MIT Sloan Management Review

Should You Build Strategy Like You Build Software?

Magazine: Spring 2008 • Research Feature • April 01, 2008 • Reading Time: 17 min

Keith R. McFarland

The strategic planning model is due for a "new release," one that enables companies to keep pace with changing environments, quickly create and adapt strategy and empower people throughout the organization to make effective choices.

In many companies, the corporate planning department has gone the way of the dot-matrix printer. But if the analyst-driven, top-down, formal process of the previous era is dead, what have companies replaced it with? Not much, according to recent surveys of executives and managers of global companies. Many companies still hew to the decades-old annual strategic planning process, only now the responsibility for creating plans falls on group managers and department heads, with little central support. Other companies have jettisoned formal strategic planning altogether — centralizing strategy discussions to a few top executives or replacing planning with a hodgepodge of informal or semiformal retreats, executive leadership councils and board committees. Some observers have gone so far as to argue that strategy making is simply too uncertain and complex to be handled by a defined process. Unfortunately, senior managers may be throwing the proverbial baby out with the bath water. The fact that the process is failing doesn't mean that strategy making is resistant to a process approach.

Consider software development. Around the same time that managers were losing confidence in strategic planning, software development went through its own crisis, as the demand for faster design and integration of increasingly robust systems began to make the traditional "waterfall" approach to software development obsolete.(See "The 'Waterfall' for Software Development/Strategic Planning.") The crisis in

software prompted a few visionaries to rethink how software gets built. They didn't abandon a process approach to the problem; rather, they invented new development processes, such as rapid application development, extreme programming and agile software development, to confront the new realities.

The pressure to accelerate and at the same time cope with increased uncertainty and complexity led to the decline of traditional software development in some settings — and similar pressures seem to have played a pivotal role in the demise of formal strategic planning. What's interesting, though, is how differently people in these distinct worlds responded. Software developers went to work rethinking how software should be created. But strategy making didn't really change.

The insights upon which new software development approaches are based may point the way for the development of newer, faster and more effective strategy-making processes. Over the past six years, my colleagues and I have experimented with some of these principles to streamline the strategy-making processes at around 60 companies, in industries including hospitality, heavy manufacturing, financial services, education, healthcare and distribution. What we've learned has changed our view of strategy and how it can be created.

THE 'WATERFALL' FOR SOFTWARE DEVELOPMENT/STRATEGIC PLANNING

The "Waterfall" for Software Development/Strategic Planning

Like the conventional "waterfall" techniques for software development, the traditional strategic planning model makes it difficult to jump back to a previous stage and integrate new learning.

Adapted from S. McConnell, "Rapid Development" (Redmond, Washington: Microsoft Press, 1996), 139.

Many software development and strategic planning processes fall on a continuum from predictive to adaptive. Traditional approaches tend to be highly predictive in nature, in that they attempt to plan for the future in detail. With predictive approaches, planning is optimized for the original targets; it is generally difficult to change direction once implementation is underway without throwing out the work that has been done and starting over. (See "Shortcomings of Traditional Software Development and Strategic Planning Models.")

Management guru Peter Drucker pointed out the problems with predictive approaches years ago: "Most discussions of the knowledge worker's task start with the advice to plan one's work," he noted. "This sounds eminently plausible. The only thing wrong with it is that it rarely works. The plans always remain on paper, always remain good intentions. They seldom turn into achievement." [2]

Adaptive approaches, on the other hand, focus on adjusting quickly to changing realities. In software development, they provide teams with the flexibility to revise software programs as designers learn how people actually use them or according to changing environmental forces. They do more than just tweak traditional models — they allow for a fundamental rethinking of the process from the ground up. A similar change for strategy is long overdue: Among 72 companies we surveyed with revenues exceeding $250 million, 80% still use a traditional strategic planning model, despite its recognized shortcomings. [3]

An Adaptive Approach to Strategy Development

In 2001, a group of software industry veterans convened in Utah to examine the commonalities of the new approaches that were springing up to address the challenges in the software industry. The group produced what became known as the *Manifesto for Agile Software Development*, [4] which highlighted, among other things, the importance of emphasizing individuals and interactions over processes and tools, of working software over comprehensive documentation and of responding to change over following the plan. These themes are equally applicable to strategy.

SHORTCOMINGS OF TRADITIONAL SOFTWARE DEVELOPMENT AND STRATEGIC PLANNING MODELS

Shortcomings of Traditional Software Development and Strategic Planning Models

The need to more rapidly integrate new learning and adapt to changing environments demonstrates the limitations of conventional software development and strategic planning processes.

