



5

Learning in Projects

Much has been written in the academic and popular business press about learning in organizations, and learning has been given many definitions.¹ For our purpose, we use a very specific definition of learning, one that emphasizes adjustment in response to emerging unknowns in a project:

Learning in projects is the flexible adjustment of the project approach to the changing environment as it occurs; these adjustments are based on new information obtained during the project and on developing new—that is, not previously planned—solutions during the course of the project.

The essence of this approach to project management is that each new activity will provide new insights and information, which can be used to review and revise the project plan, the resources required, and the stakeholders to be dealt with. While each of the changes may be minor, the project itself may look quite different at the end from the original plan and intention. This type of learning “as we go” should not be confounded with projects as instances of learning about the organization

(because you do something out of the normal activities of your organization) or post-project audits that are used to improve future projects.

To understand learning “as you go,” let us briefly return to the simple mountaineering example that we used for illustration in the introductions to Parts I and II. There we contrasted two mountaineering expeditions, one up a known mountain for which we have a map and a weather forecast and another up an unknown mountain with unknown weather conditions. The essence of the comparison is between executing an existing plan versus developing the plan as one goes along and learns about the terrain en route. Mastering the unknown mountain requires more sophisticated mountaineering skills, and more experienced and flexible people who can observe the terrain and keep an eye on the weather conditions during the expedition and make decisions in response to what they learn. More information needs to be gathered, and coordination among all stakeholders must be more flexible.

To demonstrate the project learning approach to unk unks, we continue the Escend Technologies example from Chapter 4.² There we showed how the presence of unk unks was diagnosed, and now we show how learning was managed in this project of rescuing this startup company and shepherding it to the next financing round. After this example, we discuss in further detail the process of project learning.

5.1 Learning at Escend Technologies

When Escend Technologies was founded in 1999, it was like most other startups in that it had a business plan. The business plan was an essential part of selling the Escend business proposition to potential investors, as is common practice in new venture funding (see Table 5.1 for a view of how venture capitalists expect a new venture to progress through various development stages to the initial public offering, or IPO). The business plan is not unlike a project plan in that it contains, in addition to a description of the market and core product, a development plan consisting of key milestones to be met by the startup at key points in time.

Monthly board of directors’ meetings track progress according to the key milestones. The board aims at providing guidance and continuity in governance but often considers only what has occurred since the previous meeting. The board hears management’s version of the world—that is, problems, progress, development, cash requirements versus burn rates, and so on—and if it sounds logical, the board may tolerate delays in achieving the milestones. If not, actions may be taken to correct the situation, assuming nonexecutive board members have the power to do so.

5.1.1 Planning and Firefighting

Recall that Escend was built on the idea to help semiconductor and electronic component manufacturers connecting and collaborating with their extended sales force, namely the manufacturers’ representative that sold their components to electronics OEMs. The founders originally conceptualized

Table 5.1: Project Stages of a New Venture Project Reaching IPO

Criteria	Seed	Startup	Early Stage	Expansion Stage	Later Stage
Definition	Research and develop initial concept	Concept or product under development, but not fully operational; usually in existence less than 18 months.	Testing or pilot production; may be commercially available and generating revenues; usually in existence less than three years	Production and commercial availability with significant revenue growth but may be without profits; usually in existence more than three years	Wide availability with ongoing revenue; likely profitable with positive cash flow; includes spin-outs of operations of existing private companies
Milestones	Business plan	Full team; prototype	Beta test; booking orders	Customers	Profitability
Major source of financing	Founders, family, and friends; angels	Angels and venture capital	Venture capital	Venture capital	Venture capital and buy-out funds
Financing required	R&D	Product development	Capital assets	Capital assets and working capital	Capital assets and working capital; product development
Operating cash flow	Negative	Increasing cash burn rate	Increasing cash burn rate	Decreasing cash burn rate	Positive
Risk	Higher	—————→			Lower
Valuation	Lower	—————→			Higher

Source: PricewaterhouseCoopers MoneyTree

the opportunity as one of collaboration among industry players who would want to be part of Escend’s B2B (business-to-business) community. Figure 5.1 illustrates Escend’s initial understanding of the market opportunity. In retrospect, two things become painfully obvious from this figure: First, the web of relationships among the various industry players is complex, and second, Escend had no clear idea of its value proposition to its potential customers.

In a way, it is not surprising that by mid-2003, Escend was on the brink of bankruptcy. But how did the company get into such a state? One could discuss specifics, but the general answer is that the planning approach, as laid out in Table 5.1, was not suitable for a startup like Escend. Escend faced too many unk unks: in its technology, in the industry, and in the customer needs. The milestones laid out in its business plan were simply unrealistic, not in the sense of too much stretch, but simply by virtue of the fact that no one had done this before and so no one knew what to expect. Faced with time pressure and market dynamics that they did not understand, the team was forced to improvise around the plan, and they “[found] themselves frustrated by the simultaneous pressure to act and the inability to understand what [was] going on around them.”³

This is not unusual for startup venture projects. Drucker observed already in 1985, “When a new venture does succeed, more often than not it is in a market other than the one it was originally intended to serve, with products and services not quite those with which it had set out, brought in large part by customers it did not even think of when it started, and used for a host of purposes besides the ones for which the products were first designed.”⁴

