the scientific method and create change. The Coaching Kata consists of a daily standup meeting in front of the Learner's Storyboard and PDCA sheet, where the Coach asks the Learner the *Five Questions*.

1. What is the Target Condition?
2. What is your Actual Condition today?
 - a. What was your last step?
 - b. What did you expect?
 - c. What actually happened?
 - d. What did you learn?
3. What obstacles are preventing you from reaching the Target Condition?
Which *one* are you addressing now?
4. What is your next step (next experiment)? What do you expect to see?
5. How quickly can we go and see what we have learned from that step?

Let's listen to a coaching session between Mark and Katie.

Katie: "What is the Target Condition?"

Mark: "We have a method to classify defects by causes."

Katie: "What is your actual condition today?" (See Table 1).

Mark: "I finished the checklist yesterday, so I guess we have a method now."

Katie: "Will the answers from this method be useful for reducing patch failures?"

Mark: "Yes. The categories are kinds of mistakes, like Missing Requirements or Coding Errors."

Katie: "Will different users get the same answers from this method?"

Mark: "They should."

Katie: "How could you find out quickly if they will?"

Mark: "We'll see when people use it to do the root cause analysis."

Katie: "A full analysis will be quite a bit of work. Is there a way to find out sooner?"

Mark: "I guess I could test it on just a few defects with one or two other people."

Katie: "That would be a good next step. When you do that step, what do you expect to see?"

Mark: "It will work."

Did the new process work? Here's what happened next:

1/11	5	Try checklist with a couple of example defects.	It will work.	I didn't find a category for defect #652. It was caused by bad data.	We need to add "bad data in database" to the possible causes.
1/12	6	Give checklist to Rosa to try.	Rosa will put the defects in categories and we will see a pattern.	Rosa put all the defects into just one category: "coding mistake".	Need to help developers understand that not all defects are coding errors.
1/13	7	Talk to team then try checklist again.	Defects will be in more than one category.	Defects sorted into five categories.	This process is workable and gives us some useful info.

Mark's experience is typical of what we observe with our clients. The first attempt at measurement or process change often doesn't work, but the very small steps allow people to quickly and safely learn from the failure and rapidly proceed in a more productive direction. The conscious reflection required by "What did you learn?" changes the perspectives of the Learners. For instance, simply asking "What have you learned?" can quickly turn grumbling about difficult people who just don't want to adopt a new process into thoughtful inquiry about the difficulty of the process being proposed. (See Light 2015.)

3.9 Success Card

Back to our story! Mark has reached his first Target Condition. Katie asks him to write a Success Card, which she posts on the wall next to Mark's Learner's Storyboard.

While success cards didn't originate at Toyota, we find that they help Learners and teams recognize and celebrate progress. Because change happens in such small increments, it can be hard to see the progress without a physical record.

Learner: Mark	Coach: Katie
Target Condition: A method for root cause analysis that works.	Results: We developed a process that fits our types of defects & we can train people to use it.
Elapsed time for the PDCA cycles: ~10 days	Benefit: Now we can do root cause analysis and find out why our patches fail.

Figure 6 - Success Card

3.10 The Next Iteration

Now Mark meets with his Coach, Katie, to set a new Target Condition. The Challenge remains the same, but the Current Condition has changed. Mark and Katie set a new Target Condition and brainstorm obstacles.

Focus Process: Patch development	Challenge: A known distribution of causes for patch failures.		
Target Condition We have a known distribution of causes. Achieve by: Now + 1 wk	Current Condition: <ul style="list-style-type: none"> We don't have a known distribution. We know how to do the root cause. 	PDCA Cycles Record Learner: Mark Coach: Katie	
	Obstacles Parking Lot No list of failures to analyze. Nobody assigned to do the analysis.		

Figure 7 - Mark's second Learner's Storyboard

Katie notices that one of the obstacles – “No list of failures” – was previously assigned to Bob and his system test group. She contacts Bob, who provides a list of failures found in the previous month complete with defect identifiers. The Improvement Kata structure ensures that managers of different teams maintain alignment so that they are able to help each other.

