

Decision-making in product development: Are you outside-in or inside-out?

(Star Decision Science Working Paper #2)

Published in *Management Decision*, vol. 46, no. 9, 2008.

© 2008 Emerald Group Publishing Limited

ABSTRACT

We explore the relationship between decision-making and business systems by contrasting the decision-making in two well-known systems: MRP and JIT. MRP is a push system; JIT is a pull system. MRP seeks perfection; JIT seeks consistency. MRP gives decision makers great discretion; JIT constrains decisions. We propose that these contrasting qualities, and others, arise from two general approaches to decision-making: Outside-In and Inside-Out. While both approaches have merit, the Inside-Out approach plays an important role in business systems. As the difficulty of decisions increase, relative to a decision maker's ability, the cost of mistakes becomes significant. The Inside-Out approach addresses these situations by constraining decision-making, and as a result, the cost of errors decreases. In contrast, the Outside-In approach embraces complexity, which can cause more mistakes. In product development, the Lean and Agile systems exploit the Inside-Out approach. They constrain decisions and reduce the cost of errors that arise from two sources. Lean addresses interactions, which add complexity to business systems. Agile addresses unpredictability, which adds uncertainty to business systems.

Keywords: Decision-making, product development, MRP, JIT, business systems, agile, lean, Heiner

Gary J. Summers, Ph.D.
CEO, Star Decision Science
8 Pine Drive
Port Washington, NY 11050
gary.summers@StarDS.com
516-570-2017

Christopher M. Scherpereel, Ph.D.
Associate Professor of Management
Northern Arizona University
The W.A. Franke College of Business
PO Box 15066, Flagstaff, AZ 86011-5066
chris.scherpereel@nau.edu
928-523-7831

Decision-making in product development: Are you outside-in or inside-out?

Introduction

Business systems strongly affect results, but the various systems view businesses differently. Consider several business systems used in product development (PD). A commonly used PD system called Stage-Gate envisions businesses as a sequence of work organized around decision points.¹ Both Agile and Lean PD systems see businesses as a collection of practices and principles. The theory of constraints views businesses as resources and tasks organized around bottlenecks. System dynamics models view businesses as nonlinear systems of feedback loops. Benchmarking and best practices view businesses as a collection of practices. Is a business system a flow of objects, a sequence of tasks and decisions, a collection of practices, a set of principles, or some combination of all these things?

The PD business system creates ideas and embodies those ideas in a product or service that provides a customer with value. This undertaking requires both creativity and discipline, and combining these two requirements makes PD difficult. Being creative without discipline and being disciplined without creativity is much easier than being creative and disciplined, simultaneously. In PD, creativity and discipline must work together to achieve a goal. They answer the question, “How can we ...?” In this context, creativity and discipline are qualities of decision-making. The essence of PD is not found in a process, flow, bottleneck, feedback loop, principle, or practice; the central element of the PD system is people making decisions with creativity and discipline.

We wish to learn how decision-making affects product development systems, but where should we start? To develop insight we contrast two common manufacturing systems: Just in Time (JIT) and Materials Requirement Planning (MRP), which grew into Manufacturing Resource Planning (MRP II). Obviously, these are not PD systems, but they have been studied and compared far more extensively than PD systems. For this reason, these manufacturing systems provide a fertile opportunity to learn about decision-making and its effects on business systems in general. Based on these insights, we show how two different approaches to decision-making affect business systems. We name the approaches Outside-In and Inside-Out. Having learned how these decision-making approaches affect business systems, we analyze three popular PD systems.

¹ Stage-Gate is a registered trademark of R. G. Cooper & Associates Consultants, Inc., a member company of the Product Development Institute.

Decision-making in MRP and JIT

To identify relationships between decision-making and business systems, we compare two well-known manufacturing systems: Material Requirements Planning (MRP) and Just In Time (JIT). We chose these systems because (1) they solve the same problem in vastly different ways and (2) each system has been thoroughly studied. The quality of prior knowledge about the systems facilitates our comparison. In contrast, much less is known about PD systems, which are comparatively newer. While presenting our comparison, we describe only enough about JIT and MRP to support our insights. Readers desiring a full introduction to these systems can read a good textbook on manufacturing systems, for example: Hopp and Spearman (2008); Vollmann, Berry, and Whybark (1988); Browne, Harhen and Shivnan (1988); or Ohno (1988).

MRP and JIT have been compared many times from a manufacturing perspective (Krajewski et al., 1987; Lee, 1989; Rees et al., 1989; Sale and Inman, 2003; Sarker and Fitzsimmons, 1989; Spearman and Zazanis, 1992; Steele et al., 1995). We compare the systems' impact on decision-making. We do not consider specific decision-making techniques, but instead, how the systems impose fundamental orientations on decision makers. Through the comparison, we identify seven key qualities where MRP and JIT influence decision-making differently.

