4

Diagnosing Complexity and Uncertainty

We saw in Chapter 2 how the presence of unforeseeable factors made traditional PRM techniques insufficient for the Circored project. We then argued in Chapter 3 that a broader view of PRM is necessary, one that allows for unk unks combined with project complexity and sets up a different project management approach, selectionism or learning, at the outset.

But the million-dollar question is this: How can unk unks be diagnosed at the outset if they are, by definition, "unforeseeable"? Our answer is that, in essence, managers must ask themselves, in all honesty: "What do I know, and where do I have fundamental knowledge gaps?" In every knowledge gap lurk potential unk unks.

In this chapter, we show how to diagnose unk unks and complexity at the beginning of the project. First, we illustrate how diligent and frank self-examination can enable project management to recognize the areas of knowledge gaps and focus their attention on them. We illustrate the diagnosis of unk unks and their subsequent flexible management with the crisis situation faced by one Silicon Valley startup company, Escend Technologies. A startup venture is a special type of project that starts with the first funding decision, follows a sequence of delivery and funding milestones (see Table 5.1 in Chapter 5), and ends when a merger, acquisition or IPO, or a transition to a self-financing and profitable ongoing concern has been achieved—or when the business closes. In Section 4.5, we turn to the diagnosis of complexity, using a systematic tool, the design structure matrix. At the end of the chapter, we link complexity back to the Escend example.

4.1 Diagnosing the Unforeseeable at Escend Technologies[1]

4.1.1 Background

Escend Technologies was founded in 1999 to enable electronic component manufacturers to connect and collaborate with manufacturers' representatives (reps), intermediaries that sold the components to electronics OEMs.[2] During the 1990s, component manufacturers, OEMs, distributors, and contract manufacturers in consumer electronics were becoming increasingly fragmented and disconnected as a result of outsourcing and globalization. There was a need to create a common customer record that tracked new products through the design, prototype, and manufacturing phases across the multiple companies involved.

By mid-2003, Escend was floundering, having burned through $16 million in venture funding, and requesting $6 million more to continue operations. Escend had won 20 rep firms as customers, who were supposed to pay—but did not—ongoing usage fees of about $800,000 per month. Sales had stagnated a year earlier, and the cash burn rate had grown to $650,000 per month. The CEO had no concrete plans, only vague promises, for any improvement. However, he reported progress on the product and the sales pipeline and maintained that it was only a matter of time before the company would be on its feet. When the initial response for the $6 million request was a resounding no, the board thought that the message, not the business or the management team, was the reason and behaved as usual—packaging the company for another financing round (i.e., improving the message by rewriting the presentation) and making introductions to their networks of contacts.[3]

Before completing the next round, the board (comprising the key investors) had to critically assess management's capability and the cash burn rate. It quickly became clear that the leadership had to change, as there was no convincing planned way out of the crisis, and business irregularities were emerging. Elaine Bailey temporarily stepped in as the CEO in mid-July 2003 to turn around Escend.[4] She was a general partner at Novus Ventures, a small business investment corporation, and one of the key investors. Within three weeks, she had to decide whether or not to recommend that Novus participate in another round of financing for Escend. A "yes" decision would give Escend a chance to fix its problems, become successful, and provide Novus with a positive return on investment. A "no" decision would end Escend's existence when the remaining cash was spent, and Novus would lose all it had invested in Escend. Writing off Escend might reduce Novus' chances of drawing on additional SBA funds for future investments. But investing in the next round and then losing everything (including other Escend investors' funds and Elaine's credibility in the VC community[5]) would be much worse.

4.1.2 Diagnosing the Unforeseeable

By examining the companies' expense reports and telephone records and by interviewing the entire team (35 employees) and four key manufacturers, Elaine quickly acquired a feel for the situation. She realized what a mess she had inherited. The CEO had invited buddies into the management team who seemed more interested in paying themselves perks than in pushing the business. The product functionality had major flaws (for example, customers could not specify custom reports), the architecture was inflexible (running in batch mode, not real time) and difficult to scale, and the Oregon-based design team was uncooperative. The phone logs revealed that the salespeople spent little time talking to customers and prospects. Elaine discovered that the $15 million pipeline was grossly inflated (the actual figure was $256,000) and that no potential customers existed. Additionally, morale had hit rock bottom.

