

A Pattern System of Underlying Theories for Process Improvement

MICHAEL VAN HILST AND EDUARDO B. FERNANDEZ, Florida Atlantic University

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 six patterns for theories that underlie many of the well known practices in process improvement. The patterns are Plan, Best Practices, Flow, Feedback, Systems Thinking, and Living System. These patterns are found in ISO 9000, PMBOK, CMMI, SPICE, Lean, TQM, Six Sigma, and Agile. Like all patterns, the patterns in this system are structural configurations that solve a problem in a context.

Categories and Subject Descriptors: **D.2.9 [Software Engineering]:** Management—*Productivity, Software Process Models (e.g. CMM, ISO, PSP)*

General Terms: Economics, Management, Theory

Additional Key Words and Phrases: Process improvement, best practices, lean, agile

ACM Reference Format:

Van Hilst, M. and Fernandez, E. B. 2011. A Pattern System of Underlying Theories for Process Improvement. Proceedings of the 2010 Conference on Pattern Languages of Programs. (October 2010), 24 pages.

1. INTRODUCTION

The literature on software process improvement offers a cornucopia of competing initiatives, methodologies, certificates, practices, and standards. It is difficult to choose which ones to apply. Which one is better than the other, and what kind of improvement do they offer? In trying to make sense of it all, we decided to step back, to reflect on the underlying theories and assumptions, and to search for foundational patterns. In so doing, we followed the advice of Don Schon.

When faced with competing explanations of a problem, the reflective practitioner, in (Schön 1983), steps back to look at the underlying theories in which the problem is set.

“When we set the problem, we select what we will treat as “things” of the situation, we set boundaries of our attention to it, and we impose upon it a coherence which allows us to say what is wrong and in what directions the situations need to be changed.”

Like a patterns, an underlying theory is a framework of goals, solutions, and assumptions that guide how we observe situation, define problems, and construct solutions. In science, the underlying theories, like particles, waves, relativity, plate tectonics, and evolution, are well known and understood. For software process improvement, the underlying theories are not so well known. To address that problem, we present this system of patterns.

Presenting these patterns as a system, as is done here, makes it easier to promote reflection and insights, and highlight differences. In preparing the material, we also found that each pattern has its own unique history. We present some of that history, along with each pattern, to help explain how the pattern came to take hold, and in what form.

In all, we found six underlying patterns. The six patterns are Plan, Best Practices, Flow, Feedback, Systems Thinking, and Living System. A process can be improved by having better plans (Plan), copying the most successful practices (Best Practices), reducing waste (Flow), using newer information

Author's address: M. Van Hilst, College of Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431; email: vanhilst@alum.mit.edu; E.B. Fernandez, College of Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431; email: ed@cse.fau.edu

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers' workshop at the 17th Conference on Pattern Languages of Programs (PLoP). PLoP'10, October 16-18, Reno, Nevada, USA. Copyright 2010 is held by the author(s). ACM 978-1-4503-0107-7

(Feedback), addressing more factors (Systems Thinking), and having the process improve itself (Living System).

The same six patterns are found in many of today's best known process improvement methodologies. Plan is emphasized in the ISO 9000 standards for quality management and in the PMBOK (Project Management Body Of Knowledge). 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). Flow is emphasized in Toyota Lean practices. Systems Thinking is emphasized in Total Quality Management (TQM) and Six Sigma. Feedback is emphasized in Agile practices. Finally, Living System is a goal in TQM, Lean, and Agile process improvement. In a more general sense, it is also the subject of Christopher Alexander's most recent work.

Here is an overview of highlights for each of the six patterns:

- The main idea in Plan is that teams are most efficient when everybody knows what to do. Management's use of plans borrows from the military, where plans are separated into strategic, operational, and tactical.
- The main idea in Best Practices is that one should do the things that are already known to work the best. But its history in Positivism may lead some to treat practices as laws, and ignore more situated types of learning.
- Flow is best understood by comparing the work of Frank Gilbreth with that of Frederick Taylor. While Taylor looked for the fastest worker, Gilbreth looked for the one who used the least effort. By eliminating motions that cause fatigue, and not value, even the fastest worker becomes more productive.
- Feedback as a theory owes a debt to automating gun mounts in World War II. You have to deliver the right ammunition to the right place at the right time. All three matter, and the target is not sitting still.
- What is notable about Systems Thinking is that its major protagonists, Ludwig von Bertalanffy and Edwards Deming both had backgrounds in agriculture and biology. You can't tell wheat how to grow or cows how to make milk. Yet, by studying the factors that contribute to yield, the field of agriculture has achieved tremendous improvements in productivity.
- Finally, while mechanistic systems are subject to the 2nd Law of Thermodynamics, Living Systems have negative entropy. A Living System self-organizes and creates its own order.

Before discussing the six patterns, we need to say something about measurement, and what it is that software process improvement is supposed to improve. Traditional measurements focus on lines of code, hours of effort, and numbers of defects. More recent measurement focuses on value and knowledge. Value is whatever properties of the process, and its output, increase value for the customer. It includes features, cost, timeliness, and quality.

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. This perspective was introduced by (Patterson 1992), and later expanded by (Reinertsen 1997) and others. 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.

It should be noted that the patterns presented here apply to processes as much as they to process improvement. Process improvement is about making better processes and making processes better. Thus the patterns should be read as guidance on what it is about processes that can be improved, as well as guidance on how to do so.

As foundational patterns, the patterns apply generally to processes and product development. However, while the patterns are not specific to software product development, most of the examples, and the discussion, describe the patterns in the context of their application to software process improvement.

2. PLAN



Wikimedia photo

... the project proceeds day to day. It is not always clear what to do. Team members don't always know what each other is doing, or what to expect. Progress is erratic and commitments are not always met. The customer is not happy.



For the project to make effective progress everyone should know what to do, and how to do it.

Deciding what to do in each situation costs time and effort. Decisions must then be communicated to the 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.

2.1 History

Rituals and recipes have been around since the earliest recorded times. But our understanding of planning in business derives largely from military experience. In war, large numbers of people must act in coordination. The consequences are life and death. Sun Tzu's *The Art of War*, written in the 6th Century B.C., devotes the first chapter to planning. The word "plan" itself comes from the Latin "planum" or flat surface, and means a drawing on a flat surface.

Planning appeared in management literature early in the 20th Century and gained momentum with the production demands of World War I. Gilbreth's *Primer of Scientific Management* includes the following quote from Louis Brandeis, "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" (Gilbreth 1914, p.3). Henri Fayol's *Administration Industrielle et Générale* described planning as one of five core principles of management (Fayol 1916). Fayol's plans anticipated needs and provided coordination, while allowing flexibility to deal with situations as they arose. In *Organizing for Work*, Henry Gantt criticized the failures and inefficiencies of American wartime production. As a solution, he presented a method of planning, using what we now call Gantt Charts. "With a plan of this sort the Ordnance Department would be in a position to state at any time its immediate and probable future needs in men, materials, transportation, and equipment" (Gantt 1919, p.80).