A briefest introduction to MRP

Suppose a company makes four types of chairs: types A, B, C, and D. In a particular month, customers order 480 of A, 640 of B, 320 of C, and 800 of D. MRP recognizes that, *given* these customer orders, the parts, machining, assemblies, and, in general, manufacturing operations, are known. With this knowledge, MRP considers the entire problem of producing all 2,240 chairs optimally. To achieve this goal, MRP uses a scheduling algorithm that includes enough variables, relationships, and information to explicitly model production.² For MRP, the best plan is one that completes all orders on time while minimizing costs. Costs are minimized by grouping the manufacturing of parts into lots of various sizes. Large lots reduce costs by gaining economies of scale, while small lots reduce costs by reducing inventory in the factory. MRP strives to make the best trade-off. After the plan is created, MRP sends the plan to the factory where it is implemented. This approach separates the decisions that control the factory from the factory itself. This is why MRP is called a push system. Production and the movement of inventory respond to signals (the plan) that come from outside of the production system and are pushed onto the system.

MRP systems do not implement their plans perfectly. Problems arise from unpredictable events, such as machine breakdowns. Problems also arise from flaws in MRP's algorithm. MRP does not consider capacity constraints, essentially assuming that capacity is infinite. For this reason, its plans can create bottlenecks. Additionally, MRP does not

² The model focuses on parts rather than machines. For this reason, MRP's model looks and is different than what people usually think of when modeling production: a discrete event simulation.

consider important factory conditions, like queue lengths, so some of its planning assumptions do not match reality. Of course, the assumption that demand is a *given* is violated when customers change their orders or forecasts are wrong.

How does MRP mitigate these problems? First, it tries to prevent them by anticipating problems. For example, better forecasting of demand will create a schedule that more closely matches actual orders. Better forecasting of long run capacity can prevent factories from bumping into capacity constraints. Second, MRP can insulate itself from problems by building robust schedules. By using appropriate lot sizing rules, it can create a schedule that works reasonably well when things go wrong. Finally, MRP responds to problems by adjusting its schedule to the current state of the factory and demand. This approach creates a curious phenomenon when the deviations are produced by MRP's flaws, such as ignoring capacity constraints. Rather than fix its flaws, MRP incorporates their symptoms into its revised plans—the tail is wagging the dog.

To obtain and improve these abilities, MRP was expanded to include additional software modules, and the total package was called MRP II. MRP II made some adjustments to MRP's algorithm, but mostly the additional software augmented MRP.³ Additional software improved demand forecasts, so MRP's assumption that demand is given would be more accurate. MRP II included software for long-range capacity planning, so MRP's assumption of infinite capacity would be less harmful. Supplementary software compared MRP's schedules to factory capacity to catch and fix problems before plans were implemented. Still other software managed the factory floor, so that MRP's plans were easier to implement. In all, MRP II models many more variables and relationships, especially variables from MRP's environment, such as capacity, demand, and shop floor management.

A briefest introduction to JIT

JIT uses a markedly different approach. Rather than consider all the orders at once, JIT breaks the large problem in a smaller one. Again, illustrating with our chair example, if there are 20 working days in a month, a factory can fill the orders by producing 24 of A, 32 of B, 16 of C, and 40 of D, each day. If there are 8 hours in each work day, the company can achieve a day's allotment by finishing a batch consisting of 3 of A, 4 of B, 2 of C, and 5 of D each hour. JIT organizes a factory so that this unit of production exits the assembly line each hour, with the goal of achieving this result consistently, eight hours a day, for twenty days.

Unlike MRP, JIT does not control production with plans. It builds control into the production system itself. Its famous Kanban system provides decentralized, local control that coordinates production, workstation to workstation. With Kanban, the same workers who build the products manage the flow of work and inventory. Decisions are made

³ MRP was fixed in a manner analogous to “fixing” the mirror on the Hubble telescope. Rather than replace the flawed mirror, a lens was added to compensate for the mirror's flaws.

locally by applying decision rules, and since workers must control the system while making products, the rules are simple.

Like MRP's plans, many problems can prevent JIT from producing the hourly unit of production with consistency. Machine breakdowns, long setup times, complex routings, complex product designs, difficult product assemblies, unreliable supplies of raw materials, and fluctuations in demand are some of these problems. To address the problems, JIT emphasizes quality control. Quality is even more important than throughput. Machines are built with the capacity to identify problems, and upon identifying a problem, to halt production. Workers are trained in quality control, and when problems are found, workers have authority to stop production. To help identify even more problems, JIT successively reduces the inventory in the factory. With less inventory, problems cause greater disruption, so they are more easily noticed. Once problems are found, they are diagnosed and fixed. The fixes modify the production system, and often the fixes simplify production. Fixes include reducing setup times, simplifying workflows, simplifying product designs, standardizing product components, simplifying product assemblies, extending the Kanban control system to suppliers, and negotiating with customers to reduce volatility in demand.

