A Pattern System of Underlying Theories for Process Improvement

Michael Van Hilst and Eduardo B. Fernandez
Florida Atlantic University

Abstract
An underlying theory is a framework of goals, solutions, and assumptions that guide how we observe situations and define problems. In science, the underlying theories, like particles, waves, relativity, plate tectonics, and evolution, are well known and well understood. In process improvement, the underlying theories are not so well understood. To address that problem, we present a system of patterns. The system includes five patterns for theories that underlie many of the well known practices in process improvement. The patterns are Have a Plan, Copy What Works (aka Best Practices), Eliminate Waste (aka Flow), Consider All Factors (aka Systems Thinking), and Incorporate Feedback and Learning. These patterns can be found in ISO 9000, PMBOK, CMMI, SPICE, Lean, TQM, Six Sigma, and Agile. We also present 4 anti-patterns common in critiques of process improvement. The four anti-patterns are ATAMO (and then a miracle occurs), Buy A Silver Bullet, All Problems Are Nails, and Solutions Must Be General. Like all patterns, the patterns in this system are structural configurations that solve a problem in a context.

Introduction

Process improvement has a rich history in manufacturing and the military. More recently, it has been a topic of much discussion in software and product development, as well. Process improvement as a field made significant gains during the two world wars, when “business as usual” was severely challenged by unprecedented demands for productivity and innovation. The 60’s and 70’s saw a focus on quality, and a concern with high rates of failed projects, particularly among large projects involving large amounts of software. Today, process improvement faces new challenges as the focus shifts from getting the job done to improving return on investment (ROI), reducing cycle time, and delivering customer value.

Process improvement inspires controversy, discussion and debate, both in business and in software development. Proponents of the different approaches often divide themselves into factions, promoting the strengths of their approach, while dismissing those of others. In software engineering, Boehm and Turner (2004) and the SEMAT initiative (2010) have called for reconciliation and efforts to harvest all that is good. But differences have not been resolved and the debates continue.

Any discussion of alternative approaches and their divergent views should be grounded in reflection on the underlying theories and assumptions. For a given perspective, the underlying theories define what is relevant and valid in a situation. Theories frame and identify the problems and the solutions. Different theories frame the problem in different ways. In science, theories like relativity, evolution, big-bang, plate tectonics, germs, genes, waves, and particles are well known and used explicitly to frame situations in different ways. In process improvement, the theories behind the views are not so well known. Their application is largely implicit.

By reflecting on underlying theories, we build an understanding of the role of each theory in shaping the different approaches to process improvement. Perhaps, through reflection on the theories, we can find common ground and reach a level of mutual understanding where the discourse turns to how the different theories actually work together. With that goal in mind, we present this system of patterns of underlying theories for process improvement. The system includes five basic patterns for process improvement: *Have a Plan*, *Copy What Works*, *Eliminate Waste*, *Consider All Factors*, and *Incorporate Feedback and Learning*. These five theories underlie many of today’s best known process improvement practices. Have a Plan is emphasized in the ISO 9000 standards for quality management and in the PMBOK (Project Management Body Of Knowledge). Copy What Works (aka Best Practices) is emphasized in the Software Engineering Institute (SEI) Capability Maturity Model Integrated (CMMI) and the ISO 15504 standard for Software Process Improvement and Capability Determination (SPICE). Eliminate Waste (aka Flow) is emphasized in many of the Toyota Lean practices. Consider All Factors (aka Systems Thinking) is emphasized in Total Quality Management (TQM) and Six Sigma. Incorporate Feedback and Learning is emphasized in Agile methods and practices. We also include four anti-patterns that are common in critiques of process improvement. The anti-patterns are: *ATAMO* (and then a miracle occurs), *Buy a Silver Bullet*, *All Problems Are Nails*, and *Solutions Must Be General*.

The patterns presented here are not recent inventions. All nine are widely used. Like all patterns, they capture the wisdom of experts in a form that is easy to understand. Also, like most patterns, they are frameworks for solutions, and not solutions themselves. They are presented here in a form that is meant to improve the understanding of the solutions in which they are applied, and also to be useful for making comparisons among competing approaches to improving processes. All of the patterns apply to human processes in general, and development projects in specific. Most of the examples used in the discussion, however, are drawn from the domain of software development.

James Coplien gave the following definition of a pattern: “A recurring structural configuration that solves a problem in a context, contributing to the wholeness of some whole, or system, that reflects some aesthetic or cultural value” (2005, p.4). The 5 patterns here frame solutions to aspects of the same problem, but with different structural configurations reflecting different aesthetic and cultural values. It is these differences that make them especially interesting. By presenting the patterns together as a system, we highlight their differences, and also their commonalities.

There already exist a good number of patterns for processes and organizations. Scott Ambler (1998, 1999), for example, has two books on process patterns. James Coplien (2004) also has a book on organizational patterns and several articles. Those patterns are about processes and practices for developing products. The patterns here are different. They are about processes and practices for improving processes.

In discussing each of these patterns with practitioners, a common response is, “Oh, it should be divided into this pattern and that pattern.” That is precisely the point. They are seminal patterns upon which the more specific approaches and practices in process improvement are based. They are meant to capture common underlying history, theories, and assumptions of the numerous, more familiar forms in which they are used.

Each of the 5 presented patterns has a characteristic goal and a corresponding solution.

- For *Have a Plan*, the goal is preparation and consistency. The solution is to create and have available more and better plans, and to make them known and followed.
- For *Copy What Works*, the goal is doing as well as other organization or industry leaders. The solution is to adopt the same practices as the others. The focus is on adoption rather than refinement.
- In *Eliminate Waste*, the goal is high efficiency. The solution is to focus on eliminating waste in all its forms, and institutionalizing a commitment to efficiency that involves everyone.
- For *Consider All Factors*, the goal is complete optimality. Without completeness, opportunities may be missed, causes overlooked, and quality, safety or security compromised. The solution is to look at every factor, internal or external, that contributes to deviation from the ideal, and to find and fix the root causes that allow or encourage problems to occur.
- Finally, for *Incorporate Feedback and Learning*, the goal is to deliver what customers need, when and where the customer needs it, even when the definition of what they need changes fast and often. The solution is to focus on sensing, feedback, and responsiveness, to shorten feedback loops, allow frequent input, and remove barriers to change.

Each of the 5 patterns also has its own distinct foundation and history.

- *Have a Plan* is grounded in a notion of defining processes in terms of smaller well defined and repeatable steps. In process improvement it has a military origin where success, and survival, depends on everyone knowing what to do and how to do it.
- *Copy What Works* is grounded in a view of science where theories are found and confirmed by repeated observation. In process improvement, its historical roots can be traced to Positivism and Scientific Management.

- *Eliminate Waste* uses an analogy in fluid dynamics, called flow, where any slowing, stopping, or changing of direction reduces overall efficiency. Its use in process improvement can be traced to Frank Gilbreth, who, as a young man, had observed brick layers getting tired from unnecessary motion.
- *Consider All Factors* originated with observations in biology and agriculture. You can't tell plants and animals how to grow. But you can improve their growth by identifying and controlling factors that affect how well they grow.
- *Incorporate Feedback and Learning* is grounded in the learning and action theories of John Dewey, and the practical experience of building guns in World War II to track and hit a moving target under automatic control.

Each of the anti-patterns presented here has a characteristic form of mismatch between the goal and the solution.

- In *ATAMO* (and then a miracle occurs), the characteristic is a well defined goal and a poorly defined solution or solution process.
- In *All Problems Are Nails*, the characteristic is a solution appropriate to an old problem, that doesn't fit the new problem.
- In *Buy a Silver Bullet*, the characteristic is a solution that is dictated or applied with little or no analysis of the actual problem or the fit between the solution and the situation.
- In *Solutions Must Be General*, the characteristic is a search for solutions that work in many places, while discounting details and solutions specific to the situation at hand.

Terminology

Before presenting the patterns, we must first define some terminology used in the discussion of processes and process models. These terms are significant when we compare different theories and discuss their assumptions. The terms include strategic, operational, and tactical levels of concern, general vs. situated reasoning, inclusive vs. generative plans, open vs. closed systems, and deterministic vs. non-deterministic processes. We also expand upon uses of value and knowledge in process improvement.

Discussions of process plans and controls divide concerns into three levels: strategic, operational, and tactical. *Strategic* concerns are long term and long range. They might deal with how an organization defines its mission, how it sets priorities, and how it wants to approach its customers. *Operational* concerns are more day-to-day and deal with who will do what and when. *Tactical* concerns deal with how tasks should be performed and focuses more on skills and practices. Getting the job done is an operational concern. But improving value and efficiency can often involve strategic and tactical issues, as well. Some approaches to process improvement are criticized for being strictly operational and ignoring other levels of concern (Purvis, Santiago, and Sambamurthy, 1999).

Decision making in processes can be general or situated. *General* means valid for many or all cases. *Situated* means valid only for the immediate, specific case. The issue of general vs. situated occurs in discussions of validation, and again in discussions of effectiveness and applicability. Put simply, should we demonstrate validity by showing that it works for us, or do we demonstrate validity by showing that it works for everyone else? In development processes much that is learned may be situated and hard to generalize. But Reinertsen (1997) observed that the three main sources of variability are: the type of product being developed, the scope and complexity of the product, and the people doing the work. Within an organization, these three factors show little variation. Thus the issue of general vs. situated can be different within an organization than across organizations.

An *inclusive* plan describes a solution to a problem. Both the problem and the solution are known in advance. A *generative* plan describes how to create the solution to a problem. The actual problem and solution are not known, at least not in sufficient detail. But the method of solution is known. The plan guides a process to create solutions, typically through composition, synthesis, or specialization. Situated solutions require a generative approach. Have a Plan is applicable to both inclusive and generative plans. This distinction between inclusive and generative plans is also significant in the ATAMO and Solutions Must be General anti-patterns.

Closed systems are systems that can be understood independent of their context. Their behavior is not affected by outside context. *Open* systems are systems that affect, and are affected by, their environment, and thus cannot be fully understood in isolation. The “closed world” assumption is a simplification that makes modeling and analysis tractable, but sometimes has undesired consequences when applied in the real world.

Deterministic processes consistently and reliably produce the same output for a given set of inputs. *Non-deterministic* processes do not reliably produce the same result. The result of a non-deterministic process cannot accurately be predicted. But non-deterministic processes may be bounded, meaning that the result reliably falls within a specific range. In complex systems, non-deterministic effects are significant and often have more impact. It is a mistake to dismiss these effects. Different assumptions or expectations about the deterministic nature of a model or process call for different forms of validation and statistical testing.

Two additional concepts merit explanation, *value* and *knowledge*. Customer value is a common metric by which all aspects of a process can be judged. It is common to see the statement, “Automate or eliminate non-value-add” as part of a process improvement initiative. Customer value places the focus on whatever properties of the process, and its output, increase value for the customer. It includes features, cost, timeliness, and quality. The motivation for focusing on value is that increasing customer value is often the primary goal of a business process – it’s what the customers pay for. Organizational value is like customer value, but directed at the growth and well being of the organization in which the process occurs. Given that the organization is ultimately dependent on the customer, the two are closely linked.

Knowledge is the ability to produce (or predict) a desired result. Knowledge is the output of product development. Manufacturing produces a product; development produces the knowledge of how to make the product. In product and software development, process improvement should be focused on the production and handling of knowledge. This perspective was first introduced by Patterson (1992), and later expanded by Reinertsen (1997) and others. It has significant implications for process improvement in product development. It shifts the focus away from counting artifacts and provides a metric by which to analyze the inputs and outputs of intermediate processes. Tracking the production and handling of knowledge plays an important role in several of the patterns presented here.

Have a Plan

0. Name

Have a Plan

Also known as

Documented, Planning (Koskela & Howell, 2002)

Other variants

Checklists

1. Intent:

Deciding what to do in each new situation costs time and effort. In making decisions there is always the hazard of making a wrong choice. New decisions must be communicated to many people whose actions must be coordinated. Having plans in place institutionalizes the decisions that have proven effective in the past and builds consistency and predictability into the process. When everybody knows what to do, the process becomes smooth and orderly.

2. Example

Sally's team has been turning out products for in house use. But management is considering outsourcing some of the work. At the same time, there may be opportunities for Sally's team to get extra work by bidding on other contracts. But to compete for project work, Sally's team must be able to reliably predict time and cost, and to deliver on their commitments.

3. Context

Documented plans are helpful when a process has many details that must be communicated and are easy to forget or overlook. Plans are especially important when the inclusion or omission of a detail can significantly impact a project's outcome.

History

Rituals and recipes have been used since the earliest recorded times. But our understanding of business planning derives largely from military origins. In the military context large numbers of people must be coordinated, and plans are put to the test, with life and death consequences. Chapter 1 of Sun Tzu's *The Art of War*, written in the 6th Century B.C., concerns planning. The word "plan" itself comes from the Latin "planum" or flat surface, and means a drawing on a flat surface.

Military planning divides plans into three levels: strategic, operational, and tactical. Strategic plans are concerned with the setting of broad goals and policies, the direction to take, and how to organize resources – i.e. to win the war. The name "strategy" derives from the Greek word "strategos", which means "army leader." Operational plans address a particular mission, or theatre of operation including logistics, and are concerned with who will do what, when and where – i.e. to win the battle. Tactical plans address the means used to gain an objective. Tactics address challenges posed in each situation and concern what actions to take – i.e. to take the hill, or not get shot.

When the United States entered World War I, its industry performed poorly at meeting the needs of the military (Gantt, 1922). The managers were largely trained in finance and economics. Resources were not well allocated and decisions were poorly made. French industry, from which the US bought guns and aircraft, was organized around the ideas of Taylor and Fayol, where planning was central. In the U.S., Henry Gantt responded with improved methods of planning.

During WWII, the United States faced severe shortages of both skilled labor and skilled supervisors, for work that posed daily challenges. The War Production Board developed a program called Training Within Industry that trained workers and managers in the best ways to perform and communicate the work to be done. This approach was not continued in the U.S., but was adopted by Toyota under the names Standard Work and J-Programs (see Implementation below).

In the 1950's, Booz Allen Hamilton developed Program Evaluation and Review Technique (PERT) for the US Navy's Polaris Missile program – an ambitious project with considerable uncertainty. PERT established many of the practices we now consider standard for project planning, including the Work Breakdown Structure and various methods for assembling small plans into large projects.

Foundation

The application of plans in project management is grounded in a transformation theory of projects (Koskela & Howell, 2002). In the transformation view, a project is a process that transforms a set of inputs to a desired output. The top level process is divided into smaller steps, called tasks. The tasks are assigned to resources and arranged in a schedule to address dependencies and resolve conflicts. The process is run by executing the tasks according to the plan. In this view, the challenges for process improvement include: making sure that all the tasks that are needed are in the plan, eliminating tasks that aren't needed, standardizing task execution (see Copy What Works), accurately predicting the needs and duration of each task, scheduling tasks so workers don't sit idle, and controlling deviation (see Copy What Works, Eliminate Waste).

4. Problem

Organizations cannot function effectively to achieve goals if the people involved do not know what to do. The more complicated the activity, the more there is to know about how to do it.

