Lean Methods for Creative Development

How to rapidly deliver solutions and capture knowledge using lean techniques.

Tim Schipper and Ryan Schmidt

Steelcase, Inc. has been applying lean principles to its manufacturing facilities since 1996, with favorable results, including a dramatic shrinkage of its operations footprint. To build on its experience with the lean methodology, Steelcase decided to apply lean principles to other areas of the company. The initial applications of this effort were areas characterized by similarly repetitive and sequential processes (order processing, for example). In particular, we looked first at "order-to-cash" transactional areas — areas assigned to develop quotes, take orders, and pursue customer payments.

As we looked more closely at the waste involved in these order-to-cash processes, we realized that much of the waste in those areas was caused further upstream, in product development. This article describes our first steps in transitioning to a lean development process, the means to capture related knowledge, and examples of lean creative development at work.

In Brief

Steelcase, Inc. has been applying lean principles to its manufacturing facilities since 1996, with favorable results, including a dramatic shrinkage of its operations footprint. Building on its experience with application of lean methodology in manufacturing, Steelcase decided to apply lean principles to other areas of the company. While examining ways to eliminating excess cycle times in order processing and other "order-to-cash" transactional areas, they realized that much of the waste in those areas was caused further upstream, in product development. This article offers insights on their continuing transition to a lean design process, the means to capture related knowledge, and examples of lean creative development at work.

Product Development and Launch

Steelcase has long been known as an innovation leader in the office environments industry. The introduction of the Leap® chair in 2000, with its patented Live Back™ system, was widely acknowledged as a breakthrough in the industry. More recently, Steelcase introduced the Think® chair, which was the first product to receive McDonough Braungart Design Chemistry's (MBDC) new Cradle to Cradle™ Product

Certification. However, these kinds of accomplishments are too few and far between to satisfy the increasing demands of our customers and dealers. Steelcase recognized the need to shorten product development time without sacrificing quality in order to improve its customers' experiences.

If we step back and look at product development not just for Steelcase, but for most companies, we see some pretty disturbing trends. For example, a 1994 study by IBM for IT Application Development[1] reported: 1) 55 percent of projects go over budget, 2) 68 percent of projects exceed schedule due dates, and 3) 88 percent of projects require major redesign. Informal research at Steelcase and other firms revealed similar statistics.

A 2000 study by the National Center for Manufacturing Sciences (NCMS)[2] found that some companies, namely Toyota and Nippondenso, have dramatically better results. Some key findings in the study:

- Toyota takes half the time of its U.S. competitors in terms of time to market, with an average of 18 months and a goal of 12 months.
- Toyota consistently earns the highest quality ratings from J.D. Power and *Consumer Reports*.
- From a productivity standpoint, Toyota is nearly four times more productive than its U.S. competitors. As an example, Toyota assigned 150 product engineers to a car program versus 600 for Chrysler (and the Chrysler program took twice as long).
- Nippondenso has $15 billion in sales with 45,000 people versus Delphi with $30 billion in sales and 210,000 people (less than half the productivity of Nippondenso).[3]

Realizing that huge productivity gains were possible and knowing the nagging problems facing product development at Steelcase, we enlisted the help of an outside consultant, Jim Luckman (a member of the TWI network, an affiliation of practitioners), and set out to design and test a lean product development process.

Steelcase

Steelcase, a global leader in the office furniture industry, helps people have a better work experience by providing products, services, and insights into the ways people work. The company designs and manufactures architecture, furniture, and technology products. Founded in 1912 and headquartered in Grand Rapids, MI, Steelcase serves customers through a network of more than 800 independent dealers and approximately 13,000 employees worldwide.

Lean Product Development: First, the Quick, Iterative Proccess

Lean in product development has two basic components: 1) a quick, iterative process to achieve flow, and 2) a knowledge capture process to improve quality, reuse, and repeatability of information developed in the iterative cycles.

To develop and achieve a quick, iterative process similar to the process used at Toyota, we focused on these five techniques (based on Toyota's product development model):

1. Separate development from execution.
2. Manage the process with regular checkpoints within the iteration process.
3. Define the scope of each phase of iteration with a cross-functional meeting.
4. Draw the organization together to integrate learning for process governance.
5. Manage interrupts to keep the process flowing.

Figure 1 illustrates some key differences between the traditional and lean approaches to product development. The top illustration depicts a typical phase-gate approach. Concept development leads to a period of engineering refinement, which leads to prototyping and testing, which eventually leads to a launch. Traditional managers in product development encourage the design freeze as early as possible, stating, "If marketing would just get the requirements up front, I could get the design done much faster."

We observed the same pattern in Information Technology (IT) application development. IT traditionally attempts to gather all of the requirements up front (typ-

Traditional product development assumes that a batch process allows a team to execute well. We know from manufacturing that producing in large batches leads to excess inventory, quality problems, and frequent rework.