Elaine sought the advice of her coauthors, two academics with whom she regularly exchanged ideas. "What should I do? Some of the problems and urgent moves are crystal clear. But I have no idea where the real fundamental problem lies. Perhaps it's just execution, but perhaps it's something to do with the industry. It's all opaque, like trying to see through a rock."

Her coauthors advised her against asking the question of what to do at this stage, as the situation was too uncertain and unstructured. They recommended first that she ask herself a fundamental question: "What do I know, what do I think I know, and where am I completely ignorant and unsure?" In other words, they recommended questioning all assumptions and identifying areas of the mess where she was on firm ground, as opposed to areas where she had major knowledge gaps.

This prompted Elaine to examine each part of the business and identify open issues. Some, such as reducing the burn rate and the head count, were obvious for an experienced VC partner, and the actions to address them were very clear. However, she began to realize that the fundamental concern was not within the company but in the question of, "Is this a fundable business?" Writing down the problem areas resulted, after some consultation and iteration, in the diagnosis summarized in Table 4.1.

Problem area 10 wasn't a problem at all; rather, Elaine found out that the customer support team was one of the key assets of the company. Problem areas 4 through 9 were relatively straightforward—they contained risks, but it was quite clear what had to be done. Problem area 3 was more difficult: The software product had some functionality gaps, and it was not entirely clear how to plug them. However, a hypothesis was already on the table, namely, that the rigidity might be mitigated by an interactive shell behind which the batch program would run.

Table 4.1: Problem Areas with Uncertainty Type

Problem Area	*Situation*	*Uncertainty*
1. Customer need (external)	Would customers buy Escend's products? Why? What is the customers' pain?	High potential for unk unks
2. Industry readiness (external)	No collaboration SW play has succeeded, willingness of players' openness questionable, what do other players want?	High potential for unk unks
3. Product functionality (external)	Holes in functionality, too rigid, based on once-per-day batch mode when customers wanted real-time product	Foreseeable, possible unk unks
4. Cash burn rate (internal)	$650,000 per month	Variation
5. Top management team (internal)	Complacent and dishonest, risk of lawsuit	Foreseeable
6. Sales team (internal)	Not sufficiently active; inflating results	Foreseeable
7. Head count (internal)	35 employees (70 at the high point, including 50 software developers), lack of performance	Foreseeable
8. Geographically dispersed operations (internal)	Sales offices in Alaska, California, Massachusetts, New Hampshire, and Texas; development team in Maine, Nevada, Pennsylvania, and Oregon	Foreseeable
9. Design team (external and internal)	"Old style" with outmoded (monolithic) architecture approach, uncooperative	Foreseeable
10. Support team (external and internal)	Strength of the company	Not a problem

The fundamental knowledge gaps lay in areas 1 and 2. The technology enabling Escend (XML and Rosettanet) had not existed before 2000. XML made collaboration possible from one database to another, and Escend aimed not only at supply chain management but also at demand creation across multiple countries. There were no competitors in this space; no one had defined the problem before, and no analysts were covering any companies in this part of the industry. It was uncharted territory—no one had gone there before. As a consequence, customer needs were undefined, and the ability of the players in the electronics industry to collaborate via common software was unclear. Thus, the required functionality was unknown. Customers could not articulate their needs, and different players would name mutually incompatible benefits because no one understood where the product would ultimately create the most value.[6]

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

During her first three months as CEO, Elaine traveled frequently to interview enterprise firms, end customers, analysts, consultants, VCs who did not invest (through not believing in either the management or the business model), managers of various collaboration startup companies, and academics. She searched and probed to find out why collaboration solutions had not succeeded, what the needs of the enterprise customers were, and what problems the complex industry structure really posed. In face-to-face meetings, people opened up and provided useful nonverbal cues—information that could never be obtained by only attending board of directors' meetings. Slowly, information emerged, but the information kept changing as the industry evolved.

After an intense three-week probing of the problem areas where unk unks threatened, Elaine made a leap of faith to conclude that: (a) demand for Escend's product existed; (b) the potential market was large enough to have the potential of a sufficient return on investment; (c) the competitors were far enough behind for the opportunity not yet to have been tapped; and (d) Escend represented one of the last remaining large enterprise software "plays" for VC investors. Elaine recommended that Novus provide additional funds, and in August 2003, Escend closed a $7 million round from Novus and NIF Ventures, an affiliate of Daiwa Securities.[7]

Thus, Escend was allowed to continue. According to the problem characterization in Table 4.1, Elaine Bailey managed part of it as planning projects, and parts of it as a learning project, in which she did not know what she would find. The management of the learning project is described in Chapter 5. In the remainder of this chapter, we generalize from Escend's initial turnaround phase and suggest how a project management team can go about diagnosing unk unks, as well as complexity.