Forces

- Projects often have commitments to reach a certain point of progress by a certain day. Sometimes these commitments are commitments to investors and customers. Other times, these commitments involve coordination with persons and resources whose availability must be scheduled in advance.
- Decision must be made before progress can continue. If the decision has not been considered before the situation occurs, the process must wait while decisions are formulated and then communicated. Corresponding arrangements must still be made, adding to the delay.
- Different decisions have different impacts on process time and cost. Allowing too much variation in choice, especially for commonly occurring decisions, makes it difficult to predict a project's overall time and cost.
- Some decisions are too dependent on the specifics of a situation to be made in advance. Yet, even in such situations, the process for gathering facts, making choices, and getting approvals should be known in advance. A range of viable choices available may also be selected in advance.
- Standards or certifications often require the organization to show that it has documented processes in place and that members of the team know their corresponding roles in the documented process.
- Process knowledge is an organizational asset that should be captured and reused to maximize its value. Team members gain experience and get better at doing things over time. New members must be taught this experience, while old members may need some form of reminder about which solutions have proven successful in the past.

5. Solution

Keith Moore, one of HP's top internal process consultants, once said that the key to being productive is to always know exactly what you need to be doing. To achieve that goal at the team level, good plans must be defined, learned, and followed. When faced with a new situation, team members should have a plan to follow that covers that situation. At each decision point in its execution, the plan indicates which decisions to take and which steps to follow. Plans define the roles that different members of the team should play and the responsibilities involved. Even for experienced developers performing routine tasks, following a consistent set of steps can avoid oversights. In more challenging situations, "knowing the drill" improves outcomes and success. As new knowledge and experience becomes available, plans should be improved. The improvements should be communicated.

6. Discussion

Anne Pankhurst (2007) provides the following definition of a plan: a systematic approach of organizing work to be performed to go from a present state to a defined desirable state. In debates about planning, the argument is not really about whether or not to have a plan. Nobody would argue that we shouldn't know what to do. The question is what type of plan is appropriate.

Planning occurs at multiple levels. Tactical plans and policies describe how to perform tasks or parts of tasks. Operational plans describe what steps to take, when, and by whom. Scheduling, risk management and change management are all operational. Strategic plans make sure everyone is working with the same objectives and vision. Plans can span years and hundreds of people. But a plan can also involve only one person for an hour or less, like the plan for creating a document or installing a device.

Plans can be fragmented, covering interchangeable parts of a larger plan. With plan fragments, new plans do not have to be constructed from scratch. Frameworks, like the Rational Unified Process (Peraire, 2007) and the evolving SEMAT framework, have plan and method fragments that can be composed to form different plans depending on the situation.

Plans can be complete, incremental, or generative. A complete plan covers everything needed to achieve a goal from start to finish. The desired state is to be done. In dynamic situations, it may not be possible to plan everything in advance. An incremental plan covers only a short period in the process, after which another plan is needed to cover the next period. The desired state is to have made significant progress. A generative plan covers the steps to be taken to create another plan. A generative plan enables the team to move smoothly to the creation of a needed plan, when the time is right. The desired state is to know what to do next.

Plans can be "followed to the letter" or serve as points of reference. In Wysocki's Continuous Process Improvement Model, teams are monitored for compliance, but are allowed to deviate. If they choose to deviate from the standard plan, they must document their reasons for doing so. Coplien (2007) observed that most of the work in the organization he was studying did not follow the written plans: "80% of the work was being done under documented waivers."

Plans can be rigid or flexible. Rigid plans must be followed regardless of the circumstances. Flexible plans allow for situated adjustments. The SPI Manifesto advocates flexible models in process improvement: "the best models with highest utility are dynamic models. They have built-in ways to take circumstances and contingencies into account, and they change behavior dependent on the status of your […] effort."

Plans can be prescriptive, as in "the thing that must be followed," or descriptive, as in "the best way to solve a problem." A project schedule is prescriptive. Patterns are descriptive. Coplien (2005) draws this distinction in his book under the heading "Organizational Patterns are Inspiration Rather than Prescription."

Plans can be supporting. The goals of a supporting plan may differ from the more traditional goal of transforming inputs to outputs, or achieving a new state. In the PMBOK, for example, communications plans and risk management plans organize policies and behavior to support the process and avoid problems.

Plans can be incomplete and lack important detail (see ATAMO) or overly complete with too much detail. Both Wysocki (2009) and Coplien (2007) describe "shelf feet" or "decimeters" of plans that are never read because there is too much, or the level of detail is too specific for any real situation. Wysocki recommends a library with plans documented at three levels. The three levels are a one page executive summary, a two page descriptive reference, and a full tutorial (Wysocki, 2009).

Plans can be written or learned. At Toyota, plans are descriptive and learned (see Implementation, below). The learned approach is also emphasized in Coplien's (2005), Domain Expertise in Roles, and Apprenticeship patterns. Certifications, like ISO 9000, often require both written evidence of plans and demonstrated knowledge.

Plans are easy to criticize for their necessarily simplified view of reality. Every situation has something that is unique. But plans are still important. To paraphrase George E.P. Box's quote, "All models are wrong – some are useful," all plans are wrong, but some are useful.

7. Implementation

The creation of plans, and plan templates, requires expertise and foresight. Smaller plans can be made by an individual who documents the steps and processes he or she has applied to address a problem or achieve a goal. Bigger plans are often made in a planning meeting with representatives from different areas of expertise. A planning meeting typically uses an *affinity diagramming* technique, with cards or sticky notes, to identify, group, and scope the elements of the plan. Participants write each candidate plan element on a sticky note. The notes are all posted on a wall or board. During the meeting, the notes are grouped into collections of similar or related elements. Elements in each group are combined, or subdivided, to form sub-processes and tasks.

Toyota takes a different approach to plan development, dissemination, and enforcement. Plans are developed by a pilot team who experiment with how best to perform each task. It is then the responsibilities of managers to teach each worker how to perform the work to achieve the best results. The relationship between managers and workers is much like that on a professional sports team, with managers as trainer and coach, and workers as skilled experts. How to train the workers is itself a plan, with managers receiving instruction and supervision. Instruction takes time. Eventually, managers and workers alike look for ways to improve the standard practices. It is a part of their commitment to continuous improvement. Proposed changes are reviewed and tested, before being accepted for Standard Work (Huntziger, 2006).

8. Example resolved

Sally's team has documented many of their activities as plans. All members of the team are aware of these plans and refer to them in project planning meetings and in their work. Team members have a clearer understanding of their roles and of their relationships to other members of the team. Communication has improved. New members become productive more quickly. Of particular value have been the plans for how to proceed when difficulties arise. As a result, deadlines are being met with greater consistency and problems are being addressed earlier. Using the plans for reference, Sally has been able to create better schedules for new projects. Upper management has gained confidence in the in-house team and continues to use them for important projects.

9. Known uses

PMBOK

The Project Management Institute's Project Management Body of Knowledge (PMBOK) places a heavy emphasis on planning. The PMBOK forms the basis for Project Management Professional (PMP) certification. The PMBOK is itself a plan for how to manage generic projects. Plans specified in the PMBOK include: requirements management plan, staffing management plan, procurement management plan, communications plan, schedule management plan, change management plan, risk management and response plans, contingency plan, and quality management plan. Project execution in the PMBOK consists almost exclusively of dispatching plan elements, and monitoring their performance.

Szymanski and Neff's definition of process improvement

In a CrossTalk article titled, Defining Software Process Improvement, Szymanski and Neff (1996) proposed the following definition:

> *"Software process improvement is a deliberate, planned methodology following standardized documentation practices to capture on paper (and in practice) the activities, methods, practices, and transformations that people use to develop and maintain software and the associated*

products. As each activity, method, practice and transformation is documented, each is analyzed against the standard of value added to the organization."

The authors proposed the following three steps (plus additional details for each step):

1. Prepare and document a flowchart of the entire process from beginning to end using conventional flowcharting diagrams.
2. Analyze each activity documented in the flowchart to determine if it is "value-added".
3. On a continuous basis in a structured, repeatable fashion, identify ways to eliminate or modify non-value-added activities in the process. Add any activities that would attach value to the process.

Situational Method Engineering

In their Developer Driven Approach to Situational Method Engineering, Jarvi, Hakonen, and Makila (2007) discuss a variety of approaches to constructing plans from method fragments. Some approaches select the methods and construct plans before the project begins. But they also describe approaches that construct plans from pre-existing components at the time that the need arises, and also allowing methods to be changed in the middle of the project. In their discussion, they emphasize the need to adapt plans to the specifics of each situation.

Louis D. Brandeis quoted in Gilbreth's *Primer of Scientific Management* (1914, p.3).

"Scientific Management means universal preparedness, the same kind of preparedness that secured to Prussia a victory over France and to Japan a victory over Russia. In Scientific Management nothing is left to chance; all is carefully planned in advance."

10. Consequences

Advantages

- Plans capture an organization's knowledge and can guide new workers to follow established and well proven practices.
- Plans can be refined and improved over time, thus institutionalizes learning for everybody's benefit.
- Well made plans allow resources and events to be scheduled in advance so they can arrive when they are needed, and not be wasting time when they are not needed.
- Established plans promote consistency in the process, which enables commitments to be made and met.
- By anticipating situations with a plan, everyone will know what they are supposed to do and the process will proceed smoothly.
- Contingency plans and plans for developing a response can avoid the situation where everyone stands around not knowing what to do next.
- Having documented plans that are known and followed demonstrates to others that the team is prepared.
- Often repeated plans, or parts of plans, can become opportunities for automating parts of the process.

Disadvantages

- Planning in advance may use information that is incomplete. "Premature planning" can result in sub-optimal or even wrong decisions being made. (See Herbert Simon's Bounded Rationality.)
- The plans for a general or imagined situation may not fit specific details of the actual situation. Following the predefined plan may not be the best solution in that case.
- Relying on preset plans may discourage workers from paying attention to the peculiarities of a situation that represent risks or opportunities.

- Plans take elements of decision making and control out of the hands of the individuals involved. Over-reliance on plans treats workers like interchangeable machines. Workers who do not feel valued for their intellect and creativity lose motivation.

Many of the disadvantages of plans are not disadvantages of plans, per se, but rather consequences of too much or too little specificity or a lack of flexibility.

11. See Also

Copy What Works, Domain Expertise in Roles and Apprenticeship (Coplien, 2005)

Copy What Works

Name

Copy What Works

Also known as

Best Practices, Standardization.

1. Intent

There are many ways to perform a task or complete a project – some better than others. Businesses can’t afford to try too many practices to discover which ones work best, and they can’t afford to fall behind the competition. When others perform better than your, copying what they do is a way to bring parts of your own process up to the same level.

2. Example

John’s company has gotten a foothold in the market. They put out their initial product two years ago, and have since issued their first update. John put together a good team of developers who worked hard to get that first product finished and working. But progress has always been chaotic. For the long run, they can no longer rely on heroic effort. They would like their process to be more stable and consistent. They also need to focus more on quality.

3. Context

Copy what others do can be used to jump-start or leap-frog process improvement when an organization is lagging in one or more practices. Adopting best practices should be considered when others in a similar setting are performing better. Adopting best practices is done in the context of a process improvement initiative. The initiative requires staff to perform an analysis to assess the current practices and the appropriateness of any new practices. The organization must be willing to change and it must have commitment at all levels. A period of training and adjustment may be required. Continuous management attention and a follow-up assessment should be expected.

History

Adopting best practices in process improvement can be traced back to Frederick Taylor and Scientific Management. While working at Midvale and then Bethlehem Steel in the 1890’s, Taylor used a stopwatch to find workers who performed each task the fastest. When he found the fastest worker, he made that worker’s practice the standard for that task. Sometimes he combined elements from two workers. All workers were then trained to follow the standard practice. Productivity doubled. Taylor collected his writings and talks in a book, called *The Principles of Scientific Management* (1914). The goal of scientific management was to improve every aspect of the process by finding the “one best way.”

Foundation

Best Practices is grounded in a philosophy called Positivism. Developed by August Comte in the 1830’s, Positivism sought to apply scientific principles to all human endeavors. In the Positivist view, scientific knowledge is discovered and validated through the empirical observation of repeating patterns associated with a desired outcome. Once validated, this knowledge is codified and used by practitioners to solve problems in the field. Problem solving in this model consists of identifying a problem and selecting the corresponding known best solution. In the early 19^{th} Century, practices in many fields involved quackery, superstition, and myth. By the late 19^{th} and early 20^{th} Centuries, this Positivist model of knowledge and practice brought discipline to many professions (Schön, 1983).

4. Problem

Organizations that want to improve may not know what to do or in what ways to improve. If facing a choice of alternative practices to apply, they may not know which practices will work the best. When investing in process improvement, they should try to do at least as well as their competitors.

Forces

- Gradual improvement is possible. But the team's own practices are ad hoc and incomplete. Improvement would be faster if complete proven practices were adopted and made standard.
- The businesses must not put its processes at risk by applying changes that have not already been shown to work. Management requires any new practices to already have a proven record of success in other projects and organizations.
- Management prefers solutions that are recognized as industry standards. Such solutions can be justified using general guidelines and do not require deep understanding or analysis of the situation.
- Management decision making models and practices often require all options to have quantifiable probabilities and a defined expected value. Managers are trained to weigh all options and make decisions using models with numerical results.
- Process improvement initiatives are costly and disruptive. Allocating dedicated staff, performing measurements and analyses, and conducting interviews and questionnaires all involve substantial time and cost. The organization expects a measurable and significant improvement that will justify the cost of the initiative.
- The organization would like their improvement efforts to be recognizable to the outside world in the form of a certificate or award. To achieve formal recognition, the organization must undergo an audit and demonstrate that they apply recognized best practices.

5. Solution

Practices that are applied by the most successful organizations, or the most successful groups within an organization, are considered "best practices." Unless there is a good reason for doing something different, a team that wants to achieve the same high levels of success should apply the same practices.

6. Discussion

In discussions of certifications and standards, the term "best practices" often implies "industry" best practices. In the 1990's there were many efforts to reduce the number of projects in military contracts that were behind schedule, over budget, or failed. The preferred solution was to promote the adoption of industry best practices. The idea was that companies that experience serious failures should learn from those that don't. Industry best practices are often promoted in professional standards and industry publications. Certification standards themselves often contain "reference" practices.

To adopt a new best practice, one must first be aware of it. In addition to published standards, reading trade publications and attending conferences are common ways learning about practices that are benefitting other organizations. Direct interaction with members of outside teams, as occurs in conferences and networking events, allows team members to discuss specific challenges and experiences, and exchange best practice ideas.

Industry best practices are not the only source of best practices. Frederick Taylor originally used organizational best practices for his standard practices in the steel industry. Within an organization there may be variations in how activities are performed. Some variations will prove to be better than others. Wysocki's Continuous Process Improvement Model also encourages organizations to allow some variation and reap the most successful variations to improve their own best practices. From this perspective, best practices can be seen as an extension of planning. The important point in this case is that the standard practices are documented and/or communicated, and that there is a process for advancing the organization's standard practices to capture the best practices demonstrated to work in their situation, however they may be found.

Simply adopting industry best practices carries some risk. Practices that worked for others may not work well in all situations (see Solutions Must Be General). Adoption should be accompanied by local validation that evaluates how well it is working, and not just how well it is being followed. If the practice is not being followed, there may be a good reason that an alternative path was chosen.

Even in the general case, some practices gain popularity and strong support without proper validation. Such practices are called "fads." The software industry is not the only industry that faces this danger. In June of 2006, shortly before the market crash, Federal Reserve Chairman Ben Bernanke was quoted referring to high risk financial instruments as "best practices" that should be "adapted and disseminated to a broader array of financial institutions" (Kling, 2010).