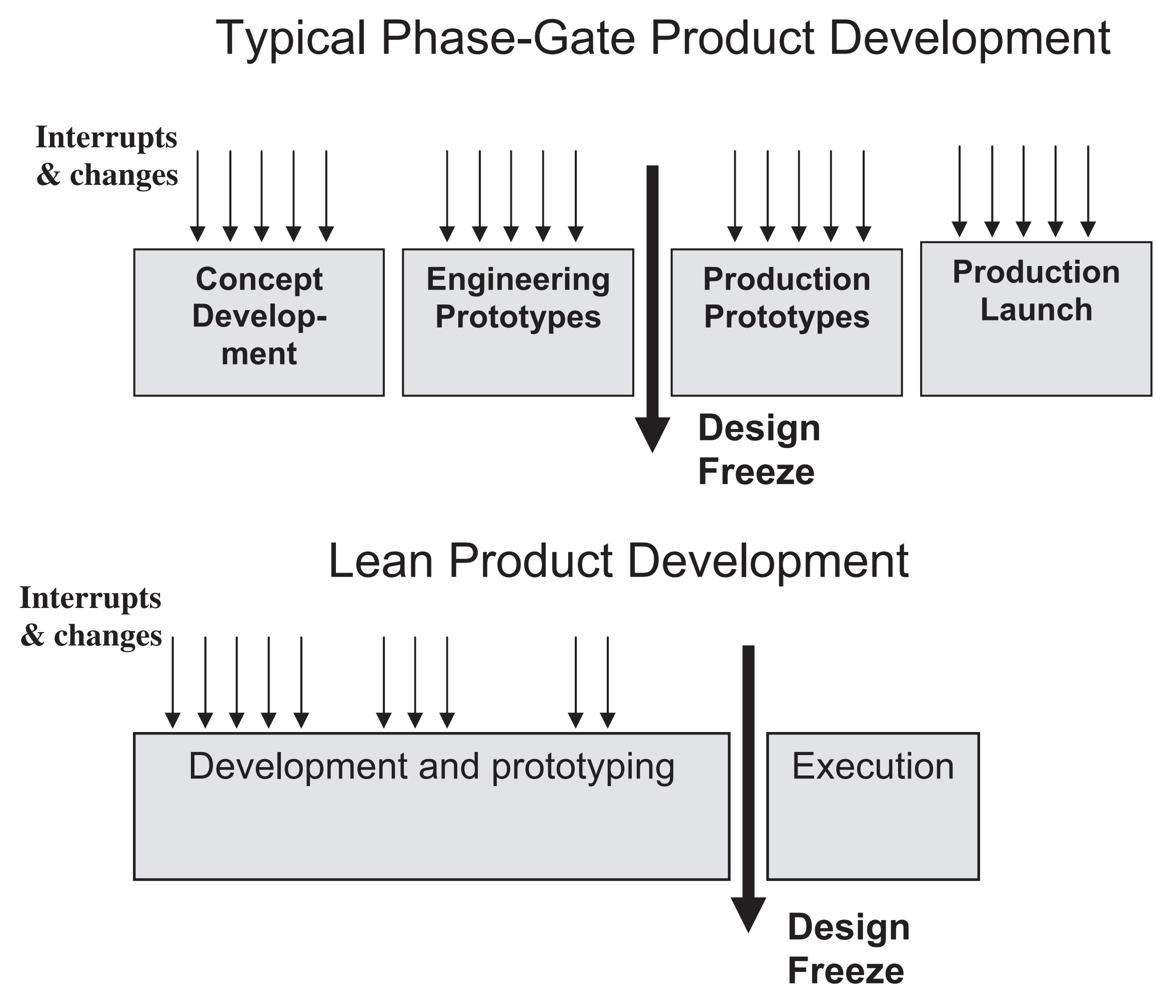

Figure 1. *Phase-gate product development versus lean learning cycles. Copyright Luckman Consulting. Inc.*

ically asking, "What do you want?" when the user is asking, "What can it do?"). Finally, IT will ask the user, "What do you think?" and the user inevitably say, "That's not what I wanted!" There was never enough customer interaction on an intermediate basis.

Traditional product development assumes that a batch process allows a team to execute well. We know from manufacturing that producing in large batches leads to excess inventory, quality problems, and frequent rework. In traditional product development, teams attempt to gather all requirements, design, and test in a batch, and then execute. However, poor quality of the initial requirements typically leads to a lot of rework. We documented many cases in which teams went through the entire design phase, only to find that the unit was too costly, or that it couldn't be manufactured; starting over wasted time and effort.

In addition, teams have constant interrupts and changes during the development process (marketing and user specifications change, manufacturing methods change, etc.). Therefore, we wanted to design and process that followed reality, instead of trying to ignore it. As Figure 2 shows, the requirements in a product development process become more and more detailed as the development process