How MRP and JIT affect decision-making

MRP and JIT address the production problem with remarkably different strategies. For the current planning period, MRP seeks the optimal solution, although the implementation is imperfect. By breaking production into hourly units, JIT accepts a suboptimal solution, but it seeks perfect implementation. This is our first distinction regarding decision-making in MRP and JIT.

Distinction 1: MRP aims for perfection; JIT strives for consistency.

The next distinctions focus on system control. In a push system, work is triggered by signals that come from outside of the system: a plan. A push control mechanism plans the behavior of a system. MRP is a push system. It plans the behavior of the factory and pushes the plan into the factory. In a pull system, work is triggered from the behavior of the system itself. A pull control mechanism responds to the behavior of a system. JIT is a pull system; a workstation does no work until the queue in front of the next (downstream) workstation becomes short enough. Additionally, JIT manages system development with a pull mechanism, because it responds to the system's problems. Push vs. pull provides distinctions 2 and 3.

Distinction 2: MRP controls the system from the outside; JIT controls the system from the inside.

Distinction 3: MRP organizes decision-making with planning and pushes those decisions into the system; JIT has the system show where and when decisions are needed.

The next two distinctions consider how decisions are made. MRP explicitly models production, and considers many variables and relationships. Meanwhile, JIT's Kanban system guides decisions with rules, and these rules sharply constrain decision-making. Essentially, workers react to the flow of kanban cards.

Distinction 4: MRP explicitly models decisions; JIT guides decisions with rules, tools and policies built into the system.

Distinction 5: MRP gives decision makers many "degrees of freedom;" JIT constrains decision-making.

Table 1: How MRP and JIT affect decision-making

Distinction	MRP	JIT
1	Seek perfection	Seek consistency
2	Control system from the outside	Control system from the inside
3	Plan and then push decisions into the system	Allow system to show where and when decisions are needed
4	Explicitly model decision situations	Guide decisions with rules, tools, and policies built into the system
5	Give decision makers many "degrees of freedom"	Constrain decision-making
6	Adjust to problems	Eliminate problems
7	Embrace complexity	Seek simplicity

MRP's and JIT's management of problems provides two additional distinctions. MRP's primary response to problems is adjustment. Deviations from the production plan are batched and adjustments are made weekly or even daily; complete remaking of the schedule is performed monthly or even weekly (Browne et al., 1988). MRP II tries to reduce the number of problems. It tries to anticipate problems by accurately forecasting demand and capacity, and it seeks robust schedules. Still, MRP II's primary method of managing problems is adjusting its schedule. In contrast, JIT halts production, diagnoses problems and fixes the problem. This is distinction 6:

Distinction 6: MRP adjusts to problems; JIT eliminates problems.

The development of MRP II is telling. MRP did not improve by changing the production system. Rather, it changed its planning process, and this change made planning more complex. In contrast, JIT sought improvement by changing the factory. These changes tended to simplify manufacturing. Products were made with fewer parts and from standardized parts. Assembly and routings were made simpler. Variability in demand was decreased. The different methods of seeking improvement comprise distinction 7.

Distinction 7: MRP embraces complexity; JIT seeks simplicity.

Outside-In and Inside-Out Approaches to Decision-making

Table 1 summarizes the seven ways that MRP and JIT affect decision-making in a production system. Do the same sets of qualities occur in other systems and in other business functions, such as PD? We suggest that they occur in other business systems, because we propose that the qualities arise from the way systems approach decision-making. Approach decision-making one way, and a system becomes more like MRP. Approach it another way, and the system becomes more like JIT. To understand how the seven distinctions arise, we classify the decisions affecting a business process into two types, and illustrate each, using examples from PD.

- Level 1 decisions constitute the work in the process. For example, in PD level 1 is PD itself: product definitions, concept testing, technical research, design, etc.
- Level 2 decisions create the environment in which level 1 decisions occur. For PD, level 2 decisions include project management, standardizing architecture, setting team structures and rules, distributing power between project leaders and function managers, and tools that guide product development work. An example of a simple tool is a check list that reminds workers of issues they must address.

Level 1 and 2 decisions work together. A decision can be made with great discretion at level 1 or be constrained by policies set at level 2. Consider PD. Should engineers develop the architecture for a product or adhere to an imposed standard? Should team members communicate freely, or with the aid of standard forms and charts? Should project teams address risk, case by case, or should they follow well-defined practices? While level 2 decisions affect the structure of a system, one should not confuse level 1 and level 2 with the hierarchical structure of business systems. The level 1 and level 2 distinctions apply to all decisions, made at any level of a hierarchy.

What determines how decisions are shared between level 1 and level 2? We propose that it is a system's approach to decision-making. For example, MRP and JIT illustrate different ways of approaching and organizing decision-making. They are examples of two general approaches: Outside-In and Inside-Out.