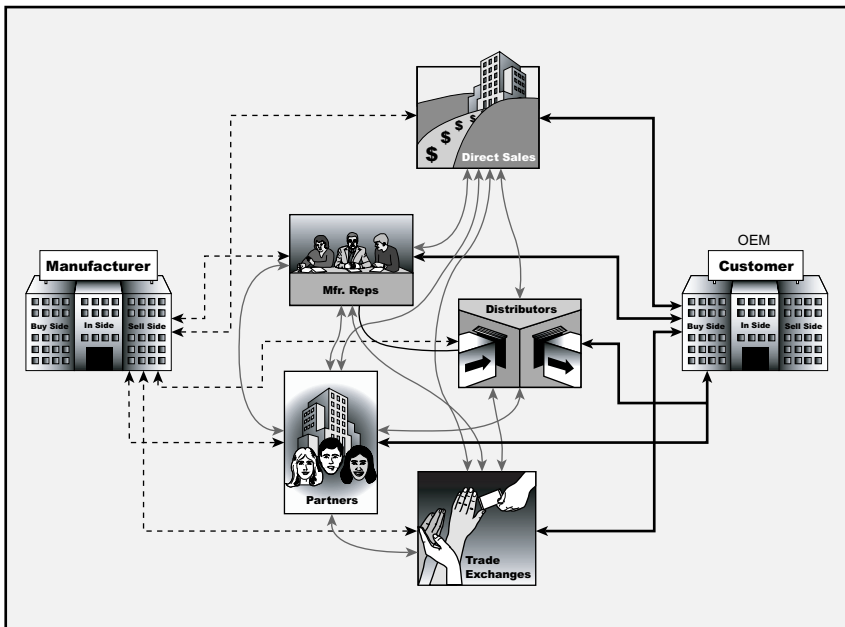


Figure 5.1 Escend Technologies’ initial view of the extended sales organization

February 2000	Escend builds B2B communities that connect manufacturers with their outsourced sales channels and distributors through its sales information network.
May 2001	Escend provides the only system that overcomes the competitive disadvantages of a many-to-many business environment by speeding communication, normalizing data interchange, and connecting an entire industry.
September 2001	Escend provides the only online CRM application that includes the infrastructure and data exchange translator required for independent companies to collaborate on any aspect of the order life cycle.

Figure 5.2 The evolution of Escend's business model before 2003

During this time, Escend's value proposition went through various evolutions (see Figure 5.2). These evolutions in its value proposition were not the result of systematic investigations into the industry or the needs of its customers. They were simply reactions to events that occurred around them. They lost sight of the original objective of making money in a market opportunity and instead focused on trying to implement the business plan. The business plan became the objective, and the message was changed from time to time to help get the business plan back on track.

In fact, when the initial reaction of outside investors to the \$6 million request for additional funds was negative, the board thought that the message, not the business or management team, was the problem. The board continued to behave as usual by rewriting the message to investors, without critically examining the management team or the approach to the business opportunity.

It was not until the funding situation reached a critical point that the board woke up to the situation at hand and things began to change. Unable to raise outside money, the board, including the key investors, had to assess critically both the management team and the business opportunity. The task fell to Elaine Bailey, a general partner at Novus Venture, one of the original funding partners. She faced a difficult decision: Should Novus participate in another round of funding for Escend—thus risking throwing good money after bad—or should they simply pull the plug and cut the losses already incurred in funding Escend to this point?

5.1.2 Diagnosis and Learning

In July 2003, Elaine Bailey stepped in as interim CEO of Escend Technologies to assess the company and recommend the next steps. It became clear to her relatively early on that major changes had to be made in order for the company to survive. The trouble was that it was far from obvious what these changes were or how they could be identified. As she put it, it was like “trying to see through a rock.”

Fortunately for Elaine, and for Escend, she was sent in to assess the situation for purposes of generating a yes-or-no answer as to the additional funding. This perspective led her to diagnose the current situation, something the previous management team should have been doing all along. Chapter 4 already described in detail her steps to diagnose the situation. Her key insight was not to attempt to diagnose what needed to be done to implement the business plan, but to remind herself of the objective (to make money in a particular market opportunity) and to diagnose what they knew and what they did not know about the market opportunity and Escend's ability to take advantage of this market opportunity.

She recognized that there were significant knowledge gaps and, thus, major unk unks that had to be dealt with (see Table 4.1 in Chapter 4). The three main areas with the highest potential for unk unks were (1) customer needs, (2) industry readiness, and (3) product functionality. Customer needs had the greatest knowledge gaps because customers themselves could not articulate their needs. No one had yet understood where the "product" would ultimately create the most value. Thus, the required product functionality was unknown and the underlying technology (XML and RosettaNet) was fairly new. Finally, as there were no competitors yet in this space, no one had defined the problem before, and no analysts were covering any companies in this part of the industry. In other words, Escend was a pioneer in uncharted territory.