4.2 Diagnosing Complexity

In addition to uncertainty, we argued in Section 3.4 that complexity is an important characteristic of projects. Complexity captures the number of interactions among different project pieces, and high complexity hinders the team's ability to "optimize" the overall solution. Thus, the project team needs to diagnose complexity at the outset, just as it diagnoses the uncertainty.

A practical tool, in the form of the *design structure matrix* (DSM), exists to map and illustrate the complexity of a project.[8] The DSM maps interactions among pieces of a project in matrix form, by listing which task needs input from which. Interactions may take the form of, for example, *information* (one task requires input from the other), *space or resources* (both tasks take place in the same spot or use the same resource), *material* (one task provides material that the other uses), or *constraints* (one task may produce a system parameter that constrains the things that are allowed in the other task).

Figure 4.1 presents a brief example of the application of this tool. In the figure, *impacted* tasks are listed in the columns, and *impacting* tasks along the rows. Crosses (x) mark dependencies. Task B is *sequentially dependent* on task A, as the impact goes only one way; for example, task A produces information that task B needs, or task A occupies space that task B must then avoid. Tasks B and C are *independent*. Tasks C and D impact each other—that is, they might require mutual information input. Thus, they are *coupled* (interdependent). The matrix suggests an ordering of the tasks: A, then B in parallel with C and D, the latter two being performed in a closely coordinated way.

Interactions occur in three domains simultaneously: the *system domain*, in which components interact, the *task domain*, in which project activities interact, and the *organizational domain*, in which teams and stakeholders interact. For each of the three domains, a separate DSM interaction matrix may be drawn up. All interactions are relevant for the project and may contribute to overall complexity, and yet it is useful to distinguish the domains because they are concerned with different types of interactions. Table 4.2 lists examples of interactions for the three domains.

	A	B	C	D	
Task A	A				A-B: Sequential
Task B	x	B			B-C: Independent
Task C			C	x	C-D: Coupled
Task D	x		x	D	A-D: Sequential

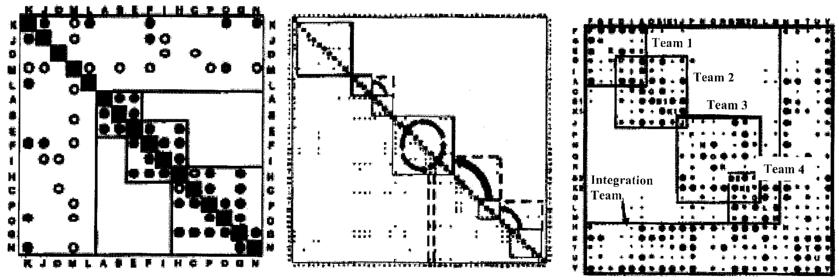
Figure 4.1 Example of a design structure matrix, DSM

Obviously, the three domains are not independent; they strongly overlap and influence one another. Linking the system and task domains, many tasks have a one-to-one relationship with system components (a task comprises work on a system component), and other tasks relate to interactions among system components (for example, coordination or integration tasks). Relating the task and organization domains, tasks are typically assigned to individuals, and highly interdependent task groups are arranged in organizational units. As a result of these relationships, the DSM interaction matrices for the three domains often look related, as Table 4.2 shows.

Figure 4.2 uses three DSMs to describe the interactions in a climate control system development project.[9] The first DSM lists the components (16 major subsystems in this case) and their interactions; the second lists the task interactions (there are many more tasks than components) and the interactions among the organizational groups, the subteams. Some subteams were assigned to component groups of the system, with some overlap across teams (members belonging to multiple teams) in order to ensure integration and representation of all expertise. One team worked on the integration of the components; therefore, it interacted with all the other teams simultaneously.

