9

Putting the Infrastructure in Place: Management Systems

After Chapter 8 has established the "mind-set," or "culture," of managing a novel project with selectionism and learning, Chapter 9 discusses how to manage the project, in terms of systems and methods, which represent an important part of an organization's capabilities. The system offers managers tools that they can use to put selectionism and learning in practice.

9.1 Managerial Systems in Project Risk Management

Managerial systems are necessary for members of an organization to successfully collaborate, so that their efforts complement one another and are aligned to achieve an overall system performance. In novel projects, the environment is too ambiguous and complex for any individual to fully understand. Human rationality is "bounded": Decisions in such environments are too complex for any one individual to comprehend. Project teams cannot "optimize" over the set of all conceivable alternatives, and so management systems are needed to help guide their actions.[1]

Managerial systems consist of routines and decision rules that are analogous to "skills" or "intuition" of individuals. They help in the smooth execution of certain sequences of actions that are automated, and thus executed without apparent effort or conscious decision making. They are also tacit—that is, not articulated: Often, individuals cannot even explain them. Such routines represent stored experience and are effective in the "normal context," in terms of the actions of the surrounding employees and the characteristics of the project environment.[2]

Routines are repetitive patterns of activity in an organization; processes are examples of routines.[3] Routines are de facto organizational memories (analogous to the intuition of an individual)—each organizational member remembering his or her part of activities, and the context of behavior of the actors around him or her, is enough to enable the organization to behave reliably over time, without the need for any individual to fully understand the entire system. Some individual understanding of the entire system is always desirable, but in many cases, it is simply not feasible. Knowledge in organizations is usually distributed and "routinized."

Project teams can perform completely new and original problem solving only in bursts, during short intervals of intense effort by a small group. Original problem solving puts such a level of stress on the team that it cannot be a continuously ongoing process. Original problem solving may result not only in a new solution but in a change of the routines by which the team operates, and thus, in an innovation. Most people find it too stressful to constantly innovate over long periods without the security of routines. This was illustrated by our discussion of the map and sensemaking in Chapter 8, which showed how the map, even the wrong map, provided a sense of routine to the group, allowing them the security to innovate a solution to finding their way back home.

All this has important implications for a project team that must deal with unk unks: The fundamentally nonroutine activity of responding to unk unks must somehow be routinized at a higher level, by learning the (automated) skill of challenging assumptions and experimenting, or of trying out multiple solution approaches in parallel.

Managerial systems are an important component of organizational capability and form the basis of competitive advantage: "Competitive advantage of a firm lies within its managerial and organizational processes . . . the way things are done in the firm, or what might be referred to as its routines, or patterns of current practice and learning."[4]

Managerial systems consist, first, of the formal and informal ways for guiding action and learning. This can happen, for example, through formal problem-solving methods, information collection protocols, such as benchmarking or competitive intelligence, or prototyping guidelines. Managerial systems also provide tools for control through procedures, incentives and rewards, monitoring, and information exchange.[5] Managerial systems are the organization's means of encouraging (or forcing) employees to engage in certain behavior and take certain actions. Due to their routinized and "automated" nature, managerial systems, and the capabilities that they incorporate, may *hinder* a project, if the project has requirements that are incompatible with, or in contradiction to, the organization's managerial systems and routines. As Dorothy Leonard-Barton (1992) noted, "Core capabilities can become core rigidities."

To summarize, a large body of management research has found strong evidence that a firm expects too much of its employees if it simply demands to "solve the problems that come up" and assumes that this can be done from scratch at will. Over time, humans can solve problems only "on the margin"; they need to build upon a foundation of a stable structure of procedures and success criteria. An organization that wants its project teams to successfully deal with unforeseeable uncertainty must give them managerial systems that specify a skeleton of search and decision procedures.

An infrastructure of such management systems must include five areas, which are all heavily affected by the presence of unk unks:

- *Planning.* What do we plan, and what does a plan look like when it is only stakes in the ground, as opposed to a detailed specification of all foreseen actions?
- *Monitoring and progress measurement.* What do we measure? Milestones? Achievements? Knowledge obtained? Or actions (such as experiment cycles) executed?
- *Coordination and relationship management.* How do the various parties in the project adjust their activities with respect to the other parties? In other words, what *actions* should the parties take in response to the information that has been exchanged about their respective status and situation?

- *Information management.* Strictly speaking, this is an aspect of the coordination management system. However, it is so important that we want to discuss it in its own right. What is the relevant information that must be communicated across subprojects in order to allow them to adjust to one another?
- *Performance evaluation.* What aspects of progress, process, and results should be used to evaluate the performance of the teams and individuals involved in the project? This system is concerned with a mixture of measuring process and output, and a combination of rewards for target fulfillment and upside incentives.

The infrastructure must differ between planned, selectionist, and learning aspects of projects and subprojects. This is briefly summarized in Table 9.1, and we discuss the differences in more detail in the remainder of this chapter.

The instructionalist approach outlined under the "Planned Projects" column of Table 9.1 summarizes the management systems that are well known and established in projects with moderate foreseeable uncertainty. Tasks and targets can be planned, their fulfillment can be monitored, deliverables and structured information is exchanged among parties, and fulfillment is used to evaluate people.

Table 9.1: Infrastructure for Planned, Learning, and Selectionist Projects

	Planned Projects	***Learning Projects***	***Parallel, Selectionist Projects***
Planning systems	· Plan tasks and targets · Work structure and defined responsibilities · Buffer against risk	· Overall vision, intermediate targets · Tasks to learn · Rapid turnaround of experiments to learn	· Collective vision, different roles across projects · Intermediate diagnosis criteria of the potential of an individual project
Monitoring systems	· Target achievement · Progress tracking (e.g., % complete, or deliverables)	· Track "experimentation" · What has been learned? · What problem to solve next?	· Project stopping criteria (relative potential) · Information to be shared among peer projects
Coordination systems	· Fulfillment of deliverables · Coordination via work structure in hierarchy · MBE (management by exception) · Little decision power necessary	· Dynamic and less formal · Long-term trust-based relationships handle changes · Decision power to change approach or targets · Higher problem solving necessary	· Relative progress of the projects · Sharing of learnings · Stopping decisions
Information systems	· Progress, deliverables, actual outcomes of events	· Richer, unstructured information exchange and mutual adjustment	· Overarching over peer projects
Evaluation, incentives	· Target fulfillment · Measurement of output	· Upward incentives on output · "Process quality" incentives	· Shared incentives on output · "Process quality" incentives

Management systems must be adapted in projects (or subprojects) with unk unks. *Planning systems* emphasize milestones of learning, versus delivered results. *Monitoring* must track what has been learned, rather than progress along the planned tasks. *Coordination* must be richer and more flexible, as opposed to a work structure with deliverables. It must include the possibility of rearranging the responsibilities as the character of the project changes. This flexibility must be supported by richer and less structured *information systems*. Finally, the project team should not be *evaluated* on target fulfillment—as the outcome is not under the control of the team, this would cause withdrawal of the best people from learning projects. Rather, upward incentives and process measures should be used.

The right-hand column of Table 9.1 offers a few indications of the differences between planning and selectionist approaches. The overarching idea is that one needs to keep a global view of the various projects while tracking what is being learned in the individual projects. The global tracking allows insights to be shared and individual projects to be stopped when they no longer contribute to the whole. Systems and incentives need to keep the whole of the organization committed, even when the projects to which some of the individuals are allocated have to be stopped.