The three unk unk problem areas required a different mind-set at Escend than had been previously established. Elaine had to determine whether Escend made sense in the first place, what assumptions potential success depended on, and what different angles might offer new opportunities. This involved an open-ended search with an unknown result, requiring "switching gears" as compared to the execution mode that characterized the other problems (and most of what VCs usually do).⁵

Escend was creating a new market niche "where no one had gone before." Thus, unk unks had to be lurking in the unmapped terrain. The goal had to be to turn unk unks into known unknowns (foreseeable uncertainty). This cannot be done in a classic, straightforward "analysis"; it is a process of discovery over time. Table 5.2 shows some of the questions that Elaine and her team used to investigate assumptions Escend was making and to kick-start the discovery process.

The table provides a column for assumptions about the relevance of channel collaboration software for the business of the customer (that is, the electronics component manufacturer). Escend's value proposition is based on this assumption and is where Elaine and the team initiated the process by formulating probing questions. Two example assumptions and a few probing questions are listed in the table. The full list covered a large whiteboard, which was maintained in a meeting room. The management team would meet daily and weekly for one to two months to nail down the unk unks.

As Elaine traveled extensively during her first three months as CEO, interviewing enterprise firms, end customers, analysts, consultants, VCs who

Table 5.2: Probing Questions to Query Assumptions

Assumptions

Product is not a commodity/not custom silicon.

Channel is very diverse in skills, characteristics of firms

Escend's Value Proposition for Its Customer (Component Manufacturer)

Keep customer from losing orders (*Customer need statement: "I'm losing orders from design win to production. I want my channel to give me a competitive advantage."*)

Provides in-depth visibility about design-win process (*Customer need statement: "I don't understand the breadth and depth of the relationship with my customers."*)

Probing Questions

- Once you have a design win, what's the likelihood that you'll get the order?
 - How much do you lose (leave on the table) for design wins that don't translate into orders?
 - What influence does your channel play in securing the order once it is designed in?
 - What impact do you have in moving it from design win to order? (scale 1–10; least–most)
 - What needs to happen to improve your design win to order conversion rate? (. . . and so on)
 - On a scale of 1–10, how much visibility do you have of your customer base?
 - What types of information are important to know about your customers?
 - On a scale of 1–10, how well do you "service" your customers?
 - What type of service (questions) are your customers requiring/demanding?
 - What information is the customer requesting that is out of the control of the sales department?
-

did not invest (either because they did not believe in the management or the business model), managers of different collaboration startup companies, and academics, information slowly emerged, but the information kept changing as the industry evolved. As we described in Chapter 4, Elaine concluded that it was worth continuing and achieved a financing round in August 2003.

In October 2003, Elaine convened a “no good news” board meeting. She had realized that truly learning and using the unk unk concept was not easy, and letting the unk unks assume shape takes time. The table with assumptions and questions was erased and rewritten again and again, as new information came in. As Elaine put it, “We kept putting our ear to the ground, and we heard nothing. Slowly, we began to hear some faint hoof beats; then they became louder and louder.”

In parallel to the planning subprojects, the knowledge gap around customer needs and the readiness of the industry for a player like Escend became a learning project. Elaine reserved part of her and the Escend team’s time to reflect and to gather information from multiple parties about that problem area, not knowing what to expect. She remained open to finding nothing of significance in this inquiry or something that might prompt her to fundamentally rethink the business model—or even to shut Escend down.

5.1.3 First Results and Further Adjustments

In developing the business over the 12 months, two large business changes emerged, both of which changed Escend’s strategy and neither of which could have been anticipated. First, Elaine learned how fast the electronic components market was becoming global. The “technical vertical” (industry-speak for the end-user industry application) had to be a complex product that was global in nature.⁶ This implied a global platform and design-win tracking for component manufacturers when they were bidding for their components’ incorporation into end products. A global platform, another product redesign, would consume precious funds and resources. Thinking that potential competitors would also have to redesign their product, which would put Escend in the lead, Elaine decided to bite the bullet. For instance, the redesign incorporated multiple languages and currencies, and multiple access points per customer, into the product. This also meant that Escend’s target customers and growth strategy changed.

Second, any firm in the industry network (Figure 5.1) had limited visibility of the entire network. Component manufacturers, through their rep firms, could not track either sales or the process and, thus, could not ensure full and timely payment. OEM buyers cared about the design win but not about tracking it, because they wanted to buy at the lowest possible price and had to give extra profit margin to the contract manufacturers and design-win firms. OEM buyers often gave the job to *different* manufacturers

who offered lower costs *after the design was “won.”* As a result, manufacturers were changing the way they sold products, shifting away from reps toward relying on distributors and in effect taking back some of the activities that reps were offering. Therefore, Escend would have to build distribution functionality into the product. To this end, in October 2003, Escend produced a prototype that offered shipping and debiting, samples management, and pricing and quoting functionality. Coding would take another 12 months; the plan was to go live in January 2005.

In late October 2003, Elaine’s search for information had led her to another startup (run by a competitor VC) that had a collaboration software product covering the demand cycle of the industry. The two had the potential to form a perfect match if their products could be made to work together. Elaine was confident that the two software products could be made interoperable (this was a problem area with foreseeable uncertainty). But the merger fell through because the investors of the other startup got some of their money back and wanted out.