7. Implementation

The best practices approach to software process improvement has four basic steps. First, identify the gap between the existing "as is" process (the baseline) and the "to-be" best practices (the benchmark). Second, develop a plan to close the gap. Third, execute the plan. Finally, check the results. The problem is the gap. The solution is the best practice. Before applying a new practice, an analysis is performed to assess the expected value of the change, whether the organization is sufficiently capable and committed, and whether the change is consistent with business goals. After the change, an analysis is performed to assess the level of conformance to the proposed practices, and whether the expected value was achieved.

For an alternative approach to adopting and disseminating standard practices, see the Implementation description in Have a Plan.

8. Example resolved

John's company adopted best practices in several areas where their earlier practices had been weak. Their process is now more reliable with less chaos. New releases of their product are now being delivered closer to schedule and they see an improvement in product quality.

9. Known uses

SEI CMMI

The Software Engineering Institute's Capability Maturity Model – Integrated (CMMI), uses best practices to guide and appraise organizations. The CMMI divides process concerns into 25 areas. Each area defines goals for different levels of maturity, and recommended practices to achieve those goals. "CMMI models are collections of best practices that you can compare to your organization's best practices and guide improvement to your processes. A formal comparison of a CMMI model to your processes is called an appraisal." (http://www.sei.cmu.edu/cmmi/)

Enterprise Resource Planning

Enterprise Resource Planning (ERP) systems integrate all of a company's sales and production information to facilitate decision making across departments and across the enterprise. They are also used to comply with regulations like IFRS, Sarbanes-Oxley, or Basel II. ERP systems tend to be big and complex and are notoriously difficult to customize to fit existing processes. The situation is described in Wikipedia as follows:

> "Prior to ERP, software was developed to fit individual processes of an individual business. Due to the complexities of most ERP systems and the negative consequences of a failed ERP implementation, most vendors have included "Best Practices" into their software." (http://en.wikipedia.org/wiki/Enterprise_resource_planning)

From the SAP website:

> "For the past three decades, SAP has collaborated with our most successful partners and customers to develop what has become one of our key packages, SAP Best Practices.... Our best

practices can help you control costs, reduce risk, and drive more value from your SAP solutions." (http://www.sap.com/services/bysubject/servsuptech/index.epx)

Extreme Programming

In the design of Extreme Programming, Kent Beck and Ron Jeffries adopted a carefully chosen set of best practices and combined them to create a new process. To quote from the Wikipedia entry for Extreme Programming, "The methodology, after all, takes 'best practices' to extreme levels." These practices include test driven development, unit testing, pair programming, continuous build and integration, iterative and incremental development, prioritized requirements, time boxing, design simplicity, and patterns based design. In books and in workshops, Beck and other consultants teach organizations how to adopt these same best practices. Scott Ambler also lists the best practices of Agile modeling in a diagram on his website: http://www.agilemodeling.com/essays/bestPractices.htm. Many of the practices in Extreme Programming are generative – ways of finding and solving problems quickly.

10. Consequences

Advantages

- By adopting practices that worked well for other projects and organizations, a team should realize similar benefits.
- Proven best practices carry fewer risks than unproven practices.
- It is easier to justify the choice of an industry recognized best practices, than a practice that is custom developed or designed on the spot. It is easier to convince certification assessors, and upper management, of their value.
- Adopting a new practice can often achieve a quicker improvement than the gradual evolution of an existing practice.
- Industry standard practices are often well supported with available tools and consultants.
- In some areas, developing good practices involves a steep learning curve. Security practices and meeting new government regulations are two examples. Learning from others can provide a significant jump-start in bringing an organization up to standards.

Disadvantages

- Judging a team on how well it follows standard practices discourages innovation. "Thinking outside the box" is discouraged.
- Practices that work well in a variety of situations may not be the optimal choice for a specific situation, especially if they are adopted without local (situated) adaptation to the situation on the ground (see Solutions must be General). Best practices as "the one best way" can be over-enforced by "the process police."
- Over-reliance on best practices can be counter-productive. If an organization is already applying best practices, additional best-practice initiatives will focus more on enforcing the process than finding solutions to problems (see All Problems Are Nails).

11. See Also

Have a Plan, Buy a Silver Bullet, All Problems Are Nails, Solutions Must Be General

Eliminate Waste

0. Name

Eliminate Waste

Also Known As

Flow, Lean Development

1. Intent

Work adds time and cost. A process that delivers the same result with less work saves both time and cost. Unnecessary work consumes time, energy, and resources. Unnecessary work also distracts attention from essential goals and tires the workers. It is not always easy to identify or eliminate work that isn't needed. Focusing on throughput and flow draws attention to details of a process that are likely signs of unnecessary work which could or should be eliminated.

2. Example

Betty's team has been developing variations of the same product for years. Their processes are highly repeatable and well documented. They have achieved high levels of certification for quality and maturity and can predict within a month how long it will take to develop the next product – 23 months from commitment to production. Lately, their competition has been getting better and is turning out new products in 14 to 16 months. By the time Betty's products reach the market, their "new" technologies have already been available from these other companies for months, and at lower price. Betty's company is losing money.

3. Context

Flow views a process in terms of the flow or trajectory of parts, workers, tools, and machines. A more efficient process has smoother flow, with fewer stops, waits, restarts, and changes of direction.

History

Flow can be traced to Frank Gilbreth, a contemporary of Frederick Taylor, and is best understood by contrasting his work with that of Taylor (see Copy What Works). Taylor looked for the worker who performed each task the fastest by measuring performance with a stopwatch. The other workers were then trained to duplicate the fastest worker's behavior. By this method, the best that can be achieved is to reduce the variance in the time it takes to perform the task, and shift the average to its lower bound. In contrast, Gilbreth was looking for workers who performed each task with the least effort. As a young brick layer, he had observed brick layers stooping down to pick up the next brick and then standing up to put it in place. By placing the bricks on a raised shelf at the same level as the work being done, those motions could be eliminated. Stooping down and standing up was wasted effort and caused fatigue. By reducing waste, productivity improves for even the fastest worker.

The goal in flow is to identify signs of waste, and find ways to avoid it. Gilbreth used high-speed cameras to study how each task was performed. He eventually realized that the highest efficiency is characterized by smooth flows of parts, workers, and machines. Stopping, waiting, restarting, and changing direction are all signs of wasted effort. Gilbreth called his method Motion Study, in contrast to Taylor's Time Study.

In the 1950's and 60's, Taiichi Ohno and Shigeo Shingo applied flow in Toyota's manufacturing. They developed a series of methods and practices, which, in the West, have come to be known as Lean. The Lean approach identifies seven forms of waste: defects, inventory, waiting, transportation, motion, over-production and over-processing (see below). They also promoted the ideas of just-in-time (JIT) and one-piece flow.

Flow analysis was used to reduce pit stop times in car racing to under 5 seconds, and by athletes who broke numerous records in the 2008 Summer Olympics. Hilbreth personally changed the way doctors and nurses exchange tools in the operating room – a practice that remains to this day.

Foundation

The concept of flow has a physical basis in fluid dynamics. In the flow of liquids and gases any slowing, stopping, or changing of direction reduces overall efficiency. Even small perturbations produce instability, turbulence, and drag.

Common techniques for measuring flow apply models from queuing theory. In queuing theory the steps in a process are modeled as a series of queues. Work makes its way through the process by advancing from one queue to the next. Process performance and the interaction among process elements can be analyzed in terms of starvation (empty queues) and waiting (queues with backlogs).

4. Problem

To improve efficiency, processes need to be more streamlined. Traditional process improvement, based on Plans and Best Practices, often seems to do just the opposite, adding new practices and overhead.

Forces

- To improve a process, the process itself must be analyzed. Systematic process improvement requires a theory of how to look at a process, what to find, and how it can be improved.
- The goal of process improvement is to produce more value for less cost and in less time. To gain everyone's commitment to process improvement, the effort should be clearly connected to that goal.
- In fast changing markets, windows of opportunity are short. Technology becomes obsolete very quickly. Success goes to organizations that deliver newer products in less time than the competition.
- Shortening development time should not come at the expense of worker stress, fatigue, and overtime. Making workers work harder, faster, or longer is not a viable long-term solution.
- Management accounting principles, and the standards for CMMI level 5, both call for improvement to be metrics driven. Ideal metrics should be objective. Beyond inputs and outputs, it is not obvious what else to measure. The cost of collecting some metrics can be high.

5. Solution

Flow focuses directly on efficiency. It views the process and frames the analysis in terms of reducing effort and waste. For flow, the process in question is the process on hand. Flow analysis can guide the replacement of practices to achieve more efficiency. Once practices are in place, an awareness of flow can guide continued refinement and adaptation to the situation on hand.

6. Discussion

In process improvement books aimed at managers, eliminating waste is often defined in terms of automating or eliminating non-value-adding steps and activities. For example, steps 2 and 3 of Szymanski and Neff (1996), quoted as an example for Have a Plan, state: "Analyze each activity documented in the flowchart to determine if it is 'value-added'," and then "Identify ways to eliminate or modify non-value-added activities in the process." This approach reflects a strictly operational view. Much of the value in flow, achieved by Gilbreth, Toyota, and others, is more fundamental and requires a more detailed and intimate understanding of the process.

Flow is best known for revolutionizing practices in manufacturing. But it does not only apply to manufacturing. In product development, issues of flow are found in the creation and handling of information or knowledge. Examples include looking for information, putting information down, leaving it idle, picking it up again, waiting for it to be available, and passing it from one group to another. Knowledge is lost, depreciates, and incurs added costs to be found, transferred, or recovered. Some processes are more efficient at generating knowledge than others. Applying flow to development means improving efficiency in generating and handling knowledge.

Ohno's 7 Wastes in manufacturing have direct correlates in development.

- *Defects* refer to defects that are not corrected at the time they occur. Defects require later rework, which adds cost but no new value. Defects which remain in the product reduce value to the customer.
- *Inventory* is work that is produced before it is needed, and sits for a period as inventory. In-process inventory adds throughput time with no increase in value. Items in inventory lose value and may increase cost at their time of use. Unused knowledge is inventory. It fades from memory and becomes increasingly expensive to recover. Queues of change requests and problem/bug reports are a form of inventory.
- *Waiting* occurs when work that could be done is not being done due to delays or bottlenecks in other parts of the process. Testing is often a source of bottlenecks in development processes. Defects that should be acted on must wait for testing to reveal them. Tasks waiting for developers, who must first be freed from other tasks or projects, is another bottleneck.
- *Transportation* refers to a variety of problems that occur when materials move from one group to another. Transportation in development is called a hand-off. The recipient of a hand-off spends additional effort to acquire knowledge that the prior handlers of the artifact already had.
- *Motion* refers to a workspace that is not well organized with tools and materials ready-at-hand when and where needed. In development, motion includes searching for information, a problem that often accompanies task switching. Developers must recover information that they themselves may once have had, but have now forgotten, or at least have lost the train of thought needed to take the next step. Often, they must also repopulate their workstations with tools and documents for the task.
- *Over-production* refers to things that are produced, but never used or needed. In development this can apply to features, options, and artifacts. The likelihood of over-production, or over-engineering, increases with the remoteness of the intended use or customer. The developer will over-produce when trying to anticipate what might possibly be needed. In the terminology of Agile development, over-production is called YAGNI, short for you ain't gonna need it.
- *Over-processing* refers to activities that have little benefit in terms of customer value. It is often applied to activities that serve the process, but not the customer. Over-processing also describes anything that would involve, cause, or require less work if addressed in some other way. Candidates could include ineffective testing, brittle architecture, outdated practices, or poor tool use. Developer education can remedy some forms of over processing.

Toyota itself has applied the same Lean thinking to product development. They claim that their engineers can achieve 80% value-add efficiency, compared with an industry norm of 20% efficiency (Kennedy, 2003). Lean development practices are the topic of a growing number of books.

Allen Ward, who studied the Toyota development process, defines three knowledge wastes (2007).

- *Scatter* characterizes poor and disrupted flows of knowledge due to physical, temporal, and cultural distances, changing roles, or chaotic practices.
- *Handoff* wastes knowledge due to the gaps and barriers between channels whenever knowledge, responsibility, action, and feedback are separated.
- *Wishful thinking* occurs when decisions are made without data, or knowledge is lost. Responsibility is given, or taken, without all the data or knowledge that goes with it. Knowledge that is present at one moment is lost for lack of a good way to capture it.

Flow treats defects as a problem of value (they reduce value) and cost (they require time and effort to find and repair). Their treatment is integral to every step of the process. In contrast, when defects are viewed in isolation, they are more likely to be treated and scheduled as external accidents, and addressed by separate steps in the process.

In a flow analysis the process to be analyzed is not a plan or a document. It is what you are currently doing. The actual process must be observed.

7. Implementation

Flow can be applied directly, by studying actual process flows, or indirectly, by looking for signs of waste. Flow can also be applied implicitly by adopting practices that are known to improve flow (see Copy What Works).

Process flow can be modeled using any of a number of process diagramming techniques. A value stream map, for example, identifies the activities in a process that are directly involved in delivering value. The process is then analyzed by inspection to identify delays and bottlenecks. Two different views are appropriate. One view looks at a workstation and examines the flow of work as it builds up or passes through. The activities at that point in the process are studied to determine how they contribute to value or impede flow. The other view follows a unit of work as it passes though the process from beginning to end, examining changes in its velocity at different points along the way. In the development process, a unit of work might be a feature, task, or requirement.

Measurements of flow can be analyzed using queuing theory. Optimal flow minimizes both time-in-process or takt time, and work-in-process. Time in process is the time it takes a unit of work to pass completely through the process, from its first appearance to its final exit (e.g. from work order to tested, inspected, repaired and approved for release). Work-in-process is a measure of the amount of work present in an intermediate state at a given point in time. Waterfall processes typically have a large percentage of work in process during the life of a project. Iterative and agile processes stage more of the work and typically have much less of the work in process at any one time.

Progress in the process can be displayed as an S-curve, showing cumulative progress versus time. Deviations from a smooth flow appear as bumps in an otherwise smooth curve. Work-in-process and time-in-process can be derived directly from the two progress S-curves for work started and work finished, by applying Little’s Law. It is sometimes hard to trace rework to the requirement or task whose completion is being delayed. Little’s Law avoids this problem.

Signs of waste can be found by direct inspection, or by directly asking people what they are doing. In a Lean organization, workers are trained to identify signs of waste on their own. In this way continuous improvement is institutionalized at all levels.

Toyota’s Lean methodology has yielded a variety of practices that improve flow. These include one-piece-flow, just in time (JIT), mistake-proofing, and source inspection. One-piece-flow is a practice of keeping work progressing through to completion, as opposed to letting work accumulate in batches. Just-in-time has work arriving just as it is needed for the next step of the process. In development, JIT is applied by generating knowledge where and when it is needed and where it will have the best impact. Mistake-proofing, or poke-yoke, designs parts so that they can be assembled without hunting and possibly making mistakes. Color-coding connectors and plugs is an example. Source-inspection inspects work immediately at the point where a mistake is possible. In Extreme programming, the practice of pair programming creates source inspection to catch typos and other kinds of errors. Test-first development provides both a mistake-proof objective (it has to fit the test), and source inspection.

8. Example resolved

By studying flow and looking for waste, Betty’s team has been able to identify several ways that they can reduce cycle time in their processes. They were able to identify many of the seven forms of waste, and changed the way they approached testing and rework, which had relied on batching. They are now looking at some of Toyota’s methods for speeding up development. With everyone involved, they continue to improve their processes. Their products are now reaching market earlier, and with more innovative features.

