7

Selectionism and Learning in Projects

We have introduced learning and selectionism as fundamental responses to the presence of unk unks, and we have demonstrated how they work in projects. But up to now, we have presented them in isolation of each other—we have not discussed under what circumstances each approach may be more promising, nor have we said anything about how they complement each other in a useful combination.[1] These are, therefore, the two goals of this chapter: comparing selectionism and learning, based on a priori identifiable project characteristics, and demonstrating how they can be combined.

We remind the reader that we want to propose a priori choice criteria for a combination of selectionism and learning, at the outset of the project, before unk unks have been revealed. At the outset, we can only identify knowledge gaps, or the potential for unk unks, but not the unk unks themselves. Therefore, the framework must be qualitative, a rough decision guideline.

Selectionism and learning can be applied at the level of an entire project, or at the level of subprojects, or at individual problems to be solved (we saw this in the diagnosis of the startup company, Escend, in Chapter 4). Clearly, selectionist trials of entire projects, even entire companies, are constantly conducted by markets. For example, one engineering provider may compete with proprietary technology, another with low-cost execution by "off-shoring" the labor contracts, and a third may have access to attractive financing structures. The market chooses which approach is most successful. Competitors learn from one another by copying what works and avoiding the mistakes that they see others make. This is the essence of competition.[2] The same holds not only for projects but also for companies and their strategies in general.

Venture capitalist companies also apply selectionism at the level of entire startup projects. They take an explicit portfolio approach, looking for tenfold returns and killing the startups that do not perform. For the entire portfolio, returns of 20 to 40 percent are often achieved. It is not unusual for a large VC to have several portfolio companies that pursue the same target market. Learning occurs when each startup evolves as it goes through successive financing rounds, sometimes with substantial changes in its business model.

Large technology companies (such as IBM, Xerox, AT&T, and Siemens) have repeatedly tried to replicate the VC approach of project selectionism inside their organizations. However, for economic and organizational reasons, there are difficulties in applying VC-style selectionism to entire projects within established organizations. By definition, selectionism means that most of the trials are abandoned. In the eyes of controlling management, abandoning projects is inevitably synonymous with failure and wasted resources, and thus becomes increasingly difficult to justify when times are tough. Compensation in large companies is geared toward continuity of careers and fairness, while it varies much more in startups. VCs are focused on financial returns and ruthlessly kill projects that do not appear to deliver. In large companies, the continuity and strategic rationale of the business are almost impossible to ignore. As a result, project cancellation is harder in large organizations, which compromises the VC business model and, ultimately, may compromise both financial returns and strategic rationale.[3]

This is felt not only by large companies but also by the VCs themselves—their investments have turned increasingly to proven and more incremental startups since the burst of the bubble in 2001. Independent of the economic cycle, abandoning projects as trials is affordable only when the cost of each trial is small relative to the overall portfolio.

Thus, many companies that execute major projects find that they cannot apply selectionism at the level of entire projects alone but must also consider selectionism at the subproject level, and learn within each subproject how to improve the chances of success. In complex and highly novel projects, we typically see a combination of both approaches: learning and selectionism. Section 7.1 describes such a typical example. In Section 7.2, we discuss principles of how selectionism and learning complement each other, and how they might be combined. In Section 7.3, we reexamine the Circored example to illustrate how our framework could have been applied to this project. We present our conclusions in Section 7.4.

7.1 Selectionism and Learning at Molecular Diagnostics[4]

The example is a European pharmaceutical startup company that raised €7 million in 2002 to develop a new technology for a highly sensitive diagnostic of various diseases (the first round was intended to get the company to the proof of concept milestone). The technology consisted of identifying a modification of one of the four bases in DNA, cytosine, which may make the gene in which this base is included dysfunctional (this kind of modification of a base in a gene is called "DNA methylation"). Identifying a possible dysfunction in a gene via this marker is highly sensitive and can serve as a powerful early indicator.[5]

Fortunately, management was well aware of the unforeseeable nature of events that might hit them. They put several things in place to deal with unk unks. First, they instituted a "strategic pot of money," a (small) budget to deal with unplanned events, "just enough so we can quickly hire an expert or buy a license."[6] This played a role analogous to Metal Resources Co.'s residual uncertainty management described in Chapter 1.

The company also made explicit efforts to scan the market environment and diagnose trends as well as uncertainty. They conducted weekly patent screening and employed an academic advisory committee to learn about new technologies and bounce ideas; the input from the advisory committee also helped them to chart the initial course. They also built up intensive contacts with pharmaceutical companies and clinics to learn about potentially interesting products and technologies. This diagnosis effort did not come without cost—the patent screening alone cost them €400,000 per year. The advisory committee and industry contacts were cheaper but cost a lot of scarce management attention. It turned out that no patents forced them to fundamentally change their plans; they bought a few patents but essentially went on as planned.

The company used both trial-and-error learning and selectionist trials. They foresaw trial and error from the beginning, albeit only within a well-defined area: They were willing to experiment with the clinical indications to be diagnosed; essentially, all autoimmune diseases (such as cancer, diabetes, etc.) were considered fair game. But the diagnosis technology was fixed, namely, DNA methylation.

They conducted parallel trials by exploring several potential technologies to read DNA methylation patterns and narrowed them down to two.

Some did not work, and among the remaining candidates, they used comparative tests. Again, these parallel trials were anything but cheap, costing about €2 million per year.

On the market side, they talked to several potential partners to fund further product development for each potential indication, although they often asked for exclusive contracts. Across the parallel negotiations, learning was fostered: Some team members served on several teams, and they systematically maintained intranet documentation of the lessons learned, the test results, and the meeting protocols for each cooperation attempt.