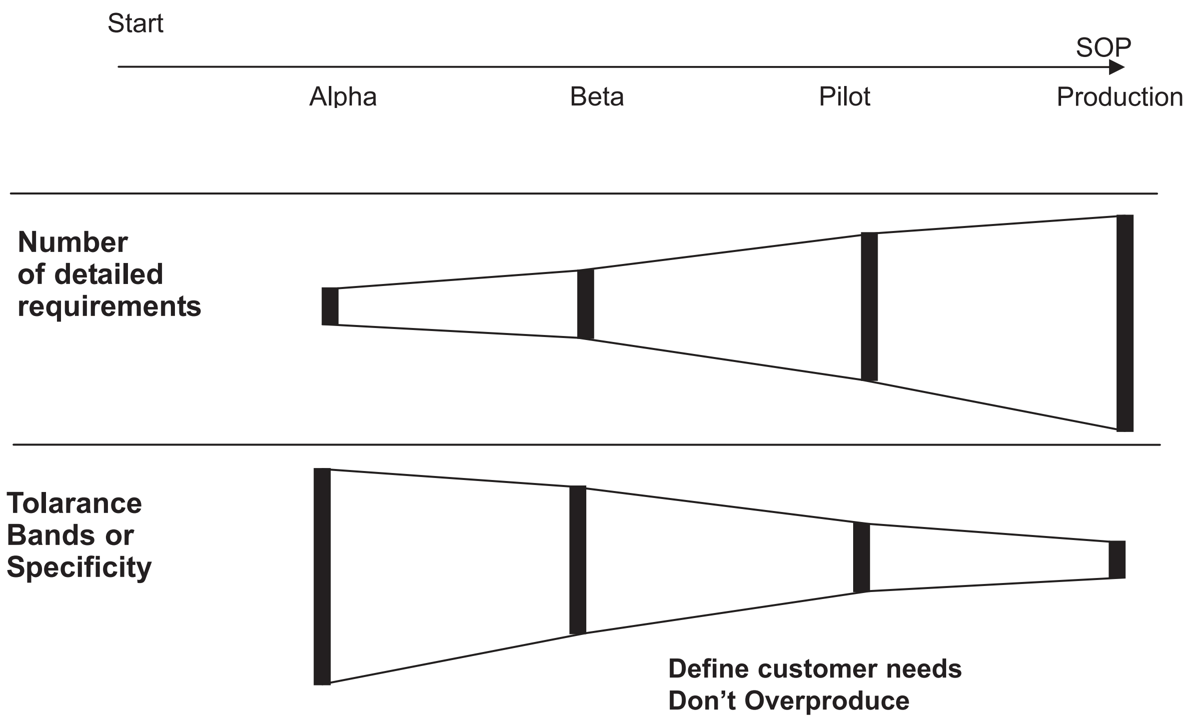

Figure 2. *Requirements become more complete as the launch date nears. Copyright Luckman Consulting, Inc.*

moves along, and the amount of variance (tolerance) decreases. Information becomes more complete as the development process progresses. Delaying the design freeze (versus trying to move it up), therefore, allows the team to execute flawlessly (less rework because of increased knowledge). Toyota, in fact, delays decisions until the latest possible point, setting final specifications until close to starting production.[4] Yet Toyota rarely has a major redesign after a vehicle launch.

Toyota utilizes quick, iterative cycles of learning to maximize the accumulation of knowledge about the product prior to launch. Figure 3 introduces quick, iterative cycles, which we refer to as "learning cycles." Learning cycles at Steelcase typically last two-four weeks. They comprise five basic steps:

1. *Plan* what the team needs to accomplish in a learning cycle.
2. *Design* a solution (done at the component or sub-system level).
3. *Build* up part or all of a solution.
4. *Test* part or all of a solution.
5. *Review* the knowledge gained and capture it in some form. The key is to synchronize learning across functions.

Learning cycles for each function can vary, depending on which questions need to be answered. Steelcase developed a project management structure to support and enhance learning cycles for improved development and knowledge capture. The remainder of this article focuses on how Steelcase has integrated these learning cycles into our product development process.

Planning: Dividing up Development Duties

The first step in Figure 3 defines the scope of the learning cycle by means of a cross-functional meeting. This is the "plan" portion of a learning cycle. In this example, the cross-functional team consists of players from product design, sample build, testing,

