Test management: The greatest risk

Risk expert Felix Redmill returns to consider in detail the influence of decisions made by the test manager on projects, teams and deliverables

Testing the software of a system under development is a project in its own right. To be effective, it requires careful planning. As there are many factors that force testers to deviate from test plans, it requires dynamic project management. The test manager needs to be a project manager, and a good one - not merely an adequate inline manager.

All managers need to make decisions. They must plan and monitor progress and direct staff to where they are most needed. They must expect change and, when it occurs, they must make decisions - to reverse it, adapt to it, or capitalise on it. A job in which decision-making is not prominent may be supervisory, but it is not managerial.

A project is a dynamic approach to achieving objectives, and decision-making is critical to it. Change is likely to be frequent and deviation from plans almost inevitable. For a project manager, controlled reaction to change - often sudden change - is crucial, but the ability to predict variations and act early to neutralise their effects is even more so. The rate of change can be overwhelming and often traps project managers into a wholly reactive mode. But it is possible for good project managers to detect the causes that will lead to changed circumstances, foresee their likely effects, and act to counteract them - or even to derive advantage from them. Such project managers plan carefully but recognise their plans as being only tools for meeting their objectives. They continually check their plans against the objectives, detect when the plans diverge from them, and replace obsolete plans with new ones. This takes courage, for having invested in drawing up plans, the instinct of all but a few is that their prime objective is to meet them.

All managers should be aware of the risks that threaten their goals. Projects, by their nature, can be particularly risky. Testing projects, if not set up appropriately, if not properly integrated into their parent development project, and if not subjected to the highest quality of dynamic management, can be more risky than most. Both time and budget allowances are often too small to start with, and the need for re-testing can rapidly stretch them beyond their defined limits. Schedules are difficult to plan, and late delivery of software by the development team can throw them into chaos. Changes in staff requirements can occur suddenly and frequently, and their amplitude can be considerable.

All project managers need, at some time, to make decisions based on inadequate information; but it is often possible to foresee the need for a decision and set out to obtain the necessary information in advance, even when time is short. Good decision-making is a requirement of all project managers, and the attribute should be exceptionally well developed in the managers of testing projects.

In the context of such projects, there are three principal categories of risk to be considered. First there are the risks related to the operation of the system - from the tester's perspective, the 'forward risks'. These may be used as a basis for the planning of 'risk-based testing'. Second, there are the 'backward risks' - those imposed on the testing project by the software developers, through uncertainty and poor quality in all its forms. Third, there are the risks created within the testing project itself. These arise from the potential for deficient leadership, teamwork, training, allocation and co-ordination of effort, prioritisation of work, and so on.

If the forward risks are to inform test planning, they must be understood. The test manager must identify and analyse them, use them in test planning, and then, when things go wrong, re-employ them as the basis of re-planning. This is not trivial and has been the subject of previous articles.

The second category, the backward risks, may at first seem beyond the control of the testing project. But if the test manager has a seat on the project board, and is involved in project planning, she can influence the co-ordination of development and test planning. Moreover, the test manager's relationship with the development manager is not limited to that of a colleague but extends to those of both supplier and customer. She provides a testing service, which should be planned and managed, and so is a supplier to the development team. She is a customer in that she relies on the development team for the supply of software, which should be delivered in accordance with time and quality criteria. Unstructured, badly documented, and inadequately inspected code, delivered late, does not have to be the norm. Nor does it have to be accepted. Professionalism is not limited to 'doing a good job'; it demands a good and reliable job of others on whom we depend. Professionals should specify requirements for good quality in the transactions between them. Appropriate advance agreements can make the rejection of bad quality a normal response to an 'exception' or 'mistake' rather than an embarrassment. If the test manager interacts regularly with the development manager to agree delivery and acceptance criteria, discuss problems, make and receive suggestions, and plan improvements, she not only reduces risks but also acquires information to inform her management of those that remain.

Thus, there are many ways in which the test manager can significantly reduce the backward risks to the testing project. By taking a risk-based approach, deciding that all risks should be targets for reduction, and actually taking steps to identify and reduce them, the best test managers can overcome problems that others would consider external, inevitable, and uncontrollable.

Risks in the third category - those created within the testing project - have various causes, but all are within the ambit of the test manager's decisions (which is not to say that the test manager can eliminate them). Projects, and indeed all fields of management, are subject to risks that arise from decision-making. As testing projects are decision-rich, this applies especially to them. Some decisions carry risks of the first category - forward risks. For example, we must decide not only what to test and how to test it, but also what to test first. What, because of the risks attached to it,

must be tested thoroughly even if we are short of time? But then, how do we know? Do we understand the potential consequences of not testing these modules or subsystems? A test manager who has not analysed the consequences cannot honestly claim to know the risks involved and so lacks the basis for confidence in her decisions.

Other decisions, such as those concerning the scheduling of work, carry risks of the third category - to the testing project itself. For example, how do we know how much time to allow for testing a subsystem? A first shot may be to estimate how long or how many man-hours it will take to execute its test plan. But this is the minimum allowance. And anyway, what confidence does the test manager have in the estimate? Is it an 'educated' guess, is it derived from experience of testing many similar subsystems, is it based on knowledge of the testers assigned to the task? If it is an optimistic estimate, and the job is given to inexperienced testers, a delay in the schedule would almost inevitably result - and is therefore predictable.

The scheduling of time and resources for re-testing creates other problematic and risky decisions. If these things depend on the development team, how can we know? If the development manager assigns an inexperienced or rogue designer or programmer to the job of correction, or neglects supervision, or fails to enforce verification, re-testing may be prolonged - and may even need to be repeated. What is the probability of this? The risk cannot be avoided, but it can be reduced if the test manager addresses the matter not only within her team but also with the development manager.

> "A project is a dynamic approach to achieving objectives, and decision-making is critical to it"

Then there is the matter of contingency. How much should be included in the test team's staffing and resources? Too much would increase the budget unnecessarily and, perhaps, ruinously. Too little could cause small delays to accumulate and cause project overrun, inefficient testing, and a failure to meet test objectives. It is never possible to predict future contingency needs accurately, but the test manager needs to amass evidence from both sides of the question so as to balance the risks and make decisions. Decisions are both the job and the scourge of the test manager.

Typically, software development projects are problematic, and the testing projects within them particularly so. The fact that a major cause is inferior project management is often overlooked. Habit leads delays and inefficiencies to be perceived as 'the sort of thing that you can't foresee' rather than as the results of decisions that might have been better. Every decision carries risk, and decision-making should be informed by advance information gathering, supported by an assessment of the risks, and protected by risk-management activities when appropriate. In simple circumstances intuition may suffice, but many project situations require an understanding of risk analysis and management.

The test manager's decisions are crucial to the success of a testing project. They affect not only system operation but also the efficiency and effectiveness of the test team. Does senior management understand the importance of appointing an able and appropriate test manager, and of then providing timely and ample support? Do they realise that their selection is likely to be the decision that carries the greatest risk to the testing project? PT