In World War II, 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 both perform and teach the work to be done. In this approach, the supervisor acted as trainer and coach. The approach was not continued in the U.S., but was later adopted by Toyota under the names Standard Work and J-Programs (see Implementation below).

In the 1950's, Booz Allen Hamilton developed the 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.

2.2 Theoretical Foundation

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. Gantt’s method of planning is operational. 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 and not get shot.

The application of plans in project management is grounded in a transformation theory of projects (Koskela and 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 Best Practices), accurately predicting the needs and duration of each task, scheduling tasks so workers don’t sit idle, and controlling deviation.

2.3 Strategy and Variation

Anne Pankhurst 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 (Pankhurst 2007). 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 argument is about what type of plan is appropriate. Types of plans from which to choose include prescriptive, descriptive, adaptive, generative, incremental, and partial. Plans can also be written, learned, or both.

A plan can be prescriptive and rigid, or descriptive and flexible. A prescriptive plan gives the “right way” to perform a task and must be followed to the letter. In contrast, a descriptive plan provides a reference that worked in the past, but allows for deviation. (Coplien 2005) makes this distinction under the heading, “Organizational Patterns are Inspiration Rather than Prescription.” Prescriptive plans can be seen as overly bureaucratic and restrictive. People don’t like to be told what to do. But for situations involving compliance and safety prescriptive plans can be appropriate. In Wysocki’s Continuous Process Improvement Model (Wysocki 2009), teams are monitored for compliance, but are allowed to deviate. When a team chooses to deviate from the standard plan, however, it must document the reasons for doing so. When preparing a plan care should be taken to assure that the level of detail is sufficient to achieve the goal, but not so specific as to be overly restrictive. In process improvement, the use of waivers can indicate problems with plan detail. In one case, (Coplien 2007) observed that “80% of the work was being done under documented waivers.”

Plans cannot anticipate every detail of every situation. Three variations of planning address situation-specific needs: adaptive, incremental, or generative. An adaptive plan includes methods to adapt the plan to a given situation. The SPI Manifesto (Pries-Heje and Johansen 2010) advocates adaptive plans: “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.”

An incremental plan covers only a short period in the process, after which another plan is needed. After each increment, new knowledge is obtained. Things may also have changed (see Feedback). Choosing plans between increments, rather than at the very beginning, can produce better plans. But some level of long term planning will still be needed to address commitments. Agile planning has a strategic plan that is comprehensive. But the operational plans are incremental and applied in short cycles. When changes are frequent, incremental planning is more cost effective. Rewriting a plan that was already written takes extra effort (see Flow).

A generative plan is a plan for making other plans. It answers the question, “If you don’t have a plan, what would it take to get a plan?” In a generative plan, the “defined desirable state” is having a plan that works for the situation at hand. Generative plans are especially appropriate for process improvement, since most improvements must be found, and cannot be known in advance. In process improvement, a generative plan is not the improvement itself, but a method of finding things to improve for which plans are then developed.

With a collection of 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 (www.semat.org), have plan and method fragments that can be composed to form different plans depending on the situation. 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. Other approaches construct plans from pre-existing components at the time that the need arises. They also allow 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.

Plans that are known and used do not always have to be written. At Toyota, plans are descriptive and learned (see Implementation, below). The learned approach is also emphasized in Coplien's Domain Expertise in Roles and Apprenticeship patterns (Coplien 2005). Certifications, like ISO 9000, often require both written evidence of plans and demonstrated knowledge.

In addition to having plans for directly achieving the goal, processes should also have contingency plans, procedural plans, management plans, and improvement plans. Contingency plans include alternatives to deal with changing conditions. Procedural plans provide structure and consistency for how members of the team should behave and interact. To improve communication, the procedural plans in extreme programming prescribe the layout of the room. Management plans describe how the project should be managed – managers need plans, too. Management decision making can be the most significant factor in success (see Systems Thinking). Improvement plans concern how to make things better. Process improvement involves both better plans, and having plans to make things better.

The Project Management Institute seeks to improve the practice of project management through practice standards documented in the Project Management Body of Knowledge (Project Management Institute 2008). The PMBOK places a heavy emphasis on plans. The PMBOK is itself a plan for how to manage generic projects. Plans specified in the PMBOK include: the requirements management plan, the staffing management plan, the procurement management plan, the communications plan, the schedule management plan, the change management plan, the risk management and response plans, the contingency plan, and the quality management plan. Project execution in the PMBOK consists almost exclusively of dispatching plan elements, and monitoring their performance.

2.4 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 uses a specific 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 (Hunziger 2006). As mentioned in history (above), the Toyota approach is a variation of the Training Within Industry program used by the United States in World War II.

In general, plans are more effective, and more willingly applied, if those who have to use them are involved in their creation. Frederick Taylor believed, on the other hand, that plans should be left to specialists, and that workers would be more willing to follow plans if they were paid more to do so (see Best Practices).

2.5 Risks

There is a danger in following plans too closely, and not allowing the plan to adapt and change. More adaptive or incremental planning, as well as contingency plans, should be used where either the situation or the goal is likely to change.

Plans can be over specified with too much detail. Both (Wysocki 2009) and (Coplien 2007) describe “shelf feet” or “decimeters” of plans that are never read because they are much too long and the level of detail is too specific for any real situation. Wysocki recommends a library with plans documented at three levels: a one page executive summary, a two page descriptive reference, and a full tutorial (Wysocki 2009).

Plans can also be underspecified and miss important detail. Such plans are called ATAMO which is short for “And Then a Miracle Occurs”. ATAMO is a reference to a famous Sidney Harris cartoon that 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 “then a miracle occurs” 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.”

A typical ATAMO plan for process improvement is characterized by its emphasis on the before and after, with little to say about how to design the change or actually change the process. The plan 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.

The problem occurs when tactical details for performing step 4 are missing – the plan is incomplete. The ATAMO label has been applied to the Software Engineering Institute's IDEAL model for process improvement (Ouellette 2007), the Information Technology Infrastructure Library (ITIL) (Doerscher 2008), and Business Process Reengineering (Gerrits 1994) (Sharp and McDermott 2001).

When processes have repeatable elements, it is easier to plan them in advance. Process improvement initiatives, however, are often not repeated. Solutions tend to be specific to the situation. Once the improvement is made, there is no reason to repeat it. The methods of looking for things to improve, and steps to create specific kinds of solutions, can and should be planned. The patterns for Best Practices, Flow, Systems Thinking, Feedback, and Living System all provide non-ATAMO strategies for process improvement. Having plans is only the beginning.

2.6 Therefore

Have plans for what to do and make sure everyone knows their part.

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. “Knowing the drill” improves outcomes and success.



Best Practices can be used to find plans that have worked well for others. Flow can improve the plans currently being used, as can Feedback and Living System. Having better plans is not the only way to improve performance. Systems Thinking addresses other issues that affect performance.

3. BEST PRACTICES



... your team is experienced and can execute on plans. But some competitors are doing better than you. You want to improve. But change involves risk. You want some assurance that any changes you make will work.



When others in a similar setting are performing better than you are, copy their best practices.