	Problems in software development	Problems in strategic planning
Speed	**Symptom** System delivered is already significantly obsolete at system release date. **Process root cause** "Cascading" sequential process (requirements definition, design, implementation, integration, testing, deployment) is too slow.	**Symptom** Realities in the marketplace call for a change in strategy before the ink dries on the strategic plan. **Process root cause** "Cascading" sequential process (environmental analysis, corporate goals, corporate strategies, business unit goals, business unit strategies, budgets, action plans) is too slow.
Substance	**Symptom** Systems fail to deliver on their potential, often due to a disconnect between those building the system and those who will use it. **Process root cause** Cumbersome process and extensive documentation rigidly enforced by information technology heads despite system users being increasingly under the gun to deploy better systems faster. Users blame system failure on bad developers. Developers blame system failure on poorly defined or "moving target" requirements.	**Symptom** Strategies fail to deliver on their potential, often due to a disconnect between those creating the strategy and those tasked with executing it. **Process root cause** Limiting strategy discussions to top managers reduces leaders' access to "rich" and "soft" data, and that fails to give implementers (who have that rich data) the opportunity to rapidly adapt strategy. Implementers blame failure on bad strategy. Strategists blame failure on bad execution.
Scope	**Symptom** "Kitchen-sink" approach to software development, in which users try to spec and developers try to build the perfect system with every feature users can imagine ever wanting. **Process root cause** Methodology fails to recognize and incorporate how the world really works. Users won't know up front exactly what kinds of functionality they will need, nor exactly how they will need it to work. Speed and substance problems above create the "development doom loop." Users know that if something is not in the spec, they are not going to get it. As specs balloon, development cycles lengthen, the wait in the cue for development resources grows long as well — prompting a user to throw everything he can imagine into the spec document when his number finally comes up.	**Symptom** "Kitchen-sink" approach to strategy development in which each year the company begins an analytical search for the "correct and complete" strategy, which is expected by senior management to be reflected evenly and meaningfully across the organization. **Process root cause** Methodology fails to recognize and incorporate how the world really works. Strategies built on an annual basis and treated as "complete" reduce the organization's ability to make changes on the fly. There is little opportunity to collect, codify and expand on incremental opportunities to shape the strategy as learning occurs in the organization.

Strategy is a mechanism through which a company makes sense of the world around it. It is a collection of ideas about how the company intends to win, the source code upon which everything else depends. Because strategy can only capture a company's best thinking at a given point in time, strategy (like a software program) needs to be refined and improved as people gain and distribute new experience and knowledge.

Given this reality, sound strategy development processes should enable a company to create and adapt strategy quickly and iteratively, so that people can triage issues effectively and allocate resources in changing environments. Strategy development should be optimized for two important outcomes: surfacing the best ideas for improving the company's current and future market positions; and ensuring that individuals throughout the organization have access to the latest version, so that what people do every day is aligned with the most important strategic insights. The "customer" in this process is the entire organization — everyone who is involved with making strategic choices and allocating resources. Since people at many levels of an organization make daily tradeoffs that impact the company's strategic success, the process needs to be designed to tap into ideas from all corners of the organization more than just the

top executives. As my colleagues and I have helped companies work on strategic issues, three major themes have emerged: the importance of having an iterative, or "spiral," approach as opposed to a linear approach; the importance of organizing the strategy-making process around people rather than processes; and the recognition that in strategy there is no such thing as a "silver bullet."

Adopting a Spiral Rather Than a Linear Planning Model New software development techniques recognize that many software programs can be developed more quickly and successfully when they are developed iteratively. Software developers realize they can't have all the knowledge they need to produce the perfect system up front, so they get working models into the hands of users as early as possible. By interacting with users in a "spiral" process, developers can integrate their feedback in future versions.[5] (See "The Spiral Model of Software Development.")

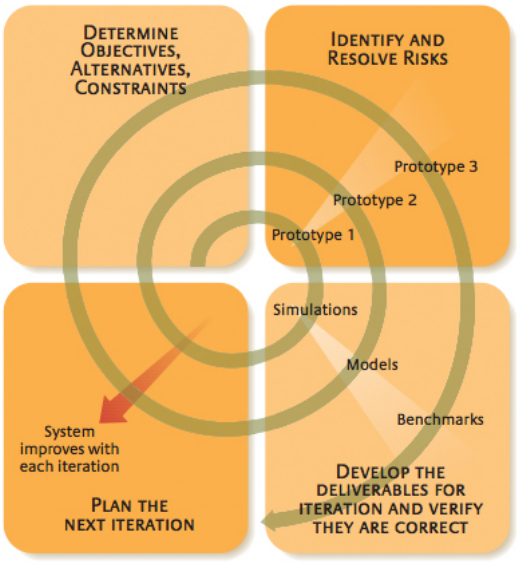
Image courtesy of Flickr user darijus .