9. Known uses

Toyota Product Development

Toyota applies flow in its product development. Their focus is on delivering a steady flow of new products. They had observed that organizing the work around separate products resulted in delays, rework, and duplication. Problems found in prototype testing required doubling back to earlier points in the process. By organizing work around subsystems, Toyota smoothed the flow of development. Using a process called Set Based Concurrent Engineering, Toyota develops high-risk and low-risk subsystems in parallel. "If a subsystem proves unworkable, a proven subsystem is always available, which eliminates the need to double back" (Kennedy, 2005).

Poppendiecks' Lean Software Development

The Poppendiecks apply flow in their consulting work and in their books (2003, 2007). Their work uses value stream maps, Pareto charts, and Little's Law. In their book, they identify churn as a significant source of waste in software development projects.

Reinertsen's Design Factory

Reinertsen shows how and why to reduce cycle time and work-in-process. In his books, he explains how to apply accounting principles and queuing theory to the creation of knowledge in product development (1997, Smith & Reinertsen, 1998). He also explains various models of cycle time, work-in-process, and Little's Law. Reinertsen has a later book called *The Principles of Product Development Flow* (2009).

10. Consequences

Advantages

- Flow puts attention directly on how the process delivers value. There is clear guidance on what to look for, and the kinds of changes to be made.
- The basic concepts in flow are understandable by everybody involved. Everyone is encouraged to participate in the effort to improve and see the results. Fine grained improvements in every role across the process are valued and supported.
- Flow directly addresses time. The types of improvements made are likely to reduce process time. Progress is measured by the amount of time that has been saved.
- Flow is supported by direct measurement of the process. Progress is measured continuously at every station and can be automated.

Disadvantages

- Flow requires everyone's participation. It cannot be dictated from above or performed as a short term initiative (see Buy a Silver Bullet).
- Flow requires situated analysis. The improvements to be made must be discovered by observation of the actual process by engineers who understand the process. The solutions to be applied cannot be taught in a class or found in a book (see Solutions must be General).
- Improvement is focused internally and may suffer from NIH (not invented here). Bigger opportunities from wholesale replacement of parts of the process by Best Practices created elsewhere may be overlooked.

Consider All Factors

0. Name

Consider All Factors.

Also known as

Systems Thinking, Systems Theory, Quality Management.

1. Intent

The performance of processes and people are affected by many factors, including the organization's culture, external events, properties of the environment, and the process itself. Changes in any of these factors can affect performance either positively or negatively. Systems Thinking looks at the context, people, and processes, and how they interact, to discover factors that, if changed, will lead to better results.

2. Example

Phil's company has applied documented best practices and follows a complete set of plans. They even have long checklists for each process step. But they still make costly mistakes. With each mistake, they add more detail to the plans and create new items to be checked. But the situation isn't getting better and morale is starting to suffer.

3. Context

The Systems Thinking views processes in terms of the interrelationships among factors, subsystems, and results. Careful analysis, and attention to detail, identifies factors that negatively or positively contribute to a desired result. Systems analysis is inherently situated and focuses on the process in place and its immediate context.

History

Systems Thinking can be traced to the Austrian biologist, Ludwig von Beralanffy, who created General Systems Theory in 1928. Von Bertalanffy wanted to explain the behavior of systems and organizations that didn't fit conventional closed and deterministic models.

In management, W. Edwards Deming applied systems thinking to improving quality in business processes. In the late 1940's, Deming started teaching his approach to Japanese businesses.

It is notable that Deming and von Bertalanffy both started out studying plants and animals. Deming worked at the U.S. Department of Agriculture from 1927 to 1939. You can't make plants grow faster or cows produce more milk by giving them a checklist or telling them what to do. Yet, by studying the conditions under which plants and animals perform their best, the field of agriculture achieved tremendous growth in productivity throughout much of the 20th Century. Several of the statistical methods and models used in Systems Thinking also have their origins in agriculture. (By contrast, Taylor, Gilbreth, and Gantt were all mechanical engineers.)

The systems approach to human factors gained attention in the aircraft industry during World War II. During the war, half of all aircraft losses occurred in non-combat accidents. Accidents are often blamed on pilot error. Yet, by taking a systems approach and carefully studying the relationship between cockpit design and pilot behavior, the rate of pilot error could be reduced. Deming viewed all worker error as ultimately the responsibility of management for creating the conditions under which errors occur.

In 1985, Bill Smith of Motorola observed that products that required the least rework in development performed the best in the field. Together with Mike Harry, they developed the Six Sigma approach as a better alternative to "test and fix" for eliminating defects. Like Deming, their approach seeks to identify "intermediate" factors that contribute to defects.

Foundation

Systems Thinking is grounded in a family of theories that share certain characteristics. Systems are viewed as complex, open, non-deterministic, and adaptive. Complex systems are modeled at many levels and viewed as interrelating systems of systems. Unlike closed systems, open systems interrelate with their environment, and must be understood in context.

Causality in complex systems can be based on non-deterministic effects of contributing factors. The factors are discovered by observing bounded statistical patterns, rather than repeatable fixed patterns. A factor's weight, central tendency, and variance can be computed and tracked using statistical techniques like fractional factorial models and Shewhart control charts.

4. Problem

Properties like efficiency, quality, reliability, safety, security, and value are compromised by errors and weaknesses in every part, level and step in processes and systems.

Forces

- Traditional point solutions lead to plugging leaks and fighting fires. Checklists just get longer and plans become more numerous.
- Even with all the best practices being followed, there are still many opportunities to improve a process.
- Many of the factors that contribute to process behavior and outcomes must be observed in place. They cannot be discerned from general principles or read about in a book.
- Contributing factors are often subtle and difficult to identify. The common approach of fractional factorial models requires large amounts of data and many runs. Common uses of control charts assume system stability. Many aspects of development projects are not repeated or stable.
- Simple decisions can have consequences beyond just the intended result. Like other factors, identifying these consequences and bringing them to attention can be difficult.
- Management practices often assume a centralized model of control. Such models are easier to use and understand than distributed models of control. In distributed models there can be many sources of decisions. The results depend on many factors and interactions.
- The behavior of organizations and workers often deviates from that which is predicted by overly simplistic (reductionist) models.
- Workers are intelligent social beings with human needs and feelings. They resent having excessive constraints imposed upon them from outside and not being rewarded for their own ingenuity. A common response to perceived external locus of control is to become less active and less committed to the work being done. Yet, when properly empowered, workers can actively solve problems and improve processes.

5. Solution

Systems Theory values solutions that identify and fix root causes. A root cause analysis starts with a specific process event or outcome in need of change. It looks for factors around the event that may have contributed to it. In particular, factors of interest are those that changed at the same time, or slightly before the event in question. These related changes may, themselves, have their own contributing antecedent factors. Eventually one or more chains of contributing factors are traced back to an initial deviation or root cause. The root cause can then be fixed or improved. An Ishikawa or fishbone diagram is often used to draw a tree of contributing factors.

In the systems approach, a deep analysis of the factors contributing to a single metric, like quality or time, can reveal much about a process. But identifying the contribution of factors in random data can require a substantial amount of data and special techniques, like the Fractional Factorial experiments and the Taguchi method. In some cases, simulations may be used in place of running tests on the actual process. Looking at factors around an obvious change is easier than trying to account for small variations. It is important to keep the tests simple, and look for confirmation that a contribution exists (the null hypothesis), rather than trying to assign a numerical weight to that contribution. In some cases, a qualitative analysis is sufficient to identify what changed.

Systems Thinking looks for contributing factors beyond just the direct, internal factors assumed in closed deterministic and mechanistic models. The system in question exists within a broader system-of-systems. These systems may be considered as a hierarchy. Contributing factors may come from any level. The factors themselves may be part of a culture, chain or hierarchy.

Some factors are autonomous actors (agents) capable of adaptive (or erratic) behavior. An important avenue of improvement is to improve an actor's ability to adapt.

6. Discussion

Consider All Factors is the only pattern of the five that looks at external factors as being part of the system. Even Cybernetics, which is a branch of Systems Theory, views feedback in a closed system where external events are "sensed." While common applications of Systems Thinking are associated with the statistical control of variance, multi-level analysis of factors is also a major contribution. In situations where any error leads to compromise, like safety, quality, and security, multi-level analysis is a must.

The application of Systems Thinking to process control, as in Statistical Process Control and Quality Management, often involves some form of Shewhart control charts. The metric in a control chart is expected to stay within an acceptable range, allowing for normal "common cause" variance. Observing metrics outside of that range indicates that one or more factors have changed beyond acceptable limits. An excessive deviation is attributed to a "special cause" and warrants attention. The errant factor could be a machine part that is beginning to wear out or workers becoming lazy. But control charts are not just about drift in repeating processes. They can also identify recent change. Perhaps a new employee was not properly trained or members of the team have adopted a different practice.

Another application of Systems Thinking is multi-level analysis. In multi-level analysis, behavior in one level is influenced by conditions, constraints, and events (or lack thereof), in other levels. The Leveson and Weinberg examples in Known Uses (below) demonstrate this kind of analysis.

7. Implementation

An analysis can begin with a goal, like reduced defects or improved security. More commonly it begins with an event, like a particular defect (or set of defects) or a schedule slippage. The situation is then analyzed by observation and measurement. Observation requires walking around and talking to people involved in (and affected by) the process. The analysis uses both intermediate factors internal to the process and external factors in the environment to identify indicators and sources of change. Measurement includes not only the systems inputs and outputs, but also intermediate indicators of conditions, behavior, and effects throughout the process. Having more indicators contributes to gaining a better understanding of the behavior and relationships in the system. Statistical analysis is used to confirm the significance of possible contributing factors. The solution should be applied to the earliest or most basic factors in the causal chain.

Several tools are commonly used in the application of Systems Thinking to process analysis. Two of the most common are Shewhart Control Charts and Ishikawa Fishbone Diagrams. In non-deterministic systems that naturally exhibit noise, control charts provide early indicators of a deviation from the baseline levels of variance. An Ishikawa Fishbone Diagram is a qualitative way to identify and group potential contributing factors from all sources and levels. In an Ishikawa diagram, the issue or event is positioned at the head. Branches, shown as ribs off the spine, represent contributing factors from different sources, such as technology, workers, management, regulators, customers, competition, etc. Branches are, in turn, decorated with further lists of detail and sub-branches. As an example, contributing factors to a deep sea oil well disaster could include limitations of technology, lax regulators, and cost-cutting from management. The important observation is that no factor is viewed in isolation – it takes a combination.

8. Example resolved

Phil and his team analyzed the factors associated with persistent hot-spots in the code. They discovered that the code that was later involved in the most defects was often written after hours, during periods when the team was under time pressure from upper management to meet a deadline or a milestone. Based on this analysis, new policies were established with several options for dealing with "charettes". Being aware of the situation made it possible to better manage a major source of defects and reduce its impact on production code. The team felt more in control and upper management gained a new appreciation for developer well being.

9. Known uses

James Coplien's Organizational Patterns

James Coplien developed his organizational patterns from a systems view of the organization. He argues that organizational improvement is impossible to "master-plan", and argues instead for an approach of local adaptation and piecemeal growth. "Improving an organization requires systems thinking" (2007). Instead of focusing on practices and behavior, his patterns emphasize roles and relationships.

Nancy Leveson and systems safety

Nancy Leveson (2004) applies systems thinking in her work on systems safety and factors that contribute to failure. In her studies of failures at NASA she shows that technical and operator failures were part of a broader system of contributing factors at many levels. Technical errors were preceded by changes in management policy and other changes in institutional behavior. Had any one of those other factors been addressed, the outcome would have been different. The root cause was higher up. Her message is that addressing the problem of safety requires a complete culture of safety.

Gerald Weinberg and the soda machine.

Alistair Cockburn (2004) quotes an extract from an article by Gerald Weinberg, about the relationship between a soda machine and the work load of a help desk at a university computing center.

A manager received a complaint about the noise around a group of vending machines. After the manager moved the vending machines to a remote location, the work load at the help desk increased to where they couldn't keep up. Eventually the cause was determined. When the vending machines were near the work area, students congregated by the machines and often discussed their programs. In these discussions, they often learned the solutions to common problems. The students did not congregate by the machines in their new location. When they have problems, they were now more likely to seek advice from the help desk. The role of the vending machine placement was determined through a systems view.

Vending machine placement is unlikely to become part of a Plan, rise to the level of a Best Practice, be viewed as a solution to waste in Flow, or be found through better Feedback. It does however, share features in common with Coplien's (2005) Hallway Chatter, and The Watercooler patterns

Six Sigma

In Six Sigma training, the emphasis is on measuring intermediate factors in order to identify where in the process a deviation causes a decline in customer satisfaction. The example, used in the Motorola training for Six Sigma, is a coffee machine in a coffee shop. The manager finds that the customer satisfaction with the coffee is not always what it should be. After a standard statistical analysis, the deviation is traced to the fact that the machine was brewing water at a slightly different temperature than before. The manager finds a solution to achieve a more consistent water temperature. Unfortunately, standard Six Sigma training does not yet extend this kind of analysis to development processes.

Li et al. Tracking Projects in China

Li et al. (2007) describe "a project tracking process … to guide analysts to locate potential causes for different kinds of variations, and effectively correct them in real projects." They use Control, Pareto, and

Scatter charts to analyze “factors involved in project and specifying impacts between these factors”. When applied to a software development team in China, rated at CMMI maturity level 4, they identified a bottleneck caused by intermediate batching in the team’s test confirmation practice.

10. Consequences

Advantages

- The approach to process improvement involves more bottom-up participation, and less top-down command than standard best-practice initiatives.
- Sources of problems that are ignored by other approaches to problem improvement can be identified and resolved using a broader and more analytic systems approach. While standard forms of assessment focus on worker behavior, assessment in the systems approach includes many other factors, including factors in the environment, the process itself, and management decisions and behavior.
- The approach can be seen as more focused on finding solutions and less on blaming individuals when compared to process improvement based on compliance assessments, even if the solution eventually turns out to be the same as improving compliance.
- Solutions are likely to be adopted and persist because they directly respond to actual problems.

Disadvantages

- Tools that promise the kinds of metrics needed to measure internal, external, and intermediate factors can create a higher reporting burden, and/or generate metrics that are subjective and hard to analyze.
- It sometimes takes an outside consultant to see details in the culture and environment worthy of investigation. Such details may escape the notice of people who see it every day, and thus take it for granted.

Incorporate Feedback and Learning

0. Name

Incorporate Feedback and Learning

Also known as

Adaptation, Value (Koskela & Howell, 2002)

1. Intent

Projects require correction. Some of the requirements may not be known or understood at the start. Requirements may change over time. Errors may occur during execution. All of these situations require corrections or adaptations. But first, they must be detected. Feedback focuses on those aspects of the process that help, hinder, speed up, or delay dynamic adaptation to change.

2. Example

Jean's current process works well if the requirements remain fixed. But changes cause instability and add significant costs. The last time they introduced an important change, test failures shot way up, and not just around the immediate change. Even with intensive rework, it took months before failure rates were back to expected levels. Jean would like to incorporate more high-value changes without sacrificing quality or control.

3. Context

Incorporate Feedback and Learning views all parts of a process in terms of actively delivering value to the customer, where value is a moving target. The question for process improvement is how well the process can track and respond to the business goal of delivering the right product to the right customer at the right time. While understanding customer value plays a major role in Planning, Flow, and Systems Thinking, it is the view of value as something that is dynamic and requiring constant adjustment that makes the feedback view unique. The problem is easily understood in terms of the military problem of delivering the right ammunition to the right place at the right time – all three matter!