The Outside-In Approach to Decision-making in Business Systems

MRP follows an Outside-In approach to decision-making. The Outside-in approach seeks the best solution to a problem. To find the best solution it increases the range of choices for level 1 decisions and the ability to select from those choices. It accomplishes this by explicitly modeling the decision (distinction 4). With a model, decision makers consider many variables and relationships (distinction 7), so decision makers can consider many solutions (distinction 5). To model more variables and relationships, elements of the decision environment are often included in the environment, so the environment becomes an input to decision-making. This approach subordinates the environment (level 2) to decision-making (level 1).

If the Outside-In approach is used to control a system, additional distinctions emerge. Controlling the system by modeling it moves control out of the system (distinction 2), and the decisions (plans) are pushed into the system (distinction 3). If the system becomes out of control, the decision is adapted or remade for the new situation (distinction 6). When the Outside-In approach is applied to system control, it may or may not be applied to other level 1 decisions. We chose the name Outside-In to highlight its application to system control, but we believe its preference to increase the sophistication of decision-making is equally, if not more, important.

The Inside-Out Approach to Decision-making in Business Systems

JIT follows an Inside-Out approach to system design. The Inside-Out approach has two components. The first component is a pull system of development. The system shows where improvement is needed (distinction 3). As these problems are identified, the system strives to eliminate the problems (distinction 6) with the second component. The second component is its emphasis on level 2 decisions. For each problem, the approach asks, "How can we structure the decision-making environment to better guide (constrain) decision-making?" It then provides level 2 structure to constrain level 1 decisions (distinction 5). The Inside-Out approach emphasizes level 2 and subordinates level 1.

Because level 1 decisions are constrained, perfection is impossible. Instead, the approach aims to reduce problems; its goal is consistency (distinction 1). The constraints themselves are the rules, tools, and policies that guide and assist level 1 decisions (distinction 4), and these tools often simplify decision-making (distinction 7). We named the approach for its pull development mechanism, but for reasons we present in the next section, we believe emphasis on restricting choice is equally important.

The performance of Outside-In and Inside-Out Approaches

Until now, we have not suggested how the two approaches affect system performance or the situations for which each approach is suited. A new theory of decision-making under uncertainty, developed by economist Ronald Heiner provides some insight to these issues (Heiner, 1983). Heiner proposed the concept of comparing the difficulty of a problem with the competence of a decision maker. Suppose that one could measure these qualities on an interval scale. Heiner defines a competency-difficulty gap, called a C-D gap, as $C-D \text{ gap} = \max\{0, (\text{Difficulty} - \text{Competency})\}$. If the decision maker's competence equals the difficulty of the decisions, the decision maker can make perfect decisions. If the difficulty of the problem exceeds the decision maker's competency, the decision maker makes mistakes. As the C-D gap widens, the frequency and severity of mistakes increases as well. The cost of these errors can be significant, especially for a large C-D gap. When facing such situations, decision makers should adjust their behavior to limit the cost of decision errors. Heiner derives two theorems that show how decision makers should adjust behavior:

- Heiner (1988a, 1988c) proves that as the C-D gap gets bigger and mistakes occur more frequently, some choices (actions) are used inappropriately too frequently, compared to their correct uses. The opportunity to select these choices does more harm than good, so a decision maker should not consider these choices. As the C-D gap grows, the decision maker's choice set should get smaller. The same is true with signals from the environment. Considering both environmental signals and choices, as the C-D gap grows, successful decision-making becomes less like optimization and more like rule-based decision-making.
- Heiner (1988b) considers how a C-D gap > 0 affects a decision maker's ability to react to change. Compared to a decision maker with C-D gap = 0, when facing a C-D gap > 0 , a decision maker must respond more slowly and cautiously to change.

Considering Heiner's theorems, when the C-D gap > 0 decision makers must constrain behavior to prevent the cost of decision errors from harming performance.

Successful gamblers illustrate this relationship when playing blackjack. Blackjack players can select from many card-counting schemes. The most powerful schemes are complex, requiring the player to keep running totals of several types of cards. These schemes are mathematically (theoretically) superior to less-sophisticated schemes; they have greater expected values. Yet, anecdotal evidence shows that players perform much better with simple schemes such as hi-lo, which keeps only one running count. A member of MIT's famous card-counting team declares:

Hi-lo is not the most powerful or even most advantageous counting method. However, card-counting mistakes made in the casino are infinitely more disastrous than a less-than-perfect method of counting. My teammates and I felt comfortable with our ability to execute the hi-lo system flawlessly, and did not feel that the added

value of more advanced systems outweighed the potential risk of casino error. (Mezrich, 2002, p. 254)

The superior schemes are more difficult to use, increase the C-D gap that players face, and cause errors. The cost of these errors overwhelms the theoretical superiority of their greater expected values. By reducing the C-D gap and limiting errors, the “inferior” hi-lo scheme outperforms mathematically superior schemes.