The flexible way of proceeding, including repeated unplanned product changes and three major strategy changes (counting the aborted merger), was very stressful, and possible only because Elaine, combining the roles of chairperson, CEO, and partner of one of the major investors, was leading it. However, while it was Elaine, who had authority, access to the investors, and prior operational experience, it was the new Escend management team that implemented the process. Elaine assembled a new team in the late summer of 2003 but had to make further changes, including replacing the VP of sales again in June 2004. Over time, the team became fully engaged in the learning process. “Unk unks” is now a well-known term in the Escend offices, reflecting the new mind-set Elaine has instilled in the company.

Escend’s business model slowly crystallized. Figure 5.3 is taken from a white paper that Escend published in the spring of 2004 and has a clarity that stands in contrast to the complex Figure 5.1. Tracing the flow of the design-win process is easy in Figure 5.3, and the demand cycle and demand fulfillment (i.e., supply) cycle is also clearly delineated. Unk unks no longer dominate; the industry structure now looks understandable, and the effect of actions taken can be traced. Foreseen uncertainty has replaced unforeseen uncertainty.

In the fall of 2004, it seemed that Escend had turned the corner and was becoming an excellent bet for the investors. At the end of October, the company obtained another \$3 million from the existing investors, Novus and NIF Ventures. Daiwa Securities now intended to translate the product into Japanese and to go into Japan in December 2004, having targeted 12 Japanese companies. In November of that year, the plan foresaw cash flow breakeven occurring in Q2 2005. A new investor had signed a letter of agreement (term sheet) to invest another \$1.5 million in Escend. And Elaine, signaling her confidence, was getting ready to replace herself as CEO.

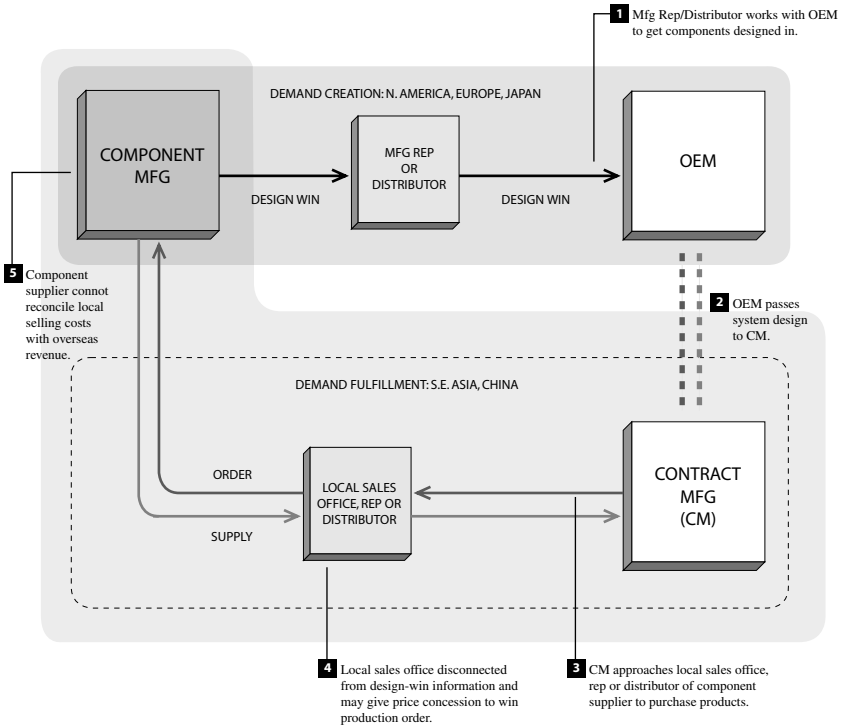


Figure 5.3 Escend’s description of the industry network, spring 2004

5.2 Types of Project Learning

5.2.1 Three Types of Learning

Classic typologies of organizational learning⁷ distinguish between three levels of learning: (1) single loop, (2) double loop, and (3) deuterio learning. In *single-loop learning*, the organization detects “errors” and makes corrections according to existing plans and policies. This is consistent with our discussions of contingency planning and classic PRM in earlier chapters (see, for example, Figure I.1 in the introduction to Part I). In contingency planning, the organization creates policies consisting of contingent actions, should “events” arise, and then implements these planned actions if and when events *do* arise. While specific actions might change, no modification is made to the underlying policies and plans that generate these actions.

In *double-loop learning*, when an organization detects errors, it takes action to correct these errors in ways that involve the modification of the existing plans and policies. The term “double-loop” implies a correction not only in response to errors but also in how the response is made. This is consistent with the type of learning we will be discussing in this chapter. As the organization modifies its project plan in response to acquiring new information (as it “detects errors”), it is creating new policies and implementing these new

policies as it proceeds. Thus, actions, and the policies and plans that generate these actions, are changing over time.

The final level of learning, *deutero learning*, involves changing the learning system by which organizations detect errors and take action. In order for an organization to move from contingency planning to project learning, it will have to change not only what it does but also how it goes about doing it. Thus, the organization will have to shift from a project mind-set, infrastructure, and governance geared toward contingency planning to one geared toward active project learning. This level of learning will be addressed in Chapters 8 through 10.