Beyond its planned experimentation and residual risk management, the company was twice forced to change its plans in a major unforeseen way. First, they ran into a firm in the United States that worked in the same area but had fewer resources. Each had two working methylation technologies, so they merged (formally, the European firm was the acquirer). Of the four technologies, they picked the best two, one from each company.

The second major deviation from the plan was forced by a failure—they had planned to form many clinical development and marketing partnerships, but these partnerships did not materialize. Therefore, the company ended up in one big partnership with a major pharmaceutical company.

The project was successfully completed in mid-2004 with an IPO and became an ongoing concern. In January 2005, the company reported successfully passing a major clinical development milestone, identifying a biomarker that closely correlates with prostate cancer aggressiveness. This concluded successful market identification for all five initial products in its diagnostic collaboration with the pharmaceutical partner; it triggered an undisclosed milestone payment from the partner.

The Molecular Diagnostics example illustrates how an overall project—"get the startup from technology to proof of concept"—is broken down in pieces, and how the pieces are attacked with a combination of established PRM methods, selectionism, and learning, depending on the uncertainty, complexity, and cost structure of each piece. The example also tells us that it is beneficial to allow the pieces to feed off one another. The parallel partnering trials shared information and lessons that helped each to negotiate better, and learning and modifications were applied to each of the methylation technology candidates as they evolved; indeed, after the merger, a different final set emerged than could have been foreseen.

7.2 Choosing and Combining Selectionism and Learning

As we already discussed in Chapter 6, Japanese consumer electronics companies used a "product churning" strategy in the early 1990s. They introduced scores of trial products into the market, "mutant products such as the refrigerator with the built-in microwave oven, the ambidextrous refrigerator whose doors could be opened from either the left or right, etc."[7] The market determined the winners, and the losers were withdrawn. Many of these products failed, but the survivors covered the market and produced new niches. In this "Darwinian" selection, a trial is launched, the unk unks are revealed, whether they stem from technologies or customer behavior, the successes and failures of the trials are unambiguously observed, and then the "losers" are culled from the "winners."

The Japanese manufacturers could do this ex post selection because they had developed the capability of producing product variants quickly and cheaply, and their customers, at first, did not mind buying products that were subsequently no longer available. However, eventually the Japanese consumer electronics manufacturers were overwhelmed by the cost of launching, and then servicing, so many trial products, and pulled back from the approach in the mid- to late 1990s. Instead, they emphasized selection at the subproject level.

As the Darwinian selection of final products became too expensive, they would, instead, pursue competition within the lab—that is, they would not go through the expense of launching complete product alternatives but would explore and select alternatives before they launched them. This *early* selection was less expensive, but it had one major weakness that the Darwinian selection did not have: incomplete information. In the lab, not all unk unks will reveal themselves like they do in the marketplace. It is our task to explain how the interaction between unk unks and complexity systematically influences the value of the information produced in a way that we can understand. Thus, we can offer a causal framework and build understanding and intuition of how selectionism and learning compare.

Figure 7.1 illustrates four canonical examples of learning and selectionism in projects: instructionalist, Darwinian, sequential, and exploratory.[8] In the traditional PRM box, neither learning nor selectionism is used extensively. This, in fact, is the standard contingency approach to project management; the plan is followed, and preplanned contingencies are implemented as foreseen uncertainties arise. We have already discussed the *Darwinian* example. This is the pure selectionist strategy; projects are run in parallel and allowed to compete, unk unks are revealed, and the best project is chosen ex post. The *sequential* example is the pure learning strategy. Parallel trials are not used, but the project is simply modified over time as unk unks are revealed. The final example, the *exploratory* strategy, combines both learning and selectionism. Parallel trials are conducted at the subproject level, when competition amongst alternatives takes place early, before unk unks have fully emerged, and are thus based on incomplete information. These trials are then incorporated into the overall learning strategy where the project is modified over time.

By comparing these canonical combinations of selectionism and learning, we will consider the differences in the *costs* of applying each approach, and the differences in the *value* of the information that each approach tends to produce.

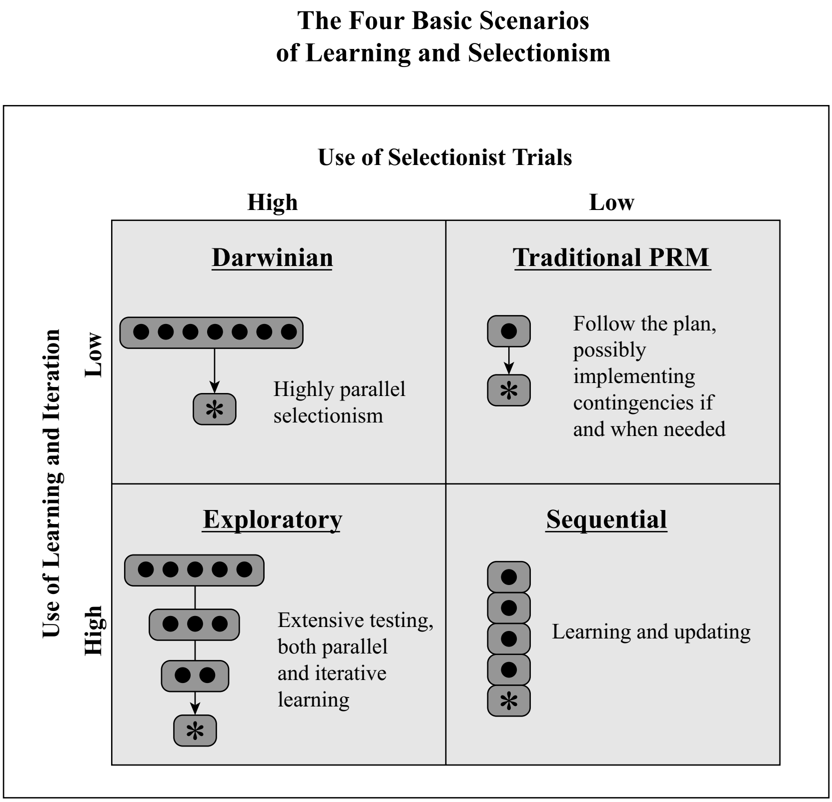


Figure 7.1 Four canonical examples of learning and selectionism

7.2.1 Understanding the Cost of Darwinian Selectionism and Sequential Learning

We start with the cost side because, with managers, this springs to mind immediately—if parallel trials or repeated iterations are very expensive, they are not affordable. Project managers have intuition about this comparison; indeed, managers sometimes ask us, "How can you possibly afford to do something multiple times in parallel? That's too expensive," not realizing that selectionism can be done on a scale that may well be affordable.