There are many ways to perform a task or complete a project. But some are better than others. Businesses can't afford to try too many practices, and may not have the same expertise as others to discover which ones work best. When others are clearly performing better than you are, copying what they do can bring parts of your own process up to their level.

3.1 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* (Taylor 1914). The goal of scientific management was to improve every aspect of the process by finding the "one best way."

3.2 Theoretical 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 leading to a specific 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 19th Century, practices in many fields involved quackery, superstition, and myth. By the late 19th and early 20th Centuries, the Positivist model of knowledge, with its proven practices and standards for validation, brought discipline to many professions (Schön 1983).

3.3 Strategy and Variation

Many well known methodologies build on collections of best practices. In manufacturing, American companies adopted the best practices of Toyota, along with their Japanese names. These practices include just-in-time (kanban), mistake proofing (poka yoke), and continuous improvement (kaizen).

The practices in Extreme Programming are a set of best practices, carefully chosen and combined from the earlier experience of Kent Beck and others. 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 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 fell behind schedule, went over budget, or failed. While some projects were successful, many more were not. The idea was for companies that experience failures to 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.

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.” (SEI 2010)

In order to adopt a best practice, one must first be aware of it. Some practices are published as standards. Trade publications often have articles about best practices, and consultants provide advice on practices. Some practices are also published as patterns. Direct interaction with members of outside teams, as occurs at conferences and in networking events, allows team members to discuss challenges and exchange best practice ideas.

Vendors can be a source of best practices, which they learn from existing customers. Enterprise Resource Planning (ERP) solutions are difficult to build and pose many risks. “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.” (SAP 2010)

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 (Wysocki 2009) 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.

3.4 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 set of 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.

3.5 Risks

Some practices gain popularity and strong support without proper validation. Such practices are called “fads.” For example, 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).

There is also a risk of using “Best Practices” as a buzzword, without actual analysis of the problem or its solution. We find this criticism in James Bach’s No Best Practices (Bach 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.

There is a danger of putting too much faith in best practices – treating them as Universals or Scientific Laws. Universals are assumed to be irrefutable, correct in every detail, and applicable in all situations.

Universals are held as the ideal in academic research and in vendor marketing. To understand this tendency, and its impact on the social sciences, it is worth revisiting the history of Positivism.

August Comte created Positivism in the 1830's. At that time, Europe and the Americas were swept up in a series of seemingly never-ending revolutions, each one claiming to replace the old order with a new and better order. Comte blamed this situation on the dominant philosophical tradition of the time, Idealism, and its mode of analysis through dialectical reasoning.

To Idealists, the goal was the creation of ever more perfect institutions. The way to improve institutions was through analysis and refinement. This view is embodied in the first line of the United States Constitution, "in order to form a more perfect union", and its mechanism of refinement through amendment.

In dialectical reasoning, an initial principle, or thesis, is always flawed by virtue of being abstract. When applied in the real world, the flaws create a negative tension – the antithesis. The initial abstraction must be refined or synthesized to resolve the tension and be made concrete. The synthesis itself becomes a thesis, or abstraction, and the process repeats. A similar concept is found in Herbert Simon's concept of bounded rationality and the process of "satisficing" (Simon 1956). While today we know this sequence as Thesis-Antithesis-Synthesis, Hegel referred to it as Abstract-Negative-Concrete.

To Comte, the antidote was Positivism – the search for invariant laws governing the social and natural worlds. It "will supply in every department of public and private life fixed principles of judgment and of conduct" (Comte 1907, p.4). Comte saw Positivism as the ultimate stage of human progress, 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. Comte chose the name Positivism specifically as a rejection of the negative in dialectical modes of thought. When governed by natural law, there could be no negative. This view of progress reached its pinnacle, perhaps, in the motto of the 1933 Chicago World's Fair: "Science Finds; Industry Applies; Man Conforms."

To Positivists, the goal is the creation of perfect knowledge. Ultimately, there is only one truth needing simply to be discovered. The method of discovery is the creation and testing of hypotheses, with validation by empirical observation. Once discovered, empirical truths are universal – true for all times and places. Progress has a single path leading ever closer to perfect knowledge.

Don Schön, who argued for more reflection in problem solving, termed the Positivist model, "Technical Rationality" (Schön 1983). John Stuart Mill had noted the danger of an excessive faith in Positivism for the social sciences already in the 1800's.

"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" (Mill 1865, p.37).

In his position paper for the Software Engineering Method and Theory initiative (SEMAT), Scott Ambler argues that "Practices are Contextual, never 'Best'" (Ambler 2010). 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. In the 20 page SEMAT vision statement, the word "universal" appears 10 times (Jacobson et al. 2010).

A Best Practices philosophy can create a kind of reverse NIH (Not Invented Here). If the strategy is to only consider improvements that have been proven in 1000 other places, you should ask yourself, "Do I want what works for us, or what works for everyone else?" Best Practices augments, but does not replace, the need to analyze the situation at hand and to search for continuous process improvement.

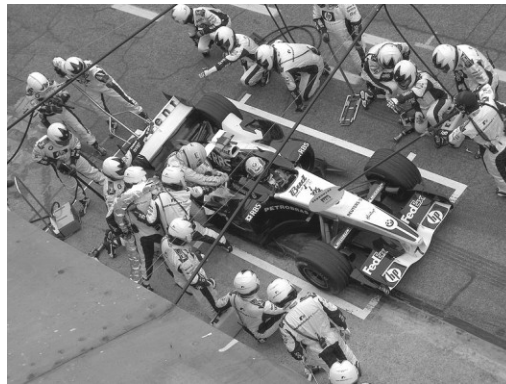
3.6 Therefore

Find and follow the best practices for the tasks to be done.

Copying what works for others can jump-start or leap-frog process improvement. 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.



4. FLOW



Wikimedia photo by Alexandre López

... plans are in place and everybody knows what to do. We have a good sense for each project how much effort it will take. But to be competitive, we need to deliver faster and at less cost, without compromising quality.



View the 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.

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.

4.1 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 Best Practices). 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 (Gilbreth 1914).

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 less than 5 seconds, and by athletes who broke numerous records in the 2008 Summer Olympics. Gilbreth personally changed the way doctors and nurses exchange tools in the operating room. Before Gilbreth, doctors often took their eyes away from the patient to rummage through tools.

4.2 Theoretical 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 can 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 bottlenecks (queues with backlogs).

4.3 Strategy and Variation

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.

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* require rework, which adds cost but not value. Defects which remain in a product reduce its value.
- *Inventory* is partially done work. When produced before it is needed, it sits idle. In development, knowledge that sits idle fades from memory and becomes increasingly expensive to recover. Untested work, test reports, and task queues are all forms of inventory.
- *Waiting* occurs when work is postponed 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.
- *Transportation* refers to problems that occur when materials move from one group to another. Transportation in development is called hand-off. The recipient expends effort to acquire knowledge that the previous handler already had.
- *Motion* occurs when things are not well organized and ready-at-hand. In development, motion includes searching for information. In task switching, developers must recover information, or even just the train of thought, that they themselves once had.
- *Over-production* refers to things that are produced, but never used or needed. Over-production can include unused features or options. 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 be needed. In Agile development, over-production is called YAGNI (for You Ain't Gonna Need It).
- *Over-processing* is activity with little benefit to the customer. It can apply to activities that serve the process, but not the customer. Over-processing also describes anything that would involve less work if done 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.