5.2.2 Double-Loop Learning through Improvisation or Experimentation

So how do organizations carry out double-loop learning? In other words, how do they go about acquiring new information and how do they respond to this information, once acquired? There are two types of double-loop learning that organizations can undertake, which increase in the amount of effort required and responsiveness achieved: improvisational learning (learning in real time from action variations) and exploratory or experimental learning (stretching from trial-and-error to purposeful experimentation).

Improvisational Learning

In improvisational learning, real-time experience drives novel action at the same time that the action is being taken.⁸ That is, planning and execution occur simultaneously, typically in response to problems or opportunities created by unanticipated events.

Improvisation can be in the form of new behaviors or product features, but they can also be in the form of new interpretive frameworks; problems can become opportunities when seen in a new light, thanks to changes in how the “problem” is framed.

The metaphor of improvisation comes from the arts. Take jazz, for example: Each musician varies his or her tune around a common and stable plan of rhythm and harmony. While the musicians vary their tunes spontaneously, in real time and in unplanned ways, they must strictly adhere to the rhythm of the ensemble. This limits the range of variations that are possible; thus, creativity is limited. In other cases, the time and coordination pressure is not so high, and so more creative variations can be sought. Painters, for instance, while loosely collaborating, work alone and can widely vary their individual styles.

Thus, improvisation can comprise a differing mixture of spontaneity and creativity. As the degree of uncertainty increases, in particular, as unknowns and complexity become important, creativity of improvisations becomes critical in order to be able to respond to the unknowns as they emerge.⁹

In jazz improvisation, spontaneity is high, as the jazz performer is creating in real time, but uncertainty is low, and unknowns nonexistent. The basic “plan” is known to all performers, and improvisation is mostly ornamental. This also applies to what we referred to as “residual uncertainty” in Chapter 1. In the PCNet project, the basic plan was known and the risk management office (RMO) was simply there to respond to unknown unknowns as they arose. The team had to be improvisational in that they had to respond to unexpected events quickly, but mostly this improvisation was driven by spontaneity to time pressure, rather than open creativity.

Rapid response to events is also critical in control-and-fast-response situations. With complex projects, it is critical to stay on a known control path. Any deviations must be dealt with quickly, often requiring individuals to simultaneously develop and implement a solution. However, again we see that unknowns are relatively minor as long as one stays within the known region of control. Improvisation is still based more on spontaneity than on creativity; the overriding goal being to keep the project within the known region of control. As a case in point, when the PCNet project management team discovered that local changes to the e-mail system created unanticipated system instability, they quickly moved to preempt this behavior at all local installations, thus keeping the system in its stable configuration.

When uncertainty is higher and becomes unforeseeable, improvisation must vary more widely, incorporating more on creativity than on spontaneity. Austin and Devin (2003), in a collaboration between a technology management professional and a theater professional, pointed out a very close parallel between managerial “knowledge work” and the work of artists, a parallel that is underutilized by project and new product development organizations. Describing a theater production and a software development project, they show how creative improvisation around a theme is necessary not only to react to external changes but also to *create* a superior solution that could not be foreseen or planned because the project is too complex—too many variables interact in determining how good the solution is. The authors see four qualities that make for successful improvisation of a team:¹⁰

1. *Release*. A method of control that accepts wide variation within known parameters. “Release” contrasts with “restraint.” This quality is essential for the other qualities to work. It allows the creation of variety, which is the core of creativity.
2. *Collaboration*. The requirement that each party, in language and behavior, treats the contributions of the other parties as material to create with, not as positions to argue with, so that new and unpredictable ideas emerge.
3. *Ensemble*. The quality exhibited by the work of a group in which individual members relinquish sovereignty over their work and thus create something none could have made alone, a whole greater than the sum of the parts.

4. *Play*. The quality exhibited by a [theater] production while it is playing to an audience, or by the interaction among members of a business group, and ultimately between the group and the customer. This quality builds upon intimate interaction between the group and the audience, or customer, so the customer closely relates to the production.

These qualities capture the essence of improvisation; it means creative variation within a known range. When uncertainty and unknowns are high, a project team may “find [itself] frustrated by the simultaneous pressure to act and the inability to understand what is going on around them.”¹¹ Here, a one-off improvisation is not enough to resolve the uncertainty and to find a good solution. Indeed, Austin and Devin emphasize that improvisation must be *embedded in a systematic process*, which essentially repeats cycles of creativity. This is what we describe next, as experimentation.

Experimentation

The most basic form of experimentation is trial-and-error learning, in which the organization develops and implements its plan but then closely monitors the situation to constantly evaluate whether and how the plan should be modified. The more systematic and exploratory experimentation becomes, the more it contains the *purposeful* search to uncover unknowns, without regard to the success of the trial.

The basic building block of experimentation is the Plan-Do-Check-Act (PDCA) cycle, often seen in operations (see Figure 5.4). The key success factor for learning is to keep the PDCA cycle small and fast.¹² That is, “failures” should come early and often, before they become catastrophic. Changes early in the project are always less costly than those that come late in the project. The ability to create early, small failures runs contrary to traditional project management mind-set, infrastructure, and governance. Many projects fail to implement the PDCA cycle because they use a model of project management where all tasks and requirements must be defined *before* the project can even get under way. The standard response is “How can you manage a project if you do not know exactly what the project involves?”