We saw in Darwinian selection that the value of the information is quite good, as the unk unks are revealed in the harsh light of the marketplace, but the costs of launching these products could be quite high. The costs of Darwinian selection include the costs of pursuing multiple solution candidates—which include the cost of personnel and material—and the potential negative impact on the brand. Trials that "do not work"—and typically, most of the solution candidates do not work—may generate negative reactions by the employees and customers involved with the products.

The cost of learning includes activities to identify unk unks, such as experimentation, hiring experts to design experiments, or screening the environment. Also included is the cost of running the iterations themselves, such as testing facilities, personnel and equipment, information leakage to the outside, negative word of mouth if an iteration has a negative result, and the delay from repeated iterations, such as deadlines missed and penalties incurred, a seasonal window that is missed, or the project being outraced by a competitor.

These costs are not to be found in a "universal handbook"; rather, they depend on the organization's capabilities. For organizations that do not have the capabilities of trying out experimental solutions and incorporating feedback from the environment, or that cannot run parallel solution teams, the costs of either approach may well be prohibitive. Moreover, the costs are not precisely known, because the precise nature of unk unks cannot be foreseen. Still, when differences in costs between selectionist trials and learning over time can be estimated, they must play a role in the decision.

7.2.2 Value Comparison of Darwinian Selection and Sequential Learning

In addition to, and separate from, the *costs* of obtaining a problem solution, we must consider the *value* of the solution found. Value refers to the quality or performance of the output achieved by the (sub) project, for example, the quality of the technological solution developed, or the customer response achieved by a new configuration. The value comparison may be even more important than the cost comparison; we discuss it second only because it is less familiar to project managers. They have intuition about costs, but to date, there exists no framework for value comparison.

In this section, we first examine the "pure" case of Darwinian selection, where unk unks are revealed in competition among final project outcomes, the best of which is chosen ex post. We compare this to sequential learning, where a project is changed over time in response to unk unks as they are revealed. Comparing these two relatively pure examples will allow us to set up the more typical situation where selectionism is conducted at the subproject level, where unk unks are not fully revealed.

Clearly, Darwinian selection will be favored whenever parallel trials are cheap and/or delays are expensive (upper left box in Figure 7.1). Learning is minimized within each project, and instead, speed and cost are emphasized. If time is all-important, and the organization has the resources, multiple projects are undertaken so as to increase the likelihood that at least one will be a success. For example, the successful credit card company Capital One uses parallel experimental products. It rapidly develops many new ideas, tries them out in the marketplace, sees what works and what doesn't, backs the winners, and ruthlessly kills off the losers. In this way, it generates more "hits" than its competitors.[9]

If, however, parallel trials are prohibitively expensive or ex post selection is not possible, then sequential learning will be favored (lower right box in Figure 7.1). For example, Internet browser development in the 1990s permitted concept modifications until a short time before launch, enabled by fast prototyping of feature changes. Unk unks were to be expected more on the market than on the technical side—browser use was emerging, and the market was still learning how best to use it. The software kernel and architecture remained stable, but component reuse and quick prototypes allowed the developers to customer-test a new version within a few weeks. Thus, time delay costs were kept low. Parallel trials, in contrast, would have required putting an entire additional team on the project, a very expensive proposition in the face of scarce capacity, and releasing multiple versions of the product onto the market simultaneously would have been confusing to the market.[10]

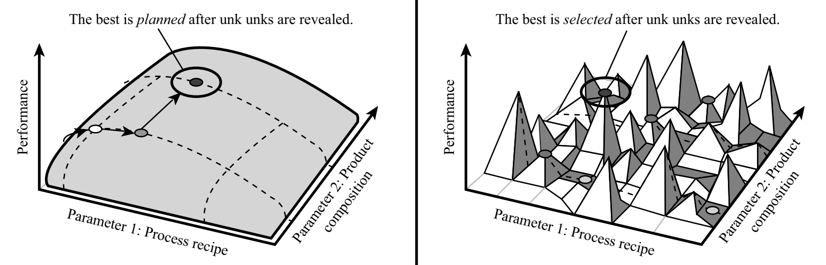
But what happens when the cost situation is not so obviously in favor of one or the other, or when either Darwinian selection or learning offers a much higher value? In this case, we need to consider the value of information obtained. As we stated previously, in Darwinian selection, the unk unks are revealed and then the best alternative is selected. In this case, we choose the trial based on complete information. But what about learning? So far, we have taken the availability of full information about the emerged unk unks for granted. But this is by no means always the case.

Let us first examine the value of selectionism and learning in the complete information case (ex post selection after unk unks have emerged). Figure 7.2 provides a simple illustration of a project performance landscape, as was introduced in Figure 6.1. The landscape is reduced to *two* influence factors, in order to allow for a three-dimensional graphical representation (the third dimension being the performance, or quality, of a solution).[11]

The left-hand picture shows a simple landscape with only one performance peak, and the right-hand picture shows a complex landscape with many peaks. In projects where complexity is relatively low (see Figure 7.2a), the performance landscape is more easily understood. As unk unks are revealed, we can utilize this new information to change the project plan such that project performance is likely to be improved. In other words, it is relatively easy to chart an optimal path as unk unks emerge. As different influence parameters do not interact, the team can make incremental changes (one parameter at a time), first improving one and then the other. Thus, information revealed early in the project tells us much about what to do later in the project.