Allen Ward, who studied the Toyota development process, defined three knowledge wastes (Ward 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 that should be used. In wishful thinking, responsibility is given, or taken, without all the data or knowledge that goes with it.

The Poppendiecks identify *churn* as a source of rework. "When requirements are specified long before coding, of course they change. When testing occurs long after coding, test-and-fix churn is inevitable" (Poppendieck 2007, p.24).

4.4 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 Best Practices).

Two different views of flow 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 through 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. In a development process, flow can also be studied by following the movement of information through the process.

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.

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 often have a large percentage of work in process during the life of a project. Iterative and agile processes stage more of the work and have typically have lower percentages of work in process.

Progress in the process can be displayed as an S-curve, showing cumulative progress versus time. Work-in-process and time-in-process can then be derived directly from progress S-curves by applying Little's Law (Reinertsen 1997). Using aggregate progress data, Little's Law gives generally comparable efficiency metrics against which to gauge improvement (VanHilst and Huang 2011).

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 (poka 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 reduces over-production, and provides both mistake-proof objectives (it has to fit the test) and source inspection (tests are run as code is written).

4.5 Risks

The Lean focus on queues should not be mistaken with queuing practices in traditional assembly line manufacturing. The notion that "when all workers are busy, the process is optimally efficient" leads to increased backlogs for a ready supply of work. We see this in some doctor's schedules where you have to make an appointment months in advance. This strategy builds in delay, and is very inefficient when faced with change (Reinertsen 1997).

Process improvement books aimed at managers may define eliminating waste in terms of automating or eliminating non-value-adding tasks or steps. For example, (Szymanski and Neff 1996) suggest, "Analyze each activity documented in the flowchart to determine if it is 'value-added'". In the next step, they suggest, "Identify ways to eliminate or modify non-value-added activities in the process." This approach reflects a strictly operational view (see Plans). Much of the value in Flow is more fundamental and requires a more detailed and intimate understanding of the process.

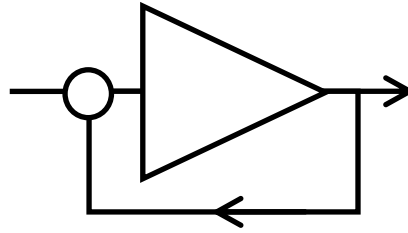
4.6 Therefore

Reduce waste and improve flow on a continuous basis.

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.



5. FEEDBACK



... the project is off track. The situation has changed. Some of the requirements were not known or were poorly understood at the start. Other requirements have changed over time. Errors were made during execution and some plans were ill conceived to begin with. All of these issues require corrections and adaptations. But first, they must be detected.



Improve those aspects of the process that help, hinder, speed up, or delay learning and dynamic adaptation to change.

Response to change has two parts: detecting the need for change, and executing a response. Failure of either part perpetuates the problem. Even without change, the process should be continuously observed and analyzed to find things to improve, and to monitor improvements once they have been made.

5.1 History

The study of feedback in controls dates back to 1788, when James Watt and Matthew Boulton applied a centrifugal governor to control speed in steam engines. The design was based on a mechanism used in windmills since the 17th Century. In 1858, Alfred Russel Wallace noted a similar principle working in nature.

“The action of this principle is exactly like that of the centrifugal governor of the steam engine, which checks and corrects any irregularities almost before they become evident; and in like manner no unbalanced deficiency in the animal kingdom can ever reach any conspicuous magnitude, because it would make itself felt at the very first step, ...” (Wallace 1858).

In 1868, James Clerk Maxwell published “On Governors” (Maxwell 1868), which gave a mathematical treatment to the relationship between feedback and stability. In 1927, Bell Labs’ Harold Stephen Black applied negative feedback to amplifier design. Without feedback, equivalent fidelity required complex designs and layers of compensation. At the time, the concept of using the output as an input went against established beliefs – the patent wasn’t granted until 1939.

During World War II, work on auto-aiming servo-mechanisms, for radar antennas and gun mounts, led to an even greater understanding of the role of feedback in complex adaptive systems. The experience of targeting in systems with significant lag, and ideas learned in the earlier efforts, formed the basis of a branch of Systems Theory called Cybernetics. After the war, MIT’s Servomechanisms Laboratory became the Laboratory for Information and Decision Systems.

Much of the work on feedback in learning derives from John Dewey. Dewey believed that learning and action were interdependent. 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 Dewey – Schön did his dissertation on Dewey.

5.2 Theoretical Foundation

Systems are modeled, and improved, through the design of sensing and feedback loops. A more formal understanding of feedback is found in control systems theory, which addresses system stability, or the lack thereof, and the affects of various kinds of feedback. In the design of systems, the system may under-correct – leading to continuous degradation, or over-correct – leading to instability and possible loss of control. The response may also exhibit distortion and lag, which effects quality and efficiency, and can lead to chaos.

In their work on learning organizations, Argyris and Schön (Argyris and Schön 1976) describe two kinds of learning: single loop learning and double loop learning. 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.”

5.3 Strategy and Variation

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 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.

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.

In *Effective Project Management*, (Wysocki 2009) describes the feedback loops in different project lifecycle models. All of the lifecycle models have stages of scoping, planning, launching, monitoring and controlling, and closing – the five process groups in Version 3 of the Project Management Body of Knowledge (PMBOK). The traditional linear lifecycle model ignores feedback, except between adjacent stages. The PMBOK, itself, mentions iteration only once, as a possible response to risk, with no discussion. In the Wysocki's *Incremental* model, the stages from launch to close repeat in cycles, allowing for adjustments in execution. 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” (p.321).

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.

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 proposed by Steward (Steward 1981), applied at NASA (Rogers 1997), and advocated in the Harvard Business Review (Eppinger 2001).

Toyota uses Feedback in its Set Based Concurrent Engineering approach to reducing risk while promoting innovation. In a time-boxed SBCE cycle, 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.

(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.

5.4 Implementation

Adaptation occurs in a cycle. The process moves, makes some progress, and 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 practice to improve feedback is to create short, coordinated cycles, called iterations. At the end of an iteration, current work is completed, assessments are performed, and changes can be made.

A major impediment to change is work-in-process. Work that is underway is not in a known state. It is hard to engineer a change on work which state is unknown. For tasks in process, a change in plans makes their original begin and end points invalid. The result can be confusion and 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 achieves this goal.

Shortening the test-and-fix cycle should be a priority. 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. This cost is especially evident in queues for test and rework.

In our own study of a traditional project, with batched testing and a queue waiting for rework, more than 40% of new tasks were built on code that already had identifiable defects. We also found a positive and significant relationship between lag time and subsequent effort. Moreover, since defects naturally occur in the most error-prone modules, the repair work is itself likely to cause defects. In short, there is a larger multiplier for the lag time in repair cycles, than for lag time in new work. (Van Hilst, et al 2011).