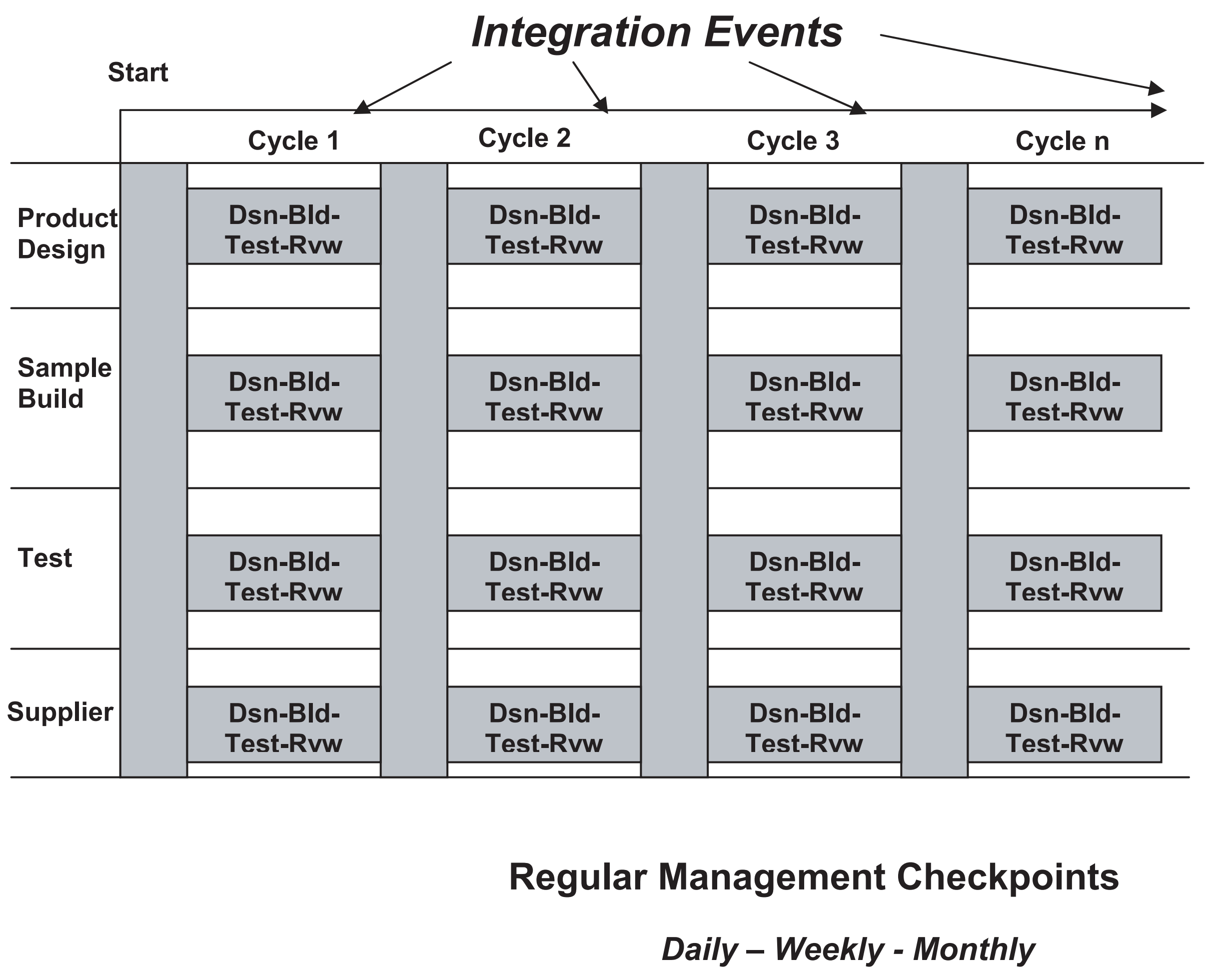

Figure 3. *The learning cycle concept. Copyright Luckman Consulting, Inc.*

and suppliers. Team composition varies from project to project. For example, in IT projects, the team ideally includes users in the up-front planning step, which is typically a one or two hour meeting. In this session, the team develops a small number of objectives for the next learning cycle and determines how long that cycle will be.

Typical learning cycles last about two-four weeks. However, they can be as brief as one day, depending on the questions to be answered and the knowledge to be gained in that project phase. After the team decides on objectives, they discuss what questions must be answered to achieve them. Owners (from the project team) are then assigned to objectives and questions. After the planning meeting, the owners of various questions list daily tasks needed to answer the questions. For example, if a team member was assigned to the question, "What is the maximum load that a shelf needs to hold?" then that question can be broken into a series of tasks — test competitor 1's product, test competitor 2's product, review building code laws, examine municipality requirements (for major cities such as Los Angeles), etc. With the plan complete, the team members go back to their functional areas for the duration of the learning cycle, but they still meet once or twice a week to check their progress. Visual controls are used for the check-in meetings to provide project status at a glance.

Visual Controls and Daily Accountability

The team uses what Steelcase calls an "accountability board" to track daily tasks (it can be done using Excel or Post-it notes).[5] The accountability board is the place where teams meet to discuss progress. A daily task is given a green or red dot depending on whether or not it was completed. Team members are held accountable for these daily tasks by their management.

Elements within Every Learning Cycle

After planning, each learning cycle also includes the following elements: design, build, test, and review results. Every group is expected to complete these steps, at some level, in every learning cycle. For example, if an engineer needs to determine if the shelf he is designing will hold the desired load, she or he will need to create a design (hand sketch, model, or full drawing set, depending on which learning cycle she is in), build that design (electronic or physical prototype), and test the design (using Finite Element Analysis [FEA], for example, or physically testing, or marketing validation). It is *crucial* that some aspect of prototyping and testing be used in *every* learning cycle to learn as quickly as possible, which aids in evaluating design options and shortening development time.

In IT, prototypes can be PowerPoint mock-ups or full-fledged working models in an offline environment. The key is to employ any tool or technique needed to increase the speed of learning.

Bringing It All Together with an Integration Event

The final portion of the learning cycle is to review the findings, make key decisions, and capture the related knowledge. This is done in what we call an integration event. At Toyota, the event would be led by the chief engineer. At Steelcase, the event is led by the project manager and attended by the entire team and often business leadership as well. Lasting about 90 minutes, integration events follow a standard format (to avoid events that extend for several hours). At the end of the event, the team often holds its planning session for the next learning cycle and the process starts all over again.

Changes to Project Management Techniques and Tools

Figure 3 showed a high-level illustration of how a team could achieve quick, iterative cycles in product development. Another concept appears at bottom right of Figure 3, the idea of having regular checkpoints for the development process. Conventional product development projects go through a tollgate process, with leadership review at the end of every phase. The typical gap between tollgates may be three to six months, so if leadership decided they did not like the project direction, several months' rework would be in store. Although a project manager typically fills out an extensive Microsoft Project plan noting all the tasks and related milestones, projects often miss their targets. If milestones were not hit, and the team proceeded anyway, a crunch would occur at the end of the development process, costing added person-hours and dollars.

In contrast, lean product development consists of daily or bi-weekly checkpoints where daily tasks are coded green or red to denote whether or not a task was accomplished. Instead of working from a Microsoft Project file, the team only generates a detailed plan for the current learning cycle — typically two to four weeks of tasks. It seems clear that there is too much variation in requirements to plan at a detailed level beyond that. The approach Steelcase takes is to estimate the number of learning cycles at the very beginning, knowing we will get better at planning as we complete more projects.

Another important difference between traditional and lean product development is that the traditional approach allows interrupts at any point during the process. For

Instead of working from a Microsoft Project file, the team only generates a detailed plan for the current learning cycle — typically two to four weeks of tasks. It seems clear that there is too much variation in requirements to plan at a detailed level beyond that.

example, Steelcase was developing a piece of furniture using sheet steel construction. Eight months into the project, a market manager decided they wanted wood, and the entire project went in a new direction. There was no validation with the rest of the organization or with customers, just engineers flying down a different path because of a lone comment.

Many such comments come during the product development process. They add confusion because no one person is fielding or prioritizing these types of requests. No validation is performed on many of these comments. At Toyota, the chief engineers have a great deal of knowledge about the market (having spent time there). They are responsible for fielding these requests and deciding whether or not they should enter into the development process. At Steelcase, the project manager now filters these requests and decides if and when they impact the development process; validation is done by consulting with other salespeople, asking that a request be validated by marketing, etc.