Self-correcting systems require feedback. Better sensing of differences between the actual state and the desired goal leads to better adaptation. Shorter feedback cycles and less resistance to change lead to smoother adjustments, less disruption, and higher efficiency.

History

In the 1920's and 30's, Bell Labs developed amplifier technologies that could better follow a desired fast-changing signal by using feedback in the design. Without feedback, equivalent fidelity is much harder to achieve, requiring complex designs and layers of compensation. During World War II, work on the automatic control and direction of anti-aircraft guns lead to an even greater understanding of the central role of agility and feedback in complex system design. The experience and ideas learned in these earlier efforts formed the basis of a branch of Systems Theory called Cybernetics.

In the 1940's and beyond Edwards Deming preached the Plan, Do, Check, Act cycle of continuous improvement, which he learned from Walter Shewhart. Later, in the 1970's and 80's Don Schön, and others, wrote about learning organizations and the importance of reflection in action. Shewhart and Schön were both inspired by John Dewey's ideas on the interdependence of learning and action.

Foundation

Feedback, and its role in self-regulating control systems, is the central concept in Cybernetics. Systems are modeled, and improved, through the design of sensing and feedback loops. But feedback and self-adaptation can also lead to instability. A deeper understanding of feedback relies on control systems and chaos theory, both of which are concerned with system stability, or the lack thereof, and the affects of various kinds of feedback.

4. Problem

The ability to change is a problem when the process should respond to corrective input, but does not. Response to change has two parts: detecting the need for change, and executing a response. Failure of either part perpetuates the problem.

Forces

- Today, changing requirements is the norm rather than the exception. Many customers are themselves responding to change. Delivering value requires processes that can respond to changing needs and expectations.
- In today's "Internet age," the rate of innovation is very high. New technologies become obsolete often within a matter of months. Windows of opportunity are short and hard to predict. Even when a new technology is known, its impact on customers and the market is hard to foresee.
- In competitive markets, companies can survive by targeting products at a niche. But addressing a new niche requires products to be customized and adapted to a unique set of requirements. Companies that cannot adapt lose business to competitors who make product adjustments that deliver higher value to specific customers.
- In security and defense, adversaries are as smart as you, if not smarter. Maintaining the advantage requires innovation. Smart adversaries quickly adapt to each new situation or innovation as it appears.
- Not all change is external to the process. When a defect is introduced, an artifact that was supposed to have certain properties suddenly takes on different properties. These new properties may affect on other parts of the process. Even if the defect is eventually fixed, negative effects can accrue during the period that it is present.
- Tasks that have been started, but not completed, are impediments to change. Plans require a known starting point, and a known ending point. Until the current work is completed, or abandoned, the project has an unknown state.
- Changing circumstances may render a project no longer profitable. In such cases, the appropriate response may be project termination. Any delay in detecting the situation, and responding, costs money.

5. Solution

Adaptation occurs in a cycle. The process moves, makes some progress, perhaps causes a defect. Other changes occur and new information becomes available. The difference between the current state and the goal is reassessed. Adjustment is made. The cycle repeats. Feedback focuses on how and when the assessment and adjustment steps are performed.

The common solution to improving feedback is to create shorter, coordinated cycles, called iterations. At the end of an iteration, current work is completed, assessments are performed, and changes can be made.

Feedback also concerns the test and repair cycle. Since work is not really completed until all fixes are done, shortening the repair cycle shortens all other cycles.

6. Discussion

All project management practices employ some form of monitoring and control. The Feedback view focuses on how feedback is being applied, and how the process design enables or impedes adjustment.

The basic goal in Use Feedback is to detect changes as soon as possible and respond as quickly as possible. Processes and practices are judged by their ability to respond to change, and by the timeliness and appropriateness of their response. Change can come from many sources. It can be a deviation from plan, as in a defect. It can be new information that wasn't known before. It can be changes in the goal or a requirement. In any case, the process should adapt. Ideally, it should adapt as the change occurs, or very soon afterwards.

Robert Wysocki (2009) makes the following two points about cost and feedback. If the project starts with a complete plan, then each adjustment or change invalidates the remainder of the plan. After each change, the plan must be redone from the current point forward. If changes happen often, the repeated overhead of redoing the remainder of the plan can be substantial. With frequent change, it is more economical to defer planning until closer to when it is needed. Changes may also cause a once viable project to become uneconomical. With short and frequent feedback, the decision to kill the project can be made much earlier.

In *Effective Project Management*, Wysocki (2009) describes the feedback in different Project Management Lifecycle Models (PMLC). All of Wysocki's PMLC Models go through stages of scoping, planning, launching, monitoring and controlling, and closing – the five process groups in Version 3 of the PMBOK. (The PMBOK, itself, mentions iteration only once, as a possible response to risk, with no discussion.) The traditional linear lifecycle model ignores feedback, except between adjacent stages. In the Incremental model, the stages from launch to close repeat in cycles, but scoping and planning are not revisited. In the Iterative and Adaptive agile models, the iterations also repeat the planning stage, to allow for new or changed requirements. Finally, in Wysocki's Extreme model, even scoping is repeated allowing later inputs and experience to revise the overall goal. "The bottom line is this: what will deliver business value is a moving target. [Traditional Project Management lifecycle] models aren't equipped to assure the delivery of business value" (p.321). When the project is defined in terms of the number and duration of its cycles, it becomes schedule driven – the scope is dependent on the schedule rather than the other way around.

A major impediment to change is work-in-process. While work is underway, it is not in a known state. It is hard to engineer a change on work which state is unknown. Moreover, if the commitment to change is made, for tasks in process, both their begin and end points will be invalid. The result is instability. The alternative is to minimize work-in-process and synchronize tasks to start or end at fixed times – i.e. between iterations. Time boxing tasks achieves this goal.

Conventional process wisdom, inherited from mass production, holds that a process is optimally efficient when overhead is minimized and all resources are busy. Under this assumption, optimal efficiency can be achieved by batching. The overhead occurs less often, and upstream queues assure a steady supply of work for every resource. But batching and queues increase the lag time in feedback loops, and long feedback loops increase cost

In our own studies on traditional projects, with batched testing and a queue waiting for rework, more than 40% of new tasks were built on top of code that already had defects. We also found a positive and significant relationship between lag time and subsequent effort. Most significantly, since defects often occur in the most error-prone modules, repair work is itself likely to cause defects. There is a large multiplier for the lag time in repair cycles. (VanHilst & Huang, 2009).

7. Implementation

There are three basic types of feedback loops in processes: build-validate-revise, make-verify-fix, and act-reflect-learn. The build-validate-revise cycle can be improved with continuous builds, short iterations, incremental development, and pilot or spike projects. Management-by-walking-around and customer-on-site also improve validation feedback. The make-verify-fix cycle can be improved with practices like source inspection, continuous testing, unit testing, test-first, and test driven development. Practices that improve the act-reflect-learn cycle include lessons-learned and practices associated with learning organizations or organizational learning. At an individual level, the learning cycle can be shortened by replacing annual reviews with mentors and continuous coaching.

8. Example resolved

Jean's group adopted an iterative, time-boxed approach to scheduling, with feedback and a reassessment of the requirements at the beginning of each iteration. Changes are now introduced in a more coordinated way with less disruptive consequences. When problems do occur, they are detected earlier and resolved within a single iteration. Defects are no longer allowed to persist, and are also more under control. With added confidence in their ability to handle change, Jean’s team interacts more with customers and seeks out high value ideas to incorporate in each new iteration.

9. Known uses

Agile Methods

XP, Scrum, and other Agile methods apply many practices to improve feedback and adaptation. These practices include short iterations, incremental development, test-first, and continuous build. Pair programming is a form of source inspection, where the second programmer is watching as the code gets written, and catches simple mistakes as they happen. Lessons learned are also performed on each iteration, for short act-reflect-learn cycles of improving the process.

Design Structure Matrix (DSM)

The Design Structure Matrix is a tool for handling iterative sub-cycles in complex development processes. The goal is to make the feedback loops as tight, or short, as possible. It maps information flows, rather than work flows, and identifies the information flows among tasks in a project. Feedback loops, such as those involved in complex learning, appear as entries in the upper triangle of the matrix. Using the matrix, co-dependent tasks are grouped closer together by changing the order. It is also used to help localize the impact of change. DSM was first proposed by Steward (1981), applied at NASA by Rogers (1997), and advocated in the Harvard Business Review by Eppinger (2001).

Parallel Development at Toyota

Cutting edge technology creates substantial benefits and substantial risk. If a product has three new innovations, each with an 80% chance of success, the combined likelihood of success is only 50%. Toyota manages the tradeoff with a time-boxed innovation feedback cycle. During this cycle in the process, two alternative technologies are developed in parallel – the high risk innovative solution and a low risk traditional solution. At the end of the time box, the innovation is assessed. If the new technology proves itself ready, it continues into the product. If it is not ready, the more traditional solution is used. The innovation is not wasted. It is simply deferred for a later product.

Organizational Learning

To quote from Wikipedia, "Organizational learning is an area of knowledge within organizational theory that studies models and theories about the way an organization learns and adapts." It was first proposed by Argyris and Schön (1976), and introduced the concept of "double-loop learning." Again, quoting from Wikipedia, "In single-loop learning, individuals, groups, or organizations modify their actions according to the difference between expected and obtained outcomes. In double-loop learning, the entities (individuals, groups or organization) question the values, assumptions and policies that led to the actions in the first place; if they are able to view and modify those, then second-order or double-loop learning has taken place. Double loop learning is the learning about single-loop learning."
See http://en.wikipedia.org/wiki/Organizational_learning

10. Consequences

Advantages

- Using explicit feedback loops helps organizations structure their processes in ways that support adaptation and change as an integral part of the process, rather than as an external event or problem.
- With adaptive processes, companies can start projects before all the details of the requirements are known, and respond when new opportunities arise after a project has already started.
- Iterations in the process allow customers' special needs, or adversaries' recent advances, to be considered and addressed later in the process.
- In cyclic processes, maintenance can be viewed as simply later iterations of the same project.

- Incremental planning minimizes the lost overhead of creating plans for late stages of the process that become obsolete and inoperative due to midcourse changes and corrections.
- With short iterations, lessons learned during process execution to be applied to improving later iterations within the current process.
- If a project becomes unprofitable, the situation will be detected during the assessment for the next iteration, and allow the project to be terminated or re-directed long before delivery of the final product.

Disadvantages

- Allowing adaptation makes it difficult to predict outcomes, and to make and meet hard commitments at the beginning of a project.
- In adaptive processes, management and execution are more closely linked, requiring managers to be more knowledgeable and aware of development issues. It may also require customers to be more closely involved in the development process.

ATAMO

0. Name

ATAMO (And Then A Miracle Occurs)

Also known as

Where's the beef?, Hand waving.

1. Intent

Often there is an intention to include a particular step or activity as part of a process, without a clear idea of how the step or activity should be performed, or perhaps even could be performed. By invoking ATAMO, the description of the process can appear to be complete and even formal, while leaving crucial implementation details to be figured out by others at the time of implementation.

2. Example

Fred has been told by his management to undertake a process improvement initiative. He has undergone training in a well known process improvement methodology where he learned methods of planning, organization, and statistical analysis. He has been given 9 month and a team of 3 to achieve the specified level of improvement. But he has no specific problem and doesn't know what to do. The analysis methods he has learned were presented in a way that doesn't seem to fit the current situation.

3. Context

ATAMO solutions occur in situations where the goal is more ambitious, or the solution has to appear more substantive and formal, than can be supported by the capabilities currently at hand.

History

"Then a miracle occurs" is a reference to a famous Sidney Harris cartoon. The cartoon first appeared in American Scientist, in 1977. It depicts two men wearing ties standing in front of a blackboard. The blackboard has a series of complicated looking equations. In the middle there's a gap where, instead of equations, the four words appear in block letters. The one scientist is pointing at the words. The caption reads, "I think you should be more explicit here in step two."

"Where's the beef?" is the punch line from a commercial for Wendy's hamburgers shown in the US and Canada in 1984. In the commercial, three old women order hamburgers and get sandwiches with massive, impressive buns, but only a tiny piece of meat. The expression is used to describe a solution that leaves you hungry.

4. Problem

Forces

- Many organizations are pressured to undertake process improvement initiatives to satisfy strategic directives from above.
- Managers with business degrees, and little understanding of the actual work in the process, are often tasked with leading initiatives for process improvement. They need a method they can understand at the level with which they are comfortable – typically organizational or operational.
- There is a market for solutions that can be applied in all cases. But real solutions are often situated – specific to the situation. Finding such solutions requires detailed analysis and insight. To sell a solution as universal, one must focus on those parts of the solution that are general.
- Decision makers who are accustomed to decision models using numbers are uncomfortable with making an investment if the resulting value cannot be quantified up front. ATAMO allows for models that are [almost] complete, and with numbers already assigned.

5. Solution

There is none – it's missing.

6. Discussion

ATAMO has been used to point at gaps in many kinds of processes. In some cases, bridging the gap between two steps requires an intellectual leap, such as the gap between analysis and design. In other cases it requires a solution involving expertise from a separate domain, other than the one which defines the process. In the simplest case, details are left out because they are troublesome and cannot be provided in the same degree of thoroughness as the surrounding steps.

Process improvement is rarely the kind of project in which the problems and solutions are completely known at the outset. ATAMO situations arise in process improvement when quantified objectives are set prior to knowing a problem or a solution. If the method of achieving measurable improvement is obvious, then why hasn't it already been applied? If the method of achieving measurable improvement is not obvious, then how do we know how much improvement can be achieved? Goals should create the motivation and set the direction. But for realistic improvement, measureable objectives must be formulated in conjunction with, and correspond to, the design of the initiative.

ATAMO is sometimes applied by variations on the original Six Sigma, for non-manufacturing situations. The original Six Sigma training involves statistical methods to identify factors and find the root cause, and is presented in a form that is rather specific to manufacturing. In the new variations, the DMAIC (design, measure, analyze, improve, and control) form of Six Sigma is duplicated without a specific, statistically based approach to find the problem and define a solution.

Other problem solving frameworks, when applied without a solution methodology, can also result in ATAMO. The SEI IDEAL (initiating, diagnosing, establishing, acting and learning) (see below), and ITIL's and ITIL's Continual Service Improvement 7 Step Improvement process (define should, define can, gather, process, analyze, present, implement) are two other examples.

7. Implementation:

The typical ATAMO implementation is characterized by its emphasis on setting goals and collecting data, with little to say about how to analyze the data or how to actually change the process. The plan for an ATAMO solution has the following steps.

1. Create an initiative and get commitments.
2. Measure the "as is" process and define a baseline.
3. Define the "to be" goals and set quantifiable benchmark metrics.
4. Plan and implement the improvement.
5. Assess compliance and long-term commitment to the solution.

8. Example not resolved

Fred has found baseline measurements on the process that are easy to make, but have no clear relationship with the goals of the business. He then implemented new process practices to show a clear difference in the measurements he defined. Responsibility for carrying out the plan and achieving results was delegated to lower level staff. When upper managers ask about the lack of business impact, Fred gives vague explanations of the complexity of the situation and external factors beyond their control, and blames workers for their lack of commitment to the solution.

9. Known uses

Business Process Reengineering

Several authors have criticized presentations of Business Process Reengineering for applying ATAMO. In Hammer and Champy's original book, Business Process Reengineering [1993], only 14 of its 250 pages present a prescriptive method to redesign a process, and 11 of those describe a case. "In the literature on BPR, examples of successful BPR implementations are given. Unfortunately, the literature restricts itself to descriptions of the 'situation before' and the 'situation after', giving very little information on the redesign process itself" (Garrits, 1994). "How to get from the as-is to the to-be isn't explained, so we conclude that during the break, the famous ATAMO procedure is invoked – And Then, A Miracle Occurs" (Sharp and McDermott, 2001).

