Sources of Scope Risk

Scope risks are quite common in the PERIL database, representing almost 40 percent of the data. Even more important, risks related to scope accounted for well over 40 percent of the total schedule impact. The two broad categories of scope risk in PERIL relate to changes and defects. By far the most damage was due to poorly managed change (about three-quarters of the overall scope impact and roughly one third of all the impact in the entire database), but all the scope risks represented significant project exposure. While some of the risk situations, particularly in the category of defects, were legitimately "unknown" risks, quite a few of the problems could have been identified in advance and managed as risks. In this table the two major root-cause categories for scope risk are separated into more detailed subcategories.

Scope Root-Cause Subcategories	Definition	Count	Cumulative Impact (Weeks)	Average Impact (Weeks)
Changes: Creep	Any nonmandatory scope change	141	1218	8.6
Changes: Gap	Legitimate scope requirements discovered late in project	203	1639	8.1
Defects: Software	System or intangible deliverable problems that must be fixed	67	524	7.8
Defects: Integration	Program-level defects that require scope shifts in projects	15	101	6.7
Defects: Hardware	Tangible deliverable problems that must be fixed	58	389	6.7
Changes: Dependency	Scope changes necessary because of external dependencies	23	115	5.0

Scope changes due to gaps were the most frequent, but scope creep changes were the most damaging on average. A Pareto chart of overall impact by type of risk is summarized in Figure 3-1, and a more detailed analysis follows.

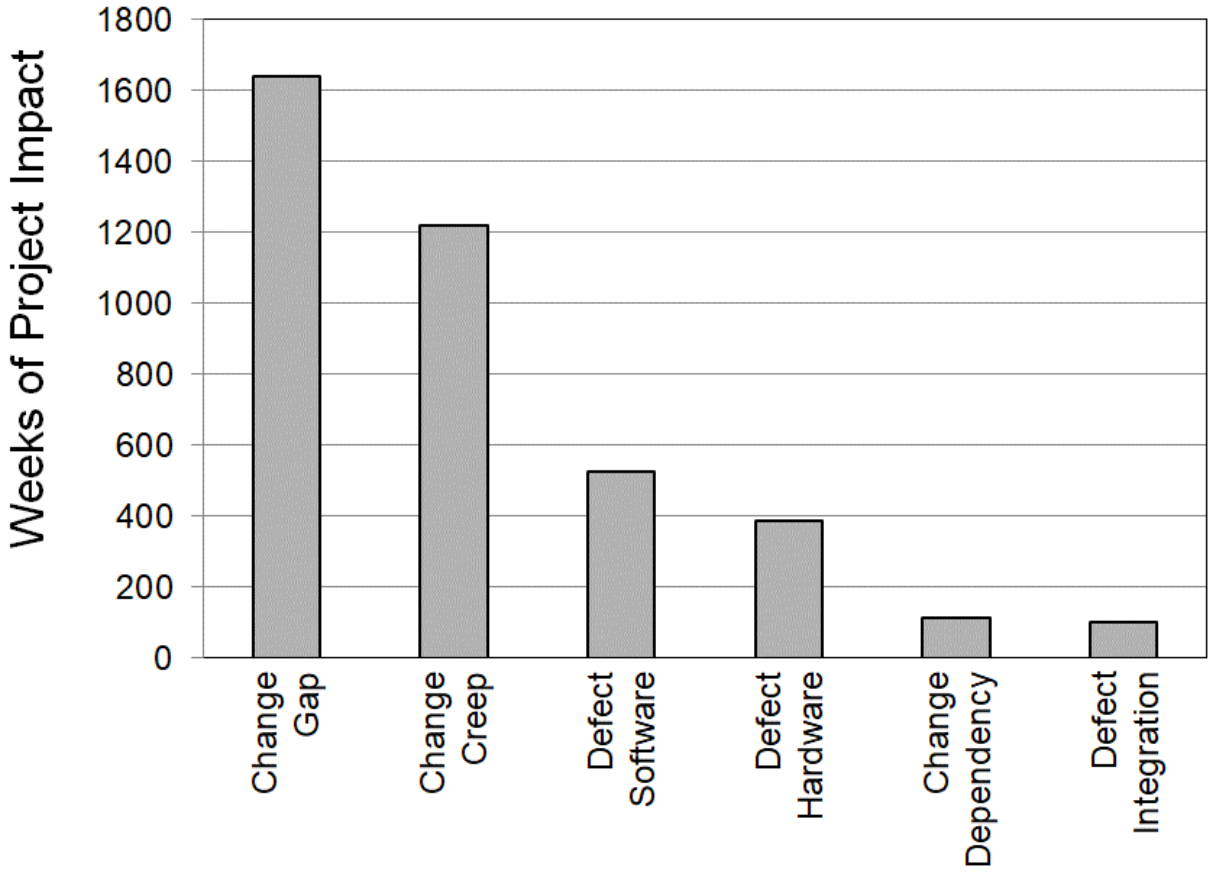

Figure 3-1: Total Project Impact by Scope Root-Cause Subcategories

Change Risks

As they say, "shift happens." Few if any projects end with the original scope intact. Managing scope risk related to change relies on tying up loose ends on requirements at project initiation and having (and using) a robust process for controlling changes throughout the work. In the PERIL database, there are three categories of scope change risks: scope gaps, scope creep, and dependencies.

Scope creep was the most damaging type of change risk, resulting in an average schedule slip of well over eight weeks. Scope gaps were only slightly less damaging, and these gaps were both more common and represented greater total impact. These two subcategories alone account for nearly a third of all the reported impact across the PERIL database.