Although many interrupts may be encountered in the process, the project manager does all that she or he can to keep the interrupts from entering into a learning cycle. A learning cycle has already been scoped out, and we need to let the project team "focus and finish." Since learning cycles are short, there is minimal risk to waiting until the start of the next learning cycle. Interrupts are ideally held for discussion and evaluation at the next team integration event. Keep in mind that this approach is based on reality, so there may be times when interrupts enter the middle of a learning cycle. The key is to minimize these occurrences.

Concurrent Set-based Design

Using lean learning cycles in shorter time frames leads to faster development, but this is not the whole story. Another element involves creating multiple concepts early, and then "narrowing" those concepts based on objective evaluation. Toyota refers to these design concepts as sets or concurrent design sets. Moving multiple sets forward is called set-based concurrent design.[6] Each set has multiple concepts, and all the concepts are moved forward and evaluated against the requirements. As an example, in an engine design, multiple fuel system concepts would be examined simultaneously. In a different set, multiple piston designs would be evaluated. As development continues, the different concept ideas are refined and eventually eliminated until the best solution is reached.

While our products are not nearly as complex as an engine, we have been using the concept of set-based design to examine multiple concepts simultaneously. In one application for the health care market, multiple versions of a concept for a shelf label were examined simultaneously. In another product concept, 12 options were examined for a keyboard-mounted mouse surface. In the early learning cycles, the multiple concepts are all simple CAD (computer-aided design) models or even sketches. As the learning cycles progress, the concepts are compared against the requirements and steadily reduced and refined. An excellent technique for comparing design ideas is a decision matrix, similar to the methods used in Quality Function Deployment (QFD). The decision matrix allows requirements to be weighted, enabling comparison of products against one another. The matrix provides a more scientific and accurate approach than relying totally on informal judgment or intuition about which design best matches the requirements. The decision matrix also allows the concepts to be narrowed much more rapidly. It also may be worth your while to rate your competitor's product through the decision matrix alongside your own concept designs.

It may seem counter-intuitive that the process is faster by starting with more concepts. However, there are several ways that multiple concepts help to speed the process along. By applying the tools of value stream mapping (VSM) to several development projects, each map showed the project was delayed because the single design being pushed forward was stalled or even dropped when it missed a key requirement.

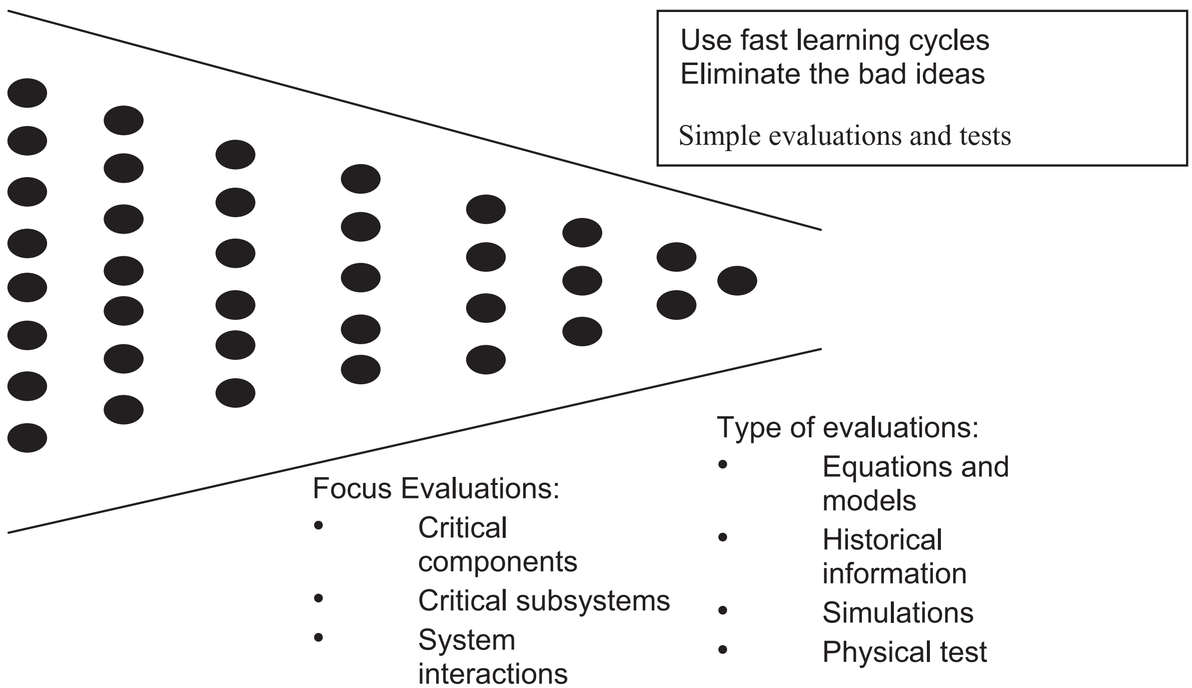

Figure 4. *Narrow the concepts. Copyright Luckman Consulting, Inc.*

The missed requirement could be any number of things: missing the right cost targets, not meeting the customer's quality expectation, not passing a critical test, etc.

The batch approach to development will inevitably cause delays in the project timeline as well as rework. The rework can be significant (as noted in earlier project examples).Rework on projects is measured in months, not days, and the true cost of the rework is never really fully documented or accounted for in the cost of the product. The rework, associated additional costs, and project delays can be eliminated by moving multiple concepts forward together. With each learning cycle, the concepts can be evaluated against key requirements of cost, customer expectations, and even confidence in meeting the test specifications. Some concepts are quickly eliminated, while others will merge into new or better concepts. The synergy achieved by moving multiple projects ahead in parallel provides greater efficiency and eliminates common development process pitfalls.