Table 4.2: Examples of Interaction Types in Three Domains

System Domain (among system components)	***Task Domain (among project tasks)***	***Organizational Domain (among actors or groups)***
Spatial (e.g., components occupy the same space) Energy (e.g., one component supplies the other with electricity) Flows (e.g., one component transfers fluids or other materials to the other) Force (e.g., one component exerts pressure on the other) Information (e.g., in software, or control units) Synchronization (e.g., two components must deliver a flow of signals simultaneously)	Spatial (e.g., two tasks work on the same system component and cannot, therefore, proceed simultaneously) Resources (e.g., two tasks compete for access to the same test facility, or need the same expert personnel) Artifacts (e.g., a task delivers a prototype for further development) Information (e.g., one task delivers information that serves as input for the other task)	Groups are affected by the task domain interactions because they must carry out the tasks In addition: Power, status, influence (e.g., the group that "wins" in a resource competition has more status) Goals (e.g., groups' interests are incompatible: one group wants higher performance; the other wants lower costs)



System Component Interactions (Components × Components) | **Project Task Interactions** (Tasks × Tasks) | **Organizational Group Interactions** (Individuals × Individuals)

Figure 4.2 Interactions in three domains in a system development project

It is not surprising that the three DSMs in Figure 4.2 are related—component interactions drive task interactions, which, in turn, are mirrored in team interactions.[10] However, the similarity is not perfect; some task interdependencies are introduced by the way the project is organized, and the project tasks ignore some component interactions (for example, those that are judged less critical). Similarly, interest conflicts among teams may be independent of the system that is being built.

The DSM becomes heavy and cumbersome when it is used in a large project to assign tasks and personnel in detail: When the matrix grows beyond a size of 50 x 50, it becomes difficult to administer. However, the DSM is an excellent tool for diagnosing complexity at the outset of the project, at a semiaggregated level:

- Identify the main subsystems (or components) of the system that the project builds (no more than about 20).
- Use the work breakdown structure to identify major (aggregate) tasks of the project (no more than about 40).
- Use a stakeholder mapping to identify the major project subteams (no more than about 5) and the other stakeholders (no more than about 15).
- Construct the three DSMs, even if the interactions have to be estimated.

Complexity can now be roughly estimated as a product:

$$\text{Complexity} = (\text{Sum of all project elements}) \times (\text{Sum of all interactions})$$

Here, the sum of all project elements includes all three domains, subsystems, tasks, *and* stakeholders; similarly, the interactions include all three domains. First, this reflects the fact that complexity can arise from different sides, the system architecture, project organization, and the stakeholder interest constellation. Second, this complexity measure reflects the fact that complexity is driven by the combination of the number of elements and their interactions. In other words, if there are very few interactions, complexity is low: Even a large project (with many elements) is not complex if the elements do not interact; the elements can be tackled one by one. If the project is very small (few elements), complexity is low, even if those elements heavily interact. On the other hand, the number of elements and their interactions compound one another; adding a few interactions to a large system makes complexity *much* worse.

This measure of complexity refines the distinction of "task complexity" and "relationship complexity" that we made in Chapter 3. Moreover, this measure can be estimated (albeit roughly) at the outset of a project, when a first work breakdown structure and project organization have been determined. Complexity heavily influences the solution approaches that are feasible in a project, and, moreover, complexity and uncertainty interact in their impact on the best project approach, as we will discuss in detail in Chapter 7.

4.3 A Process for Diagnosing Uncertainty and Complexity at the Outset

In Sections 4.1 and 4.2, we showed how unk unks and complexity can be identified at the outset of a project. We now attempt to make this identification systematic in a *diagnosis process*. Identifying complexity requires listing the project elements and estimating their interactions. Identifying unk unks poses the apparent paradox of identifying something that is, by definition, unforeseeable. In essence, resolving this paradox requires asking: *What do I know and what do I not know?* and *Where are the major knowledge gaps?* Every knowledge gap poses the potential of unk unks. The diagnosis process is outlined in Figure 4.3.

1. First, identify the problem structure, as Elaine Bailey did when she identified the mess she had inherited at Escend: Do we understand the ultimate goal for the project as it is currently defined? Who are the stakeholders that may influence the outcome of the project? Do we have some understanding of the causality of actions and effects in the project?
2. Then, break the overall problem ("How do we generate enough sales?") into pieces: What are the modules or subprojects that require our attention? Who are the major players and/or stakeholders? At Escend, Elaine broke the problem up according to the market forces (customers, competitors, and partners) and the functional teams in the organization (sales, cash management, design, and so on; see Table 4.1.
3. For each piece, perform risk identification. In other words, assess the scope of variation in budgets, schedules, and achieved performance, and identify major risks that need managing (with standard PRM methods). Also, identify knowledge gaps, by probing assumptions and asking what you know and what you do not know. This identifies areas of potential unk unks. As the events at Escend demonstrate, this is a highly iterative and gradual process.