Scope gaps are the result of committing to a project before the project requirements are well understood. When legitimate needs are uncovered later in the project, change is unavoidable. Some of the overlooked requirements are a consequence of the novelty of the project, and some are because customers, managers, team members, or other project stakeholders were not available (or not consulted) at project start. Although some scope gaps are hard to avoid, most of these cases are due to incomplete or rushed analysis. A more thorough scope definition, backlog inventory, or project work breakdown would have revealed the missing or poorly defined portions of the project scope. In the VUCA model, most scope gaps are a result of ambiguity.

Scope creep plagues all projects, especially technical projects. New opportunities, interesting ideas, undiscovered alternatives, and a wealth of other information emerges as the project progresses, providing a perpetual temptation to redefine the project and to make it "better." While some project evolution of this sort may be justified by clear-eyed business analysis, far too much nonmandatory change sneaks into projects because no analysis is done or because the consequences are drastically underestimated. The damage done by scope creep is often increased because the assumed benefits of the change are never fully realized. In retrospect, much of scope creep delivers little actual added value, and in extreme cases it can result in enough delay that the ultimate deliverable has no value at all. Scope creep requires unanticipated additional investment of time and money, generally demanding new effort and redoing already completed work. To manage scope creep, you need an initial requirements definition process that thoroughly considers potential alternatives, as well as an effective process for managing specification changes throughout a project.

Scope creep can come from any direction, but one of the most insidious is from inside the project. Every day a project progresses you learn something new, so it's inevitable that project team members will see things that were not apparent earlier. This can lead to well-intentioned proposals to "improve" the deliverable. Sometimes, scope creep of this sort may become part of the project without much broad awareness and within a portion of the project where the shift seems harmless. Only later may real, and sometimes catastrophic, unintended consequences emerge. Particularly on larger, more complicated projects, all changes deserve a thorough analysis and public discussion, with a particularly skeptical analysis of all alleged benefits. Scope creep is a good example of the first element of VUCA—volatility. Both scope creep and scope gaps are universal and pervasive issues for technical projects.

Scope dependencies make up the third subcategory of change risk. These scope risks are due to factors external to the project that affect scope. (Dependency risks that are primarily due to timing rather than requirements issues are defined as schedule risks in the database and will be explored in the next chapter.) Although less frequent in the PERIL database, scope dependencies did represent an average slippage of well over a month. Admittedly, some of the cases in the database involved situations that no amount of realistic analysis would have uncovered in advance. Most examples, though, resulted from factors that should not have come as complete surprises. Although legal and regulatory changes do sometimes happen without notice, a little research will generally provide advance warning. Projects also depend on infrastructure stability, and periodic review of installation and maintenance schedules will reveal plans for new versions of application software, databases, telecommunications, hardware upgrades, or other changes that the project may need to anticipate and accommodate. From a VUCA perspective, most scope dependencies are a result of complexity.

Defect Risks

Most projects rely on many complicated things to work as expected. Unfortunately, new things do not always operate as promised or as required. Even normally reliable things may break down or fail to perform as desired in a novel application. Defects represent over a quarter of the scope risks and almost one-eighth of all the risks in the PERIL database. The three categories of defect risks are software, hardware, and integration. From a VUCA perspective, all relate to complexity.

Software problems and **hardware** failures are the most common types of defect risk in the PERIL database, roughly equal in frequency. As to impact, software defects resulted in almost eight weeks of delay, and hardware problems were a bit under seven weeks. In several cases, the root cause was new, untried technology that lacked needed functionality or reliability. In other cases, a component created by the project (such as a circuit board or a software module) did not work initially and had to be fixed. In still others, critical items such as purchased components or custom integrated circuits failed and

had to be replaced. Nearly all of these risks are visible in advance, at least as possibilities, through adequate analysis and planning.

Some hardware and software functional failures related to quality or performance shortfalls. Hardware may be too slow, require too much power, emit excessive electromagnetic interference, or exhibit other unacceptable behavior. Software may be too difficult to operate, have inadequate throughput, or fail to work in specific circumstances. As with other defects, the definition, planning, and analysis of project work will help in anticipating many of these potential performance issues.

Integration defects are the third type of defect risk in the PERIL database. These defects relate to system problems above the component level. Although they are not as numerous in the database, they are quite damaging. Integration defects caused an average of about seven weeks of project slip. For large programs, work is typically decomposed into smaller, related subprojects that can progress in parallel. Successful integration of the deliverables from each of the subprojects into a single system deliverable requires not only that each of the components delivered operates as specified but also that the combination of all these parts functions as a system. All computer users are familiar with this failure mode. Whenever all the software in use fails to play nicely together, our systems lock up, crash, or report some exotic "illegal operation." Integration risks, though relatively less common than other defect risks in the PERIL database, are particularly problematic, as they generally occur near the project deadline and are never easy to diagnose and correct. Again, thorough analysis relying on disciplines such as software architecture and systems engineering can ensure timely identification and management of possible integration risks.

Scope Risk and VUCA

Although nearly half of the impact in the PERIL database overall relates to uncertainty, scope risks mainly originate with the other three VUCA factors. Ambiguity (a lack of clarity in objectives) leads the way with about 40 percent. Not far behind, and causing approximately equal damage, are volatility (shifting goals) and complexity (insufficient understanding of problem characteristics). Being aware of these VUCA considerations can help in finding and documenting scope risks. You can minimize the effects of VUCA through better definition of requirements, disciplined assessment of potential alternatives and options, feasibility studies and iterative development techniques, and thorough analysis of project complications. Reviewing the example scope risks from the PERIL database listed in the Appendix can be a good starting point for uncovering possible scope-related problems on your next project.