Capturing the Knowledge: Maximize the Flow

Since the development process is a discovery process, the knowledge generated in each learning cycle is a key deliverable. The knowledge must include both product insights and reflections on the process used to deliver the knowledge. If the product team does not capture the knowledge, they lose valuable leverage of the ideas and concepts for the next learning cycle or for the next development project. A lean development effort needs to combine quick, iterative learning cycles and knowledge capture. The lean principle is to maximize the flow in the creation of knowledge.

Discovering How to Manage a Discovery Process

If transactional processes (such as our "order-to-cash" lean projects) are examples of standard repeating processes, product development is the complete opposite.

… the application of lean is different in development because of the heuristic nature of the work. We call development a heuristic process because it calls for discovery and learning.

Generally speaking, nothing in the development process is repeatable day to day. Instead, the project team is attempting to create something new that has not existed before. The development process is a creative one. Since creativity and discovery are the key ingredients in the development of new products or systems, that means *knowledge* (rather than tasks, as emphasized in a conventional stage gate development process) is the main deliverable.

Therefore, the application of lean is different in development because of the heuristic nature of the work. We call development a *heuristic* process because it calls for discovery and learning. Webster defines heuristic as *"involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods"* So, lean applied to product development projects causes us to look for ways to enhance generation of knowledge and creativity. The engineer or designer is no longer just a player in the process. Since they are key knowledge producers, they become the hub or center. The supporting processes around the engineers must be changed to increase the pace of developing knowledge, and the knowledge gathered by the team must be captured.

Standardize Learning Cycles

Unless the process of knowledge capture is standardized, the knowledge from development is difficult to capture. It will exist in the brainpower of the engineering staff, and only fragments of that knowledge might be found in project notebooks. Head knowledge is transitory, fading with time or even lost as people move from job to job or leave the company. Project notebooks are rarely up to date, and once they are archived, the information is soon forgotten and difficult to retrieve. Even during the life of a project, information is not freely shared because everyone is so busy delivering their piece of the project and attempting to do it under heavy time pressures. By contrast, the standard capture of knowledge follows the lean principle of creating standardized work for the lean process.

Knowledge captured using a standardized method within the learning cycle has two uses: 1) for application in the immediate learning cycle and 2) for reference in future learning cycles. It is the only way that companies can ensure knowledge is available for current and future use in order to reuse that knowledge and speed development in the next cycle and next project.

At Steelcase, we have adopted a simple method to collect learning cycle knowledge. At the start of each cycle, a standard scoping document captures both the learning cycle objectives and related questions that need to be answered. The questions to be answered define the tasks to be completed during the learning cycle, as noted earlier in this article. At the end of each cycle, the knowledge is captured in the scoping document in the form of answers to the questions for that cycle. Questions in the scoping document are formulated by the team members (engineering, marketing, etc.) along with the project manager. Team members fill in a brief answer to the questions during the learning cycle and the project manager is charged with ensuring that all questions are answered by the time of the integration event. The process also calls for the team to reflect on and record any general team learning which might point them in new directions for discovery. The integration event at the end of the learning cycle, described earlier, brings together the multi-functional development team to collaboratively share and build in the knowledge developed in the cycle. It also sets up a framework for the next learning cycle.

Preserving Knowledge for Future Applications

To capture knowledge and pass it along to the next teams, Steelcase expanded on questions answered in the scope document and adopted a knowledge capture document. Prepared by the project manager and the project engineer, this document contains all the learning from the previous learning cycle. The knowledge capture document covers a wide range of information including: general learning, test results, competitive analysis, performance curves, cost targets, concept pictures, etc. The format is standardized and easy to read, offering a repeatable way for each team to capture knowledge. These files can be placed in a database or in an open text search environment to allow for later retrieval.

The knowledge capture sheets are recorded in a Microsoft PowerPoint format. If they are placed in a document management system, they can be searched and retrieved. Keywords can be added to the documents to help with later indexing and retrieval. The documents could be manually indexed using a simple index card system, but a full open text search engine with advanced search capabilities provides the best way to capture and later use these documents.

The knowledge capture document has a huge advantage over project notebooks, as it captures both team and individual knowledge, and it is done at more frequent intervals. The information on the knowledge capture sheet is concise, allowing for easier storage, retrieval, and reuse. Compare this approach to traditional development, where information is only gathered at end-of-project post mortems, or not at all.

Refinements to Requirements

While systematically narrowing multiple concepts is key to learning cycles, the method to collect requirements is equally important. As requirements evolve from general to more refined over the course of multiple learning cycles, the team goes through the creative discovery process. While this approach may seem too loose and open-ended, the team reacts positively to the approach because it allows them to start with general requirements and then move to more refined requirements as the project moves along. For instance, the team might initially focus on the cost target for the product, but later they will narrow the question to determine the optimal cost for each component. Likewise, the cost targets in the early cycles are very rough, say plus or minus 20 percent, while in later cycles the cost estimates are much more accurate.

Test requirements, market validation, aesthetic appearances, and other elements may evolve during a series of learning cycles. The goal of the requirement gathering process is to narrow the variation over time as more discoveries are made. Each discovery allows the team to ask more specific and detailed questions for the next learning cycle. The key lean principle here is to *not* over-produce, but to only produce the information needed at the time.

Reuse

A big part of knowledge capture is reuse. An often-told story is that while many domestic automakers have dozens of types of internal door latch systems, Toyota utilizes only a handful across its brands. This is a physical reuse of parts, but it is equally important to reuse the knowledge we capture as we go through product development.

Many people have good intentions of capturing the lessons learned from a product development project, but people get too busy to follow up. It is crucial to reuse knowledge gained regarding manufacturing processes, cost information, and testing to move through product development more quickly. Reuse of captured knowledge is a major factor in taking the time and risk out of product development programs.