We illustrate the management systems for learning projects and selectionist projects in Sections 9.2 and 9.3, and then discuss how they can be combined at the overall project level.

9.2 The Management Systems of Learning (Sub) Projects

To discuss the management systems of a learning project, we return to the Escend Technologies example from Chapters 4 and 5. Recall that Elaine Bailey divided the overall challenge of turning around the startup company into pieces, which differed along their uncertainty profiles (Table 4.1). Some pieces could be managed with standard risk management techniques, while she adopted a learning approach on others. The management systems for those parts of the project (customer needs, industry readiness, and product functionality) changed significantly. In this section, we generalize the principles.

9.2.1 Planning System

The traditional planning system of a project with foreseeable uncertainty proceeds according to the spirit of identifying all tasks to be performed, over the entire project, and foreseeing all variants and complications; PRM adds possible risks and the necessary preventive, mitigating, and contingent actions to the plan.

This is not possible in the presence of unk unks, as the tasks may fundamentally and unforeseeably change after an unk unk has emerged. Thus, a rough overall plan (compatible with the sensemaking map of Chapter 8) is necessary that outlines the major steps that promise the potential of

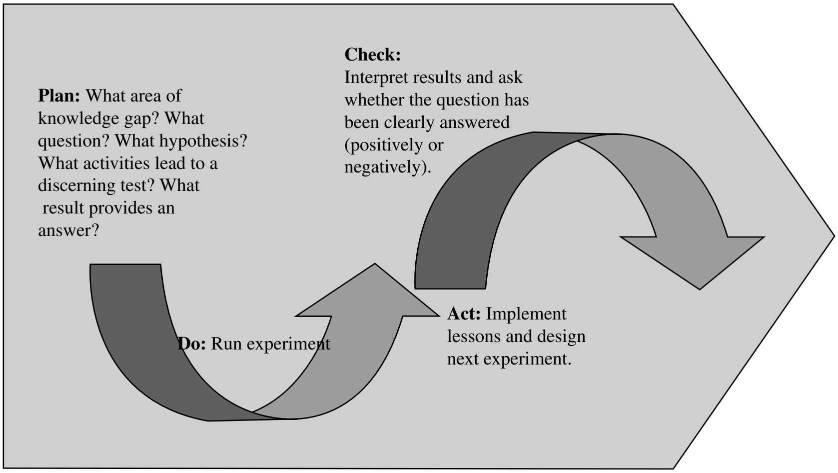


Figure 9.1 Plan the experimental cycle

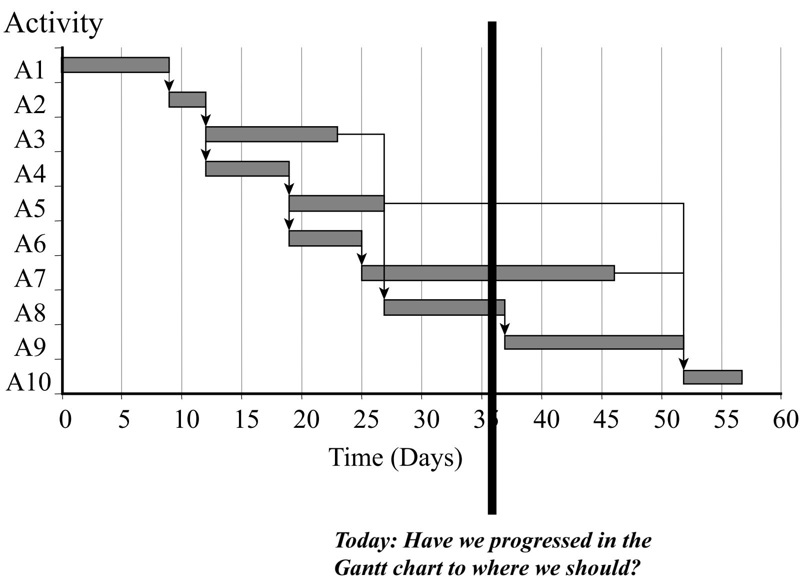
achieving the project's vision. But this overall plan should be rough, so that it does not become an end in itself. We argued in Chapter 5 that the major vehicle for progress in a learning project is the Plan-Do-Check-Act cycle (Figures 5.4 and 5.6). We reproduce Figure 5.6 here as Figure 9.1, simplified for the purpose of the planning system.

There are several studies available that suggest that effective planning and execution of this basic learning cycle maximizes learning. A detailed plan should be drawn up for the experimental cycle: What is the area of the knowledge gap? What is the hypothesis to be tested and the question to be asked? What are the activities that will illuminate the question? How can we tell whether we have an answer? Once the results are obtained, the next iteration of the learning cycle is planned in detail.

Elaine Bailey understood this. For the three subprojects with unk unks, she did not insist on a plan to the end, but took one question at a time. She planned a set of interviewing trips, an analysis by industry experts, and so on, and then reconvened her management team to reassess what the best next step was. As a result of this systematic learning effort, Escend's search for a functioning business model was very effective.

9.2.2 Monitoring System

The left-hand side of Figure 9.2 illustrates the traditional philosophy of monitoring systems in projects with foreseeable uncertainty. The project manager looks at the planning tool (here, a Gantt chart), draws a line for today's date, and asks whether all activities that are supposed to be completed are indeed complete, and whether the activities in progress are roughly at the required progress status. Admittedly, this view is simplified, but it does capture the basic logic of any tool that compares planned progress to actual progress, perhaps with some variance analysis of causes for deviations.[6]



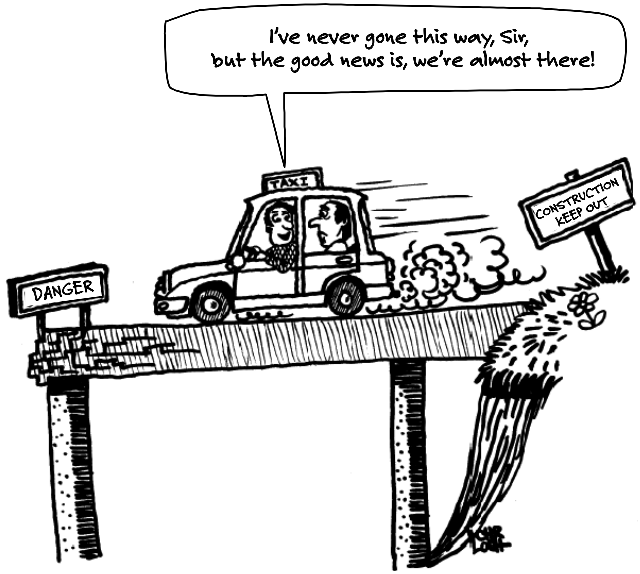


Figure 9.2 Monitoring system assuming foreseeable uncertainty

The cartoon on the right-hand side of Figure 9.2 captures the problems into which such monitoring systems run when unforeseeable uncertainty is present: Progress is often a mirage, and risks and deviations that were not included as "possible" in the risk plan accumulate unchecked (there is no tool to report them!), or are "swept under the rug." Not before the final phase of the project is reached is the team forced to face up to the accumulated problems, at which point often catastrophic deviations, and possibly a breakdown of the relationships within the team and/or with partners, occur. This danger is related to the "double-blindness of planning systems" that we discussed in Sections 3.2.3 and 5.2.2.