5.5 Risks

Development projects must have structure. Feedback must be addressed within a structure. Simply allowing every entity in a process to respond to circumstances as they occur leads to chaos (see Plan).

In development processes feedback is always a good thing. Most of the risks associated with the use of feedback are not specific to the feedback, per se, but, rather, concern the flow of information within the organization, and the ability of the organization to respond.

In organizations with hierarchical management and strict oversight, information must flow up the hierarchy to the level at which decisions are made, and then flow back down to the point at which they are implemented. This process can cause substantial distortions and lag. On the other hand, when feedback is applied directly at the source, information about a correction or change may not be disseminated to every level that needs to be informed. Having a good process to store and communicate information matters.

Applying feedback implies variations in or deviation from previously established plans. Making adjustments on a short time frame may prove difficult for workers with limited and narrow training or skills. Having good people, and raising the skill level of the people you have, is important.

5.6 Therefore

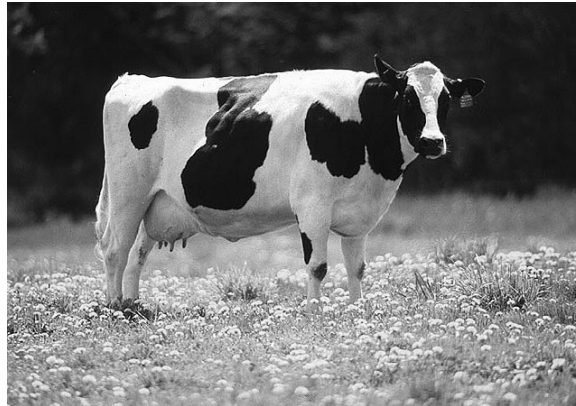
Improve the ability of the process to sense and use new information as it becomes available, and to sense and respond to new situations and problems as they occur.

For processes, and process improvement, feedback serves two roles, control and learning. For control, feedback keeps the process on track to better meet goals and objectives, even as conditions, or the goals themselves, change. For learning, feedback facilitates learning and improvement based on current experience.



Feedback should be part of a well structure process (see Plans). Many good practices already exist to support Feedback (see Best Practices). Sensing and response times can be shortened by applying Flow. Process can be affected by many factors (see Systems Thinking). Feedback is an essential mechanism for a viable Living System.

6. SYSTEMS THINKING



Wikimedia photo by Keith Weller

... you are not sure what is affecting your quality. Even with plans and policies in place mistakes are being made. You wonder if there are things you could still do better.



When the obvious solutions aren't enough, view the problem from multiple perspectives. Consider every factor.

The performance of processes and people are affected by many factors, including the organization's culture, external events, 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.

6.1 History

Systems Thinking evolved in the late 1940's from a number of thinkers in different fields who all shared a common dissatisfaction with the ability of mechanistic models to explain complex systems. Predictions were poor and traditional explanations left gaps, particularly in systems involving humans and other living things. Reoccurring problems were left to "chance" or "human error." Among these thinkers, Ludwig von Bertalanffy, an Austrian biologist, is considered the father of Systems Theory. Biological systems respond to many factors, interact with their environment, and self-regulate. Von Bertalanffy saw parallels in many fields - biology, economics, psychology, even physics - and described them in "An Outline of General System Theory" (von Bertalanffy 1950).

In management, W. Edwards Deming, a statistician, applied Systems Thinking to improving quality in business processes. Deming had worked at the U.S. Department of Agriculture from 1927 to 1939. It is notable that Deming and von Bertalanffy both started out studying plants and animals. 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 has achieved tremendous growth in productivity. In contrast, Taylor, Gilbreth, and Gantt (See Plans, Best Practices, and Flow) were all mechanical engineers.

In World War II, attention to contributing factors in human-designed systems lead to the field of Human Factors. During the war, half of all aircraft losses occurred in non-combat accidents. Such accidents were usually blamed on pilot error. Yet, by taking a systems approach and carefully studying the relationship among cockpit design, pilot behavior, and other factors, the rate of pilot error was reduced. Similar work in Britain on the operator use of radar lead to the field of Operations Research. Deming viewed all worker error as ultimately the responsibility of management for creating the conditions under which errors occur.

At MIT Systems Thinking evolved with the work of Jay Forrester and his Systems Dynamics group at the Sloan School of Management. Forrester started out simulating organizational behavior in an effort to understand what made successful businesses grow. "This modeling of corporate growth moved system dynamics out of physical variables like inventory into much more subtle considerations" (Forrester 1989). The dominant variables turned out to be the subtle influences on management decision making.

Deming applied the ideas of Walter Shewhart, a physicist at Bell Labs, including Control Charts and the notions of common and special cause variation. Shewhart was concerned with the reliability of remote and difficult to reach pieces of the phone network. He needed to predict when they would fail, so they could be replaced. But the same tools could yield insights into the causes of failure. A similar parallel can be found in the literature on process metrics for software development. Most of the literature concerns predicting completion dates and ignores the possibility of making things better (Van Hilst, et al 2011).

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 Six Sigma as an alternative to “test and fix” for eliminating defects. Like Deming, their approach tracks variations in “intermediate” factors and seeks to identify the “root causes” contributing to defects. The movement in process improvement away from test-and-fix has revolutionized manufacturing and created today’s higher expectations of quality. The term Six Sigma refers to a level of variability beyond which improvement is not cost-effective.

6.2 Theoretical Foundation

Systems Thinking involves a shift from traditional mechanistic, cause-effect thinking to a holistic view of interrelated contributing factors. Traditional thinking narrows the perspective to seek the simplest explanation (sometime termed reductionist). Systems Thinking broadens the perspective to bring in more factors including the environment. Traditional explanations assume a controlled environment. Systems Thinking assumes an open environment.

Systems Thinking is associated with the following five principles: openness, purposefulness, multidimensionality, emergent, and counter-intuitiveness (Gharajedaghi 1999).

- *Openness*: An open system constantly interacts with its environment. Openness contrasts with a closed system view, where the system only receives input at the beginning and produces an output at the end.
- *Purposefulness*: Systems thinking considers purpose and views the system itself as being responsive to its purpose. The notion of systems having a purpose actually dates to Aristotle, who included the purpose, or “final cause,” as one of the four types of causes needed to understand the world. Mechanistic thinking focuses on Aristotle’s other three causes, “material,” “formal,” and “efficient.” Much of the world Aristotle explained was biological.
- *Multidimensional*: All of the factors that influence an outcome cannot be understood from a single perspective. To find significant factors, Systems Thinking applies explanations in multiple dimensions. Related concepts are *Equifinality*, multiple paths to achieve the same goal, and *Multifinality*, multiple objectives applying to the same input.
- *Emergent*: In systems thinking, important properties of the system emerge from the interactions among its parts. Such properties cannot be understood by viewing the parts in isolation. Emergent properties are what Buckminster Fuller called “synergy”.
- *Counter-intuitive*: Systems often exhibit behavior that is unexpected or even the opposite of what was expected. Systems thinking accepts that systems themselves can respond and adapt, as can other systems in their environment. Interventions in complex systems may have second and third order effects that would not be seen without a holistic view, and are often not as intended.