Heiner’s theorems offer insight into the Outside-In and Inside-Out approaches to decision-making. Consider MRP and JIT. MRP strives for perfect solutions, but it suffers from errors in implementation (actual production deviates from the schedule). JIT accepts suboptimal solutions, but demands perfect implementation. Heiner’s theorems suggests that as problems become more complex and difficult, JIT will be superior. For these situations, reducing decision errors is more important than seeking optimal solutions. This reasoning suggests proposition 1, which addresses system design.

Proposition 1: Some level 1 decisions are best addressed via the Outside-In approach, but when the C-D gap is sufficiently large the Inside-Out approach is preferred.

System control is another difficult problem. As a system becomes more complex, an Outside-In approach to control requires a more complex model. The combination of a complex model for a complex situation is a prescription for mistakes, and it inspires proposition 2, about system control.

Proposition 2: As systems become complex, managing them becomes more difficult and reducing the cost of errors becomes more important. Complex systems will benefit from an Inside-Out approach to control, where local rules, tools, and policies respond to system events by constraining decisions and preventing mistakes.

Localized control is a coordination mechanism. It does not imply anything about hierarchical structure. Inside-Out and hierarchies are compatible, or conversely, hierarchical structures do not necessitate planning for a control mechanism.

The final proposition addresses system development. The Outside-In approach seeks development by building progressively more realistic models that address more subtleties and complexities of a system (for example, MRP developing into MRP II). Heiner’s theorems suggest that this approach has limitations. At some level of complexity, striving for realism becomes counterproductive. Conversely, an Inside-Out approach has two benefits. First, its pull approach helps managers identify the changes that truly improve the system. Second, the fixes often simplify the system, so development is not limited by complexity. These ideas suggest proposition 3.

Proposition 3: Good development follows an Inside-Out approach. It has a system identify it's own problems, so that managers know what to fix. It fixes problems by constraining decisions to prevent errors.

The Inside-Out Approach in Product Development

Having presented the concepts of Outside-In and Inside-Out approaches to decision-making, we return our focus to PD. Many companies have no formal PD system. These ad hoc systems give decision makers too much discretion, which causes mistakes. To prevent mistakes, some companies have decided to manage PD within a formal system. Three popular systems are Stage-Gate, Agile and Lean. To learn about PD systems, we analyze these systems with the Outside-In and Inside-Out concepts. In the process, we learn how the Inside-Out approaches to PD constrain behavior to manage complexity and unpredictability.

The most popular PD system is Stage-Gate.⁴ Stage-Gate standardizes the evaluation of product development projects. Rather than having the flexibility to evaluate projects whenever and however managers see fit, Stage-Gate sets required checkpoints and evaluation criteria. Furthermore, by organizing product development into stages, Stage-Gate makes the magnitude of an investment inversely proportional to the uncertainty associated with the investment. For example, at the early stages of a PD effort, uncertainty is typically high and the funding level is, therefore, low. As work and time reduce uncertainty, projects are either cancelled or they advance to subsequent stages, where investment increases. This feature gives Stage-Gate pipelines their characteristic funnel shape. This practice follows Heiner's theory. By constraining investment when uncertainty is high, the system lowers the cost of mistakes. When mistakes are less inexpensive, a company can fund more projects and thereby increase flexibility. Standardizing project evaluations and assigning funding based on a project's place in the system (pipeline) creates level 2 structure. They are part of an Inside-Out approach. However, Stage-Gate lacks the other component of the Inside-Out approach. It does not adjust these standards *and* develop additional level 2 structure in response to product development problems.

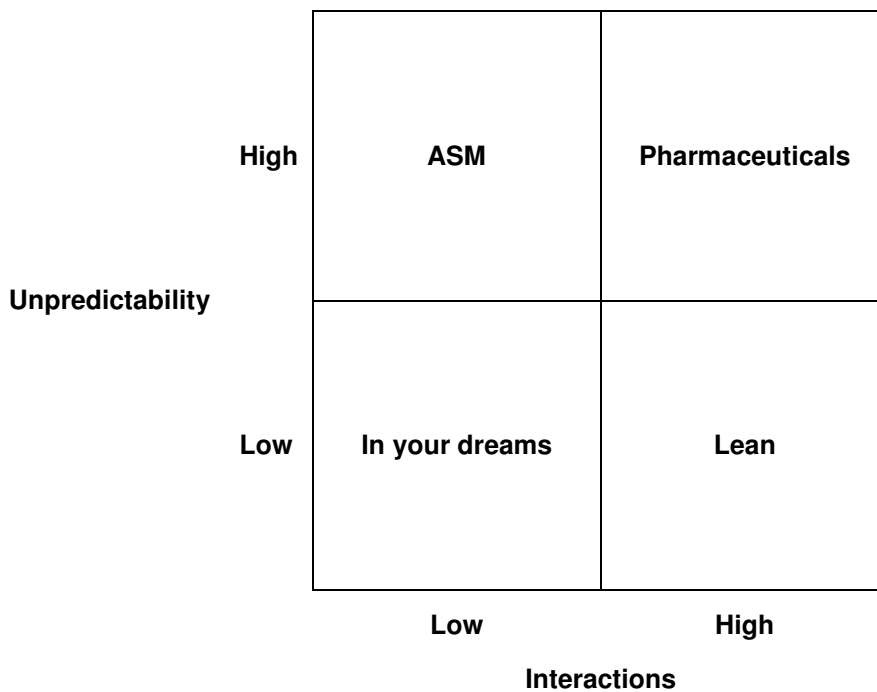
Standardizing project evaluation provides firm goals, but PD needs additional support. To fill this gap, many Stage-Gate systems use traditional project planning methods, which constitute an Outside-In approach. The combination of Stage-Gate with traditional planning methods is not necessary, and it may be problematic. Many mistakes are produced by PD itself, rather than from planning. The constraints are in the wrong place, leaving workers too much discretion and flexibility within each task. A typical solution is