By the following week, Mark’s team has used the root cause analysis method to produce a concise report on the causes of the failures found in the previous month. (Figure 8).

3.11 The Chain of Coaching

The Improvement Kata is designed to bubble up data from the process level to the levels above. Mark and his team have just provided an important piece of new data. Mike and his management team see that the largest category of issues is “unexpected user actions”. There were no user stories for these actions, the developers didn’t design or code for them, and the various levels of testing didn’t notice the gaps. Since this category is quite large, addressing it should reduce patch failures substantially.

Once again, more information on the Current Operating Pattern will be needed. Why are the user stories missing? Who might be able to predict the users’ real-life actions? (Hint: System testers often are able to predict the odd things users do.) Both Katie and Bob will ask their groups to find out more. With this additional data, Mike and his managers can define a new Target Condition to prevent the gaps or detect them before release. This could include changes in the Development process, or in System Test, or both.

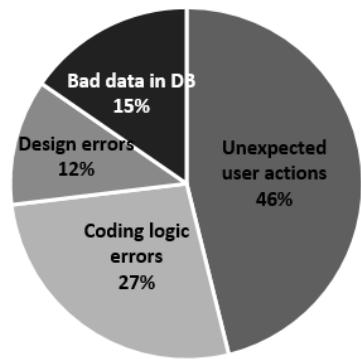


Figure 8 – Patch Failures vs Causes

Focus Process: Patch creation process.	Challenge: By end of this calendar year, only 5% of patches fail in production.	
Target Condition • Avg cycle time 10 days Achieve by: now + 3 weeks	Current Condition: • 20% of patches fail • 45% of failures due to unexpected actions by users • Avg cycle time 10 days 	PDCA Cycles Record Learner: Mike Coach:
Obstacles Parking Lot • User stories aren't fully covering what users do • Testing not reflecting real users		

Figure 9 - Mike's second Learner's Storyboard

The teams are now driving changes based on data about their own process, and the types of failures they encounter. There is a natural forum to discuss whether these changes will affect more than one team and if so, how to prioritize. The Improvement Kata gives Mike and his team a concise and current view of the probable costs and benefits of changes being proposed, and a good view of the improvements made.

4 Reflections on the Coaching Kata

The other parts of the Improvement Kata bear some resemblance to other process improvement methods such as Hoshin planning, but the Coaching Kata is unique. When I was first introduced to the Improvement Kata, I thought that the Five Questions and the daily meetings seemed unnecessarily formal and a bit silly. After all, PDCA has been taught and practiced in many other process improvement

methods without all this folderol. However, after observing numerous coaching sessions, I've come to appreciate the value of the Coaching Kata.

Learners discover things for themselves, through rapid experiments. The hands-on experience and rapid feedback makes their discoveries stick in their heads. For instance, I've seen Learners discover key points of process improvement for themselves, such as the advisability of testing processes before disseminating them. In an amazingly short time, Learners were actively applying these learnings and expanding on them as they created useful change in their own processes. I've taught the same concepts in a more traditional way to people who were trained in PDCA as part of Hoshin Planning and/or Six Sigma. They didn't progress up the learning curve nearly as fast as did people using the Improvement Kata.

Here are some of the things I observed over a year of watching Coaching sessions.

- Although most people remembered the “scientific method” from their school days, at first very few could compose a good hypothesis. It took practice to start recognizing their assumptions as assumptions rather than facts.
- Most people wanted to go directly from “what actually happened” to “what will I do next” without asking “what did I learn?” When they were required (via the Five Questions) to explicitly consider what they had learned, the next step often changed.
- If the meetings were not at a regular time and kept short, the participants stopped meeting. Once the Learner and Coach stopped meeting, progress usually ground to a halt.
- When “How quickly can we go and see” was not asked by the coach, PDCA Cycles slowed down and Learners tried to take larger steps. The sense of urgency seemed to be lost.
- Steps requiring half an hour or so of effort over a day or two fit into people’s workloads.
- Coaching meetings held 2 or 3 times a week maintained momentum, but once a week did not.
- The Coach doesn’t always need to know “the right answers”. The Five Questions will push both Coach and Learner in the right direction.