A spiral approach to strategy making recognizes that strategy can be created most effectively when formulation and implementation go hand in hand. It recognizes that if people only focus once a year on what drives the business, strategic ideas may evolve too slowly and fail to keep pace with the changing realities of the marketplace. As Jack Welch, former CEO of General Electric Co., wrote in 2000, "We've long believed that when the rate of change inside an institution becomes slower than the rate of change outside, the end is in sight."[6] The spiral approach rejects the traditional, hard-line separation between strategy and implementation, just as modern approaches to software development reject the separation between design and coding or implementation.[7] Strategy formulators and implementers need to develop mechanisms to get together in person and wrestle with the issues. Otherwise, strategy won't keep pace. In fact, the most valuable learning takes place when the people on the front lines attempt to execute the strategy.

THE SPIRAL MODEL OF SOFTWARE DEVELOPMENT

The Spiral Model of Software Development

The principles of a spiral approach to software development can be effectively applied to strategy making, where strategies are built iteratively over time as strategic assertions are tested and refined in implementation "scrums."



Adapted from B. Boehm, "A Spiral Model of Software Development and Enhancement," ACM SIGSOFT Engineering Notes 11, no. 4 (1986): 25.

Shamrock Foods Co., a leading food distributor based in Phoenix, Arizona, has used a spiral planning model to accelerate strategic change since 2004. From its beginning in 1922 as a small dairy, Shamrock has grown to become one of the ten largest food service distribution companies in the United States. The company's strategic planning activities evolved from a process introduced by the Boston Consulting Group in the early 1980s: Senior executives spent several months each year updating a five-year plan. But in 2004, CEO Kent McClelland began to have misgivings about the rigidity of this approach. Shamrock's regional organizational structure made it difficult to adapt to the quickly changing needs of its customers. He worried that Shamrock's strategy was being optimized at the regional level — with too little attention paid to the development of critical capabilities that transcended divisional boundaries. He also noticed that managers always juggled too many priorities, making it difficult to focus on the few strategic leverage points that were most important for growth.

In February 2004, we helped McClelland organize a three-day, 48-hour planning session for 45 Shamrock managers and employees. With participants from all divisions, functional areas and the front lines, the group represented a cross-section of the company. By the end of the second day, the group had zeroed in on five strategic themes: the need to revamp the way Shamrock serviced large customers like Burger King and

Subway; the need to improve order accuracy and service reliability; the need to increase revenue on Shamrock's own house brands; the need to reduce logistics costs through selected regional acquisitions; and the need to improve the company's ability to recruit, retain and develop good talent.

Selecting themes was just the beginning. Shamrock also needed to update its strategic thinking in each of the five strategic areas. While organizations can learn a lot from analyzing and discussing strategic issues, the most important learning occurs when people jump back and forth quickly between strategy formulation and implementation — testing their strategic assumptions with what they are learning. Indeed, even ideas that seem to be right on target need to be shaped and reshaped to reflect the realities of the marketplace. For this reason, at the end of each planning session, participants created and prioritized a handful of specific and measurable strategic initiatives that would advance each strategic theme. Then they built detailed action plans and set measurable outcomes they thought could be achieved within the next 90 days.

Shamrock's process revolved around quarterly strategy "scrums": Team members met at an offsite location for a day to evaluate the company's performance against the action plans from the previous quarter. We asked them to identify the most important things they had learned about the company's strategy since the previous meeting and to suggest how those insights should be integrated in the strategy going forward. The group created new action plans for the upcoming period. In addition to the quarterly scrums, the participants met every year for three days, during which time people were asked to step further back and revisit the company's strategic assumptions.

Since adopting the new approach to strategy, Shamrock has flourished. It has consolidated the part of the business that serves large national chains in one of its warehouses, which has led to new insights into how best to serve the needs of those customers. Shamrock's strategic initiative to drive world-class customer service has led it to segment its customer base and to rationalize its logistics systems in a way that will not only increase customer satisfaction but also eliminate unnecessary costs. It hasn't been a completely smooth ride — the company ran into difficulty integrating one of its recent acquisitions. But this generated some important lessons and forced management to reexamine its acquisition and integration strategies.

Organizing the Strategy Process Around People, Not Tools An effective strategy needs to magnify the efforts of people throughout the organization. It's much easier to do this if lots of people understand the strategy and are able to apply it to the various decisions they face each day. The best way to get people in the middle and on the front lines of an organization to understand and embrace the strategy is to involve them in creating it.