Significant up-front analysis and design are undertaken at great expense in terms of time and money, even in projects where there is no known feasible, let alone optimal, set of activities. This is necessary because without planning, the team has no basis on which to stand. However, by carrying out extensive risk analysis, creating risk lists and contingency plans, a team can easily fool itself into thinking that it has a robust plan. This is fiction under the presence of significant unknowns because the plan itself is not really known. The best that can be said is that the plan under consideration is simply a starting point for what is known about the project (we already touched upon this problem in Chapter 3, when we called PRM “double-blind”). Too often, the initial project plan, while flawed, becomes the new objective, while the original objective can get lost in the confusion. The project plan is seen as an end in itself, and no longer a means to achieve an end.

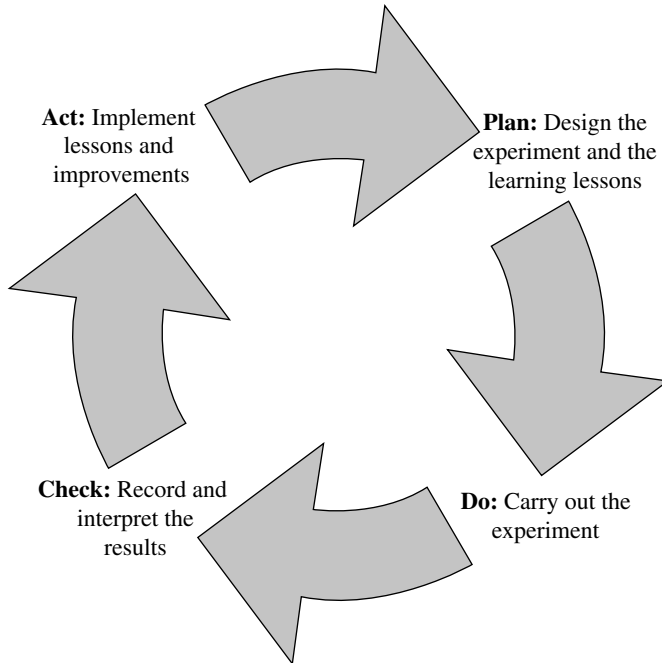


Figure 5.4 Trial-and-error learning as a Plan-Do-Check-Act cycle

Organizations must be prepared to make early mistakes, and make them often. The project should not be seen as a “single monolithic entity that is either frozen or liquid, but rather a more complex structure,” one whose specifications can be progressively frozen as the project progresses.¹³ Project commitments should be made piecewise as more information is revealed, rather than all in one go at the start of the project.

5.2.3 The Trade-off between Learning and Progress

In the extreme case of exploratory experimentation, activities are undertaken that look inefficient. The critical question is not “What are the next actions in an optimal project plan?” but instead “What are the next actions that will generate the most information about the unknowns in the project, and how can we best incorporate this information into our project?” Early actions are taken not so much as part of an initial project plan, but more a part of a learning process as in the design of experiments. At the opposite extreme of improvisation, exploration purposely maximizes variation in order to create the greatest opportunity for learning, not the greatest opportunity for immediate success. This is summarized in Figure 5.5.

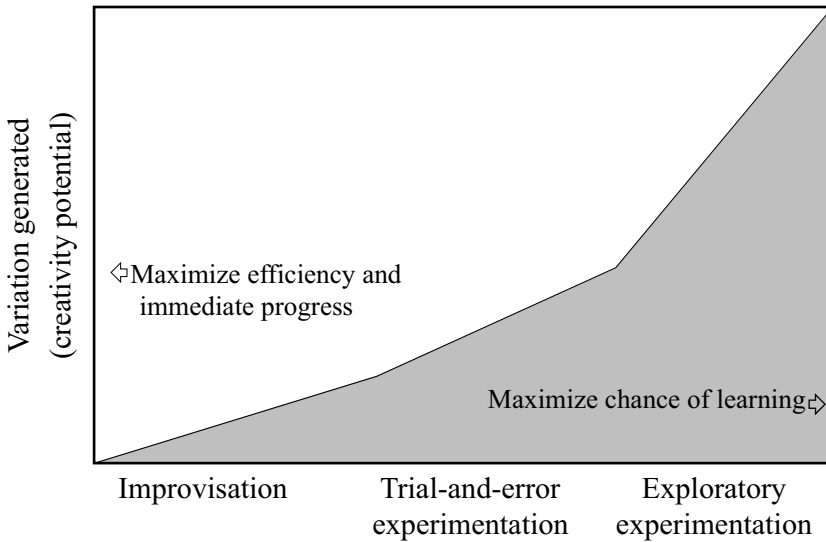


Figure 5.5 The trade-off between variation (learning potential) and progress

The more experimentation and learning the project team undertakes, the less efficient progress toward the targets becomes. However, when the plan is a fiction, progress is a mirage. The more important unk unks are, the more the trade-off is biased toward learning. And the earlier experimentation and learning can happen in the project, the lower is the efficiency loss, because early actions are usually cheap. Thus, experimentation should take place as early as possible.