Consider, for example, an engineering project where the team knows that it will have to adjust the process recipe as well as fine-tune the composition of the final product to suit emerging, currently unknown, process needs of the client.[12] In a simple landscape, the team can first adjust the process recipe parameters, and when they work well, adjust the final product composition. The second change does not invalidate the recipe choice, as Figure 7.2a shows.

However, in complex projects, new information may or may not be very useful to us. The unk unks, as they are revealed, may be important to us, but the project performance landscape is so complex that we cannot easily use this new information to improve the path that we are currently executing.



a. Sequetial learning is favored in simple performance landscapes.

b. Darwinian selection is favored in performance landscapes.

Figure 7.2 Learning and selectionism in simple and complex landscapes

In our hypothetical engineering project, complexity means that the process recipe and the composition of the outcome product interact. If the team first chooses the best recipe and then changes the product composition, the recipe now is no longer appropriate and must be changed again. Therefore, there are multiple performance peaks and valleys in Figure 7.2b; the best choice of one parameter changes with the value of the other. In this case, having multiple selectionist trials and then simply choosing the best one ex post, once all the unk unks have been revealed, offers better value than learning and adjustment.

Complex projects favor Darwinian selection, provided the selection can be made with complete information after the unk unks have emerged, while simple performance landscapes will make sequential learning more attractive. Thus, it is not surprising that well-studied engineering design problems tend to follow more sequential iteration, while more complex projects, like the search for new drug candidates, will yield more parallel search.[13]

However, as in the search for new drugs, ex post selection can be prohibitively costly, as it requires completion of multiple projects. In such cases, parallel search often takes place with selection performed early, at the subproject level before the ultimate project performance is revealed, and then the early "winners" are further refined through learning over time. We discuss this case in the following section.

7.2.3 Exploratory Learning and the Value of Partial Information

While the previous section examined the pure case of Darwinian selection, in general, selection of multiple trials may be performed at several stages of a project:

- *After design studies*. For example, multiple car concept designs are rendered in CAD models or clay. A concept is chosen based on the "holistic impression" of experts. However, even this holistic impression may not be representative of the ultimate market reaction. BMW found this out when the 7-series introduced in 2002 met a hostile market reaction to the trunk design. This reaction was not predicted after internal and customer tests, yet it forced the company to accelerate the mid-cycle face-lift of the car to 2005.
- *After technology tests*. Technology choice is often based on lab tests that cannot incorporate all aspects of the real usage environment.[14] Therefore, the chosen solution may later fail. For example, a tire tread company developed process improvements in a central R&D lab, with the aim of reducing wire breakage in the multiple cold drawing manufacturing stages. It turned out that the result of the technical change depended on the ambient temperature and air composition in the factory, which were not simulated in the R&D lab. This ultimately prompted the company to move technology tests into the factories, despite higher costs.[15]
- *After customer or client tests*. In order to predict end-user reactions as accurately as possible, many companies insist on testing project decisions with the client or end customer. But even such checks by the client are often inaccurate. Users' reactions might not be representative of their later behavior under real usage conditions, even when only some aspects of the usage environment are not correctly represented. Thus, client agreements often do not prevent later disputes in complex projects, and market predictions based on consumer feedback are notoriously unreliable.[16]
- *After launch*. The opening example of product churning concerns consumer electronics. The success of such products can be diagnosed quickly, within a few weeks after launch in a leading market (such as Tokyo's Akihabara district). This is not the case for products or projects with a longer life. In complex engineering projects or complex consumer durable products (such as cars), success may not be known until a significant part of the product's life cycle has passed, perhaps only several years after launch. Thus, even after launch, the selection decision may yield only partial information. Of course, one could delay selection even further, but that is typically not affordable.

Returning to the pharmaceutical example of the previous section, the cost structure of early lead molecule development in the pharmaceutical industry is low, relative to later stage development. Thus, many lead molecules are produced, and the promising ones are modified—that is, "optimized"—to enhance their chemical reactivity to target binding sites, and thus their potential pharmaceutical potency. This approach is a combination of the two "pure" strategies discussed in the previous section.

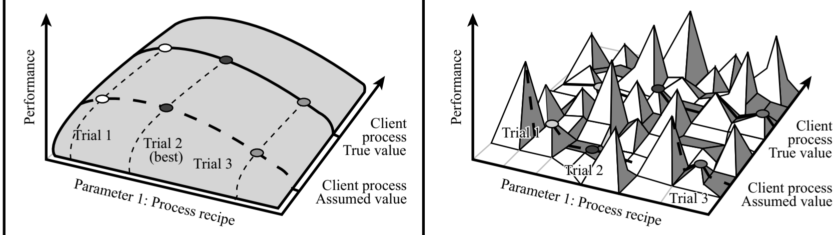
Releasing multiple trials of the same drug on the market to reveal unk unks is simply not acceptable in the pharmaceutical industry. If unk unks include death or other undesirable side effects, it is simply unacceptable to have ex post selection after the unk unks have revealed themselves in the death of patients—the costs are simply too high. In this case, the most effective approach is to *combine* selectionism and learning in "test waves" of parallel candidates that are narrowed down early and then optimized over time before project completion (lower left box of Figure 7.1).

The problem, however, with early selection is that it yields only partial information: Not all the unk unks will be revealed at the subproject level. The critical question then becomes "What is the value of this partial information?" Which conditions favor early selection and which do not? Recall that the value of information is measured by the degree to which we can improve project performance based on the information obtained. In the previous section, we saw that the value of early information in sequential learning depended on the complexity of the project. If a project is very complex—that is, the performance landscape is as in Figure 7.2b—early information is less valuable because it is difficult to use it to find the "best" solution a priori. Charting an optimal course is simply too difficult, and in this case, Darwinian selection is favored.