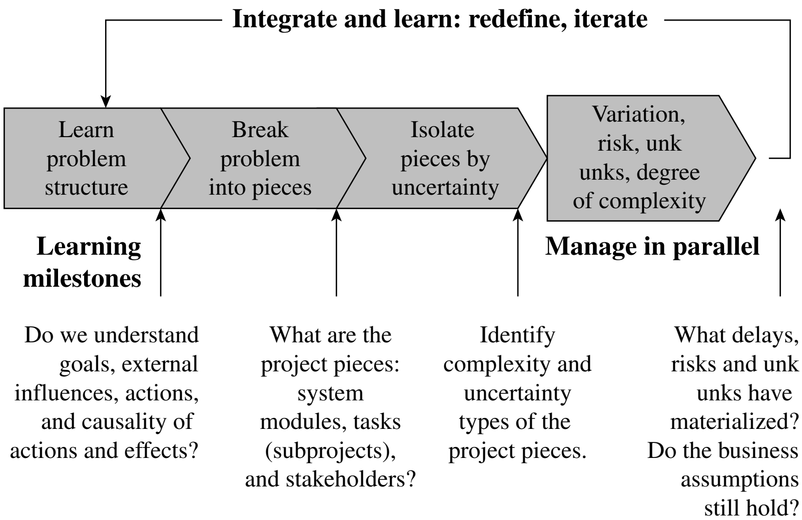


Figure 4.3 A project diagnosis process

4. Estimate the complexity of each project piece, and of the overall project, from the respective number of elements and interactions. At Escend, complexity was not a major consideration; the interactions between the system configuration and the industry players were understandable. The challenge stemmed from the unforeseeable uncertainty of not knowing where this market was going.
5. Finally, manage the pieces in parallel, using different management approaches according to the combination of complexity and uncertainty: planned execution and traditional PRM for the pieces with foreseeable uncertainty, some selectionism for the highly complex subprojects, and a combination of selectionism and flexible iteration for the pieces that are threatened by a combination of complexity and unk unks. We have not yet discussed what the "right" combination is; this will be the subject of Chapter 7. As the startup progresses toward the next milestone, or IPO, new information emerges, and the problem pieces and approaches must be updated and modified.

We have found it helpful to summarize the characterization of the project pieces (the third learning milestone in Figure 4.3) with *uncertainty and complexity profiles*. Project managers and their teams do not need to have an exact analysis of the uncertainty and complexity challenges they face to decide upon the appropriate management tools to use. What they need is a prioritization tool that helps them to define the relative importance of each source of influence at the starting point of the project, and eventually to evaluate how the portfolio of complexity and uncertainty influences is evolving over time.

After identifying the uncertainty and complexity types in Step 3 of Figure 4.3, we score the importance of each influence on a scale of 1 to 10. The importance can be a potential impact or the amount of time that the team thinks they need to spend on understanding and addressing them. Figure 4.4 demonstrates the idea with three brief examples of uncertainty and complexity profiles, summarized at the level of entire projects.

The project on the left represents the construction of a cruise ship at a major French shipyard. In naval construction, many contractors perform tasks in parallel; it is easy to imagine a site with 2,000 people from 500 subcontractors working at the same time building the restaurant, the casino, the cabins, the swimming pool; installing the wiring for the Internet system; tuning the engine; and so on.

The components of the ship do not strongly interact, and many of them can be built separately, off-site. The key interactions, for components as well as tasks, are space constraints and supply connections. Most of these interactions can be managed by defining good interfaces. A large project size, combined with relatively few interactions, implies medium to high (not extremely high) complexity. Thus, a delay or quality problem of one contractor may cause a chain reaction of problems for other contractors. This level of complexity already requires diligent coordination among many subcontractors who work on the site at the same time.

Variation due to bad weather, or quality problems or delivery delays by suppliers, may influence the execution of the plan. Variation is therefore estimated as medium. This requires buffer management to control schedule variation. However, for most of these cruise ships, the design is well known. There are really no significant gaps in knowledge, and therefore, the potential for foreseeable and unforeseeable uncertainty is limited.