⁴ Robert G. Cooper introduced the Stage-Gate approach in his book first published in the late 1980s and updated twice, titled *Winning at New Products* (New York: Basic Books, 2001). It remains the best source for information on the Stage-Gate process.

to plan with greater detail and demand allegiance to plan, but this only makes PD inflexible. This problem has motivated recent criticism of Stage-Gate systems.⁵

In contrast, the Inside-Out approach eliminates errors at the source by constraining (guiding) actual work. As previously describe, the Inside-Out approach controls mistakes by iteratively modifying tasks (actual work) with tools and rules. There are two strategies for controlling mistakes: eliminating them and mitigating their size and impact. In PD, a system has been developed for each strategy: Lean (Liker, 1998; Morgan and Liker, 2006) and Agile's Scrum Method (ASM) (Schwaber, 2004).

Figure 1: When to use ASM and Lean techniques



⁵ According to a brief article by Becker, B. (2006), ed., Management Roundtable Inc. Re-thinking the Stage-Gate process: A reply to the critics. available at: http://www.pd-advantage.com/images/RethinkingtheStage-Gate_Process_AReplytotheCritics.pdf (accessed July 3, 2007) Larry Keeley, president and co-founder of Doblin, Inc. created a controversy when he delivered a keynote address at The Management Roundtable's (MRT) Product Portfolio & Pipeline Management conference (April 24 and 25, 2006, Chicago) by stating "deeply researched, with more than \$7-million worth of data behind the assertion that the Stage-Gate model as we know it today is...going to guarantee total performance mediocrity for your businesses."

Lean improves PD by eliminating mistakes. Specifically, Lean reduces mistakes caused by interactions, which can arise in several ways. They can occur when one component affects another. Using automobile development as an example, changing the engine design may affect the performance of the transmission and drive train. Likewise, the modified engine may affect qualities of the entire car, such as vibration, noise, and manufacturability. Interactions can even arise from customer tastes. For example, customers might like limousines and red cars, but they do not like red limousines; thus, the style and color features interact. Lean PD reduces the errors arising from interactions by constraining PD decisions (level 1) with rules, tools and policies (level 2). Examples include standardizing product architectures, employing set-based design, placing the product vision with the chief engineer, using checklists, and standardizing communication practices with A3 reports.

While Lean manages interactions, it works best in predictable environments, as illustrated in Figure 1. To illustrate this quality, consider two practices. The first is the chief engineer's role in defining the product. As an authoritative book describes, "[The chief engineer] must understand what customers value ... Toyota's chief engineers and their staffs (CE team) go to great lengths to achieve this understanding." (Morgan and Liker, 2006, pp. 29-30) It is impossible to achieve this required understanding when the market is unpredictable. Examples abound: early computers, the first Xerox machines, and Sony's Walkman. For new-to-the-world technologies and products, a chief engineer cannot predict which opportunities will succeed or how each will succeed. Uncertainty cannot be resolved at the front-end, as required by Lean, but only during the actual development process and after the launch. The second practice is set-based design. Instead of designing each component independently, engineers first focus on the interactions among components. Deciding how the pieces fit together eliminates many possible designs from consideration, and once done, engineers can confidently consider the reduced set of remaining possibilities. The previously mentioned book describes this practice:

Toyota's process does not focus on the speedy completion of individual designs in isolation, but instead looks at how individual designs will interact within a system before the design is complete. In other words, they focus on system *compatibility* before design *completion*. [emphasis theirs] (Morgan and Liker, 2006, p. 50)

If unpredictable technological development or advancement changes the interactions, the effort to harmonize the components is wasted and the design process must be restarted.