In the case of early selection, where only partial information is revealed, we have the exact opposite scenario: Selectionist trials will *not* be favored in complex projects.[17] This is precisely the most challenging situation for a project team, and it is the situation in which selectionism gets into trouble: If trial selection occurs before unk unks are revealed *and* the project is complex, learning systematically promises a better solution than selectionism.

The reason, briefly, is that making wrong assumptions about an unknown project influence "disturbs" a complex project more than a simple project. Through the many interactions in a complex project, the error in one influence factor has wider repercussions and degrades the quality of the selection choice. To understand this statement in more detail, let us return to the project performance landscapes previously illustrated in Figures 6.1 and 7.2, and now reproduced in Figure 7.3.

In Figure 7.3, in contrast to Figure 7.2, we now consider a situation where one of the influence factors is an unk unk and is not revealed at the time of selection. Coming back to the engineering project example that we discussed along with Figure 7.2, let us suppose that the change in the client's process needs is not foreseeable, either to the client or to the team. In other words, the team takes the process needs as given and does not know that there is an influence that may take on different possible shapes, which have an impact on the project. Rather, the team has an implicit "default" assumption about the client's process needs, an assumption that is unconscious and nonarticulated. This default value is whatever value the parameter takes in early client discussions, although the team is not conscious of it.[18]



a. Partial information about unk unks is valuable in simple project performance landscapes.

b. Partial information about unk unks is much less valuable in simple project performance landscapes.

Figure 7.3 Selectionism with unk unks in a simple and a complex project

We have discussed real examples of this already in this chapter. Take the example of the tire manufacturer who was developing process improvements in a central R&D lab with the aim of reducing wire breakage in the multiple cold drawing manufacturing stage. It turned out that the result of the technical change depended on the ambient temperature and air composition in the factory, which were not simulated in the R&D lab because they were not known to be factors of performance by the team. Similarly, customer preferences in simulated consumption environments might differ from their preferences in actual environments.

Thus, the team's conscious project decisions, in our case the choice among selectionist trials, happen in the "sub-landscape" of the line that corresponds to the default value of the unk unk (in Figure 7.3, this is the dashed line). The trials in the sub-landscape, each corresponding to one decision along the known project influence variable, are marked in Figure 7.3, and one of them is selected as the best candidate.

Now, the problem of selectionism becomes clear. In the simple landscape with only one peak, the parameters do not interact. Therefore, the ranking of the trials is little changed when the true value of the client's process needs emerges (the solid sub-landscape line). The recipe choice is not invalidated by the product composition, and thus, the quality of the selectionist choice is high. In the complex landscape, however, the choice in the default-assumption sub-landscape does not reveal the true best choice. When the true client process needs emerge, the chosen recipe turns out to be inferior (the best trial on the dotted line in Figure 7.3b is one of the worst on the solid line that corresponds to the true client needs). That is, the value of the information yielded from early selection may be of little value in complex projects.

Learning, in contrast, proceeds precisely on the premise that the project keeps evolving, and a new choice is made, after information about the unk unk becomes available. In learning, unk unks are sought out, knowledge gaps are inventoried, and purposeful attempts are made to fill them. In learning, the project team seeks to discover and understand the unk unks. While learning may take more time, it may be more effective than early selection in complex projects.

7.2.4 The PRM Contingency Planning Approach

If both selectionist trials and learning are costly, responding to unk unks becomes very difficult. Both will have to be reduced, although their relative emphasis may remain the same: fewer parallel trials, fewer test waves, and thus, less assurance of a good solution. This means that no sufficiently effective response to unk unks may be achievable, confronting the project with an excessive danger of failure. This has an important implication: If both selectionism and learning are too expensive to be affordable, management should consider changing the project scope in order to avoid unk unks, and thus the need for selectionism and learning. This can be done, for example, by reducing functionality or using proven technologies.

One example of this is the development of the Boeing 777. Boeing developed this plane in response to the Airbus 340, and development time and cost were kept low by using previously proven technology components. Thus, it was possible to develop only one technical solution into a flying prototype; it was simply too expensive to build more than one plane in parallel. In reaction to problems, the prototype was modified, rather than building several models. However, time was very expensive, too, so testing could not go on indefinitely before the company started to earn money. Thus, the first flying prototype was later sold commercially.[19]

The development of the Concorde plane in the 1960s, in contrast, entered much more novel technological terrain. Parallel trials were not affordable (as in the case of the Boeing 777), but because of technical unk unks, the value of time was given less priority than the challenge of getting the plane right. As a result, the schedule slipped by four years due to testing needs, which contributed significantly to a budget overrun from an initial estimate of £135 million to £1.1 billion.[20]

7.2.5 A Combined Choice Framework

We now have at our disposal a logic, a set of criteria by which to choose from among the different approaches for each subproject: The decision should be driven by cost structure and complexity. We summarize these results in Figure 7.4.

When the costs of learning and delay are high, relative to parallel trials, and project complexity is high (upper left-hand box of Figure 7.4), ex post Darwinian selection is favored. Because of the complexity, sequential learning will be lengthy and difficult because causal connections are more ambiguous and harder to identify among the multiple simultaneous interactions. On the other hand, early selectionist trials may not yield any valuable information (see Figure 7.3). Therefore, selection can only be accomplished late.