We are not saying that progress monitoring should be neglected and that the tools such as the one in Figure 9.2 should be thrown out of the window. These tools are, just like thorough planning and risk assessment, indispensable as the "basic homework." They are often also a good tool to communicate with stakeholders in the project, because they are easy to understand. However, they must be complemented by a monitoring of the knowledge gaps, the open questions, the current hypotheses, and the state of verification or falsification of those hypotheses. This type of knowledge monitoring does not have to be very formal: It may be as simple as the table in Table 9.2 (the table is based on the probing questions document used at Escend, shown in Table 5.2, and develops it one step further as a systematic monitoring tool). But it is essential that the knowledge and hypotheses be explicitly tracked. Being constantly reminded of open knowledge gaps helps the team to remain focused on the large open questions. This can prevent planning and monitoring from becoming an end in itself and double-blind.

It is important that the monitoring document contain causal explanations (the hypotheses as currently available) and reports on changed parameters, as shown in Table 9.2, and not only "output" reports (costs, completion percentage, schedule, functionality achieved, etc.). Studies in complex software development projects have shown that even professional workers are not able to keep track of the causal connections among variables when only outcome information is provided, and tend to make poor decisions and, moreover, oscillate in their decisions.[7]

9.2.3 Coordination and Information System

In a planned project with foreseeable uncertainty, coordination can, at least partially, be defined at the outset, for example, in the form of design rules, interfaces, and schedules that ensure that any party or subproject stays within limits that are compatible with the other subprojects. The information system can then steer interactions during the project by defining clear information deliverables among the parties (for example, completion of activities, shape of certain shared parameters and interfaces, and deviations, or risks occurred).

Table 9.2: Monitoring Table for Areas with Potential Unk Unks at Escend, Status October 2003

Area of Knowledge Gap	***Current Hypothesis***	***Evidence, State of Verification***	***Next Steps to Close Question***
1. Who is the paying customer, and what are the customer needs?	Component manufacturers need design-win tracking, but reps are the gate-keepers and are the major party to be convinced.	Interviews suggest shift of design-win activities back to manufacturers, hypothesis may; be wrong.	Verify information flow and activities across all steps of design supply chain from component manufacturer over rep, OEM, and contract manufacturer.
2. Is the industry ready to share data for the sake of better collaboration?	Industry needs collaboration; with clearly demonstrated benefits, companies will share sensitive information.	Still plausible, but design-win seems insufficient as a benefit to get everyone excited.	Verify sales side of relationships.
3. Can the product functionality gaps be closed?	Gaps can be fixed without redesign from scratch.	The technical idea of adding a shell around the kernel (keeping the kernel stable) plugs current gaps, but if sales side becomes critical (see area 2), a new major gap arises.	Test with customers, explore effort of adding sales tracking module to product.

In learning projects, coordination and communication become more blurred because unk unks make it impossible to define coordination at the outset; coordination has to be repeatedly redefined over the course of the project. Thus, the information and coordination systems become indistinguishable.

As for the planning and monitoring systems, it remains an indispensable "homework" exercise to understand interactions across the various parts of the project (for example, with a classic work breakdown structure, or the design structure matrix that we discussed in Section 4.2) and to establish interfaces. However, information exchanges have to become more frequent and broader, and less confined to initially established criteria.[8]

Project complexity makes it particularly pressing to coordinate and share information quickly. For example, complex engineering projects typically start by cutting the complex problem into pieces. A subteam solves the design of its respective piece, and then the subteams integrate learning about unforeseen system interactions as they emerge. Unfortunately, this type of learning leads to cascading changes and oscillations: If one partner in the project changes, she forces another to change, who, in turn, forces a third subteam to change, which in the end makes it necessary for the first partner to change again. Such oscillations have been widely reported in the engineering management literature.[9]

To avoid such oscillations of system changes, it is important to share information quickly, with as little delay as possible, even if this looks like information overload at first glance. Moreover, engineers should be willing to release preliminary information (even when they are not yet quite sure), which reduces the amount of work done with obsolete information, and to be willing to make small compromises at the component level to make the overall problem solving more stable.[10]

In particular, the information system should have the following characteristics:

- *Regular general project reviews* should provide all project team members with an understanding of the status of the overall project and of the subprojects, and of important mutual interdependencies (as they may significantly change when unk unks emerge). A project with unk unks cannot succeed if everyone knows only what is required for his or her job. This includes, in particular, a kickoff workshop (or multiple workshops when the project is large), in which all project team members can familiarize themselves with the project.
- Information exchanges must be *more frequent* than in planned projects, and *broader*, which means across multiple information channels. In other words, it is insufficient to exchange information through formal channels, such as change reports; rather, management must provide ample opportunity for members of various subprojects to exchange informal information about any aspect of their work. Only such frequent *and* unstructured information exchange provides safeguards against changes that unexpectedly "ricochet" from one subproject to another. Frequent, unstructured information exchange, although it is inefficient in the short run, helps the project team to be mindful of the unknown (see our discussion in Section 8.1.2).
- Information will change over a longer period in a project with unk unks; as unk unks emerge and require a response, a "design freeze" cannot be imposed. The information system must allow project parameters to remain in flux longer, until a judgment has been made that the uncertainty has shifted from unforeseeable to foreseeable. In parallel, the project team members must be kept informed about the uncertainty status (see Chapter 8).

Empirical studies have found evidence that the speed of turning around experiments increases the performance of product development in semiconductor and software companies. Thomke shows that effective experimentation cycles require not only explicit planning and information exchange, but sometimes also an adjustment of the organizational structure of the project.[11] Figure 9.3 shows the steps of the experimental cycle with an organizational boundary: The design department poses questions and develops the logic of the test. The testing department builds the detailed test and executes it; the results then go back to the design department for analysis and interpretation. Thus, the experimental cycle has two hand-offs, which is almost a guarantee for slow execution. If a project (or subproject) needs to experiment in order to learn about unk unks, the experiments, from posing questions to interpreting findings, should all be concentrated in one responsible hand. At Escend Technologies, this was accomplished by Elaine Bailey, the CEO herself, running the three critical areas and posing the questions. In large-scale projects, the organizational unity of experimental cycles does not happen naturally, so it must be explicitly safeguarded.

9.2.4 Evaluation and Incentive System

We use the term "incentives" here broadly; we do not mean only bonuses or variable pay. We also include more subtle factors that have incentive effects on workers, such as an evaluation that influences subsequent chances of a promotion or a transfer to a more attractive location, the probability of being fired during the next downsizing, or public recognition and peer pressure.