In traditional thinking, causality is deterministic. A single counter example invalidates a causal link. In systems thinking, causality is non-deterministic. Counter examples are expected. But even with degrees of significance, improvements can be achieved.

6.3 Strategy and Variation

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.

In the following example of a university computing center, Gerald Weinberg traced a problem of knowledge transfer and help desk overloading to the movement of a soda machine (Weinberg 1971, p.47). 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 learned the solutions to common problems. The students did not congregate by the machines in their new location. When they had problems, they were now more likely to seek advice from the help desk. The role of the vending machine placement was determined through taking 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 waste in Flow, or be found through better Feedback. It does however, share features in common with Coplien's Hallway Chatter and The Watercooler patterns (Coplien 2005).

Nancy Leveson applies Systems Thinking and multi-level analysis in her work on systems safety and the factors that contribute to failure (Leveson 2004). In her studies of failures at NASA she showed that technical and operator failures reflected a broader system of contributing factors at all levels. Technical errors were preceded by changes in management policy and other changes in institutional behavior. Had any one of the earlier factors been addressed, the outcome would have been different. Her message is that addressing the problem of safety requires a complete culture of safety. The message is similar to Deming's culture of quality, and NSA standards for a culture of security. Security takes more than having a secure network and good locks.

(Li et al. 2007) describe a process "to guide analysts to locate potential causes for different kinds of variations, and effectively correct them in real projects." They used standard Control, Pareto, and Scatter charts of time and throughput to analyze "factors involved in project and specifying impacts between these factors". When applied to a software development team rated at CMMI maturity level 4, they identified a bottleneck caused by intermediate batching in the team's test confirmation practice. The problem was a problem of Flow. But they found it by tracking variance and tracing its source through Systems Thinking.

6.4 Implementation

In Systems Thinking, 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 may be 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.

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 Fractional Factorial experiments and the Taguchi method. When collecting data is hard, simulations can 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.

The application of Systems Thinking to process control often involves some form of Shewhart Control Charts. A Control Chart tracks the variance of a measured parameter. The metric 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 part that is brittle or was badly changed (analogous to wearing out), a new employee who was not properly trained, or workers or managers adopting different practices.

Ishikawa Fishbone Diagrams are used to qualitatively identify and group potentially contributing factors from all sources and levels. In an Ishikawa diagram, the issue or event is positioned at the head. Branches, are shown as ribs off the spine, represent contributing factors from different sources, such as technology, workers, management, regulators, customers, competition, etc. These branches are, in turn, decorated with further lists of detail and sub-branches. As an example, contributing factors to a recent deep sea oil well disaster included limitations of technology, lax regulators, and cost-cutting pressure from management. Sources of a problem should either be corrected, or measures put in place to reduce their impact. An important observation is that no factor is viewed in isolation – effects may take a combination.

6.5 Risks

Efforts to apply methods like Six Sigma can lead to an over-reliance on statistics, and/or be applied without having an actual problem to investigate. Before starting an investigation, the problem should be identified and alternative methods of investigation considered.

Collecting the metrics to be tracked can exact an excessive overhead. Workers and customers may be asked to spend valuable time filling in reports or answering questionnaires. Questions about levels of satisfaction are vague. After the fact self reporting of usage and time is also notoriously unreliable. Metrics are more effective if they are derived from objective events and collected non-invasively (Van Hilst and Huang 2011).

The people who are intimately involved in the process, or responsible for it, may not have the proper perspective to assess the culture or environment. Things they see every day may be taken for granted. It sometimes takes an outside consultant to see details in the practices, culture, and environment worthy of investigation.

6.6 Therefore

Apply Systems Thinking.

Systems Thinking views processes in terms of the interrelationships among factors, subsystems, and results. Multi-perspective analysis, and careful attention to detail, will identify the significant factors that can negatively or positively contribute to a desired result. Systems analysis is inherently situated and focuses on the process in place and its immediate context.



The pattern of Systems Thinking is often applied together with the patterns of Living System.

7. LIVING SYSTEM



Photo by Stefan Wernli

... the situation is dynamic. Problems and opportunities seem to crop up everywhere and all the time. Management decisions and improvement initiatives cannot fix every situation as more and more issues arise.



Cultivate a self-organizing organization.

Organizations modeled after mechanical systems assume that processes are predictable, and are limited by the laws of physics – notably, the law of entropy. Much of what happens in a development process is not predictable. Moreover, order in mechanistic systems inevitably breaks down and requires

new command inputs to be restored. In contrast, Living Systems have negative entropy. They create order on their own. Living System organizations are goal-seeking, self-organizing, and self-optimizing.

In a Living System organization, the people are seen as goal-seeking agents, acting individually or in groups, and capable of altering the process to better achieve their goals. Vesting the goals of the process in its actors leverages this capability. Improvements in a Living System align the goals of individuals with those of the process, empower them to achieve those goals, remove obstacles to their success, and protect them from external forces.

In a Living System organization, its greatest assets are its people. They are the ones in most direct contact with the work being done. By nurturing a culture that self-optimizes and solves problems on its own, management can focus on the more important job of setting a course, planning the future, and seeking new and greater opportunity.

Many proponents of Systems Thinking also apply Living System. It is in some ways an extension of Systems Thinking. Both patterns are rooted in experience and observations in biology and agriculture, and both concern emergent properties and non-deterministic effects. Here we have chosen to separate the former's focus on multi-factor analysis, from the latter's focus on autonomous action and goal-seeking behavior.

7.1 History

Since as far back as Aristotle, people have observed that living systems are different from mechanical systems. Of the four causes of Aristotle's Physics, the fourth cause was telos, meaning a purpose or goal. The other three were material, formal, and efficient. But even in Aristotle's time, the idea that there were causes beyond material and efficient was controversial (Falcon 2011).

With the mind/body dualism of (Descarte 1641), the idea of goals as causes became even more alienated from main-stream science. Since goals and mental processes are not physical and cannot be observed, they are not considered valid subjects of scientific speculation, let alone empirical research (see History and Risks of Best Practices).

In the early Twentieth Century, there was a widespread belief that everything could be explained in terms of machine models. Movements away from this belief developed and gained momentum around the experiences of World War II. Some of these trends are documented in the history section of Systems Theory. In 1945, Erwin Schrödinger observed the apparent negative entropy of living organisms (Schrödinger 1945). Today, models of economic systems and behavior commonly include goals and expectations, and view people as autonomous goal-seeking agents.

7.2 Theoretical Foundation

A process viewed as a machine is subject to the Laws of Physics – most notably, entropy. Since development is a knowledge process, we can express the 2nd Law of Thermodynamics in terms of information. Over time, without new order being imposed, the uncertainty of the information in the process increases towards randomness (Shannon 1951).

In the classical mechanistic view of management, control is needed to create order and minimize its dissipation. Predictability is also a problem. The result is a management process designed to “minimize the idiosyncrasies of human behavior” (Bartlett and Ghoshal 1994). Improvement initiatives are needed periodically simply to restore order.