	Relative Cost	
	Learning and delay more expensive	Parallel trials more expensive
High complexity (many interactions)	Darwinian selection *ex Post* Selection	Sequential learning or reduce complexity
Low complexity (few interactions)	Exploratory *Early* selection and learning	Sequential learning

Figure 7.4 Value comparison of learning and selectionism with complexity and relative cost differences

When project complexity is low and the costs of parallel trials are high (lower right-hand box of Figure 7.4), sequential learning is favored. Here, it is better to pursue a single project and to chart a new course as unk unks are revealed, or, depending on the costs of trials with early selection, to combine selectionism and learning. Early trials are selected and then improved upon over time.

When complexity is low and the costs of learning and delay are high (lower left-hand box of Figure 7.4), a combination of early selection and learning is favored. If time is critical, and depending on the relative cost differences between ex post and early trials, Darwinian selection may be preferred if the organization has the resources to complete parallel projects. As our discussion at the beginning of Section 7.2.3 suggests, late selection (after unk unks have emerged) may be very expensive, as it may require full-scale operation at the client, or a full market introduction in the case of an innovation project. In the case discussed here, complexity is low, so early selection is capable of identifying the best trials.

When complexity is high and the costs of selectionism are high (upper right-hand box of Figure 7.4), sequential learning, although difficult in this case, may be favored. Alternatively, one could attempt to reduce the complexity of the project.

One way to reduce project complexity is by reducing the level of interaction among subprojects, at the level of system components, of project tasks or of relationships (see Chapter 4, Figure 4.4). The more modular a project, the less various subprojects interact with one other, and thus, the less complex the overall project. As we have seen, complexity has a big impact on the value of information obtained, either early information in sequential learning or partial information from early selection. If we can reduce the overall project complexity, it improves our ability to do both learning and selectionism.

It is helpful to classify project interactions within three categories: integrated, sequential, and modular. Figure 7.5 illustrates the design structure matrix for each of these three types. As was discussed in Chapter 4, the design structure matrix indicates the presence of interactions among system components, project tasks, and parties involved. To keep this discussion simple, Figure 7.5 shows only the tasks.

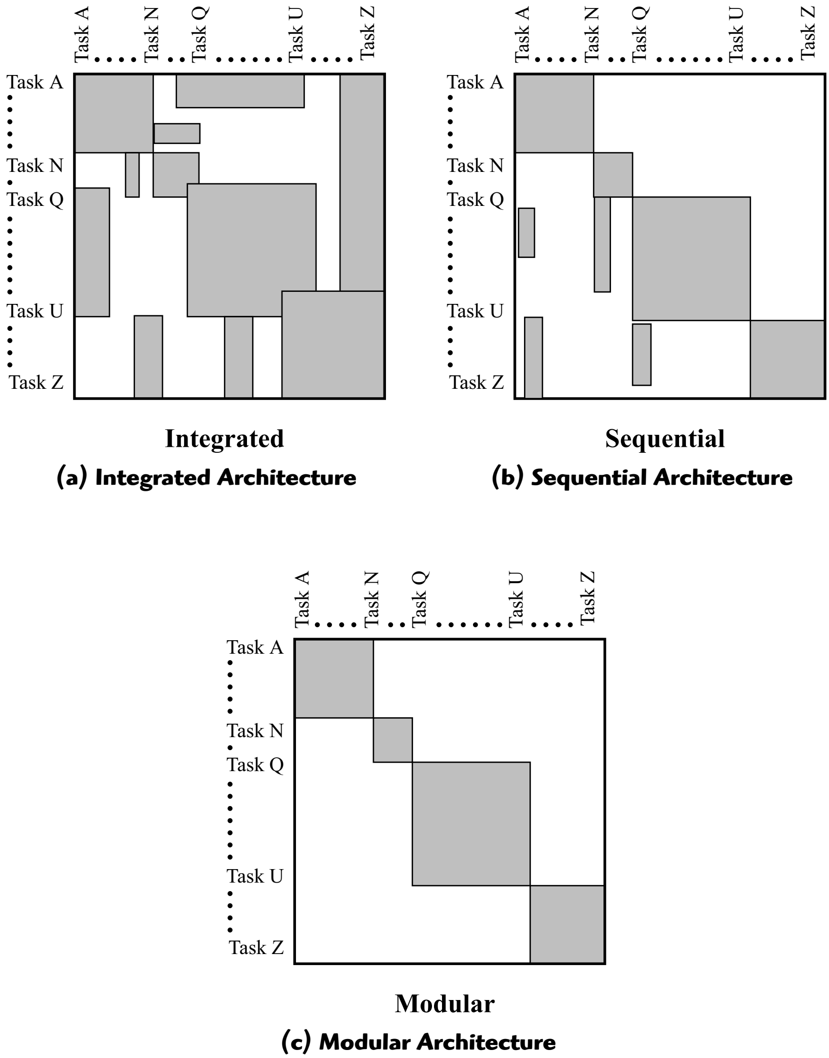


Figure 7.5 Task dependency matrices for integrated, sequential, and modular project architectures

In all three examples (Figure 7.5a–c), tasks A to M are interdependent—they are integrated—and thus are good candidates for a "subproject." We see four such subprojects in all three examples. However, in the integrated case (Figure 7.5a), subprojects are also interdependent. For example, task A depends on task Q, and task Q depends on task A, even though they are in separate subprojects. In the sequential case (Figure 7.5b), subproject dependency only goes one way. For example, task Q depends on task A, but task A does not depend on task Q. In the modular case (Figure 7.5c), subprojects do not interact. They can be optimized separately and carried out in parallel, and unk unks affecting one are not likely to affect the other (unless the project structure itself is unknown).

In an integrated project architecture (Figure 7.5a), subprojects depend on other subprojects in a complicated way. Project performance depends on a complicated interaction of the various characteristics of each subproject. Thus, the overall project can be said to be very complex. In this case, one will have to either do sequential learning or ex post Darwinian selection at the level of the entire project, depending on the relative cost and delay trade-offs.