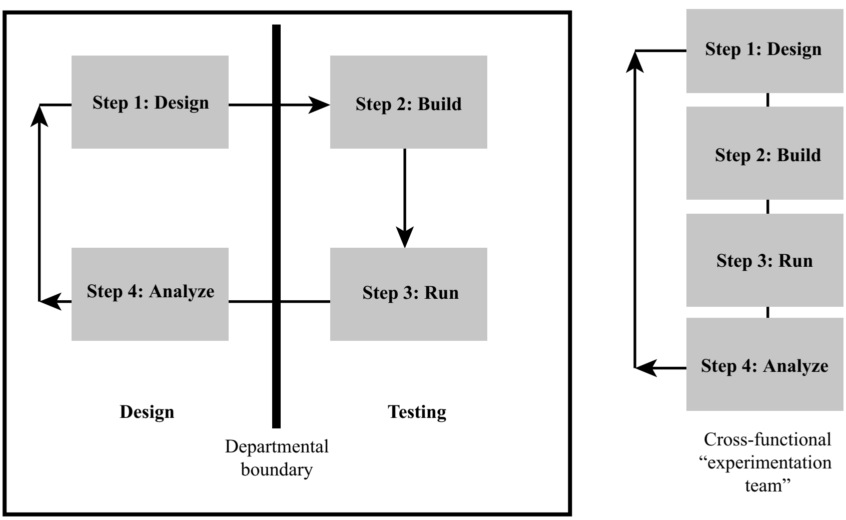


Figure 9.3 Organization and fast execution of the experimental cycle

(Source: Tomke 2003, p. 204)

It is well known from incentive research that productive evaluation needs congruence between responsibility and authority. For example, performance measurement should be supported by job design: If tasks can be separated, the responsibility for them should also be separated, and the domains of responsibility and authority should be congruent.[12] Furthermore, if a manager is responsible for multiple tasks, some of which are uncertain or difficult to measure, no "high-powered" incentives (that means no large bonuses or salary variations driven by the incentive measure) should be used. First, strong incentives would be unfair because they would place the burden of performance variation risk on the individual manager, who is typically risk-averse. Second, strong incentives on the easily measurable tasks would attract the employee's effort (toward the area where his or her effort can be predicted to have a "payoff") away from the difficult-to-measure tasks (which may be as important as the measurable tasks!).

Empirical studies have supported these principles: Performance measurement tends to vary with the character of the work. Thus, output-oriented measures tend to be used for applied projects, while *effort* is measured in risky, long-term technology development, and the issues shift to getting the right people and encouraging breadth of ideas (both from inside and outside) in basic research.[13]

In the context of novel projects, this discussion implies that project performance should be measured as *output* or *process* quality, depending on project uncertainty. For routine projects, the project team can influence outputs, deliverables, or milestones, and can thus be held responsible for them. In a project with unk unks, output measures are not under the control of the team and are thus not appropriate as evaluative measures, except if used as "upward" rewards in the case of success. An upward reward, essentially, amounts to telling a project worker, "This task is so fraught with unk unks that it will have no negative effect for you if it goes wrong; if you *do* manage to make this task a success, we will reward you" (with money or recognition).

However, judging the quality of the effort, process, or method used in a novel project by professional workers is often infeasible, and upward incentives are expensive and can become victims of "private information" and free-riding. Project workers know their tasks better than anyone else; they can hide mistakes and choose not to reveal additional opportunities that would benefit the project but would require more effort from them, or perhaps expose them to a higher risk (for example, of not being able to deliver). Moreover, unk unks are sometimes observable only (or at least long before everyone else) by the front-line project workers. The workers must *volunteer* the reporting of the unk unks (they can always claim later that they did not see it, or saw it later than they really did).

Table 9.3: Process Versus Upward Incentives in Projects with Unk Unks

	***Management* Can Monitor Process**	***Management* Cannot Monitor Process**
Management *can* observe unk unks as they emerge	Process monitoring, with incentives based on quality of action or method	Adjustment of bonus to problem difficulty & downside protection
Management *cannot* observe unk unks as they emerge		Upward incentives & downside protection

Therefore, the configuration of the reward system must be carefully chosen. Table 9.3 gives a rough guideline of when what type of reward tends to be reasonable.[14] The figure starts with the premise that the manager or team in question faces high novelty and unforeseeable uncertainty (if this is not the case, output-based rewards can be used). The choice addressed by the figure is between process-based evaluation ("has the team diligently and competently used the appropriate methods and processes?") and upward rewards (if the team succeeds against all odds, it is rewarded, while it is protected from negative effects of failure).

If management can indeed effectively monitor the process, the methods used, the diligence of execution, and the level of personal effort of the team members, process incentives should be used. In other words, team members should be evaluated on the quality of the methods, processes, and efforts (left-hand column of Table 9.3), not on the outcome. The outcome is not fully under their control in a subproject with unk unks. Holding them responsible for competent work is the best predictor of success.

Often, of course, the process cannot be effectively monitored, at least not at reasonable cost, because the work is highly professional. Team members have autonomy, they apply tacit knowledge that is difficult to articulate, and it is difficult to tell how hard they work. Sometimes the engineer sitting in his chair with his feet on the table is working the hardest. In this case, it is unavoidable for rewards to be linked to output in some way. However, the presence of unk unks requires a certain downside protection—workers must not be held responsible for unk unks preventing success. People are strongly demotivated by the threat of negative consequences that stem from causes that they cannot control.

The downside protection depends on whether management can observe the emergence of unk unks. If supervising management is so close to the project that unk unks are visible to it as they arise, the performance hurdle can actually be adjusted to the emerging situation: If unk unks made the project harder, workers are given an extra cushion, but if unk unks presented new opportunities that made life easier, or allowed higher performance, the hurdle can be raised.[15] As an example of such hurdle adjustment, consider the following interview given by GE's CEO, Jack Welch, in 1995.[16]

> If I worked for you, you'd say, "I need four!" We'd haggle all day, me making presentations, with 50 charts, saying the right number is two. In the end, we'd settle on three. We'd go home and tell our families that we had a helluva day at the office. And what did we do? We ended up minimizing our activity. We weren't dreaming, reaching. I was trying to get the lowest budget number I could sell you. It's all backward. But if instead you ask people, "give us all you can, give us the best shot at what you can do," then you can't believe the numbers you'll get. You'll get more than you need. There's a trust built that people are going to give their best.
>
> Our plastics business last year had an up year, something like 10% or 11%. But in my view, they had a relatively poor year. They should have been up 30% to 40%. They got caught in a squeeze with prices, and they didn't act fast enough. So, their bonuses were affected. Our aircraft engine business went down $50 million in earnings to $500 million. But we increased their bonus pool 17%. They had a drop, but they knocked the hell out of the competition around the world. They lost $2 billion in sales as the military market and the airline business came down. But they responded to their environment better and faster than their competitors did.
>
> Now, if we were operating under a budget system, plastics would be seen as having a nice year, and aircraft engines would have got a slap in the eye.

This interview is clearly about business units, not projects. But the spirit of Jack Welch's statement is very relevant: Do not evaluate the team on a fixed target. If the environment gets tougher, and one can observe that change, one will adjust the reward hurdle according to the circumstances under which the team was forced to operate. In the terminology of this chapter, we adjust the hurdle to the emerging unk unks.

Finally, true upward incentives should be used only when unavoidable—that is, when management cannot monitor either the process or the emergence of unk unks. In this case, upward incentives provide a motivation to strive for an upside, even if it requires effort and cannot be enforced by management, while the risk of failure is eliminated, which may discourage workers from trying. Again, we include not only monetary incentives but also intangible rewards, such as recognition for achieving something unlikely.

9.3 The Management Systems of Selectionist (Sub) Projects

For the discussion of the management systems of a selectionist approach, we will return to the example of Option International, used in Chapter 6. Recall that the Option International's CEO launched a series of parallel trials over a period of two to three years, in order to redefine the business model of the small entrepreneurial company. In implementing the parallel projects, the organization deployed a lot of good, traditional project management that we would identify with planned projects. For the individual trials, the management systems described in the first column of Table 9.1 applied. But to manage the portfolio of projects, Jan Callewaert had to apply a different management style and implement systems that correspond to what is shown in the right-hand column of Table 9.1.