The Coaching Kata reminds me of pair programming. Two people look at the same problem simultaneously, with the Coach watching for the common pitfalls (such as jumping to conclusions) while the Learner is thinking about the specific problem at hand.

Mike Rother says, “The Improvement Kata pattern is the fundamental way of working at Toyota, and there are several Toyota practices through which this pattern gets utilized and reinforced. The research found the Improvement Kata pattern underlying all of them: Daily Management, Daily Problem Solving, A3, Improvement Events, Standard Work, Quality Circles...This means copying visible Toyota activities – such as the A3s – without bringing along the enabling coaching environment is unlikely to change much.” (Rother 2015 A, Ch. 1). The Coaching Kata is used all the time, not just initially to teach PDCA.

5 The Improvement Kata and Annual Planning

The Improvement Kata strongly resembles many other process improvement methods historically used in annual improvement planning. The initial assessment, setting of targets, and the hierarchical breakdown is characteristic of Hoshin planning (Kenyon 1997, ISixSigma 2015). The “systems view” approach was a fundamental part of the Rummel-Brache methods which were popular in the 1990s (Rummel 1995), although those methods didn’t always identify the value produced by the value stream. PDCA-style thinking is used in both Hoshin and Six Sigma, albeit with considerably larger steps.

The Improvement Kata differs substantially from the more traditional methods in its use of:

- Iterative or rolling-wave planning. Rather than cascading down yearly targets to each department and each team, the first cascade sets targets only a few weeks out. Results are rolled back up to the top, and new targets cascade down, on a regular and frequent cadence.
- The Coaching Kata. Very small steps and significant emphasis on the experimental nature of PDCA.

- Active daily management involvement. Managers are a key part of the Coaching Kata and are responsible for seeing and managing the big picture across teams.

Implementing the Improvement Kata in its entirety is challenging because it changes the way process improvement is done from top to bottom in the organization. I did observe individual teams independently employ the Target Condition “sprints” and the Coaching Kata to effectively solve problems identified in their team-level retrospectives. But these teams still struggled with problems that stretched across multiple teams or departments. They couldn’t see the whole system accurately from their vantage point, so they couldn’t pinpoint the causes of the issues precisely. I saw the biggest “bang for the buck” come when the Improvement Kata was implemented across a group or department. Once managers could see flow between teams, simple but non-obvious improvements came to light. (Light 2015)

In an organization without any standard process improvement method, it would probably work to start by adopting the Coaching Kata within a single team. Once several teams know how to state a Current and Target Condition and use PDCA to get from one to the other, the teams’ managers could start iterative planning together, building the Chain of Coaching from the bottom up.

6 Conclusion

The biggest problem with annual improvement planning is that it’s, well, annual. You are expected to plan all your improvements for a whole year despite not knowing what the changes will require. You make guesses in order to plan, but then the plan has to be changed again and again. All this re-planning is very costly as well as frustrating.

Team-level agile retrospectives avoid all the “big planning up front”, but they struggle with changes that require alignment across the organization. A single team can’t see all the nuances of how work flows through multiple different teams, let alone influence all those other teams. Without a view of the entire value stream, individual teams are also prone to sub-optimizing the system, so their team runs faster but the overall delivery of value has not improved and may even have gotten worse.

The Improvement Kata provides the focus and organizational alignment expected in an Annual Plan without all the re-planning. Teams work on *different* obstacles to achieving the *same* goal at the *same* time, allowing them to build on one another’s work and assist one another, while taking advantage of each specific team’s unique perspective on the problem at hand.