In 2003, my colleagues and I were hired by Metrowerks, then a leading software development tools company owned by Motorola Inc., to develop a new strategy. Metrowerks (best known for its CodeWarrior programming product line) was plagued with problems — products were behind schedule, there were intense disagreements over where resources should be directed, employee morale was low — and there was chronic red ink. Matt Harris, the new CEO, was feeling the pressure. "Days are months in Metrowerks time," he explained. "We simply don't have time to spend months figuring out where we have to go." Harris

wanted to develop a new strategy that identified the markets, customers and products the company should focus on. He also wanted to get people throughout the organization to move beyond their differences and work together. Since Harris wanted a plan in 45 days, there was no time to waste.

Get more people involved. Motorola, the parent company, had a highly refined, top-down strategic planning process, which was conducted by senior executives and some staff analysts. We turned that model on its head. Without time for study groups and task forces, we needed to gather the information quickly and organize it for immediate action. In order to capture the hearts and minds of different factions, a cross-section of the organization had to get involved in developing the strategy. With Harris' help, we identified 60 people — senior executives, middle managers, sales and marketing people, research engineers and top programmers — to take part in a three-day session. We also included a handful of people from Motorola to facilitate strategic coordination with the parent.

Each person was invited to participate actively in setting the direction of the company. As a first step, each person identified the most important issues facing the company and what he or she thought the company should do. There was virtually no consensus, and conflict and mistrust were widespread. But much of the conflict could be traced to a lack of agreement on strategic direction. Almost everyone had a strong commitment to turning the business around.

Work the issues face-to-face. The new software development techniques stress the importance of face-to-face collaboration with users as a way to mitigate conflicts between developers and users and to get beyond the organizational silos that often interfere with multidisciplinary problem solving. The same principle applies to strategy: In a strategy process that values individuals and interactions over processes and tools, there is no substitute for face-to-face dialogue. During the three-day meeting, we discussed the big issues facing the company, including the future of Metrowerks' strategic relationships with other parts of Motorola, their markets' customers, product sets, competitive threats and internal and external challenges. To avoid getting bogged down by the size of the group, we divided it into groups of ten; each subgroup had people from different areas, and participants moved in and out of groups on a regular basis. By design, the process was iterative, moving from detailed observation to idea to strategic assertion, then back to observation. It was also disciplined, with a set of clear ground rules to keep people engaged and on topic.

Get senior executives off center stage. Someone other than a senior company executive should facilitate important strategic discussions. There are several reasons for this. First, even when they don't realize it, people naturally defer to those in authority. Also, people are less eager to promote ideas that haven't been endorsed by others (sociologists call this "social proof"). When senior executives attempt to facilitate the discussions, they are often identified with positions — even when they try hard to be even-handed. A good outside facilitator can use conflict to distill issues and build consensus; participants don't have to worry that they have supported the "wrong" side of an issue.

Trust the process. We expected that including the people responsible for implementing the strategy (the strategy "customers") in the process would make them more effective in their jobs. Among other things, we thought that an understanding of some of the nuances of the strategy would improve their decision making.

But would involving these people in strategy making actually result in better strategy? Initially, the senior people at Metrowerks were skeptical. But over time, as they learned how to integrate people into the process, they saw that opening up the strategy formulation process to include people from several levels of the company significantly improved the quality of strategic thinking in the company. They should not have been surprised. As computer programmer Eric S. Raymond wrote in his now-famous essay, "The Cathedral and the Bazaar," about the open-source community: "I believed that most important software … needed to be built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation.... Linus Torvald's style of development — release early and often, delegate everything you can, be open to the point of promiscuity — came as a surprise. No quiet, reverent cathedral-building here — rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches … out of which a coherent and stable system could seemingly emerge only by a succession of miracles. The fact that this bazaar style seemed to work, and work well, came as a distinct shock."[8]

THE FUTURE OF STRATEGY MAKING

The Future Of Strategy Making

The Future of Strategy Making

The key principles that underlie new approaches to software development point the way to further innovation in strategy making.