9.3.1 Planning Systems

One of the main challenges in managing parallel selectionist projects is to keep a global view of all the projects and to prevent them from getting into conflict with one another. Indeed, the overall goal of the parallel subprojects must be a common one. The purpose of a selectionist trial is not to hinder or kill the other trials, but to explore which avenue is the best. If a trial is stopped, the learning that has accumulated in executing it needs to be shared with the other trials.

The essence of the planning exercise is determining the ultimate goal and the collective vision for the portfolio of subprojects (selectionist trials), determining the number of projects one needs to carry out in parallel in order to have sufficient variety, and determining the criteria one will use to diagnose their progress and state.

First, determining and enforcing the ultimate vision was probably the single most important contribution of Jan Callewaert in guiding Option International through its phase of selectionism. He took it on his shoulders to carry through the project of redefining the business model and relentlessly focused the organization on its ultimate goal. While individual teams focused on delivering the best snap-on or mobile phone, he kept in mind, and under control, the overall architecture of what needed to be achieved in the transformation of the business model. He ensured that everybody understood this overall goal, because the survival of the company depended on it. It takes real leadership skills and charisma to render this ultimate goal credible.

The second important task in planning the selectionist approach is to determine the number of parallel trials or projects needed. There is no hard or fast way to determine this, but it has to balance the required variety that needs to be created by the subprojects with the cost of launching additional subprojects. The costs of these subprojects can obviously be reduced, the earlier one can stop trials. The ability to stop less interesting trials early on is thus a very important part of the management systems for selectionist projects.

Third, we already pointed out in Chapter 6 that the ability to stop trials is very important for the success of a selectionist approach. Determining up front sufficiently broad criteria of when to stop projects is an essential part of the planning process for parallel subprojects. In the case of Option International, these criteria were clearly set and evaluated by the CEO, often in collaboration with the board of directors. The evaluation process that led to the end of some subprojects was also enabled by excellent communication between the various teams: The CEO and the project managers were constantly communicating, and they could use the information collected in one subproject to improve the actions taken for another one.

It was also the CEO's strong influence that kept the various subprojects from competing with one another. A selectionist approach is not a war among projects. Success for Option International came through the organization of multiple approaches that communicated with one another, shared learnings, and brought mutual enrichment.

In larger organizations, the important role played by the CEO of Option International needs to be fulfilled by a senior manager, often called program manager, chief project manager, or chief engineer. In the case of Toyota,[17] this person was called the chief engineer and was described as the system architect or the lead designer for the vehicle. He was seen to be the most important technical decision maker on the team. It is noteworthy that this person, in the case of Toyota, had no functional authority over the engineers. The engineers continued reporting to the functional managers. However, the chief engineer was responsible for the vehicle project as a whole—that is, from the early concept stage to the launch of the car and the initial marketing campaign. Most importantly, the chief engineer performed the vital systems integration. While each function was responsible for its subsystem, the chief engineer was responsible for the total vehicle. The chief engineer implements the set-based process, described in Chapter 6, by controlling the process of narrowing the choices and number of sets, by insisting on broad exploration, and by making the decisions on competing alternatives based on the analysis of the trade-offs.

The chief engineer ensured that the overall view for the project was kept, and played a decisive role in closing down the options. This role can be performed in different organizational structures, in functional structures with project managers who have "informal" or "cultural" influence, in matrix structures with a heavyweight project manager, or in dedicated teams. What matters is that someone assumes the responsibility of planning for the selectionist trials, their sharing, and their termination.

9.3.2 Monitoring Systems

As in the learning case and the example of Escend, a Gantt chart is not sufficient to help us monitor parallel projects. Monitoring in this case will require the constant evaluation of the relative contribution of each subproject to the ultimate goal of the project. How much closer does an individual project bring us to the end goal? In the case of Option International, the overarching question was not whether the subprojects were on time (though this was an important challenge at the level of the individual trials). The key question was to what extent the individual projects had the potential to bring the organization closer to a viable and sustainable business model.

Comparing the relative progress of selectionist trials is not easy, and can even be dangerous to the overall project objective. In particular, subprojects do not move linearly at a similar pace. As we discussed previously, selectionist trials are often used for complex (sub) projects. It is difficult to track linear progress in complex projects, because just when you think one team may be about to discover the solution, an unk unk hits and changes one aspect of the project, which then changes others, and so on. Complex projects are highly nonlinear systems, and as such, they are notoriously difficult to track.

In addition, if subproject teams know that their progress is being compared directly to that of other teams, the selectionist approach will become like an America's Cup race: The objective will become, either consciously or subconsciously, to beat the other team. In America's Cup racing, the objective is not to achieve a fast time, but to block the other team, and great effort, and time, can be put into this objective.

Thus, selectionist trials should only be stopped when it becomes clear that they will not find an acceptable solution. This is not easy for organizations, as these trials can be relatively expensive, and it is tempting to shut a subproject down when it appears to have fallen behind. But in complex projects, impressions can be misleading. Monitoring of the selectionist trials should thus be based on the criteria of the project objective, not criteria relative to other selectionist trials. This is difficult for organizations to implement, but it is at the heart of selectionism.

While projects should not be "compared" to one another, they should learn from one another. Here, monitoring can help a great deal. The global project manager needs to monitor how knowledge (both tacit and codified), developed throughout the collection of subprojects, is deployed rapidly and effectively. However, caution is needed in transferring learnings from one project to another. In complex projects with unk unks, one should avoid taking the best aspects of one project and incorporating them into another. In complex landscapes, subtle differences between two projects can render these best aspects useless, or worse (see the discussion in Chapter 7). What is important is sharing the overall learnings across projects. What have we learned in one project that helps us to understand the landscape of another project?

9.3.3 Coordination and Information Systems

The fundamental difference between selectionist trials and experimental learning is the relative autonomy and independence of the selectionist trials. Selectionism is about testing alternative approaches to the (sub) project objective. Thus, selectionist trials should be given the time and freedom to demonstrate whether they will be effective in meeting these objectives or not. Thus, coordination in selectionism takes a very particular form: We need to keep the selectionist trials relatively autonomous while allowing for communication across projects to leverage key learnings.

In addition, we may need to stop some trials when it becomes clear that they will not yield satisfactory solutions. The selectionist trials, then, should not be so autonomous as to create "winners" and "losers" of the various project teams. Coordination in selectionist trials requires clearly specified "fair process" from the beginning.[18] All players must fully understand the game that is to be played. When venture capitalists fund multiple startup ventures in the same "space," the startups are fully aware that if the market place rejects their particular approach, the VC is free to walk away without offering any solace or safety net to the founders. The VC is the hub and the various selectionist trials (startups) are treated fairly autonomously.

In project organizations, the rules of the game are likely to be different, and thus, coordination systems are likely to differ as well. The "winners" and "losers" are likely to be from the same organization and are likely to continue their working relationship regardless of the outcome of the selectionist trials. It is perhaps quite easy to determine up front the criteria for ending subprojects. However, experienced project managers know that the reality of stopping subprojects is not so simple. From the general experience of how to stop failing projects, we know that this is a difficult task for many reasons.[19]

- Often, the teams and the organization do not want to accept that the subproject is not leading anywhere; teams suffer from the collective belief that the project is going in the right direction and turn a blind eye to negative information.
- Individual managers do not want to admit that their project is not delivering the optimal results, because it does not fit the image they have of themselves as good leaders.
- Even if subprojects encounter problems, the organization rarely wants to view this as a negative warning signal because novel projects are supposed to run into risks, and it is the role of a good project manager to overcome these risks. Finally, organizational bureaucracy often makes it difficult to shut projects down.