ITIL

The Information Technology Infrastructure Library (ITIL) is sometimes criticized for describing how to deploy, govern, and improve information technology services, but being silent on how services get developed. Doerscher (2008) describes the criticism as follows:

> "The ITIL Lifecycle goes from (Service) Strategy to Design to Transition to Operation and Continual Improvement. Conspicuously absent is Development. It reminds me of the infamous Sydney Harris cartoon …. So, while design considerations, deployment and release management are covered, ITIL is mute on the topic of actually creating or changing the service."

IDEAL

Use of the SEI's IDEAL model can result in an ATAMO strategy. IDEAL is a framework for SPI involving 5 phases. In the Initiating Phase, commitments are made. In the Diagnosing Phase, measurements are made and baselines established. In the Establishing Phase, measurable goals are set, a strategic plan is made, and a Technical Working Group is formed. In the fourth phase, called Acting, the TWG finds or develops solutions, a support provider may be hired, and the solution is packaged for rollout. The Acting Phase has two models, a problem solving model for less mature organizations, and a continuous improvement model for more mature organizations. The "package" is then handed off from the TWG, which ceases to exist, to the Software Engineering Process Group, who is then responsible for making sure that commitment remains high. The Leveraging Phase reviews the results.

Ouellette (2007) has criticized this model for assuming design from the outside and lack of focus on deployment.

> "A big process book is written. Deployment consists of announcing the existence of the book. The improvement team declares victory. (Here is where the miracle is planned to happen.) The process is ignored or significant resistance occurs."

10. Symptoms and Consequences

- A common symptom of an ATAMO initiative is the creation of measurable goals independent of any conception of the actual solution.
- Metrics in ATAMO initiatives often measure the process rather than value to the business. They take the "process" in process improvement too literally. A goal might be to produce more documents or perform more inspections. 10% more process is easier to achieve than either a 10% improvement in customer satisfaction or a 10% reduction in time to market (see above symptom).

Buy a Silver Bullet

0. Name

Buy a Silver Bullet

Also known as

Quick fix, Snake oil

1. Intent

Everybody likes a quick and easy fix. When faced with a difficult and potentially complicated problem, a solution that can be described with a single name and implemented with a single decision will appeal to anyone who doesn't want to get their hands dirty. The easiest way to solve an organizational problem is to buy a remedy and declare the problem solved.

2. Example

Fred's team has a problem in their process that needs to be addressed. Requirements are handled in an ad-hoc fashion, using office tools and stylized notations. Management knows that the situation is not ideal and asked Fred's manager to find a solution. Fred's manager is not intimately familiar with how the requirements are used, nor the details of their contents. But he knows that his performance depends on his finding an acceptable solution, and would prefer not to share credit for the solution with Fred or other more knowledgeable people on his team.

3. Context

Management is looking for a solution that doesn't require them to get too involved in the details of what the workers actually do. High level management has identified a problem and given the responsibility for addressing it to middle level managers. They call in vendors and consultants to make presentations on how they would solve the problem.

In a hierarchical organization, mid-level managers sit between high level managers who are responsible for corporate strategy, and the line managers who are responsible for the day-to-day work. Mid-level managers are responsible for collecting information from inside and outside the organization and advising high level management. After collecting several alternatives, they may also ask line-managers which on.

History

In folklore, a silver bullet is supposed to be the only kind of bullet that can kill a monster or demon. Fred Brooks wrote a famous paper called, "No Silver Bullet - Essence and Accidents of Software Engineering." Since then, the term Silver Bullet has been associated with the idea of a simple solution that can solve problems and revolutionize software development.

Snake oil is a reference to elixirs that will "cure all that ails you." In the 19th Century, before the Food and Drug Administration was formed to clamp down on such practices, salesmen traveled the country selling bottles of potions from the back of a wagon, with exaggerated claims. According to Wikipedia, the origin of snake oil is a Chinese remedy with Omega-3 used to treat arthritis. But those sold in America made bigger claims and had less of the original ingredients.

4. Problem

Real problems are messy. If the problem persists for a long time, there is probably a good reason why the problem hasn't been solved by an easy fix.

Forces

- Mid-level managers have professional degrees with training in business analysis and decision making. But they often lack the direct experience in development and engineering needed to gain a thorough understanding of the problem and its context. In place of hands on experience, they read trade journals and go to trade shows and expos.

- CASE (computer aided software engineering) tool vendors advertise their tools as the solution to a variety of problems. They know that a decision that requires investments of that size will be made by middle level managers and they target their publicity, literature, and presentations accordingly.
- Solving pernicious problems can require deep understanding of both the problem and the organization. It may also require reflection on the nature of the problem and how it interacts with the organization. Seriously investigating the problem may raise controversial questions about how things have always been done, and why.
- Mid-level managers need to get credit for solving problems and bringing value to the organization. Their value is not directly tied the product or service. If they involve line-level people in making decisions, their personal contribution becomes less clear.
- In an organization where the management is not very "hands-on", the appearance of a solution can be as good, if not better, than a real solution.

5. Solution

A magical solution requiring little work and no localized adaptations to details of the solution.

6. Discussion

OOPSLA 2007 held a discussion panel to commemorate the 20th anniversary of Fred Brooks' article, No Silver Bullets. A significant part of the discussion was spent talking about vendors who want to sell you a silver bullet, typically in the form of a tool or a two day class.

Process improvement is not easy to achieve. It requires understanding the nature of the problem, finding solutions that fit the situation, dealing with the local culture, and institutionalizing the learning that took place. Often it requires managers to become intimately familiar with the work done at lower levels. In the analogy of manual labor, it requires rolling up one's sleeves and getting one's hands dirty. The fathers of modern process improvement, Frederick Taylor and Frank Gilbreth, both studied how the actual workers did their jobs.

There is a certain appeal in solutions to problems that don't require serious work. They are like the get-rich-quick schemes advertised in spam and on late-night television. Managers, who view their role as making decisions, are especially drawn to solutions that can be "implemented" by making a single decision, and letting others do the work. What if the problem could be solved by buying a tool or ordering others to apply a well known practice, especially if it had worked in the past (see All Problems Are Nails)?

Tools have a valuable role in processes and process improvement. In fact tools are central to a number of best practices. But tools are just ways of automating parts of a solution. A tool is not itself the solution.

"Best practices" can become a buzzword to be used in place of describing an actual solution. We find this criticism in James Bach's No Best Practices (2005): "Best practice blather becomes a substitute for the more difficult, less glamorous, but ultimately more powerful idea of learning how to do your job." It is so much easier to declare the solution and then blame others for not applying it. Too many defects? The workers aren't following best practices. Behind schedule? The workers aren't following best practices. Declining business? The workers aren't following best practices.

The Buy a Silver Bullet approach to process improvement is decidedly top down. Management decides that there is a problem and management selects a solution. This approach is in sharp contrast to the Quality Circle approach to process improvement. In Quality Circles, a group of worker volunteers meet regularly to discuss problems and solutions. Quality Circle members receive special training and often have added incentives to find and implement improvements.

7. Implementation

Select a method or tool, claim that it will solve the problem, and declare the problem solved. If the problem is not solved, blame the team for not applying the method or using the tool correctly.

8. Example resolved

Fred's manager invites outside vendors to make presentations about the solutions they have to offer. Fred is invited, but no one else from his team. The vendors make impressive presentations about the features of their solutions, with testimonials from other customers. Nobody asks questions about how the current system works. When Fred mentions some issues, they are dismissed as typical of kinds of problems the new system will replace. A system is ordered at great cost. Fred's team undergoes training. When they find that the new system is not well suited to their needs, they are criticized for a lack of commitment. When productivity does not improve, and in fact gets worse, the blame is directed at Fred's team.

9. Known uses

Every developer who has worked in a large company long enough experiences these kinds of issues.
`

Service Desk

On IT Skeptic, Rob England (2009) describes the "buy a tool" scenario, ties it to ATAMO, and explains what should have happened.

> "We see this hope when management buy a new Service Desk product in the hope that it will introduce ITIL into an organisation. If the whole exercise is approached with most of the focus on the community and the activity, not on implementing the software, then this is possible. (But it is seldom the case: the tool gets installed and the rest is supposed to happen by magic osmosis. TAMO: then a miracle occurs.)
>
> "A new tool does give a reason for change in process, but it is an unhealthy reason: 'we have to change to this way now because of that new software' instead of 'let's do it this way because it makes more sense and gives better results and fits how we want to do things round here'".

Recommendations in Action Analytics

The first recommendation in Norris et al.'s Action Analytics article (2008) on process improvement for the education of Information Technology is: "Focus on processes, solutions, and behaviors, not just the acquisition of tools.

Jumping on the CASE bandwagon at a NASA subcontractor

While working on a project for NASA in 1984, we were told by a manager, whom we had never met, that for the next project we would have to use a particular CASE tool to produce a structure chart for every code artifact. The tool came with lots of training material. During development we were spending so much time on the tool that we requested and were granted a secretary to enter the drawings from our hand sketches. By the end of the project we had produced a large notebook of drawings. Over the next four years, including several maintenance upgrades, the notebook sat on top of a cabinet and was never touched. We found it easier to just read the code.

Alarm clocks

(This story of a case that actually worked was told by Mary Potter in her class at MIT. The source is not known.) A young researcher had been assigned to study why recently released prisoners who had been given employment had trouble holding their jobs. The men frequently arrived late and had poor work habits. After the initial interviews, the researcher discovered that none of the men in the study had alarm clocks. After buying them all an alarm clock, the situation changed dramatically. The researcher had to find another study for her dissertation.

10. Symptoms and Consequences

Symptoms

- The responsibility for process improvement is given to managers with only a shallow understanding of the process.
- The decision does not involve the people who actually perform the process to be improved.
- A detailed analysis of actual problems, and how they manifest themselves in the process, has not been performed.
- The solution involves the purchase of an outside product or service, and little else.
- Deployment amounts to reading a book or manual, or getting training on the tool.

Consequences

- A considerable amount of money can be spent with little or no improvement to be shown, or even result in making things worse.
- Practices and processes that have evolved in the organization to solve real problems are disrupted.
- Workers become cynical about any tools or practices that management supports.

All Problems Are Nails

0. Name

All Problems Are Nails

Also known as

Death Spiral (Ward 2007), Looking for the keys under the lamppost,

1. Intent

When we have a solution in which we are heavily invested, and that has worked for us in the past, we want to continue using it for every new situation that arises. There is no new learning curve, the solution has already been justified, and, if there is a cost, it's already been paid for.

2. Example

Fred's team has been through several rounds of process improvement initiatives. Each time, the process improvement team looks for gaps between the as-is process, and the industry's best practices. In the early rounds, they found significant differences. By instituting new best practices, they made real improvements. By the last couple rounds, they were already doing most of the identifiable best practices. This year, there is pressure to lower costs and improve cycle times, so a new initiative has begun. Fred is interested in trying a different approach that is specific to the current problem, but that would require convincing his managers to make a change.

3. Context

The team has had successful process improvement initiatives in the past. But now that they have solved the common and obvious problems, the remaining unsolved problems are different. Never-the-less they choose to run the same process improvement initiative yet another time.

History

"All problems are nails" is a shortened version of the statement "When all you have is a hammer, everything looks like a nail." It is sometimes attributed to Abraham Maslow, who used it in discussions of problem centering vs. method centering. The discussion came up when he described the resistance he met, from psychologists trained in studying animals and the mentally ill, when he started studying people who were healthy.

> "Many scientists disdain what they cannot cope with, what they cannot do well. … I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail. … [Many psychologists] choose to work as best they can with important problems (problem-centering) rather than restricting themselves to doing only that which they can do elegantly with the techniques already available (method centering). If you define science as that which it is able to do, then that which it is not able to do becomes 'nonscience,' i.e. unscientific." (1966, p.15)

Looking for keys under the lamppost refers to a well known joke about a drunk who is looking for his keys under a lamppost. A passerby asks him what he is doing. The conversation proceeds as follows:

"I lost my keys over there."

"Then why are you looking for them all the way over here?"

"Because the light is so much better."

"Death Spiral" appears in a book by Allen C. Ward (2007), published after his death. In the death spiral the team is under pressure and starts cutting corners. In response, management applies the usual solutions for process improvement, creating more work, which puts the team under more pressure. The team makes more mistakes, leading management to apply more of the same solution. The downward spiral continues until the organization is totally crippled and management decides that product development is no longer a core competence.

4. Problem

Forces

- The organization is heavily invested in a program of certification and process improvement. Teams and training are in place for performing that kind of initiative. Changing strategies now might raise questions about the wisdom of the earlier investments.
- Making changes to a process is already a confusing and stressful undertaking. Making changes to the process by which we go about making changes would add to the confusion and stress.

5. Solution

The same solution we always use.

6. Discussion

In Peter Senge's book on learning organizations, The Fifth Discipline (1990), the first of the 11 "laws", is that today's problems come from yesterday's solutions. The intuition is that new problems are different. They are either the problems for which yesterday's solutions didn't work, or they are created by yesterday's solutions. Either way, yesterday's solution is not the solution for today.

The idea that the application of a solution will cause or reveal new solutions was also a part of Herbert Simon's concept of "bounded rationality." Real life solutions are rarely optimal and often have "unintended consequences." Problem solving is thus an iterative progression of problems and solutions where, in each successive iteration, the situation improves and the problems are different.

All Problems Are Nails applies a practice beyond the point where it solves problems, to where it becomes just an excessive and burdensome constraint. In terms of the analogy, when all of the nails have been hammered down, the things left to hammer are thumbs.

When we mention this pattern to developers, they often say, "Best practices!" In this context, Systems Thinking is often seen as an antidote. Why don't we stop constraining our behavior and start looking for root causes? But when Systems Thinking is the dominant approach, the roles can be reversed. Why don't we stop trying to reduce the variance, and start looking for new practices? The idea that there is a reasonable limit beyond which it doesn't make sense to reduce the variance is embodied in the name of Six Sigma – it's that limit.

7. Implementation

There is no new implementation. Just do what we always do.

8. Example resolved

Fred's team applied another round of Best Practices based process improvement. This time, since all of the identified best practices were already in place, the focus was on the level of conformance to details of the best practices. Deviations from the prescribed standards, in form and duration, were measured. Punishments and rewards were put in place to discourage and reduce future deviation.

9. Known uses

The Fifth Discipline

Peter Senge in his book on learning organizations, The Fifth Discipline, warns that when things go awry, the most common response is to think of something that had worked in the past. When that strategy doesn't work as expected, the common response is to do the same thing again, but more aggressively. Senge's solution and the thesis of his book, is to apply the kind of feedback found in Argyris and Schön's Double Loop Learning (1976) or Schön's Reflective Practitioner (1983), and to view the situation through Systems Thinking. Senge lists 11 "laws" for learning organizations. The first law states: "Today's problems come from yesterday's 'solutions.'"

The SPI Manifesto

The SPI Manifesto (Pries-Heje & Johansen, 2010) emphasizes that "Change is inherently linked with change; NOT continuing as we do today." The manifesto discusses the difficulty and importance of introducing new improvement practices that go against the existing culture, as part of an SPI initiative.