ASM differs from Lean. ASM assumes that unpredictability makes mistakes inevitable, so rather than eliminating errors, it uses the second strategy for controlling mistakes: mitigating their size and impact. ASM constrains decision-making with level 2 rules that force short, quick development iterations. Each iteration comprises a few weeks and makes incremental modifications to the product. When products are too big to be considered in entirety, ASM simplifies development by ranking the features and components in order of importance. The iterative development process then works its way down the ordered list, building a few features or components at a time. Since each

iteration only makes small changes, the team can only make small mistakes (assuming no interactions—see below). Since the iterations are short and frequent, ASM can quickly discover and recover from mistakes. ASM's rule of producing a shippable product after each cycle improves the quality of feedback, so mistakes are easily identified. This strategy of reducing the cost of makes ASM flexible, a result implied by Heiner's theorems. Therefore, ASM excels in unpredictable environments.

Although ASM mitigates errors arising from unpredictability, it is weak where Lean is strong: managing interactions. ASM breaks big problems into small problems, but this decomposition is exactly what interactions prevent. When features and components interact, the ASM approach can cause excessive rework, as previous work is adjusted for interactions with the newer work. In a highly interactive system, rework would require many iterations, making an ASM approach inefficient and mistake-ridden.⁶ Even worse, interactions can bind a product to a suboptimal design. Interactions can create situations in which all possible small changes will decrease overall performance. Only large changes can break away from these situations and create opportunities for improvement. As Figure 1 illustrates, ASM is best suited for unpredictable situations, but only when interactions are few.

Figure 1 shows Lean and ASM applied to different situations. Unfortunately, most PD situations will have a mix of interactions and unpredictability, and managers may develop systems that apply Lean techniques where interactions are high and ASM techniques where unpredictability is great. This mixing of techniques may lead some people to claim that Lean and ASM are essentially the same, but although they both manage mistakes, they address different types of problems. ASM mitigates mistakes caused by unpredictability and Lean eliminates mistakes caused by interactions.

What about the remaining two quadrants of Figure 1? The lower left quadrant identifies a situation with predictable environments and low interactions. This is nirvana for PD and it occurs infrequently. The upper right quadrant identifies a situation with high uncertainty and high interactions. This is a dangerous situation. To remedy this situation, one can reduce uncertainty, reduce interactions or both. Lean and ASM techniques may facilitate this strategy. However, there are some cases where enacting this strategy is impossible. Some industries perpetually operate in the upper-right quadrant, and find solutions that are unique to their circumstance. One example is fashion. The fashion market is unpredictable and the features of fashion products are highly interactive. Designing a dress requires simultaneous consideration of shape, fabric, color, season, and the characteristics of the market segment. Change one of these product features, fabric for example, and the other features should change as well. The fashion industry solves this design problem by seeking individuals with genius. Thus, in the fashion industry, gifted designers design products.

⁶ These situations are addressed by considering interactions at the start of product development. A good software architecture is designed to minimize and coordinate the interactions. ASM is then applied within this framework. The approach is similar to Lean's techniques, where interactions are managed before components are designed. This solution is an example of applying Lean techniques to manage interactions, followed by Agile techniques to manage uncertainty.

Pharmaceutical companies face unpredictable technologies and create highly interactive products. Numerous technologies and scientific theories may offer cures for disease, but every one of them is shrouded in uncertainty. Additionally, pharmaceutical compounds are highly interactive. The compounds are made of molecules, and some molecules interact so strongly that changing even one can greatly alter the compound's properties. Facing both unpredictability and interactions, pharmaceutical companies have traditionally used Stage-Gate and project portfolio management systems. The Stage-Gate system effectively constrains investment when uncertainty is high and focuses investment as uncertainty is resolved. The portfolio management system implements strategy while managing resources and risk. Many project portfolio management practices use an Outside-In approach, and they may err too much when encountering sufficiently complex situations. Developing Inside-Out approaches to decision-making in project portfolio management is a needed area of research.

Conclusion

Decision-making is the core activity of many business systems, including PD. How does decision-making affect business systems? To gain insight, we compared MRP with JIT. The comparison revealed contrasting qualities of each system. We asked whether these sets of qualities were unique to MRP and JIT or consequences of a more fundamental phenomenon. We propose that there are consequences of a company's approach to decision-making. Specifically, they arise from Outside-In and Inside-Out approaches to decision-making.

The Outside-In approach to decision-making in systems emphasizes level 1 type decisions by seeking more powerful decision methods. This approach treats the decision environment as an input to the decision process, and in this way, it subordinates the environment to the decision. The Outside-In approach explicitly models decisions, seeks the best solutions, and pushes the solutions into a business process. This approach embraces complexity, gives decision makers many degrees of freedom, and adjusts to mistakes.

The Inside-Out approach to decision-making in systems uses a pull mechanism to identify problems. The problems are then resolved by modifying the system to constrain decisions. The approach constrains decisions with rules, tools, and policies that guide work. It emphasizes level 2 decisions and subordinates level 1 decisions. Often, the constraints simplify decision-making, and the goal is to reduce the frequency and severity of mistakes. The Inside-Out approach seeks consistency, but as a result, it foregoes optimization. The Inside-Out approach integrates control and development decisions into the system itself.