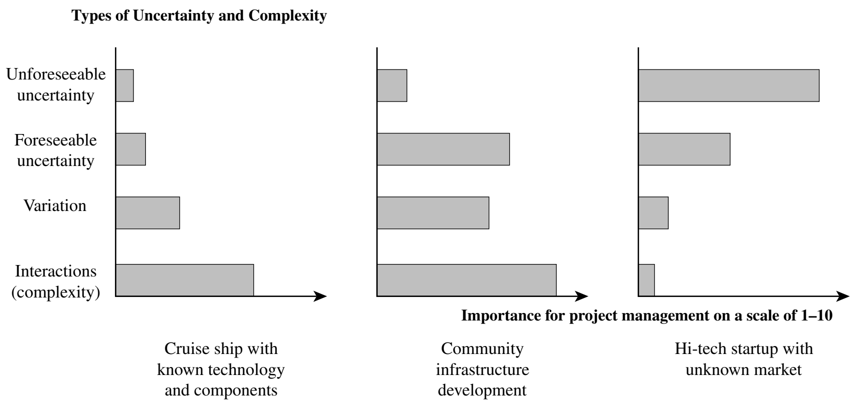


Figure 4.4 Uncertainty and complexity profiles: Prioritization tool

The project in the center of Figure 4.4 refers to a residential community construction project, Ladera Ranch in Southern California. This was a multiyear, several-hundred-million-dollar earth-moving infrastructure (water, sewers, electricity) development project to prepare the construction of a new community of several hundred houses. The project team moved millions of cubic yards of earth to provide independent builders with house pads, streets, water runoffs, landscaping, and utilities.[11] Their major objective was to plan the cuts and fills in a way that moved earth the shortest distances possible.

Since the project could be completed before other builders moved in, the team had much freedom to work at multiple places in parallel; in other words, task complexity was low, and thus, coordination was easy to ensure. However, the complexity of the various "components" (the individual lots) was very high: The efficiency of operations in such an earth-moving project was influenced by soil conditions, in particular, moisture and composition. Moist soil required more excavation and took longer to settle before one could build on it, so the project team had to consider mechanically drying the dirt rather than delay selling lots. Some types of soil required different slopes for stability, influencing the available amount of flat area for houses and streets.

This could all be planned, in principle, by drawing up a PRM contingency plan for each scenario of the type: "If the soil is moist and type x at location y, do plan A; if it is dry and soil type z, do plan B." But in practice, this contingency planning was infeasible because of the interdependent nature of cuts and fills across locations. The number of scenarios proliferated with the number of locations considered, making the moisture level and exact soil type, in effect, unpredictable. Thus, the Ladera Ranch team was forced to dig and then, based on their experience, react to what they found. In effect, component complexity forced an experimental approach. Thus, complexity and foreseeable uncertainty were closely related, and both rated high in this project.

However, truly unforeseen events, which would completely alter project execution, were rare. Examples of such events would be the discovery of prehistoric Indian ruins or a rare animal or plant species. Therefore, the score for the relative importance of unforeseeable influences, shown in Figure 4.4, is low.

The project in the right-hand panel of Figure 4.4 refers to an Internet startup company that was created in 1999 in order to apply the Priceline reverse-auction business model to the German market. A major challenge for this company was the unpredictable reaction of German consumers to this business model.[12] Despite numerous changes in the selling process to accommodate the specific preferences of the German consumer, by mid-2000, the company could see that the consumer auction boom was faltering.

Knowing that it could not survive on customer-driven pricing alone, the startup radically changed its business model. It developed software services for industrial customers and an Internet-based ticket search engine for travel agents. By summer 2001, this search engine, which dynamically optimized offers from multiple airline reservation systems, had become the most promising of the company's offerings. The change in the business model was quite dramatic. In response to the redefinition of its mission, one of its investors commented, "How can they change the business model this much? It is like we gave them money to develop a sausage factory, and now they tell us they have moved into building fighter planes." In the end, the company succumbed to the decline of the travel business after the burst of the Internet bubble and gave half of the invested money back to the investors in 2002.

In this startup company, neither complexity and coordination nor variation, foreseeable uncertainty, or managing the daily operations was the biggest challenge. The biggest challenge was to keep the finger on the consumers' pulse, to see how they perceived the business model and how their tastes evolved. The challenge was to cope with the unk unks in a new market.

Our approach, as summarized in Figures 4-3 and 4-4, is related to *discovery driven planning*,[13] which proposes to explicitly acknowledge unk unks and to uncover them with four analyses: (a) a reverse income statement calculates what would have to be achieved in terms of market share and revenues in order to reach a given return target; (b) a pro forma operations specification shows the key steps for producing the desired output and asks whether these steps can be performed with "normal" process capabilities (or whether heroic feats are required); (c) an assumptions checklist compares the plan with experiences in similar situations or with expert advice (e.g., "we assume the average selling price to be around $1.60—is that justified?"); and (d) milestone planning anticipates at what points which risks can be eliminated.