CMMI Guidance for Level 5

The standards for CMMI Level 5 emphasize continuous improvement based on measurement and analysis. Level 5 poses new challenges that differ from those of lower levels. Different approaches can be taken. In a presentation from Northrup Grumman, Hefner (2007) compares weakness in the CMMI's default guidance of continuing institutionalization of Best Practices with an alternative Lean Six Sigma approach of statistical process control and causal analysis.

Allen C. Ward's case against institutionalizing chaos management.

As mentioned in the Context section, Allen C. Ward (2007) describes this pattern as a death spiral. In a crisis situation, things no longer go according to plan and developers find themselves fighting fires. Managers "try to gain control by reorganizing, imposing arbitrary rules, demanding more reports, more tasks, …. People have to spend more and more of their time doing things that look good." The solution makes the problem worse. Ward goes on to set the groundwork for a focus on Flow, particularly knowledge flow, by arguing that solutions that work in chaotic situations should not be institutionalized since chaos should not be the norm.

10. Symptoms and Consequences

Symptoms

- Using the same method and models of process improvement repeatedly, often year after year, even after the types of problems for which the method is most appropriate have been addressed.

Consequences

- Many of the problems that remain after earlier improvement efforts get worse rather than better.
- Team members lose faith in process improvement initiatives and view them as a nuisance rather an opportunity.

Solutions Must Be General

0. Name

Solutions Must Be General.

Also known as

Technical Rationality (Schön, 1983), Scientism, Physics Envy

1. Intent

When buying a remedy, we want assurances that the remedy will actually work. Anybody can claim that they have a solution. In order to avoid making a mistake and investing in a solution that doesn't work, we demand evidence. The evidence expected is to demonstrate that the solution has been applied in many situations, and always worked.

2. Example

Fred's organization is committed to process improvement. They have had several process improvement initiatives in the past, and are highly rated. Management is confident in their processes. But somehow, their competitors seem to do better, beating them to market and with products of comparable quality. Several team members have suggested some solutions. But Fred will have a hard time convincing upper management to back the improvements if he can't show evidence that the suggested changes have worked in other organizations and quantify the amount of benefit experienced by those other organizations.

3. Context

Don Schön (1983) encountered this pattern in the resistance to his efforts to teach reflection and multi-perspective analysis. He labeled it Technical Rationality. In the model of Technical Rationality, scientists create generalized knowledge though the empirical observation of repeating patterns. This knowledge is validated and codified as universal laws. The laws are taught or disseminated to practitioners who then apply them to solve problems in the field. The role of the practitioner is to identify the problem, and select its matching solution (see the *Context* section of Copy What Works).

History

The bias towards found universal principles, and against situated analysis, can be traced to early Positivism and its project of applying "science" to all human endeavors. In Comte's own words,

> "It will offer a general system of education for the adoption of all civilized nations, and by this means will supply in every department of public and private life fixed principles of judgment and of conduct." (1907, p.4)

Comte had developed Positivism in the 1830's as an antidote to the revolutions and chaos that had engulfed much of Europe and the Americas in that period. The popular philosophy of the day, dialectics, viewed history as a never ending process of change. Comte considered dialectical reasoning a threat to order, and termed it "Negativsim". It was specifically to be rejected.

In dialectical reasoning, any initial thesis, or order, will have a flaw by virtue of being abstract. Through trial, error, and experience, this flaw will create a negative tension, or antithesis. To become concrete, the initial abstraction must be refined or synthesized to resolve the tension. But time does not sit still. The synthesis itself becomes a thesis, or abstraction, and the process repeats. While today, we know this sequence as Thesis-Antithesis-Synthesis, Hegel referred to it as Abstract-Negative-Concrete. A similar dynamic can be found in Herbert Simon's notion of "satisficing."

Comte saw Positivism as the search for invariant laws governing the social and natural worlds. He believed that, in this quest, Sociology would eventually rival and even surpass physics in its understanding of the way things actually are in the world. He saw Positivism as the ultimate stage of human progression from primitive theological modes of belief, to a metaphysical stage, dominated by abstract modes of thought, to a Positivist stage, based on universal laws of nature.

With the backing of Friedrich Wilhelm IV in the mid 19th Century, Positivism replaced Dialectics in German universities (Marcuse, 1941, p.326). American universities, which did not have graduate programs of their own, followed the German lead. British scholars remained skeptical. John Stuart Mill, for example, observed:

> “M. Comte warns thinkers against too severe a scrutiny of the exact truth of scientific laws, and stamps with "severe reprobation" those who break down "by too minute an investigation" generalizations already made, without being able to substitute others” (1865, p.37).

Positivism has never been taken seriously by the hard sciences. But its effect on social science, and the professions, has been profound and long lasting.

Schön attributed the shortcomings of Technical Rationality to Positivism’s simplistic interpretation of science. To early Positivists, there was one truth that needed only to be discovered. Once discovered, empirical truth was universal – true for all times and places. Scientific progress had a single path leading ever closer to perfect knowledge. They did not consider, or allow, alternative “truths” or interpretations. The standard for validation was limited to direct repeated observation. Ernst Mach, for example, who was both a physicist and a Positivist, refused to believe in atoms, because they could not be observed. In contrast, Laplace, following the earlier Idealist tradition, derived much of the field of statistics from first principles without resort to empirical observation. (Comte, himself, listed many principles, but never conducted experiments.)

The Positivist ideal, and faith in scientific principles leading the way to a better future, reached its pinnacle, perhaps, in the motto of the 1933 Chicago World’s Fair, “Science Finds; Industry Applies; Man Conforms.” Critics of this view label it scientism and reductionist. The recent appearance of process improvement “manifestos” makes light of this Positivist excess. Karl Marx, who we associate with manifestos, came from the competing Dialectical tradition.

Theoretical Foundation

Wikipedia describes Universality with the following passage: “In logic, or the consideration of valid arguments, a proposition is said to have universality if it can be conceived as being true in all possible contexts without creating a contradiction. … Truth is considered to be universal if it is valid in all times and places.”

The terms “formal” and “sound” represent concepts in math and logic that imply universality. Quoting from the Wikipedia definition of soundness, “a system is sound if each of its theorems (i.e. formulas provable from the empty set) is valid in every structure of the language.” Formality refers to the ability to establish correctness without dependence on specifics of the situation. From Wikipedia’s discussion of Formal Verification, “verification of these systems is done by providing a formal proof on an abstract mathematical model of the system.” The solution is provably correct if its construction is consistent with the model.

4. Problem

Forces

- Practices that work in many situations are necessarily general. Too much detail would tie them to a specific situation and reduce their range of applicability.
- Management is risk averse and demands evidence. Evidence found in literature and on consultants’ websites is easier to collect than evidence that requires data collection and analysis. Predicting value for a local solution requires intimate knowledge of the local situation.
- Given a choice, it is easier to predict the measurable value of a proven solution than it is assign a value to a method that only promises to find solutions.

5. Solution

Apply only proven general laws. If it hasn’t worked in many different places, reject it, because it’s not a valid solution.

6. Discussion

In the 19th Century Idealist tradition of reform, the goal of progress was the creation of perfect institutions, tempered by the wisdom that this goal can never fully be achieved. That ideal is enshrined in the American Constitution – in its preamble, “In order to form a more perfect union”, in its checks and balances, and in its process of amendment. A subtle distinction occurs when we shift from a goal of perfect institutions to a goal of perfect knowledge. Institutions are no longer judged by how well they perform, but by how well they apply, or conform to, universal truth.

There is a part of the community in many professions that feels that as a “science” their field should be concerned only with general principles that can be validated in large numbers of situations, analogous to laws of science. To them, following the Positivist model, knowledge consists of universal truths or laws. Universal laws are validated through consistent observation in many situations, and thus shown to be “true”. The reification of principles to the Positivist ideal of universal law creates a dangerous bias. Universal laws are not to be changed or questioned. We don’t adjust Maxwell’s equations for each situation.

Scientism and “physics envy” is blamed for the declining relevance of business school educations in How Business Schools Lost Their Way (Bennis and O’Toole, 2005). In Validity Vs. Reliability, Roger Martin (2005) similarly faults a bias towards research that yields repeatable numerical precision (reliability) at a cost of declining relevance (validity) in business theory. “A perfectly reliable system is one that produces an identical output each time if the same inputs are introduced to the system repeatedly.” “A perfectly valid system is one that produces a result that is shown, through the passage of time, to have been correct.”

The assumption of universality is an impediment to process improvement. Universal means being without contradiction in all its applications. If all our solutions are of the nature of universal law, there is no need for refinement. As described in Schön’s critique of Technical Rationality, solving problems is simply a matter of selecting the solution. Once applied, the solution is assumed to be correct. There is no need for further analysis, let alone, “too minute an investigation”.

More specifically, Solutions Must Be General impedes process improvement based on Flow, in Eliminate Waste, and Systems Thinking in Consider All Factors. Eliminate Waste studies the actual process in order to identify waste. The observations and recommendations may be specific to that context. Consider All Factors seeks to identify and fix causal factors, even if they are non-deterministic. When taken literally, soundness, and a lack of contradiction, rule out or reject factors that aren’t deterministic.

In the Known Uses section of Consider All Factors, Gerald Weinberg moved a soda machine to resolve a persistent backlog of help requests. Moving soda machines is never going to rise to the level of universal principle. In fact, it may not even work in a single other case. Yet, in that one case, it was the correct solution. What kind of solution could we have found if we were required to apply a universal principle?

In process improvement, there are very few universal truths – things that on their surface can be accepted as true without regard to details, and certainly not enough to solve every problem. In their book on empirical software and systems engineering, Endres and Rombach (2003) list 50 laws, 24 hypotheses, and 14 conjectures. But the complete set of them wouldn’t solve anybody’s problem. Many of the listed laws are very high level, including Moore’s Law and Metcalf’s Law.

Aspiring to universal principles is a laudable goal. Universal principles should be applied, when they can be found and shown effective for the problem at hand. But expecting every solution to be universal is misguided and naïve. Real problems require real solutions, not ideal solutions. Solving real problems involves analyzing the situation and crafting, or selecting, a solution to fit the problem.

The process of seeking a solution is not ad-hoc. Process improvement professionals should have a collection of tried and proven generative methods for analyzing situations and finding solutions. Several different analyses should be tried. The alternative solutions thus found or suggested should then be compared based on how well they solve the problem. The approach should be to try alternative methods of analysis, and to then compare the alternative solutions. The methods of analysis can be validated in many cases for finding a solution. The solutions themselves don't need to meet the same high bar of consistent success in multiple contexts without contradiction.

Solutions Must Be General often also requires success across multiple organizations. But experience within the organization can be more valid than experience across the industry. As explained in the introduction, within an organization, the three main causes of variation – the type of product being developed, the scope and complexity of the product, and the people doing the work – often don't apply within an organization (Reinertsen, 1997). Multiple projects within the same organization can validate a local best practice.

Traditional problem solving starts with a Gap analysis. What is the difference between the current situation as-is, and the desired situation to-be? A dialectical analysis, the kind of analysis that Positivists reject as "negative", starts in the same way. But rather than asking, "How well does the current process conform?," it asks, "How well does the current process perform?"

7. Implementation

Always assume that the current situation is comparable to the general case. If a situation poses unique opportunities or challenges, ignore what is unique about it. Before considering any change, i.e. a change suggested by members of your team, make sure that the same change has been tried in many other places and has consistently produced a measurable improvement in each of those situations.

8. Example resolved

Fred rejects any solutions found based on local observations, thus avoiding risks and staying true to sound scientific principles. In the process, he misses many opportunities for real improvement.

9. Known uses

Zimmerman et al. on general laws of defect prediction in empirical software engineering

Defect prediction is a popular area of research in empirical software engineering. The idea is that there are properties of applications and code from which general models and laws can be discovered. These laws can then be used to predict the density of different types of defects to be found in new code. In a test of generalizability by Zimmerman et al., published at FSE/ESEC 2009, they repeatedly trained a defect predictor on a code base and then tried applying it to a similar project not used in the training. Of the 622 cross-project predictions that they ran, only 3.4% actually worked.

Menzies et al. on bias against learning from local data

In a talk, titled, "Data mining: the missing link in empirical software engineering," Menzies, Brady, and Keung (2009) make a compelling case against treating general laws as the only goal of empirical software engineering. They advocate finding *general* models to support *local* learning. The conclusion suggests: "rather than try to 'clean up' empirical software engineering with more rigorous methods, we should instead explore methods for the faster generation and assessment of local models."

The above view is not universally accepted. A paper we submitted to a major software engineer conference in 2009 received a strong reject from one of the reviewers, with the following comment:

> "Data collection, mining, or visualizations themselves cannot make solid contribution to the theory or practice of software engineering. Some hypotheses need to be stated and they should be tested on a large number of products. If those hypotheses are validated repeatedly, then the authors can make a contribution to the theory. There is some mentioning of laws and testing them with the existing data, however, it was not made the major focus of the study."

The paper described a method of using data mining to automatically populate analysis models with local data, and described how to analyze the data. When applied to a several projects from a major corporation, the method identified a consistent pattern of process decisions that resulted in wasted effort. The solutions themselves were, of course, specific to that organization.

Scott Ambler's SEMAT Position Paper

In his position paper for the Software Engineering Method and Theory initiative's initial workshop, Scott Ambler argues that "Practices are Contextual, never 'Best'." He lists 10 context factors that affect the applicability of practices: life cycle scope, team size, geographic distribution, regulatory compliance, domain complexity, organizational distribution, technical complexity, organizational complexity, enterprise discipline, and paradigm. James Bach makes a similar argument in No Best Practices (2005): "Only through pretense can a practice that is interesting in a particular context become a "best practice" to which we all must bow down." (In the SEMAT 20 page Vision statement, the word "universal" appears 10 times, "sound" 5 times, and "formal" 4 times.)

10. Symptoms and Consequences

Symptoms

- Not looking at details of the current situation in any depth.
- Resistance to forms of analysis which use data that is specific to the current situation.
- Improvements found in publications are favored over improvements suggested by workers.
- Expecting the same rules to work for all cases.
- Assessments measure how well the workers are complying with the process rather than how well the process is supporting the business.

Consequences

- Solutions created or dictated may not be suitable for the current situation.
- Good improvements are ignored because they are not "standard" or have not been validated in other contexts.
- Opportunities for improvement derived from detailed or situated knowledge are rejected.

General Discussion

Applying different theories of process improvement is reminiscent of the blind men and the elephant. Each blind man describes the elephant in different terms after touching a different part. But it is all still the same elephant. Each observation gives a valid description from a different perspective.

It is not our purpose to pick favorites or draw conclusions about any of the five patterns. All five represent important contributions and deliver real results. All five can be applied in different ways. None are incompatible with the others. Thomas Kuhn (1961) described paradigms in science as being "incommensurable." But that view has long been abandoned. Scientists regularly apply multiple alternative paradigms – waves and particles being one example, and germs, heredity, and environment being another.

Some differences can be found with respect to issues and biases. Time can be addressed in all five, but is central only to Eliminate Waste and Incorporate Feedback and Learning. Issues like quality, safety, and security that are complex and require a comprehensive solution, are better addressed in Planning, Copy What Works, and Consider All Factors. Consistency is central to Planning, Copy What Works, Eliminate Waste, and Consider All Factors, but is less central to Incorporate Feedback and Learning. Innovation is, in some ways, the opposite of consistency. "Thinking outside the box" is more likely in Incorporate Feedback and Learning, but has also been associated with Eliminate Waste and Consider All Factors – which can address impediments to innovation. Eliminate Waste and Consider All Factors both require a good understanding of process details and involve situated analysis. A more hands-off approach (Silver Bullet), or a bias towards "sound principles" (Solutions Must Be General), will favor Planning and Copy What Works. ATAMO and All Problems Are Nails can occur with any of the five.