The same mechanisms can apply with subprojects in the selectionist approach if the subprojects are given too much autonomy. Thus, there needs to be sufficient coordination among the subprojects so that those subprojects that are halted should not be considered as failures, but rather as elements of the final solution. Specifically, one should

- Establish at the outset of the project clear "rules of the game," noting that many of the subprojects will have to be halted.
- Ensure that stopping a project has no negative effects on the teams and the individuals leading the subprojects.
- Use outside resources to help evaluate the progress made by the subprojects.
- Avoid all the team members of the subprojects being cheerleaders, but by all means, have a few critical members on each team.

- Have a formal review process with adequate time set aside.
- Keep a high level of common identity among the teams.
- Ensure that the whole project reminds itself constantly of the ultimate goal to be achieved by the selectionist approach.

Option International guaranteed that failing projects were stopped appropriately by constantly evaluating the contribution the projects made to the overall goal of the redefinition of the business model. The CEO used the board of directors as a sparring partner in the evaluation of the progress of the subprojects. In addition, the need to report to financial analysts on a quarterly basis gave an almost natural rhythm to the process of evaluation.

Thus, organizations are faced with an important challenge in the coordination of subprojects in the selectionist approach: How much autonomy do you give the managers of the subprojects, both in terms of the goals to be achieved by the subproject (goal autonomy) and how to go about achieving these goals (supervision autonomy)?

The degree of supervision autonomy has to do with the way in which management exercises oversight through the specification and supervision of operational activities. A project group with greater supervision autonomy has greater local discretion, permitting greater heterogeneity in day-to-day activities. Greater supervision autonomy allows for innovation in problem solving and provides an inducement for the team members to exercise greater individual discretion, leading to greater motivation and commitment. Supervision autonomy also helps to minimize the strain on the information-processing capacity of the organization.

The whole point of selectionist trials is to try different approaches, so project teams will need a sufficient level of supervision autonomy to remain independent from other selectionist trials. However, the autonomy of the trial teams must be constrained in that each team while remaining independent, must not have the freedom to mimic what might be perceived as the current "best practice" among the many selectionist trials. The America's Cup metaphor was mentioned in the previous section: One cannot allow selectionist trials to "react" to the actions of other selectionist trials in an attempt to stay ahead of the pack. Each team must fully explore the path that it has been dealt, and not hop on to another's path that the team might see as more promising at the moment. Only then can the complex solution space be properly explored.

The degree of goal autonomy has to do with the way performance goals are set. At one extreme, managers may allow a team complete latitude in terms of goals, focusing on possibilities and opportunities. At the other extreme, managers may be directive, defining very specific goals and outcome criteria. Traditional project management will argue that clarity, measurability, and unambiguity in the goals of the project are key factors for success. The value of clear authority structures and working relationships for a project is seldom questioned. Project management preaches that goal autonomy should be relatively low in order to perform well.

Selectionist trials also need clear goals, as these goals will be used to "select" the trial, or trials, that will continue. Without clearly stated goals, one cannot have a fair process by which some projects are stopped and others continued. These goals must be specific enough for all to agree as to whether one project is meeting these goals better than another. However, one must take great care not to confuse the goal of the project with the "how" of the project. Too often in project management, the project plan comes to be seen as the goal in itself, rather than a means to achieve a goal. Goals for selectionist trials must remain high-level enough to grant each trial the supervision authority it needs to explore the solution space.

In the case of Option International, the combination of both goal and supervision autonomy was realized by stimulating the entrepreneurial behavior of the subteams. The CEO, being the prototypical example of a dynamic entrepreneur in the high-tech business, set the standard for the rest of the organization: He demanded a high level of entrepreneurial spirit from his collaborators, and those who could not cope with it gradually left the organization.

9.3.4 Evaluation and Incentive Systems

The argument we developed about incentives for learning projects also applies to selectionist projects: In the context of high uncertainty, it is difficult to link incentives to output alone. The incentives need to be linked to the process quality and, as was illustrated with the quote from Jack Welch, must be stimulating to get the best out of the people, after adjustment for the level of the emerging unk unks.

In parallel selectionist trial projects, the incentives must additionally reflect a clear message that the "failure" (that is, the stopping) of one subproject cannot be the yardstick for evaluation. Incentives have to create a common commitment to the ultimate goal. But they also need to ensure that the quality with which the individual subprojects are implemented is of high standard, and that information is readily shared across the project.

One way to support this somewhat contradictory set of incentives is to create an "expedition effect." Imagine an expedition with several ships or teams. They know that their best chance of succeeding is to cooperate. Yet each of the ships or teams needs to perform at its best in order to make a real contribution to the expedition and not slow down. And they also know that their best defense against unknown dangers is to constantly communicate with one another. Expeditions succeed when the members of the team know they will all share equitably in the end result. They also cooperate because of peer pressure and because it is in their best interest to survive during the expedition. It is this feeling of being part of an expedition that one has to stimulate through the evaluation and incentive system.

A "financial incentives" approach to creating an expedition effect would contain incentives based first on the end result, such as stock options or a significant bonus that is determined primarily on the overall performance—that is, the result achieved by all parallel projects collectively—and second on the process quality of the individual trial projects.

A concern often expressed about purely group-oriented incentives is that they dull individual effort and individual stretch for creativity.[20] Incentives experts have, therefore, proposed a "win and audit" approach: Incentives *do* contain a bonus for "winning," for producing the trial project on which the final solution is based. However, if one team's trial is chosen, the bonus is not yet earned; rather, winning triggers an "audit" that examines whether the team has shared information and collaborated with the other teams (this can be done, for example, by a peer review). Only if the team has behaved collegially does it get the extra bonus.[21] However, incentives that reward individual teams, and thus cause the teams to compete, have to be viewed with caution. Much evidence shows that competition suppresses collaboration and may even push some employees to the extreme of changing their work, not to improve results but to prevent others from winning.[22]

Finally, the above outlined financial incentives approach is insufficient and may even be counterproductive. Financial rewards can bring about certain specific actions, but they are weak in producing consistently collaborative and constructive behavior whenever the employees have discretion and autonomy in their work and cannot be fully monitored.[23] Needless to say, this is the situation novel projects face.

Actual day-to-day behavior in parallel teams is driven by social interactions: status, relationships and group identity, and role models. The first concerns status and recognition, both by peers and by management. Recognition is an intrinsic need that people have everywhere (although its expression is culturally specific to countries and organizations).[24] Winning itself carries status, even without any emphasis placed on it by management. This pushes teams into competition. If, however, management consistently recognizes and acknowledges sharing and collaboration efforts, this will, over time, also carry status and counteract competition.

Second, personal relationships across teams, encouraged by shared events (especially important at the outset), and repeated emphasis on the shared endeavor and common goal of providing the best outcome for the project overall also serve to emphasize a common group identity.