In the presence of unk unks, up-front market analyses or technology performance analysis may be impossible, as critical factors affecting performance are unknown. Early experiments, generating early failures, rather than analysis, are critical for learning.

5.3 Back to Escend: Drawing the Lessons

5.3.1 Interpreting the Escend Example

The Escend Technologies story is an excellent example of what to do and what not to do in managing projects with significant unk unks. At the beginning of the story, we saw a management team attempting to utilize a planning approach to managing the project. As unk unks arose, which was inevitable given the nature of the project, the management team was faced with the pressure to act, but without fully understanding what was going on around them. They tried to improvise around a plan, but unfortunately, the plan was simply too unrealistic to offer much direction. Thus, the team became confused and began to focus everything on making the plan work, forgetting their original objective of making money from a market opportunity.

It took a crisis to bring the board around to recognizing that the original approach to the project was flawed. A change in mind-set was required. It became a priority to understand what they knew about this market opportunity and about Escend's capability to take advantage of this opportunity. The business plan came to be seen simply as a starting point, a way of asking critical questions. The important questions concerned what they knew and what they needed to know, and where the gaps were that needed to be explored.

The management team then began to systematically explore the unknowns. They could not design laboratory-type experiments—they were not in a powerful position to run trials with their clients. But they could, and did, systematically pose questions and make trial proposals to the market, as well as run technical tests with the product, in order to explore the nature of the market. In this way, Escend radically evolved its business model, not only by improvisation but by systematically exploring what it did not know.

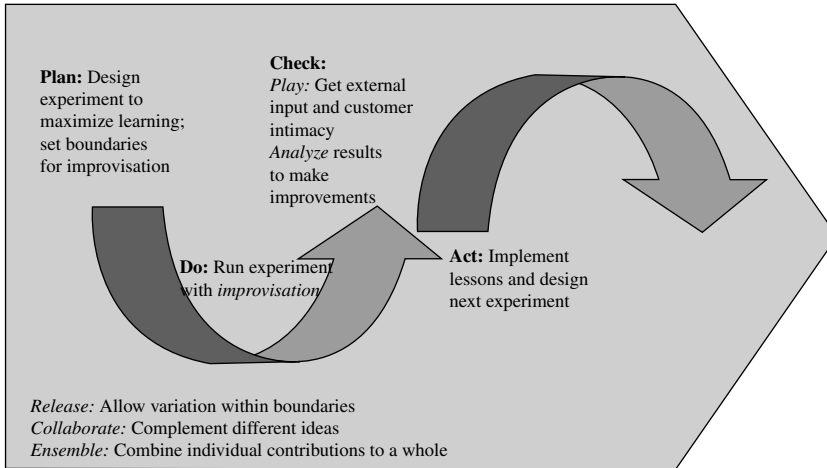
The resulting business proposition, as presented in Figure 5.3, stands in stark contrast to the confused and ambiguous proposition presented in Figure 5.1. This is the result of a systematic approach to exploring what the team did not know, improvising within the current plan in response to unfolding events, and implementing a PDCA cycle where the results of each action are diagnosed for possible learning opportunities, which then change the business plan and future action.

Implementing a learning strategy in any project is not easy. It is human nature to want a plan and milestones by which it is to be measured. However, as we have seen in both the Escend and Circored examples, a project plan should never be seen as an end in itself: It is always the means to an end. The ultimate objective, whatever it is, should always be at the forefront of the project team's mind.

5.3.2 An Experimentation Process

The learning process diagram in Figure 5.6 summarizes the discussion so far.¹⁴ The heart of the process is the experimental PDCA cycle, with embedded improvisation to maximize variety and, thus, creativity and learning potential. The “act” step emphasizes that experiments are not one-off activities but must be continuous in novel projects. The “play” position emphasizes that the evaluation of an experiment risks being insufficient if performed only by the team. Customer input is important to add fidelity to the experiment, but also to build customer intimacy, helping both customer and team to understand the other side better.

The individual PDCA cycle is embedded in a *process* of a stream of learning events. The principles of the process are, first, to accept *failure* as a source of learning. This is counter to the instincts of many managers who interpret every instance of something not working as a mistake. However, in novel projects with unknunks, you only know whether you have reached the limit of current performance when something fails. A failure is not a mistake; a mistake is a failure that produces no new information for the team.



- Design principles of the process:**
- Recognize failure as a learning opportunity
 - Experiment as early as possible
 - Organize for frequent and rapid experimentation
 - Integrate multiple experiment technologies

Figure 5.6 An experimental learning process

Second, information is most valuable if it is gained early. Chances of early success are small, but the costs are almost always lowest at the beginning, and opportunities for learning are great. Thus, the organization must *anticipate and exploit early information* if it is to benefit from early probing. If early experience cannot change later actions, the organization cannot benefit from experimentation. The important questions must be “What can be done differently?” The sooner one finds that out, the better for the project.

Third, the project team must be organized for rapid experimentation. For example, if the person, or group, who plans the experiment is in a different department than the one who executes it (and perhaps the person who evaluates the results is in yet another department), rapid turnaround becomes elusive because one party may not learn immediately when the previous step is complete, because priorities may not be aligned, or simply because the experiment enters a queue (an inbox on someone’s desk!) every time it is handed over. Or, subteams may not have the incentives to share “failures” with other subteams. As a result, the team might not be able to execute quickly enough or to draw the right lessons from the results.