Perhaps the clearest distinctions among the 5 patterns can be found in their treatment of defects. Planning treats a defect as an event that requires planning and documentation. It calls for plans to monitor, document, track, and address defects. That approach will likely increase the cost and time of testing and rework but not address causes. Copy What Works treats defects as a subject in need of corresponding practices. Since documented solutions are plans, they are most likely to follow the planning model. The Eliminate Waste pattern treats defects as a problem that is integral to every step of the process. In the work-in-process metric, Eliminate Waste views rework time, and additional rework iterations, as part of the original task. The emphasis will be on finding and fixing defects as quickly as possible, and avoiding new ones. The Consider All Factors pattern treats defects as a factor itself with causes and effects. It seeks to identify the root cause and also any contributing factors. It will also considers direct and indirect downstream costs involved in the choice of a solution. Finally, the Incorporate Feedback and Learning pattern treats defects as part of a loop where the goal is to improve sensing and shorten cycle time for detection and repair.

Pattern Relationships

The relationship among the 9 patterns is shown in Figure 1, below.

Have a Plan is an essential beginning. All of the other patterns assume some understanding of the process. If there is no shared understanding, and tasks at all levels are performed differently each time, there is no process to improve. Copy What Works is a source of plans. If one is choosing to adopt a new plan, adopting a Best Practice would be a good choice. But adopting a Best Practice is not required. A person or organization can start by documenting what they are already doing. Eliminate Waste improves the efficiency of an existing plan by reducing waste. Consider All Factors will focus on those factors that affect the optimal performance of a plan. By addressing contributing and causal factors, including those that are external, the plan is more likely to realize its full potential. Thus Consider All Factors is more likely to strengthen a plan than to change it. Even when we know the right course of action, as was the case in the BP Gulf oil spill, contributing factors can influence us to make a wrong choice. Consider All Factors addresses that problem. Finally, Incorporate Feedback and Learning draws attention to those parts of the plan that affect adaptation and agility. Changing and leveraging feedback in the plan improves plans and makes the process more responsive to changing goals and circumstances.

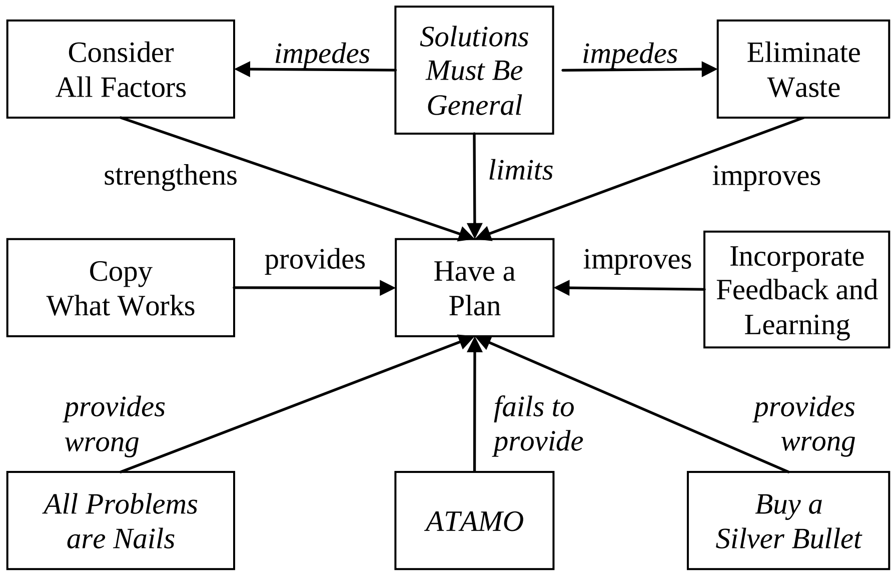

Figure 1. Relationships among the 5 patterns and 4 anti-patterns.

Among the anti-patterns, ATAMO is a process improvement initiative that lacks a plan. The problem with Buy a Silver Bullet is that it lacks the attention to details and specifics needed to make a proper choice for the desired improvement. More often it will lead to a wrong choice and no improvement at all. Similarly, All Problems Are Nails, simply assumes that the existing course of improvement can address the latest problem. Not only might this be the wrong plan, but all other avenues of improvement are foreclosed. Finally, Solutions Must Be General creates a bias against any form of improvement that involves a solution based on situated analysis. It implies an approach to problem solving based on selecting the solution rather than adapting one. This bias precludes or impedes patterns of continuous improvement such as Consider All Factors and Eliminate Waste.

Conclusion

The patterns in this collection address the theories that underlie methods of process improvement in product and software development. A method of process improvement is an activity which focuses on the process by which product and software development is carried out. The goal of process improvement is to create a better process than the one that existed prior to the activity of process improvement. There are many methods of process improvement. Many of them are well known, or at least have well known names. What we present here is a collection of basic theories behind the methods.

Awareness of underlying theory is an important step in the process of reflection. Through reflecting on underlying theories, we become more aware of the basic assumptions and biases that frame the choices that we make. They also help us make comparisons, and find common ground, among the different approaches to process improvement.

We presented 5 patterns of theories that underlie methods and practices of process improvement – Have a Plan, Copy What Works, Eliminate Waste, Consider All Factors, and Incorporate Feedback and Learning. Our intention was to show that, as theories, there is little conflict among the five. Their views are analogous to the five blind men describing an elephant after each has felt a different part. All five patterns can be used to assess and improve a process. Each of the five perspectives offers important value. It is hoped that this deeper view will inform, and support reflection, in the ongoing debate about methods and practices of process improvement.

We also presented 4 anti-patterns of theories that can be found in criticisms of methods of process improvement – ATAMO, Buy a Silver Bullet, All Problems Are Nails, and Solutions must be General. These 4 anti-patterns represent common over-simplifications or biases in process improvement.

Process improvement should be approached from multiple points of view and it should be approached with an awareness of the assumptions, strengths, and biases of each perspective. As John Tukey wrote long ago in his classic book on data analysis (1977), the evidence should be explored in as many ways as possible until plausible stories emerge. The more perspectives one is willing to apply, the more problems, causes, and solutions one will find.

Acknowledgement

The authors would like to thank Jason Che-han Yip whose suggestions have helped significantly improve this work.

References

Scott Ambler (1998). *Process Patterns: Building Large-Scale Systems Using Object Technology*. Cambridge University Press/SIGS Books

Scott Ambler (1999). *More Process Patterns: Delivering Large-Scale Systems Using Object Technology*. Cambridge University Press/SIGS Books

Scott Ambler (2010). Context counts: Position paper for SEMAT. Available at http://www.semat.org/bin/view/Main/WorkshopPositions

Chris Argyris and Don A. Schön (1976). *Organizational Learning: A Theory of Action Perspective.* Addison-Wesley.

James Bach (2005). No Best Practices, http://www.satisfice.com/blog/archives/27

Kent Beck (2000). *Extreme Programming Explained: Embrace Change*, 2nd Edition. Addison Wesley.

Warren G. Bennis and James O'Toole (2005). How business schools lost their way. *Harvard Business Review*, May2005, 83(5), pp.96-104

M.B. Chrissis, M. Konrad, and S. Shrum (2003). *CMMI: Guidelines for Process Integration and Product Improvement*. Boston: Addison Wesley.

Alistair Cockburn (2004). The end of software engineering and the start of economic-cooperative gaming. Humans and Technology Technical Report, HaT TR 2004.02, Jan 14, 2004. http://alistair.cockburn.us/The+end+of+software+engineering+and+the+start+of+economic-cooperative+gaming

Auguste Comte (1907). *A General View of Positivism*. London: Routledge and Sons.

James O. Coplien and N.B. Harrison (2004). *Organizational Patterns of Agile Software Development*, Pearson Education.

James O. Coplien (2007). Organizational patterns: A key for Agile software development. INCOSE talk. May 28, 2007

John Dewey (1933). *How We Think: A Restatement of the Relation of Reflective Thinking to the Educational Process*, Boston, Heath.

Terry Doerscher (2008). Notes from ITIL foundation training. *Planview* blog, http://blogs.planview.com/tdoerscher/2008/03/notes-from-itil.html

Tore Dybå (2003). Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context. In *Proceedings of the 9th European Software Engineering Conference*, ACM, pp. 148-157.

Rob England (2009). Community, activity, environment. *The IT Skeptic* blog, February 18, 2009. http://www.itskeptic.org/community-activity-environment

Albert Endres and Dieter Rombach (2003). *A Handbook of Software and Systems Engineering: Empirical Observations, Laws, and Theories*. Harlow, England: Pearson Addison-Wesley.

Steven D. Eppinger (2001). Innovation at the speed of light. *Harvard Business Review*, January 2001, pp. 149-158.

Han Gerrits (1994). Business modeling based on logistics to support business process re-Engineering. In *Business Process Re-engineering: Information Systems Opportunities and Challenges*, B.C. Glasson, et al. (eds.), Elsevier Science, Amsterdam, 1994, pp. 279-288.

Frank B. Gilbreth (1914). *Primer of Scientific Management*, Second Edition, New York: Van Nostrand.

Michael Hammer and James Champy (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*, Harper Business.

Rick Hefner (2007). Is CMMI high maturity worth the investment? Presentation to the Southern California Software Process Improvement Network, February 2, 2007. Available at http://www.ccpe.csulb.edu/spin/pastmeetings.htm

Jim Hunziger (2006). Why Standard Work is not standard; Training within industry provides an answer. *Target*, Association for Manufacturing Excellence, (22)4, p. 7-13.

Linda Ibrahim (2008). A process improvement commentary. *CrossTalk*, 21(8), August, 2008, p. 26-29.

Antero Jarvi, Harri Hakonen, and Tuomas Makila (2007). Developer driven approach to situational method engineering. Chapter in *IFIP International Federation for Information Processing, Volume 244. Situational Method Engineering: Fundamentals and Experiences*, J. Ralyte, S. Brinkkempter, and R. Henderson-Sellers (Eds.), Boston, Springer, pp. 94-99.

Michael N. Kennedy (2003). *Product Development in the Lean Enterprise*. Richmond, The Oaklea Press.

Michael N. Kennedy (2005). Implementing Toyota's product development system. *Appliance Magazine*, January 2005.

Barbara A. Kitchenham, E. Mendes, and G.H. Travassos (2007). Cross versus within-company cost estimation studies: A systematic review, *IEEE Transactions on Software Engineering*, 33(5), May 2007, pp. 316-329

A. Kling (2010). Do C.D.O.'s have social value? *New York Times*, April 27, 2010.

Lauri Koskela and Greg A. Howell (2002). The underlying theory of project management is obsolete. In: *Proceedings PMI Research Conference*, Slevin, D., Cleland, D. and Pinto, J. (eds.), pp.293-302.

Thomas S. Kuhn (1962). *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press

Nancy Leveson (2004). A systems-theoretic approach to safety in software-intensive systems. *IEEE Transactions on Dependable and Secure Computing* 1(1), January 2004, pp.66-86.

Juan Li, Nan Jiang, Mingshu Li, Qing Wang, and Yanwu Yang (2007). Tracking projects through a three-dimensional software development model. *31st Annual International Computer Software and Applications Conference (COMPSAC)*, 2007, vol. 1, pp.301-308.

Herbert Marcuse (1941). *Reason and Revolution: Hegel and the Rise of Social Theory*, London: Oxford University Press

Roger Martin (2005). Validity vs. reliability: Implications for management. *Rothman Magazine*, Winter 2005, pp.4-8.

A.H. Maslow (1966). *The Psychology of Science: A Reconnaissance*, New York, Harper & Row

Tim Menzies, Adam Brady, and Jacky Keung (2009). Data mining: the missing link in empirical software engineering. Invited lecture presented at the University of Texas, Dallas, October 29, 2009. Slides available online at Menzies' website: http://menzies.us./.

John Stewart Mill (1865). *August Comte and Positivism*, Project Gutenberg Ebook, http://manybooks.net/support/m/millj/millj1683316833-8pdfLRG.pdf

Donald Norris, Linda Baer, Joan Leonard, Louise Pugliese, and Paul Lefrere (2008). Action Analytics: Measuring and Improving Performance That Matters in Higher Education. *EDUCAUSE Review* 43(1) January 2008, pp. 42-64.

Anne Pankhurst (2007). *Planning and Periodisation*. Leeds, UK, The National Coaching Foundation.

Marvin L. Patterson (1992). *Accelerating Innovation: Improving the Process of Product Development.* New York, Van Nostrand Rheinhart.

Cécile Péraire (2007). A roadmap to method development. http://www.ibm.com/developerworks/rational/library/feb07/peraire/

Jean Piaget and Barbel Inhelder (1969). *The Psychology of the Child*. New York, Basic Books.

Mary Poppendieck and Tom Poppendieck (2003). *Lean Software Development: An Agile Toolkit*. Upper Saddle River, Addison-Wesley, 2003.

Mary Poppendieck and Tom Poppendieck (2007). *Implementing Lean Software Development: From Concept to Cash*. Upper Saddle River, Addison-Wesley.

Jan Pries-Heje and Jørn Johansen, Eds. (2010). SPI Manifesto, *European System & Software Process Improvement and Innovation (EuroSPI)*. Available at http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf

Russell L. Purvis, Jose Santiago, and Vallabh Sambamurthy (1999). An Analysis of Excluded IS Processes in the Capability Model and Their Potential Impact. Chapter 3 in *Software Process Improvement: Concepts and Practices*, Eugene McGuire, Ed. Hershey PA: Idea Group, pp. 31-46.

Donald G. Reinertsen (1997). *Managing the Design Factory: A Product Developer's Toolkit*. New York, The Free Press.

Donald G. Reinertsen (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development.* Celeritas Publishing.

James L. Rogers (1997) Reducing cycle time and cost through process resequencing. *Proceedings of the International Conference on Engineering Design*. Technical Report: NASA-97-11iced-jlr, NASA Langley Technical Report Server.

Don A. Schön (1983). *The Reflective Practitioner: How Professionals Think in Action*. Basic Books.

Peter Senge (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. Doubleday.

Alec Sharp and Patrick McDermott (2001). *Workflow Modeling: Tools for Process Improvement and Application Development*, Boston, Artech House Publishers.

Preston G. Smith and Donald G. Reinertsen (1998). *Developing Products in Half the Time: New Rules, New Tools*. New York, Van Nostrand Reinhold.

Donald V. Steward (1981). *Systems Analysis and Management: Structure, Strategy, and Design*. New York, Petrocelli Books.

David J. Szymanski and Thomas D. Neff (1996). Defining software process improvement. *CrossTalk*, 9(2), February 1996

Frederick W. Taylor (1914). *The Principles of Scientific Management.* New York: Harper.

John W. Tukey (1977). Exploratory Data Analysis. Reading, MA: Addison-Wesley.

Michael VanHilst and Shihong Huang (2009). Mining objective process metrics from repository data, In proceedings *21st International Conference on Software Engineering Knowledge Engineering (SEKE'2009)*, pp. 514-519.

Lev S. Vygotsky (1962). *Thought and Language*, Cambridge, MIT Press.

Allen C. Ward (2007). *Lean Product and Process Development*. Cambridge, MA: Lean Enterprise Institute.

Thomas Zimmerman, Nachiappan Nagappan, Harald Gall, Emanuel Geiger, and Brendan Murphy (2009). Cross-project defect prediction: a large scale experiment on data vs. domain vs. process. *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM 2009.