Third, the behavior of the team leaders sets the tone. If the leaders are competing and winning types, team members will take the clues and emphasize competition rather than sharing and commonly supported trial selection. If the team leaders get to that position by collaboration, the tone of the teams will be more collaborative. Personnel selection (see Chapter 8) is critical in determining the team's character and working mode.

9.4 Integrating Learning and Selectionist Pieces into the Overall Project

In the first three sections of this chapter, we discussed the management systems that support learning and selectionism in one subproject. However, we saw in Chapter 4 that a large project is not usually afflicted by unk unks in all its pieces. Every project has pieces that are well understood and can be managed without selectionism and learning. For example, the Escend Technologies project from Chapter 4 had three areas with potential unk unks and seven areas where it was clear what had to be done. And the case of Option International is a combination of a selectionist approach for the development of the business model and a learning approach for the technology development. How can these different types of subprojects be integrated at the level of the overall project?

In this section, we offer three principles: In the overall project plan, the areas (subprojects) threatened by unk unks need large buffers; in the spirit of coordination in concurrent engineering, the other subprojects should understand what deliverables and information they need from these subprojects in order to start their own work; and finally, the subprojects with unk unks need to provide uncertainty status updates and "go" signals to the other subprojects that depend on them. We explain each in turn.

9.4.1 Buffers for the Subprojects Threatened by Unk Unks

The activity areas (subprojects) that are subject to unk unks must be managed with experimental iterations. This implies directly that the duration and budgets of these areas cannot be precisely predicted or planned. Therefore, they must be given a *large buffer* that explicitly incorporates this lack of knowledge into the project plan. Figure 9.4 shows an illustrative high-level turnaround project "plan" at Escend, at the level of the subprojects from Table 4.1. From the diagnosis, we know that the first three areas were judged vulnerable to unk unks.

Areas 4 through 9 were straightforward; Elaine Bailey knew what had to be done, and it was possible to swiftly execute these areas. For the first three areas, however, no one knew how long it might take in order to get a clear picture and to understand what needed to be done. Elaine had to go on a learning mission. Large buffers (in white) in the Gantt chart graphically express this knowledge lacuna. In addition, the plan is incomplete; that is why we set the word in quotation marks. While the immediate actions to "stop the bleeding" are known, the really important actions of tapping the potential market (if it exists) cannot yet even be written down in a plan. Figure 9.4 contains a "ghost" activity whose content is concealed. In some sense, the Gantt chart in Figure 9.4 does not contain a lot of information, but sometimes it is important to clearly illustrate how *little* one knows.

9.4.2 Clarify Dependence of Other Subprojects

Given that the subprojects with unk unks have unpredictable completion times and will produce information that we cannot yet describe, the other subprojects will be affected and cannot be planned either, neither in their timing nor in their content. It is therefore important for the project team to understand which subprojects are susceptible to unk unks. In a way, a high-level design structure matrix (DSM, see Chapter 4) should be drawn up just to see which subprojects are "immune" to effects emanating from unk unks.

For example, at Escend (Figure 9.4), subprojects 4 through 9 are pretty much independent of the unk unks arising from market status and product functionality. These subprojects are concerned with stopping the bleeding from the current company situation. They represent a defensive move that must be completed anyway, independent of the findings in the first three subprojects. Therefore, areas 4 through 9 should proceed as quickly as possible. The real product development and market approach, however (the "ghost" activity in Figure 9.4), cannot even be defined (not to mention started) before the unk unks have been resolved and substantial information is available about the shape that the market is taking.

9.4.3 Transfer Preliminary Information

In order for other subprojects to start and to progress, the status of the unk unks, and the subprojects affected by them, must be communicated. The uncertainty status can be illustrated by *preliminary information.*[25] Imagine building a house: You cannot afford to delay the kitchen planning until you have put up the walls. But if you start the kitchen planning too early, using preliminary floor plans from the architect, you are likely to do it twice. You need a new way of exchanging information between the architect and the kitchen planner. Currently, your kitchen planner's idea of concurrent engineering is that he should receive the floor plans, as he did in the past, just six months earlier. He doesn't understand that the nature of the information has changed.[26]

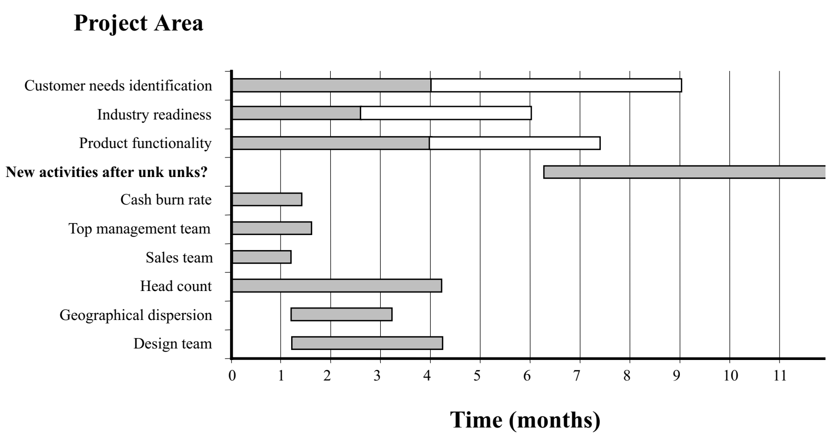


Figure 9.4 High-level "plan" for the turnaround project at Escend Technologies

The uncertain status of information that is exchanged across subprojects can be made explicit by labeling the information in terms of *stability* and *precision*. For a given amount of knowledge, information precision and information stability may be in conflict with each other, as the following everyday scenario illustrates.[27]

A traveler flying from Philadelphia to Paris wants a friend pick her up at the Paris airport. The day before the trip, the arrival time in Paris is uncertain. Thus, any information forwarded to the friend will be preliminary in nature. The traveler can ignore this uncertainty and communicate an arrival time of 14:34, which is precise information but unlikely to be stable. Alternatively, she can focus on information stability and say that she will arrive between 12:00 and 18:00. As the journey unfolds (e.g., before boarding, after take-off, at the baggage claim), the uncertainty of the arrival time is reduced and the preliminary information is revised repeatedly, until it is fully stable and precise (as she leaves the airport).

Note that the two strategies each impose different costs on the friend: The precise information may force the friend to change her own plans at short notice if the plane arrives later or earlier than planned. The stable strategy, on the other hand, forces the friend to keep her diary open for the entire afternoon, blocking any other appointments.

Initially, little information on the resolution of the design decision is available (low level of knowledge), and information is neither stable nor precise. As problem solving progresses and the level of knowledge increases, information is repeatedly communicated with changing levels of precision and stability. At the end of the problem-solving process (high knowledge level), the design solution is in place. Now, information is both stable and precise.

Figure 9.5 shows two strategies of dealing with preliminary information.[28] An iterative strategy emphasizes precision: It utilizes the information in every detail, but as the information is not stable, the response may have to be changed (which can be expensive if rework results—that is, if the response has first to be undone and then reperformed). A set-based strategy emphasizes stability over precision: It uses only "ranges" of possible outcomes, which avoids rework, but the other subproject may not be able to do useful work based on the imprecise information. When there is no unforeseen uncertainty and ranges of outcomes and their probabilities are known, a "best" combination of precision and stability can be chosen, considering probabilities and relative costs of rework versus having to wait.[29]

This definition of preliminary information, stability, and precision is, of course, based on foreseeable uncertainty, or in other words, on a situation where all possible outcomes can be described beforehand. This is not possible when we have unk unks. But the same spirit of the use of information applies to an information exchange subject to unk unks, although the form of the preliminary information cannot be "optimized."