Finally, the organization should *combine multiple technologies* in order to maximize variation, or the opportunities for learning. The best “technology” of learning changes as the project progresses. At the beginning, learning may be possible from graphs, pictures, or customer questionnaires. Then, as more information is available, mockups or realistic renderings (simulations, pictures) may be better. At some point, partial prototypes and then system prototypes are needed in order to gain additional information.

The more exploratory the experiments are, in other words, the more specialized they are toward gaining information without contributing any “progress” to the current version of the plan, the more difficult they are to implement: (1) They are off-line—that is, early probes are designed more for learning than for success; (2) they require the team to “fail” early and often, something most project management organizations are not well suited for; and (3) they require rapid and effective learning from failures to effect change in the project. Exploratory experimentation is an iterative process of successive approximation, something quite alien to most project management organizations.

For exploratory learning to work, the organization must constantly ask itself several questions: “What do we know, what do we need to know, and what might we not know that we do not know?” “How do we best go about finding answers to these questions?” and “How do we best incorporate what we have learned into the project plan?” As we have seen in the Escend example, these are difficult questions for most organizations to ask.

In the face of significant unk unks, an organization can influence the technology used to make the project flexible, so that as events arise, changes can more easily be made at all stages of the project. Project flexibility provides the ability to modify the project in response to new information as the project progresses.¹⁵ A project’s flexibility can be increased in many ways, one of the most significant being project modularity. The more modular a project’s architecture, the more one can change one aspect of the project without affecting other aspects. Technology choice can drive the underlying physical or informational architecture. A modular technology architecture allows changes in one module of the technology without significantly affecting other modules, in terms of either the physical (space, power, heat, etc.) or the informational interfaces.

Organizational processes can also have a significant impact on project flexibility. Classic stage-gate processes, while useful for classic projects where one can do up-front analysis to achieve at least a feasible, if not optimal, project plan, are poorly suited to projects with significant unk unks. Once the initial stages of the stage-gate process are completed, there is strong organizational pressure for everyone to support the project plan. When the project plan turns out to be infeasible, it is difficult to be the first to say so. In fact, it is human nature to disregard alternatives once a plan has been accepted and agreed upon. The more formal the stage-gate process, the more buy-in one gets into the resulting plan, and the more difficult it becomes to explore alternatives or to even recognize those events that are strongly indicating that the current plan may be suboptimal or infeasible.

Finally, stakeholder management is critical to maintain project flexibility. Uninformed stakeholders might put up with early changes to the project, but later changes are typically seen as problematic. Stakeholders must see the project the way the project organization does; they must be prepared for what is to come and they must have a way to measure progress.

Instead of seeing constant design changes to the agreed-upon project plan, they should see progress on learning about the ultimate solution as it emerges, or on progressively freezing the project specifications. This takes significant reframing of the project in the minds of the stakeholders. These issues will be addressed more fully in Part III.

Endnotes

1. For example, the learning organization has been defined as one “skilled at creating, acquiring, and transferring knowledge, and at modifying its behavior to reflect new knowledge and insights” (Garvin 1993, p. 80). More generally, learning has been defined as experience generating a systematic change in behavior or knowledge (for example, Argote 1999).
2. Section 5.1 is again based on Loch et al. 2005.
3. Cited from Crossan et al. 2005, p. 134. We will discuss a more systematic view of Escend’s situation in Section 5.2.
4. Drucker 1985, p. 189.
5. The open-end process worked because Elaine was the person in charge. When she became CEO, Escend’s morale was so low that no amount of cajoling was effective. After the layoffs, Elaine got the five remaining employees to make decisions by consensus (via daily and weekly meetings), which created a cohesive team that accepted both responsibility and accountability.
6. This hypothesis was confirmed in August 2004 by a report from industry experts that Escend commissioned for \$40,000. Elaine concluded that “experts are good at messaging what you already know, but not at what you don’t know.” So she convened the board to brainstorm about unkunks for other vertical markets.
7. The typology was created by Argyris and Schon 1978, in their seminal book on organizational learning.
8. This definition is taken from Miner et al. 2001, p. 316.
9. See Crossan et al. 2005. These authors go as far as recommending that an organization should shun planning and only improvise. This is wrong; without a plan, the team has no foundation to stand on. Improvisation is useful only if it happens within well-defined boundaries.
10. See Austin and Devin 2003, p. 15f.
11. Crossan, et al. (2005): 134.
12. Thomke 2003 has proposed a modification of the standard PDCA cycle for prototyping in product development projects: (1) design the experiment, (2) build a prototype, (3) run the experiment, (4) analyze the results. This modification emphasizes, first, that learning typically happens on prototypes, or incomplete approximations of the final output, and second, that the analysis step at the end is critical to ensure learning and increased understanding.
13. See Thomke and Reinertsen 1998.
14. The diagram combines the modified PDCA cycle from Thomke 2003 with process principles from Thomke 2001, and the embedded improvisation as proposed in Austin and Devin 2003.
15. This concept is based on the concept of development flexibility offered by Thomke and Reinertsen, *ibid*.