Case Study: IT Application Development

The IT application development process is well suited to benefit from the use of lean tools. It is a discovery process, and at Steelcase we used a tollgate process similar to the one used in product development. The tollgate process is called the Corporate IT Development Methodology (CITDM), with tollgates for project initiation, requirements definition, development, build, testing, and post implementation.

Problems of rework and excessive waiting time typically plague the IT projects. Many problems we saw in IT affect most every large company: quality and timing problems at handoffs; misaligned activities, events, and milestones — QA (quality assurance) reviews, design reviews, tooling initiation, testing, etc.; information not being reused; functional goals competing with each other; and defining and re-defining requirements.

Our experience showed such issues to be a normal current state. In our lean workshop for IT application development, the project team looked at two different projects to benchmark the current state condition. The larger project was to place SAP functionality into a manufacturing plant. It had been hailed by many at Steelcase as successful, although it was delayed multiple times and contained about 80 percent rework. Approached traditionally with CITDM phases done in sequence, it was tracked through periodic management reviews. The project team spent months in the requirements phase up front, yet many of these requirements were inaccurate. The end users did not agree with many of the concepts and assumptions used to build the solution, leading to the extensive rework. The project was finally finished through much effort and heroics.

Enter the lean methodology for creative processes. After mapping the initial current state process, the IT development team mapped a future state map that called for quick, iterative learning cycles. Each learning cycle was to include scoping (planning), design, build, testing, and reviewing the results. This approach was used on two pilot projects. The most significant results were achieved by applying these concepts in the next SAP plan implementation — a dramatic improvement from the earlier SAP implementation.

In the scoping phase of this successful project, the team planned out the objectives and questions to be answered in the learning cycle. They also defined the resources such as offshore development resources, software assurance, and servers/networks needed to complete the cycle. The project team gathered with the users (including IT analysts) and outlined the requirements for this phase. They used a visual control board to map their daily tasks and to track which requirements they answered and which they did not.

Once planning was finished at the start of each cycle, the team did rapid requirements gathering with the users directly involved, designed the solution to meet the requirements, tested the solution, and then, with the customer, reviewed their results and captured the learning at the end of the cycle. These activity-packed cycles lasted only three to four weeks. The IT group agreed that a cycle should last no longer than 30 days. At the end of each cycle, an integration event was held with the complete team and management to review progress, capture learning, and share knowledge. The tendency of the team was to try to delay the integration events, in order to get down to the nth level of detail. However, the integration event date, modeled after Toyota's approach, could not be missed. This discipline established the pace needed to hit the launch date.

Once planning was finished at the start of each cycle, the team did rapid requirements gathering with the users directly involved, designed the solution to meet the requirements, tested the solution, and then, with the customer, reviewed their results and captured the learning at the end of the cycle.

The integration event was also used to manage project interrupts. In a traditional IT project, the team fields various interrupts from many sources and makes individual decisions based on their perceived importance. In the learning cycles, interrupts were captured on the visual control board as issues, but the team did not stop or change the tasks for the current learning cycle. Instead, the group decided at the integration events whether or not to allow the interrupt to become part of the project tasks and redirect team activities. Since integration events occurred every three to four weeks, the team did not have to worry about waiting too long before the interrupt could be assessed. Dramatic results were achieved on the launch of SAP in the new plant. The original time frame for the launch was ten months. Applying lean principles reduced the project by four months. Also, by narrowing requirements with the system users in each learning cycle, the IT analysts rapidly learned the system. As a result, the external SAP consultants were released eight months earlier than planned, directly affecting project cost and the bottom line. Another benefit of narrowing requirements with the users during the learning cycle was that the users also increased their knowledge of the system much earlier and the duration of training for plant users decreased. Overall, project delivery was ahead of schedule, under budget, and with greatly increased quality. These results were significantly different from IT projects using traditional approaches.

Case Study: Details Group, Document the Current State

A second test case for implementing lean product development at Steelcase occurred in the Details Group. This group makes workstation accessories and tools that increase user ease and efficiency. This group approached our internal consulting group because they had intense demands for reducing development time for new products.

Detail's current state was typical of many U.S. companies. In fact, it closely mirrored the initial results from IT launches described in the previous case study. Products often missed the launch date and usually required reengineering. Some of the problems the team relayed included: 1) expensive tool modifications needed because there was never a true design freeze, and 2) suppliers were given a request for quote (RFQ) and never had a chance to offer solutions — they were not involved from the process start or when major decisions were made.

At Steelcase, we use a three-day workshop to evaluate and improve product development. The project team maps the current state the first day and a future state the second day, and then they develop a 90-day implementation plan the final day. This methodology was introduced by the TWI network consultants and improved internally. We bring leadership in toward the end of the second and third days to confirm the team's direction.

The Details team decided to look at two recent engineering design projects to see what problems existed and where the biggest opportunities lay. On Day 1, they created a current state value stream map that was astounding, for the simple fact that no one had ever seen the product development process before. Creating a value stream map for product development is quite different compared to the ones we had created for manufacturing and even from the ones developed for administrative processes at Steelcase.

The product development value stream maps broke the process down into functional "swim lanes" and captured the interrupts that typically occurred in the process. In manufacturing, it is easy to see the process. Not so in product development, where you have: greater interdependency among functions; knowledge creation is the fundamental value that flows in a product development value stream; heuristic processes versus transactional processes; less material flow and greater information flow; much harder to see the value; usually few metrics; and many operations which are not defined or standardized.

When Details mapped their current state, they found several problems consistent with many other U.S. companies' experience:

- Lengthy product leadtimes
- Too many concurrent product development projects
- High amount of rework, often due to requirements changes, specification changes, and management changes that disrupt the process
- No way to objectively evaluate concepts
- Suppliers and manufacturing involved too late in the process (causing rework)
- Testing too late in the process (leads to rework)
- Loss of project knowledge when engineers leave
- Lack of prototyping and testing leads to later problems.