Discovery driven planning is very useful, but has two limitations. Examining the estimated causal structure of profits is an obvious start but runs the risk of missing the most dangerous unk unks: The venture's plans are carefully worked out, everything is consistent and plausible, and still, unexpected factors emerge later that make the entire business model obsolete. Moreover, "checking assumptions" means that we ask for each parameter whether it is really fixed, or whether we can diagnose a relevant unk unk (an influence that may change and influence the outcome). Again, this is a good start, but may lead to the "double blindness" that we have discussed in Chapter 3, and it will not suffice when causal relationships are unknown and influences may arise that the team cannot identify through critical questioning at the outset. Thus, discovery driven planning demands a level of knowledge and structure that may not be available. It does not utilize knowledge and intuition possessed by the management team, which may be less precise than identifying individual influences, and which we propose to tap.

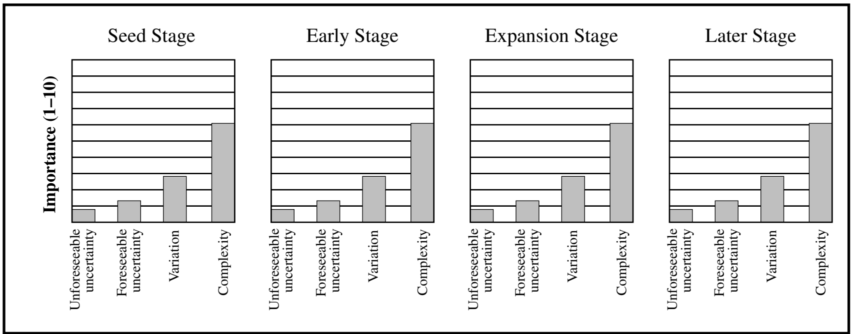
4.4 Evolve the Complexity and Uncertainty Profile

For some unk unks, identifying their *possibility* immediately identifies *them*, and thus shifts them to identified risks. For example, realizing that patients may not comply with the restrictions of use for a newly developed drug immediately transforms this from an unk unk, something the team did not consider, to an identified risk. This is, of course, precisely the purpose of the risk identification stage of established PRM methods. However, in novel projects, there are, typically, areas of knowledge gaps that can be identified as such, without being able to identify the risks themselves, as we have seen in the Escend example and in Section 4.3.

Let us consider another example. Many companies that entered the Chinese market in the late 1980s or early 1990s knew that they had little knowledge about doing business in China, and that most consultants who claimed to be able to help were unreliable. In other words, these companies were aware of their knowledge gaps and of the presence of unk unks. Some companies tried to protect themselves against the unk unks by teaming up with a "reliable" local partner. Disturbances were still to be expected in the implementation of the "project" of market entry. But they were expected to be reduced to variance or foreseeable influences, because the Chinese partner was brought on board with sufficient experience to handle these disturbances. To the dismay of many early entrants, these partners themselves often turned out to be sources of unforeseeable uncertainty: They knew far less about the market than they had pretended, they were not prepared for sudden changes in the policies imposed by the provincial or municipal governments, or they had a hidden agenda that made them unpredictable.

A key rule in using complexity and uncertainty profiles is that, over the course of the project, they change as the team learns. At critical transition points in the project, major uncertainties emerge or are resolved, and only after significant progress is made will the team know what is ahead. This is summarized in Figure 4.5.

Let us take a startup company, such as Escend, as an example. It will go through various stages of its existence. It may start as a simple idea in a university laboratory. A business angel may be prepared to put in some time and money to develop the initial concept. Once the concept proves attractive, the real startup happens. The product is developed but may not be completely operational. Customers are tested; if their reactions are positive, the company can obtain additional financing. The product is launched and tested, and the first revenues come in. Once this stage is successfully passed, the company expands and may become ready for an IPO.



Required Management Approach	Seed Stage	Early Stage	Expansion Stage	Later Stage
	• Planning • PLUS learning • PLUS parallel trials • PLUS repeated project redefinition	• Planning • PLUS learning • PLUS parallel trials • Perhaps complete project redefinition	• Planning • PLUS learning • PLUS parallel trials	• Mainly planning and disciplined execution

Figure 4.5 Evolution of complexity and uncertainty profile

The types of uncertainty change as the company progresses. In the first stage, when the idea is still the dream of a few founders, complexity is almost nonexistent, but unk unks are everywhere: How will we find the technical solutions? What shape could the market take? How do we best finance the project? And so on.