In contrast, Living Systems exhibit negative entropy, also known as negentropy or syntropy (Schrödinger 1945). Organisms naturally create order. When you plant a seed in the dirt, the seed breaks down materials from the air and soil to produce a tree. The same model can be applied to organizations. “A sustainable system has all the essential characteristics of an organism - an irreducible whole that develops, maintains and reproduces, or renews, itself by mobilizing material and energy captured from the environment” (Ho 1999).

7.3 Strategy and Variation

A Living System is goal oriented. To function, the goals of the process must be shared by the entities and individuals of which it is composed. Since external forces and other stakeholders are also capable of affecting the process, their goals must also be considered. An analysis of the process must ask, who benefits? In Forrester's simulations of successful organizations, the forces affecting goal seeking behavior in the organization played a greater role in assuring success than the mechanisms employed (See Systems Thinking).

The HP Way embodies the belief that if you nurture your people and treat them well, they will do well for the company. It required few layers of management and was all about maintaining the culture. Following those principles enabled Bill Packard and Dave Hewlett to build a great company that for many years was known for quality and integrity. The company stumbled in the 2000's when, in the pursuit of short-term profit, it separated the interests of the company (and its management) from those of its people. During that period, the number of management layers substantially increased.

The Agile Manifesto's emphasis on people over processes is an expression of Living System. Two of its principles expand on this view. "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done." "The best architectures, requirements, and designs emerge from self-organizing teams." The name "manifesto" obliquely refers to an earlier manifesto whose authors were concerned by the alienation of workers' self actualization from the means of production.

(Ward 2007) applies Living System in the context of Lean process development. "Order emerges from interactions inside certain kinds of systems, and most systems are not predictable. The lean manager's job is to continuously help order emerge by learning and helping others to learn." The book proposes a series of measures that assign clear responsibilities for results (rather than process following) "allowing (indeed, requiring) developers to decide case by case on the most efficient means available to create knowledge."

Interestingly, Christopher Alexander's latest work, *The Nature of Order*, addresses how the built environment results from, supports, and exhibits properties of living systems (Alexander 2002).

7.4 Implementation

In Living System, the people and the process share the same goals. The people are responsible for achieving those goals. Thus the goals of the people and the goals of the process must be aligned. Improvement should include appropriate incentives, and removing disincentives.

A Living System organization depends on the skill and ability of each individual actor. Coaching, mentoring, and education are all means of improving organizations and processes by improving the abilities of individuals. Other forms of empowerment, such as improving communications and access to information, can also be effective. The dissemination and widespread availability of knowledge through Patterns training and the use of Patterns achieves this kind of improvement.

Some Patterns for organizations are better for creating a Living System than others (see Best Practices). Many of the Patterns in Coplien's book on organizational patterns present specific practices for creating and maintaining organizations that function as a Living System.

7.5 Risks

Freeing individuals of external control does not lead inevitably to an effective organization. A Living System must be nurtured and cultivated. The greatest danger is posed by actors, internal and external, pursuing self-interests that are not in line with those of the organization or of other individuals involved. All levels of the organization must be committed to nurturing, maintaining, and protecting a sustainable Living System culture (see Systems Theory).

7.6 Therefore

Nurture a culture that shares goals, values people, and builds on the abilities and interests of everyone involved.

Living System views the organization in terms of the interactions of goal-seeking individuals acting alone and in groups, to achieve desired goals. An effective organization aligns the goals of the process and those of individual actors. Living System organizations are then self-organizing, and self-optimizing.



A Living System cannot be applied as a pattern by itself. It is the culmination of a well organized process that has effective Plans, functions as a smooth continuous Flow, effectively uses Feedback, and applies Systems Thinking. A Living System process incorporates all of these patterns to achieve success.

8. DISCUSSION

The relationship among the 6 patterns is shown in Figure 1. 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. Best Practices 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. Feedback draws attention to those parts of the plan that affect adaptation and agility. Incorporating feedback in a plan improves the plan by making it more responsive to the situation, and to changes in its own performance. Flow improves the efficiency of an existing plan by reducing waste. Systems Thinking focuses on those factors that affect the optimal performance of a plan. By addressing factors that support or distort performance, including those that are external, the plan is more likely to realize its full potential. Systems Thinking supports Flow in the same way. Flow strengthens Feedback by reducing delays in sensing and response. Living System extends Systems Thinking to make the process self-aware and able to organize responses on its own. The performance of Living System is improved by Flow and Feedback in the same way that they improve the performance of Plan.

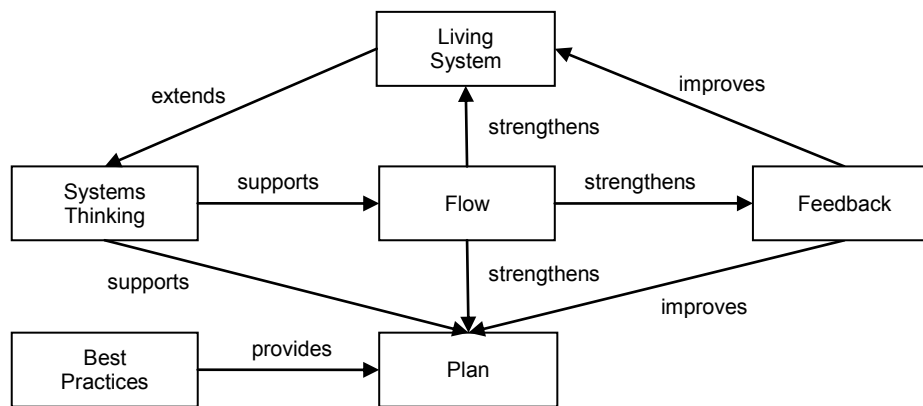


Figure 1. Relationships among the 6 Patterns.

Some differences can be found with respect to issues and biases. Time can be addressed in all six, but is central only to Flow and Feedback. Complex issues like quality, safety, and security that require a comprehensive solution are better addressed in Systems Thinking and Best Practices. Consistency is central to Plan, Best Practices, and Flow, but not central to Feedback and Living System. Innovation is, in some ways, the opposite of consistency. “Thinking outside the box” is more likely in Feedback and Living System, but has also been associated with Flow and Systems Thinking – which can address impediments to innovation. Plan, Flow and Systems Thinking require a good understanding of process details and involve situated analysis. A more hands-off approach will favor Best Practices and Living System.

Perhaps the clearest distinctions among the 6 patterns can be found in their treatment of defects. Plan 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. Best Practices treats defects as a subject in need of corresponding practices. Since documented solutions are plans, they are most likely to follow the Plan model. The Flow pattern treats defects as a problem that is integral to every step of the process. In the work-in-process metric, Flow 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 Systems Thinking and Living System patterns treat defects as a factor itself with causes and effects. They seek to identify root cause and other contributing factors. Systems Thinking and Living System also consider direct and indirect downstream costs involved in the choice of a solution. Finally, the Feedback and Living System patterns treat defects as part of a learning loop where the goal is to improve sensing and shorten cycle time for detection and repair.