With the above problems in mind and a set of kaizen (improvement) VSM learnings, the team set out to create an improved future state. (We would caution against arbitrarily applying lean concepts without going through the process of documenting the value stream.)

Clear Goals for Details Future State

The Details team laid out very clear goals for approaching the targeted future state. They sought to: Double the number of

We involved suppliers much earlier in the process versus the traditional RFQ process.

products launched per year, introduce more innovative products, shrink time to market by 33 percent, hit launch dates 100 percent with 100 percent quality, reduce operating expense, and create better visibility on project status.

To accomplish these goals, the team had to think differently. They spent most of the second day of the workshop discussing key requirements for developing a future state. Most of these requirements are discussed in the iterative learning cycle and knowledge capture portions of this article. We projected that the concept narrowing process would greatly enhance the likelihood of achieving our ideal future state, and we developed a decision matrix to aid this effort. As noted earlier in this story, our plans included (among other things) the use of built-in learning cycles, and integration events.

Details Implementation

The Details team decided to try the new methodology on two different projects, code named Ozzie (after the Red Wings goaltender Chris Osgood) and Wallace (named for Pistons star Ben Wallace). The team realized that they needed to develop more detail on the future state (what would a decision matrix look like, what does an integration event look like, etc.).

Tim Schipper and Ryan Schmidt then facilitated another two-day working event in which the Details team refined their approach to accomplish the lean goals: Define categories of requirements and weighting factors for the decision matrix to rank and narrow concepts, determine what kind of engineering tools were needed to evaluate concepts early (3-D tolerance analysis, FEA, etc.) and train our engineers on the tools, determine what kind of marketing validation we could do by involving customers during the learning cycles, determine what kind of prototypes could be done for evaluation earlier in the development process, list roadblocks that could be eliminated for a speedier process (for example, our current state mapping revealed that engineers had not been allowed to contact suppliers unless a purchasing agent was present, causing countless delays in getting crucial information; the rule was modified once management realized its impact); and define intelligent and firm launch dates for Ozzie and Wallace.

Both projects are approaching "design freeze" as this article is being written, and so far the results have been positive. Although we cannot divulge specifics on the Ozzie and Wallace projects, below is a summary of the benefits (a faster, more agile, and less risky process) from using this lean product development process:

- We involved suppliers much earlier in the process versus the traditional RFQ process.
- We utilized external experts for much more of the development process than ever before. The team is also driving for a way to get experts in place faster.
- The team used innovative validation techniques to quickly narrow concepts to the best solution.
- We were able to get cost information months earlier than we would have in the traditional method of costing at toll-gates. Using supplier-experts allowed the team to develop costing estimates earlier, later helping them to avoid all the rework of completing a design that later is found to be too costly.
- The lean approach utilized a visual display of these two projects, helping them to track weekly progress. The display indicated all the critical information from the team and executives.
- We found a way to build leadership involvement in the process for better flow. As the team moved through a learning cycle, they captured process flow interrupters (these interrupters force the stalling or slowing down). When leadership took the issues on, they sent a strong message to the team that they have management support.
- The project manager successfully managed many of the interrupts that occurred during the process and held them for discussion until the learning cycle was complete.
- Marketing has developed a product brief that combines a marketing brief and a

design brief all in one. This product brief is a key input in the product development process — how we get "quality at the source."

More Work to Be Done

Steelcase is still in its infancy regarding the implementation of lean into the product development world. Engineers, as important and creative forces of the company, sometimes avoid following a process. What helps in this case is that the process is centered around engineers and they have a big hand in developing the process.

We are also moving from a command-and-control product development process to a leaner process focused on the engineer. We are trying to ensure that the engineer has the necessary tools for improving product development flow (analogous to the plant floor, where we aim to provide the operator with parts and tools needed for product flow at takt time). With continued attention to our product development process and with time, we hope to move from an environment where our engineers spend only 20 percent of their time on value-added (VA) activity to the Toyota model, where engineers are spending 80 percent of their time on VA activity.[7]

The work in product development will inevitably point to problems in other support areas (legal, finance, etc.). This visibility will lead to a need to implement lean in additional support areas to ensure product development can flow smoothly.

Tim Schipper has worked at Steelcase for 21 years and is a lean consultant. His prior experience includes managing the Engineering Systems group in the Information Technology department and developing new products in product engineering.

Ryan Schmidt has worked at Steelcase for six years and is a lean consultant. He previously worked in manufacturing engineering, supply chain management, product development, and plant supervision at Steelcase. Before working at Steelcase, he was a launch supervisor with DaimlerChrysler.

Footnotes

1. Middleton, Peter and James Sutton, *Lean Software Strategies*, Productivity Press, May 2005, Chapter 7, pp. 89, 90.
2. NCMS study, "Product Development Process – Methodology & Performance Measures Final Report," January 31, 2000.
3. NCMS study, "Product Development Process – Methodology & Performance Measures Final Report," January 31, 2000.
4. NCMS study, "Product Development Process – Methodology & Performance Measures Final Report," January 31, 2000.
5. Mann, David, *Creating a Lean Culture; Tools to Sustain Lean Conversions,* Productivity Press, 2005.
6. Like, Jeffrey, and Allen Ward, Dorward K. Sobek, "Toyota's Principles of Set-Base Concurrent Engineering," *Sloan Management Review*, Massachusetts Institute of Technology, Winter 1999, Vol. 40, Number 2.
7. NCMS study, "Product Development Process – Methodology & Performance Measures Final Report," January 31, 2000.