	New approaches in software development	Implications for strategy making
Start small and iterate from there	Replace "big design up front" with a model that breaks the development cycle down into shorter iterations that allow users and changes in the environment to influence later iterations.	Replace "big design up front" — highly structured and analytical annual planning processes — with more frequent and compressed "strategy scrums" that enable the iterative development of strategy.
Get something into people's hands quickly	Rather than trying to build all possible needed functionality into the product at release, build quick prototypes and focus first on getting important core functionality in the hands of users; add additional functionality in later iterations.	Toss the three-ring binders filled with hundreds of pages of strategic planning data. Develop a fundamental set of strategic assumptions and get them into the hands of the users (managers) to enhance iteratively in web communities and face-to-face strategy "reset" meetings.
Get people to work together	Short-circuit developer-user adversarial relationships by making users and developers jointly responsible for the success of the project.	Broaden the group responsible for strategy to include both "thinkers" and "doers." Use streamlined strategic processes to involve people from several levels in the organization and from all the major functional areas.
Integrate the best ideas available	Whether through implementing a true "open source" approach or by generally emphasizing transparency and openness, encourage everyone in the community to help make the program as good as it can be.	Give people throughout the organization constant and up-to-date access to the "strategy source code" — the most recent set of assertions and assumptions about how the organization intends to win — and encourage them to make improvements.
Debug as you go	Bugs are identified and corrected continually through a process called "code and smoke" instead of waiting until the end of the project. People who find bugs are celebrated and appreciated.	View strategy as simply a collection of ideas that is likely to be shaped, refined and corrected as the organization learns. Don't let people who question the strategy be labeled as "not team players" — celebrate people who ask great questions that cause the company to question and refine its assumptions.

Hide Exhibit

Senior executives at most companies have yet to consider this type of approach. A 2006 survey of 796 executives by consulting company McKinsey & Co. found that in 73% of companies, strategic decisions are made by the CEO, business unit leaders or a small group of senior executives.[9] But executives who have been through a more participative and adaptive process immediately see the merits of that process. Matt Harris of Metrowerks admits he was skeptical when the new process got underway. Among other things, he

worried about the size of the group. He says: "I had already decided beforehand that if the process wasn't working by the end of the first day, I'd shut it down — it would be a great way to signal to the team that I would demand performance in everything we did — even from consultants that I myself hired. Not only was this unnecessary — the process itself was an unmitigated success. People who participated still call and write me years later, telling me how invigorating the whole experience was, and asking how to replicate it. No one was more surprised than me."

Forgetting About Finding the "Silver Bullet" In a 1986 essay entitled "No Silver Bullet," software industry visionary Frederick Brooks wrote, "However frustrated we may be in the writing of computer programs, we will never find a magic, transformational breakthrough — we should expect only modest, incremental advances."[10] Had someone said something similar about the field of strategy early in its development, strategic planning might never have declined so precipitously. Today, few people still believe that strategy is merely a matter of understanding industry structure or writing a long-term plan. Most managers operate in settings that are too dynamic and complex for simple success recipes. Instead of seeking long-term*sustainable* advantage, good managers need to create*sustaining* advantages on an ongoing basis.

Like software developers, strategists must learn to add new strategic features (capabilities, insights and tools) incrementally. Taken together, these features can add up to an advantage, but managers need to be realistic and recognize that any competitive edge may only last for a short time. An ongoing challenge will be to continue to develop and refine new approaches to strategy making that help companies bridge the gap between strategy formulation and implementation. (See "The Future of Strategy Making.".)

ABOUT THE AUTHOR

Keith R. McFarland, *a former technology CEO, is founder of McFarland Strategy Partners in Sandy, Utah, and author of* The Breakthrough Company: How Everyday Companies Become Extraordinary Performers.

REFERENCES (11)

1. S. McConnell, "Rapid Development" (Redmond, Washington: Microsoft Press, 1996), 139.
2. P Drucker, "The Essential Drucker" (New York: Harper Business, 2001), 225.
3. Unpublished survey, McFarland Strategy Partners, 2006.
4. Manifesto for Agile Software Development, 2001, http://agilemanifesto.org.
5. B. Boehm, "A Spiral Model of Software Development and Enhancement," ACM SIGSOFT Engineering Notes 11, no. 4 (1986): 21–42.
6. J. Welch, J. Immelt, D.D. Dammerman and R.C. Wright, "Letter to Share Owners," in "GE Annual Report: 2000" (Fairfield, Connecticut: General Electric, 2001), 1–7.
7. Mitchell Kapor, founder of Lotus Development Corp., said, "If design and implementation are in watertight compartments, it can be a recipe for disaster because the natural process of refinement and change is prevented." See S. Rosenberg, "Dreaming in Code" (New York: Crown, 2007), 150.
8. Ibid., 23.
9. "Improving Strategic Planning: A McKinsey Survey," McKinsey Quarterly (September 2006).
10. F.P. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," Computer 20, no. 4 (April 1987): 10–19.

REPRINT #: 49315