9. CONCLUSION

The six patterns of process improvement are not mutually exclusive. Different patterns have different strengths in different situations, or even in the same situation. In the final analysis, any improvement is an improvement. But multiple improvements add up and make for an even bigger improvement. The best

examples of process improvement apply these patterns in combination. The worst examples often occur when one theory is used, and others are excluded. When considering an initiative for process improvement, you should ask yourself, "If everything worked exactly as planned, how good would that be? Is that good enough?" Our experience has been that following a single path or methodology for process improvement is not sufficient. We hope that the insights presented here will enable organizations to reap benefits from several, if not all methods of process improvement, and not be constrained by any one camp or methodology.

ACKNOWLEDGEMENTS

The authors wish to thank the shepherds and reviewers for their considerable help, including Jason Yip, Ralph Johnson, and Christian Köppe. They also wish to thank Lew Creary and Fitz-Gerald Cushnie for help on earlier drafts.

REFERENCES

- Alexander, C. 2002. *The Nature of Order: An Essay on the Art of Building and the Nature of the Universe*. CES Publishing
- Ambler, S. 2010. Context counts, http://www.semat.org/pub/Main/WorkshopPositions/SEMAT_position_Ambler.doc, accessed December 15, 2010
- Bach, J. 2005. No best practices, <http://www.satisfice.com/blog/archives/27>, accessed December 15, 2010.
- Bartlett, C. A. and Ghoshal, S. 1994. Changing the role of top management: Beyond strategy to purpose, *Harvard Business Review*, 72(6), 79-88.
- Comte, A. 1907. *A General View of Positivism*. London: Routledge and Sons.
- Coplien, J.O. 2007. Organizational patterns: A key for Agile software development. INCOSE talk. May 28, 2007
- Coplien, J.O. and Harrison, N.B. 2005. *Organizational Patterns of Agile Software Development*. Pearson Education.
- Descartes, R. 1641. Meditations on first philosophy, in *The Philosophical Writings of René Descartes*, trans. by J. Cottingham, R. Stoothoff and D. Murdoch, Cambridge: Cambridge University Press, 1984, vol. 2, pp. 1-62.
- Doerscher, T. 2008. Notes from ITIL foundation training. Planview blog, <http://blogs.planview.com/tdoerscher/2008/03/notes-from-til.html>. Accessed February 24, 2011.
- Falcon, A. 2011. Aristotle on Causality, in the Stanford Encyclopedia of Philosophy, revised January 25, 2011. <http://plato.stanford.edu/entries/aristotle-causality/>, Accessed February 24, 2011.
- Fayol, H. 1916. *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle*, Paris, H. Dunod et E. Pinat
- Forrester, J. 1989. The beginning of system dynamics. Invited talk at the international meeting of the System Dynamics Society, Stuttgart, Germany, July 13, 1989. <http://sysdyn.clexchange.org/sdep/papers/D-4165-1.pdf>, accessed December 20, 2010.
- Gharajedaghi, J. 1999. *Systems Thinking: Managing Chaos and Complexity: A Platform for Designing Business Architecture*. Butterworth-Heinemann.
- Gantt, H.L. 1919. *Organizing for Work*. New York: Harcourt, Brace and Howe.
- Gerrits, H. 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.
- Gilbreth, F.B. 1914. *Primer of Scientific Management*, Second Edition. New York: Van Nostrand.
- Ho, M.-W. 1999. Are economic systems like organisms?, Institute for Science in Society (ISIS). <http://www.i-sis.org.uk/hannove.php>. Accessed February 26, 2011
- Hunziger, J. 2006. Why Standard Work is not standard; Training within industry provides an answer. *Target*, Association for Manufacturing Excellence, (22)4, p. 7-13.
- Jacobson, I., Meyer, B. and Soley, R. 2010. Software engineering method and theory – a vision statement. <http://www.semat.org/pub/Main/WebHome/SEMAT-vision.pdf>, Accessed December 15, 2010.
- Jarvi, A., Hakonen, H. and Makila, T. 2007. Developer driven approach to situational method engineering. Chapter in *Situational Method Engineering: Fundamentals and Experiences*, J. Ralyte, S. Brinkkemper, and R. Henderson-Sellers (Eds.), IFIP International Federation for Information Processing, Volume 244, Boston, Springer, pp. 94-99.
- Kling, A. 2010. Do C.D.O.'s have social value? New York Times, April 27, 2010.
- Koskela, L. and Howell, G.A. 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.
- Leveson, N. 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.
- Li, J., Jiang, N., Li, M., Wang, Q. and Yang, Y. 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.
- Mill, J.S. 1865. *August Comte and Positivism*, Project Gutenberg Ebook, <http://manybooks.net/support/m/millj/millj1683316833-8pdfLRG.pdf>
- Ouellette, F. 2007. The babel tower of process improvement initiatives. Keynote address at the International IT Quality Conference, Toronto, Canada, October 1-5, 2007
- Pankhurst, A. 2007. *Planning and Periodisation*. Leeds, UK, The National Coaching Foundation.
- Patterson, M.L. 1992. *Accelerating Innovation: Improving the Process of Product Development*. New York, Van Nostrand Rheinart.
- Péaire, C. 2007. A roadmap to method development. <http://www.ibm.com/developerworks/rational/library/feb07/peraire/>, accessed February 24, 2011
- Poppendieck, M. and Poppendieck, T. 2007. *Implementing Lean Software Development: From Concept to Cash*. Upper Saddle River, Addison-Wesley.
- Project Management Institute 2008. *A Guide to the Project Management Body of Knowledge*, Fourth Edition.

Pries-Heje, J. and Johansen, J., Eds. 2010. SPI Manifesto, *European System & Software Process Improvement and Innovation* (EuroSPI). http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf, accessed February 24, 2011.

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

SAP 2009. SAP Services: Tools, technologies, and methodologies. <http://www.sap.com/services/bysubject/servsuptech/index.epx>, accessed March 12, 2010

Schön, D.A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books.

Schrödinger, E. 1945. *What is life?* New York: Macmillan.

SEI 2010. CMMI: Capability Maturity Model Integration (CMMI) Overview, <http://www.sei.cmu.edu/cmmi/>, accessed December 15, 2010.

Shannon, C.E. 1951. Prediction and entropy of printed English, *The Bell System Technical Journal*, 30:50-64, January 1951

Sharp, A. and McDermott, P. 2001. *Workflow Modeling: Tools for Process Improvement and Application Development*, Boston, Artech House Publishers.

Simon, H.A. 1956. Rational choice and the structure of the environment. *Psychological Review*, Vol. 63 No. 2, 129-138.

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

Taylor, F.W. 1914. *The Principles of Scientific Management*. New York: Harper.

Van Hilst, M., Huang, S. and Lindsay, H. 2011. Process analysis of a waterfall project using repository data. *International Journal of Computers and Applications*, (33)1, ACTA Press.

von Bertalanffy, L. 1950. An Outline of General System Theory. *British Journal of the Philosophy of Science*, 1(2): 134-165. doi: 10.1093/bjps/1.2.134.

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

Weinberg, G. 1971. *The Psychology of Computer Programming*. New York: Van Nostrand Reinhold.

Wysocki, R.K. 2009. *Effective Project Management: Traditional, Agile, Extreme*, Fifth Edition. Indianapolis, Wiley.