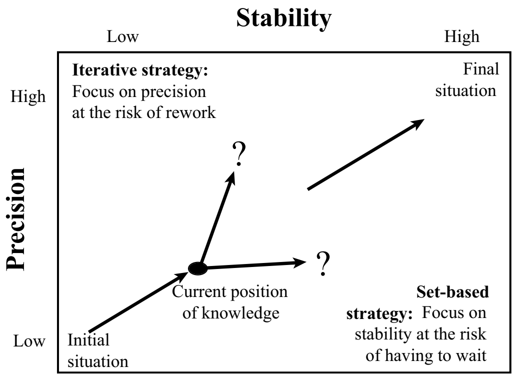


Figure 9.5 Formats of preliminary information

For example, let's again consider Elaine Bailey's situation at Escend Technologies. When it became clear in late 2003 that design-win tracking was not the only important customer benefit and manufacturer reps were not the most important target group, this had implications for the product redesign. Hypotheses arose about the "ghost activity" in Figure 9.4, preliminary information concerning the dependency between the market emergence and the product design: Escend may need to add input and reporting capabilities for manufacturers and OEMs, or ordering capabilities for the contract manufacturers, or distribution capability to enter sales data of the final product. An iterative strategy would imply that the design team work on (or at least prepare) each of these possibilities, discarding the work for the option that was finally chosen. This had the advantage of maximum responsiveness, but it was very expensive, beyond Escend's resources. A set-based strategy implied doing only preparatory work allowing *any* design change to be made, and otherwise waiting. Escend waited until Elaine felt confident, and then took a bet on the distribution capability.

The key lesson is that the overall project needs an information exchange strategy across subprojects that explicitly acknowledges the uncertainty and vulnerability to unforeseen changes. A simple tool that embodies Figure 9.5 can serve as a management system, sufficient to capture the key uncertainties. Overformalization should be avoided because formalized tools easily deteriorate into "double-blindness," claiming precision and knowledge that are not really available. The key questions that should be answered are as follows:

- What information from the unk-unk-fraught subproject does the other subproject need? (This should come out of the dependency DSM; see Section 9.4.2.)
- What does the information-delivering subproject team know? Where are the limits? Can they deliver partial information, or a range of possibilities, that are stable? Or is the status still in flux in unforeseeable ways?

- Does the dependent subproject have enough information to commit to a certain course of action? Is it feasible and affordable to start a course of action and later change it (iterate)? Or should the team keep multiple courses of action open, or simply wait until more information is available? The answer depends on relative costs and urgency.
- Update the status of information at learning points and communicate the updates to the dependent subprojects. If both sides understand the dependency, information can be exchanged in a targeted way, which helps to reduce information overload. The two subproject managers can decide together when enough information is available for the dependent subproject to start at full speed.

The benefit of this management system is not in planning, or optimizing, information exchange. The benefit lies in *making explicit* to the subprojects how they are dependent on one another, and where the dependencies are affected by emerging unk unks. This makes the teams aware that they have to adapt to one another in unforeseeable ways that emerge during the project, and helps them understand where changes come from. The key benefit, in other words, is not in planning but in sensemaking and mindfulness of mutual adaptation (see the project mind-set in Chapter 8).

The precise nature of the vulnerabilities to unk unks and of the dependencies across the subprojects is so specific to different projects that no general statements can be made. However, the need for the teams to understand the unpredictable nature of the subproject interactions, and to deal with them explicitly, is universal in novel projects. The *simplest* tools possible should be used to capture this insight.

Endnotes

1. The term "bounded rationality" was first coined by Simon (1955).
2. See Nelson and Winter 1982, p. 73; see also the discussion of intuition in Chapter 8.
3. Nelson and Winter 1982, p. 97 and pp. 99–100.
4. Teece, Pisano, and Shuen 1997, p. 518. Leonard-Barton (1992) referred to the "position" as the "technical systems."
5. See Leonard-Barton 1992, pp. 113–114.
6. The Gantt chart in Figure 9-2 is based on the unmanned aerial vehicle (UAV) project from Chapter 3.2.1; it corresponds to the network flow diagram in Figure 3.2. For an example of a more complex monitoring system, see Pillai and Rao 1996. The tool presented there is more complex than what we show in Figure 9-2 in the sense that schedule, budget, and "progress" are combined, with the possibility of cross-comparisons and, therefore, causal risk monitoring and analysis. However, the basic philosophy of comparing actual with planned progress is the same.
7. See, for example, Sengupta and Abdel-Hamid 1993.
8. This is already discussed, for example, by Shenhar and Dvir 1996, p. 618.
9. See, for example, Allen 1966, Mihm and Loch 2005.
10. See Cusumano and Selby 1995, Chapter 5, and Mihm and Loch 2005.
11. See Iansiti and MacCormack 1996, MacCormack et al. 2001, West 2000.
12. An influential article, Holmström and Milgrom 1991, established this principle in a formal model that has become widely accepted.
13. See, for example, an overview in Hauser 1998; see also Loch and Tapper 2002.
14. Table 9-3 is based on Sommer and Loch 2005.
15. Subject to respect for transparency and fair process. Fair process is discussed in Chapter 10.
16. Cited from Loeb 1995.
17. See Sobek et al. 1999.
18. We will discuss fair process in detail in Chapter 10, when we discuss collaboration with external partners.
19. See Ward et al. 1995.
20. Studies of innovativeness have repeatedly shown that group rewards emphasize execution, while individual rewards emphasize idea generation and radically new ideas. See, for example, Angle 1989, p. 142. In addition, incentive experts in economics have shown that when employees are confronted with multiple, partially conflicting tasks, strong incentives push them toward the less uncertain and more predictable tasks, because in this way, they can better guarantee some output and a positive evaluation for themselves. In economics, this is referred to as the impossibility of strong incentives for multiple tasks with uncertainty (Holmström and Milgrom 1991). In projects with unk unks, it is, of course, exactly the uncertain tasks that must be tackled.

21. This was first proposed by Sinclair-Desgagné 1999. See also Sommer and Loch 2005b for a discussion in the context of unk unks.

22. See, for example, Pfeffer and Sutton 2000.

23. We know this from empirical studies in innovation research; for example, see Angle 1989, p. 142. In addition, there is much research on incentive systems that shows this see, for example, Kohn 1993, Kunkel 1997, Pfeffer 1998.

24. See an overview of the reasons and the implications in Loch, Yaziji, and Langen 2000.

25. This discussion is based on Terwiesch, Loch, and De Meyer 2002; the rest of this paragraph is quoted from this article, p. 402.

26. Kitchen building is, by the way, a good example of a project in which unk unks can easily occur. As the aesthetics of the finished kitchen depend on subtle and complex interactions of colors and other elements, the final look is very difficult to predict and may easily come out differently than was intended. The design space is “unstructured” and must be searched (see Terwiesch and Loch 2004).

27. See Terwiesch, Loch, and De Meyer 2002, p. 409.

28. Adapted from Terwiesch, Loch, and De Meyer 2002, p. 412.

29. For a discussion of principles according to which a course of action can be chosen in dealing with the preliminary information, see Loch and Terwiesch 2005.