In a sequential project architecture (Figure 7.5b), because dependency is one way, early selection can be used at the subproject level as long as one proceeds sequentially from one subproject to the other. Thus, a combination of early selection and learning may work well in such an instance.

The "best" case, of course, is a modular project architecture (Figure 7.5c). Here, subprojects do not interact (much), and one is free to conduct early selection on the various subprojects independently of one another. The real challenge here is then to coordinate the various subprojects so that the overall project can be completed on time.

We put this designation of "best" in quotation marks because it represents the view of the project manager. Modularization of a project is often not possible, or only possible by reducing performance of the system that is produced, for example, in terms of size (to make system components separate and noninteracting), in terms of tasks (building buffers into the schedule so that tasks do not happen at the same time and cannot interfere with one another), or in terms of organization (e.g., giving subteams extra resources so they do not need to compete for the same scarce experts or facilities). Because of these trade-offs, determining the system architecture of what is built during the project is usually beyond the scope of the project team. The architecture must be set by senior management in consideration of other projects and business objectives. We will return to the role of senior management in Part IV of the book.

7.3 Reexamining the Circored Project with This New Framework

The Circored project (described in Chapter 2) underestimated the unforeseeable nature of uncertainty and did not compile uncertainty profiles for the subprojects. As a consequence, the project suffered from delays, budget overruns, and damaged careers. How would our selectionism and learning decision framework have helped them?

Let us attempt to apply our decision framework. We do so somewhat speculatively because the relative costs of selectionism and learning were never analyzed; we have to estimate them roughly, in hindsight. Still, the analysis suggests some useful changes in how the project could have been managed.

We define as subprojects the major components of the facility (preheater, bucket elevator and lock hoppers, CFB reactor, SFB reactor, discharge, briquetting), the novel materials, system integration (process control and ramp-up), and marketing (as the product had novel aspects that the customers did not understand). Drawing up uncertainty profiles suggests that the preheater, the bucket elevator and lock hoppers, and the briquetting machines used established technology and were not affected by unknown material properties, so they were not affected by unk unks. We can leave them out of the analysis. Entering the other subprojects, the ones affected by unk unks, into the framework of Figure 7.4 produces Figure 7.6.

Using selectionism for the facility as a whole was simply out of the question, as a planned construction cost of $165 million was too expensive. However, selectionism could well have been applied at the level of some of the ducts and valves made of novel materials. For example, the ducts from composites could have been ordered in two versions, to test and compare them; this would also have allowed the team to react faster when the duct broke and caused a long delay. Similarly, the valve that let hydrogen gas shoot through could have been tried in two different configurations to test for tightness and internal sticking. Double ordering would not have been expensive, as the materials themselves are but a fraction of handling, design, and partner negotiation efforts.

	Relative Cost: Learning and delay more expensive	Relative Cost: Parallel trials more expensive
High complexity (many interactions)	Darwinian (*Ex post* selection) • **Marketing**	Sequential learning/ reduce complexity • **System Integration**
Low complexity (few interactions)	Exploratory (*Early* selection and learning) • **Novel materials**	Sequential learning • **CFB reactor** • **SFB reactor** • **Discharge**

Figure 7.6 Possible learning and selectionism in the Circored subprojects

For the huge reactors and the discharge system, selectionism was, again, too expensive. There was no choice other than to try them out and modify them as the process properties emerged (for example, the walls in the SFB reactor to control retention time). However, once the team had realized that experimentation and learning would be necessary, they could have organized the experimentation very differently: They could have tested the three system components *in parallel*, feeding each with partially processed material. The runs would have been explicitly designed as experiments, not as the production runs that the team attempted almost from the beginning. In this way, the bugs and the verification could have been worked out much faster, with less frustration, and without the need to perform the fundamental verification test of the process in the summer of 2000.

System integration, that is, process control and ramp-up, were highly complex and unpredictable, as the chemical process was new, and hundreds of system parameters interacted in the ore conversion and the properties of the briquetted iron. Moreover, the system behavior could not be simulated beforehand with a sufficient degree of realism (although CAL did develop the process control software, which, after calibration under real operating conditions, was then capable of automated operation). In other words, high-fidelity tests were not available. Moreover, pursuing two parallel process control systems would have been very expensive. Thus, a sequential learning approach was unavoidable. The Lurgi engineers understood this intuitively and approached the ramp-up slowly and deliberately, but this caused some tension and debate with the Cliffs engineers.

Finally, the market introduction of the Circored HBI was also plagued by unk unks because the customers did not understand the product (remember, for example, the fear that residual hydrogen traces would make the briquettes dangerous, or the skepticism over the 2 percent carbon content, which was really a plus for the customers). Again, this was quite complex because customers interacted by word of mouth, and because the HBI entered as an ingredient into a complex steelmaking process. Thus, some experimentation and learning in the customer approach was certainly needed. However, it would have been quite cheap to prepare several advertising leaflets and several "customer briefing story templates" in parallel, trying out on the way that might work best. As the marketing approach was relatively independent of the other activities, a more ex post selection approach would have worked, although in reality, a combination of selectionism and learning would have been needed.

With such an explicit management of unk unks, choosing an appropriate approach, subproject by subproject, the project would have proceeded faster and more successfully. The same unk unks would have affected the team as before, but they would have been handled more quickly, in a more focused way, and with more confidence. Had the facility operated for two years before the 9/11 price decay, it would have already been established as a success, and the historic price drop would not have caused the same fundamental questioning. And quite importantly, the project would have *felt* different to the project team and to the supervising management; all the bad surprises would not have been interpreted, even if unconsciously, as incompetence on the project manager's side, but as normal aspects of building a first-of-a-kind facility.

7.4 Conclusion

We have argued in this chapter that our decision framework of how to choose between selectionism and learning for subprojects can help project managers to plan and execute their project better. In other words, we have proposed a "decision method" that allows the project team to estimate costs and value of selectionism and learning a priori, and then make a better choice of project approach than without the method.