Bringing in a business angel with a lot of experience reduces some of the unk unks to foreseeable uncertainty. For example, a business angel may be able to help reduce the market uncertainty and propose contingencies in the financial construction of the company (like a decision tree, as illustrated in Section 3.3): "If we can obtain this financing, we may also be able to convince that party." For products that push the technological boundary, unk unks about technology and market definition may still exist. However, the relative importance of the various sources of uncertainty is evolving.

Once first customer tests have succeeded, market uncertainty may well be redefined as variance: The direction of the market is becoming clear, but it is still uncertain at what speed customers start adopting the product. Once the startup enters the preparation stage of an IPO, management attention shifts from managing unk unks and foreseeable uncertainty to managing complex relations with investors, investment analysts, partners, and the stock exchange, not to mention the lawyers.

4.5 Conclusion

In this chapter, we have extended the risk identification phase of PRM to include the diagnosis of complexity and the identification of knowledge gaps, signifying the presence of unk unks. In the example of Escend, we have shown that, although unk unks themselves are unforeseeable, the project management team can recognize knowledge gaps. The areas of knowledge can then be searched and probed, in parallel with project execution.

The project profile diagnosis, complexity identification, and probing for unk unks can be pursued as a systematic diagnosis process. This process is essentially a search with an unknown outcome, so it is inherently iterative. The result of the probing can be summarized via a simple qualitative representation tool, the complexity and uncertainty profile, to estimate the relative importance of each type of uncertainty. The uncertainty profile allows the project management team to choose the best management approach for the project.

Endnotes

1. This section is based on Loch, Solt, and Bailey 2005.
2. OEM stands for "original equipment manufacturer." OEMs sell products to the end consumer. OEMs often perform marketing and system design but outsource component design and manufacturing.
3. The investors, accepting top management's view, thought maybe that the Escend story was just not being told well, and they, led by Elaine Bailey, rewrote the executive summary, generated a "VC presentation," and taught the CEO how to give the presentation. The result was zero interest by potential new VC investors.
4. Elaine's experience running her own rep firm in the 1980s made her the logical choice and meant that she knew the "right" questions to ask—or that she would discover what they were.
5. The lead VC is primarily responsible for performing the due diligence that supports a recommendation, and in the tight-knit VC community, bad decisions (such as "throwing good money after bad") tarnish reputations.
6. It is well documented that technologies that enable new uses in the market cause unforeseeable effects, see, e.g., O'Connor, G. C. and R. Veryzer 2001; see also Leonard-Barton 1995.
7. Elaine reports that she almost threw in the towel in the run-up to the recommendation. The situation was too confusing and opaque. The discussion of unk unks and flexible adjustment to what lay ahead gave her the mind-set of possibilities, and the will to continue.
8. The DSM was first proposed in engineering by Steward 1981. Eppinger et al. 1994 developed it further as a management tool. The illustrative example in Figure 4.1 is based on Loch and Terwiesch 2000.
9. Figure 4.2 is adapted from Eppinger and Salminen 2001.
10. Sosa et al. 2005 quantify the similarity among the three matrices in a statistical sense and find that there is, indeed, substantial overlap. Moreover, these authors show some first evidence that congruence between system, project, and organizational architectures lead to higher performance of the project team. The work by these researchers has emphasized the usefulness of the DSM for assigning tasks and teams in a way that minimizes interactions, and thus complexity. In this chapter, we emphasize the usefulness of the DSM as a diagnosis tool (at a less precise, more aggregate level) at the outset of a project.
11. For further details, see De Meyer et al. 2002.
12. For example, Germans did not want to pay via credit card (habituation took four years), and many did not want to commit to an offer without knowing the price. Other challenges were the fact that business processes could not be patented in Europe, and the relative scarcity of venture capital in Germany. The company invited "strategic investors," large companies who had an interest in developing alternative Internet sales channels, in the hope that they would support the startup's growth. This turned out to be a misjudgment, as

it dragged the startup into some large-company internal resource struggles that prevented effective support by the strategic partners. For details, see De Meyer et al. 2002.

13. Discovery driven planning was developed by McGrath 1995, and McGrath and MacMillan 2000.