How do the two approaches perform? We propose that the difficulty of a decision situation strongly affects the performance. Specifically, Ronald Heiner showed that as the difficulty of a decision situation increases, relative to a decision-maker's ability, the cost

of mistakes becomes a significant problem. Adjusting behavior to limit this cost becomes a major goal and influence on decision-making. In these situations, the Inside-Out approach will have a strong advantage. Consistent with Heiner's theorems, Inside-Out constrains decision-making to limit the cost of errors. In contrast, the Outside-In approach embraces complexity, and when facing difficult problems, this approach should cause more decision errors. Based on Heiner's framework, we proposed propositions about system design, control, and development.

Finally, we analyzed common PD systems with the new concepts. Lean, Agile, and to a lesser extent, Stage-Gate, all constrain decision-making to reduce the cost of errors. However, Lean focuses on eliminating errors that arise from interactions. In production development many interactions arise within a product, between business functions and from communications. Agile focuses on reducing the size and severity of errors arising from unpredictability. Most PD projects would benefit by combining the techniques used in these two approaches. This analysis and the concepts of Outside-In and Inside-Out will help managers manage decision-making to create successful PD systems.

References

- Becker, B. (2006), ed., Management Roundtable Inc. Re-thinking the Stage-Gate process: A reply to the critics. available at: http://www.pd-advantage.com/images/RethinkingtheStage-Gate_Process_AReplytotheCritics.pdf (accessed July 3, 2007)
- Browne, J., Harhen, J. & Shivnan, J. (1988), *Production Management Systems: A CIM Perspective*, Addison Wesley Publishing Company, Reading, MA.
- Heiner, R.A. (1983), "The origin of predictable behavior", *American Economic Review*, Vol. 73 No. 4, pp. 560-95.
- Heiner, R.A. (1988a), "Imperfect decisions and routinized production: implications for evolutionary modeling and inertial technical change", IN Dosi, G., Freeman, C., Nelson, R., Silverberg, G. & Soete, L. (Eds.) *Technical Change and Economic Theory*. Pinter Publishers, London.
- Heiner, R.A. (1988b), "The necessity of delaying economic adjustment", *Economic Behavior and Organization*, Vol. 10 No. 3, pp. 255-86.
- Heiner, R.A. (1988c), "The necessity of imperfect decisions", *Journal of Economic Behavior and Organization*, Vol. 10, pp. 29-55.
- Hopp, W.J. & Spearman, M.L. (2008), *Factory Physics*, McGraw-Hill Irwin, New York.
- Krajewski, L.J., King, B.E., Ritzman, L.P. & Wong, D.S. (1987), "Kanban, MRP and shaping the manufacturing environment", *Management Science*, Vol. 33 No. 1, pp. 39-57.
- Lee, L.C. (1989), "A comparative study of the push and pull production system", *International Journal of Operations and Production Management*, Vol. 9 No. 4, pp. 5-18.
- Liker, J.K. (Ed.) (1998) *Becoming lean: Inside stories of U.S. Manufacturers*, Productivity Inc., Portland.
- Mezrich, B. (2002), *Bringing down the house*, The Free Press, New York.
- Morgan, J.M. & Liker, J.K. (2006), *The toyota product development system*, Productivity Press, New York.
- Ohno, T. (1988), *Toyota Production System*, Productivity Press, Cambridge, MA.
- Rees, L.P., Huang, P.Y. & Taylor, B.W. (1989), "A comparative analysis of an MRP lot-for-lot system and a Kanban system for a multistage production operation", *International Journal of Production Research*, Vol. 27 No. 8, pp. 1427-43.

- Sale, S.L. & Inman, R.A. (2003), "Survey-based comparison of performance and change in performance of firms using traditional manufacturing, JIT and TOC", *International Journal of Production Research*, Vol. 41 No. 4, pp. 829-44.
- Sarker, B.R. & Fitzsimmons, J.A. (1989), "The performance of push pull systems: A simulation and comparative study", *International Journal of Production Research*, Vol. 27 No. 10, pp. 1715-31.
- Schwaber, K. (2004), *Agile Project Management with Scrum*, Microsoft Press, Redmond, Washington.
- Spearman, M.L. & Zazanis, M.A. (1992), "Push and pull production systems: issues and comparisons", *Operations Research*, Vol. 40 No. 3, pp. 521-32.
- Steele, D.C., Berry, W.L. & Chapman, S.N. (1995), "Planning and control in multi-cell manufacturing", *Decision Sciences*, Vol. 26 No. 1, pp. 1-33.
- Vollmann, T., Berry, W. & Whybark, C. (1988), *Manufacturing Planning and Control*, Dow Jones-Irwin, Homewood, IL.