We mentioned several times that project managers may not know the costs of coping with unk unks through learning or selectionism, or the precise value created by these strategies, but that they have a good feel for it. What we noticed is that managers often take decisions to cope with unk unks based on their intuition. How does this square with our proposal of a decision method? An additional use of our framework is to examine the popular rules of thumb, or intuition expressed by project management professionals. Intuition is automated experience that is no longer open to introspection. Intuition is very important for making the entire myriad of snap decisions during the day, for which one has no time to thoroughly deliberate. But intuition is also dangerous because it is based on one's own track record, which may not be appropriate for the problem in hand right now.

The CEO of a venture capital (VC) investment firm, in a conversation with us, gave the following rule of thumb: "If you come to me with a business idea and want money, don't propose selectionism early, in the technology development stage. I expect you to have the technology nailed down. If you tell me you want to try several technologies in parallel, I won't give you a penny. But later, in the marketing stage, I can see selectionism making sense to test out several market approaches in parallel."

We do not want to second-guess an experienced VC who knows what he is talking about. It is still useful to know under what circumstances this rule of thumb would hold, according to our framework. Prescribing learning (and not selectionism) early, at the technology stage, makes sense if the startup (1) is still susceptible to major unk unks, and (2) the situation is complex because technology system elements and market system elements all interact, if nothing is yet determined and frozen. If it is also true that the technology candidates cannot be fully tested for performance, we are in the lower right-hand box of Figure 7.4, and our framework concurs that learning is preferable. On the other hand, if the startup could find a way of testing the technology performance realistically with customers, the answer might well be different.

The second part of the VC's intuition is that selectionism makes more sense during the market introduction stage. Indeed, at that point, complexity and unk unks should be reduced, and so selectionism is more promising (provided, of course, several market approaches can be made at reasonable cost).

Another example of intuition, or a generalization from a small number of examples, is the conclusion in Miller and Lessard's study of major projects, which states that large-scale multiyear projects have no choice other than evolving over time. Again, this is consistent with our framework: Such projects combine major unk unks and complexity, so learning promises higher value than selectionism.[21]

The point of this discussion is not to second-guess project professionals. We are trying to demonstrate that when a project manager enters a novel project, where it is not known what should be expected, it is worth making one's intuition (the principles that one takes for granted, "this is how it's done, of course") explicit, and then to *question the intuition*. Our framework gives the project manager a tool to do that questioning: Where do we have gaps in our knowledge, and thus the potential for unk unks? What do parallel trials cost versus the delay in sequential learning? How high is the complexity of the subprojects and their interactions with one another? Where can we test our solution candidates right away, and where can we not? Can I interpret my intuition in terms of the choice between risk identification and management, selectionism and learning? Does it make sense in the light of the trade-offs?

Our framework gives an operational and conceptually sound rule of how the choices should be made. Part II of this book dealt with the conceptual tools we have in order to understand the conceptual possibilities of dealing with unk unks. Making the choice in a real project team is, of course, a more complicated matter. Solving the managerial challenges of dealing with unk unks in real projects and real surrounding organizations is addressed in Part III.

Endnotes

1. This reflects the current state of knowledge. Although project management experts have discussed selectionism and learning, as we have summarized in Chapters 4 and 5, we are not aware of a framework of how the two approaches compare or combine.
2. Dorothy Leonard-Barton (1995, p. 207) called this *vicarious learning*: "Wait and let the pioneers get the arrows in their backs and learn from their mistakes." For example, IBM, Motorola, and Compaq delayed introductions of PDA (personal digital assistant) products in 1993 in the light of extreme uncertainty about what the market wanted.
3. For further discussion on this, see Watts 2001, and Chesbrough and Socolof 2000.
4. This example is based on Sommer 2004. The name of the company has been changed to protect confidentiality.
5. The process was designed to discover changes between healthy and cancerous cells. Methylation is a natural epigenetic process that occurs when a methyl group binds to one of DNA's four bases, cytosine. The presence of methylation is responsible for controlling the activity of genes by turning them off, like a switch, when not needed. By measuring the differences in the methylation patterns between healthy and diseased tissue, a change in gene activity that could trigger diseases such as cancer is detected. The company had developed an industrial process that was able to read and interpret these methylation patterns.
6. Source: company interview.
7. The story of product churning and its demise is told in Stalk and Webber 1993. The quote is on p. 95.
8. Figure 7.1 is based on Loch, Terwiesch, and Thomke 2001, and on Sommer and Loch 2004.
9. This is reported in Beinhocker 1999.
10. This is described in Iansiti and McCormack 1996.
11. The conclusions are the same if we have many influence variables; for a full discussion, see Sommer and Loch 2004.
12. "Unknown" here means that the needs are currently unknown also to the client, so questioning them won't resolve the uncertainty.
13. See Loch, Terwiesch, and Thomke, 2001.
14. Thomke 2003, pp. 101, 119, refers to the accuracy of tests as *fidelity*.
15. For details, see Lapré et al. 2000, pp. 603 and 607.
16. See, for example, "purchase prediction adjustments" from stated consumer intentions voiced in surveys, reported in Chapter 8 of Ulrich and Eppinger 2004.
17. This discussion is based on Sommer and Loch 2004. A detailed analysis of this situation can be found there.

18. Note that we are even making the optimistic assumption that the team correctly determines the original lab value of the unk unk, here, the client's process needs. If the team does not even listen or conducts the early investigation incorrectly, the assumed unk unk value is, in effect, random.

19. The design of the Boeing 777 did not pose high unforeseen uncertainty; the development was highly complex but fundamentally well understood; see Sabbagh 1996.

20. See Morris and Hough 1987; see also Kharbanda and Stallworthy 1984.

21. See Miller and Lessard 2000.