As we’ve seen, the Improvement Kata also incorporates many of the other advantages of agile:

- It uses Lean and agile methods that have been proven to deal well with uncertainty: cadence, time boxing, and iterative planning.
- PDCA steps split the work into very small batches, which provide the quick feedback crucial to working efficiently in the face of unknowns.
- Learner’s Storyboards and PDCA sheets act as “information radiators”, providing visibility to status in a lightweight manner.
- Managers have enough visibility into the system so they can see where help is needed to make improvements happen, but the actual decisions happen at the process level where people are most familiar with the process.
- The choices in the Improvement Kata are driven by data taken from specific processes in specific organizations. The focus is on making *your* process deliver your value faster and better, not on achieving a checklist of practices dictated by experts.

I see the Improvement Kata as a promising alternative to the schedule-driven methods often used in annual improvement plans.

References

- Aulinger, Gerd. Rother, Mike. Rosenthal, Mark. 2013. "The Coaching Kata: Chain of Coaching".
<http://www.slideshare.net/mike734/the-coaching-kata-chain-of-coaching> Downloaded June 15, 2015.
- Best, M and Neuhauser, D. 2003. "Walter A. Shewhart, 1924, and the Hawthorne Factory." Quality and Safety in Health Care. 2006 April. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2464836/> (accessed July 25, 2015)
- Forss, Hakan. 2014. "Toyota Kata – Habits for Continuous Improvement". MIX-IT 2014 Lyon France. <https://hakanforss.wordpress.com/2014/04/29/toyota-kata-habits-for-continuous-improvements-mix-it>. (accessed June 15, 2015).
- Forss, Hakan. 2013. "Stop doing Retrospectives and Start your Toyota Kata".
<http://www.slideshare.net/HkanForss/stop-doing-retrospective-and-start-your-toyota-kata>
- Iberle, Kathy. 2013. "Lean in the Software Test Lab". *Proceedings of the 31st Annual Pacific Northwest Software Quality Conference*.
- ISixSigma. 2015. "Hoshin Planning: Making the Strategic Plan Work".
<http://www.isixsigma.com/methodology/hoshin-kanri/hoshin-planning-making-strategic-plan-work/> (accessed June 15, 2015).
- Kennedy, John F. (May 25, 1961). *Special Message to Congress on Urgent National Needs* (Motion picture (excerpt)). Boston, MA: John F. Kennedy Presidential Library and Museum. Accession Number: TNC:200; Digital Identifier: TNC-200-2. Retrieved August 1, 2013.
- Kenyon, David. 1997. "Strategic Planning with the Hoshin Process". Quality Digest. May 1997.
<http://www.qualitydigest.com/magazine/1997/may/article/strategic-planning-hoshin-process.html#> (accessed July 15, 2015).
- Kwinn, Kathryn Y. 2003. "Enhancing the Total Customer Experience through HP-UX Patch Quality". *Proceedings of the 21st Annual Pacific Northwest Software Quality Conference*. Pp. 235-246.
- Light, Adam. 2015. "Managing for Continuous Improvement Using the Improvement Kata and Coaching Kata". Cutter IT Journal. Volume 28, No. 6. June 2015.
- Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lead Product Development*. Celeritas Publishing. Redondo Beach, California. 2009.
- Rother, Mike. 2010. *Toyota Kata: Managing People for Improvement, Adaptiveness, and Superior Results*. New York: McGraw-Hill.
- Rother, Mike. 2015 A. "The Improvement Kata Handbook, version 31.1." The Toyota Kata Website.
http://www-personal.umich.edu/~mrother/Materials_to_Download.html (accessed June 15 2015).
- Rother, Mike. 2015 B. "The Improvement Kata Practice Guide, version 10." The Toyota Kata Website.
http://www-personal.umich.edu/~mrother/Materials_to_Download.html (accessed July 23 2015).
- Rummel, Geary A., Brache, Alan P. 1995. *Improving Performance: How to Manage the White Space on the Organization Chart, 2nd edition*. Jossey-Bass, San Francisco.