

Originally Published [MDDI](#) May 2005

Risk Management

Develop Defensively: Control Risk and Predict Results

Using risk management during product development is a given. But to achieve better and more-predictable results, it is important to take all aspects of risk into consideration.

Stan Telson

Product development is a complex activity; many important capabilities must be constantly addressed, especially in a worldwide market. It's no longer acceptable to apply the old cliché, "good, fast, or cheap—pick any two." Instead, investments must optimize many factors, overcoming the rigors of regulated product development and hurdles related to intellectual property, resource acquisition, and competition. Top-tier results for a medical device manufacturer depend on managing a richer palette of risks than just traditional regulatory issues and product liability concerns. Proper attention to the design process and to active management of all types of risk enables a company to achieve better and more-predictable program and product results, leading to higher-quality medical products and better financial rewards.

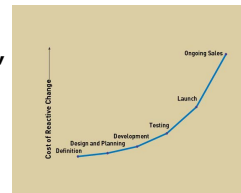


Figure 1. The cost of reactive changes increases exponentially throughout the product life cycle (click to enlarge).

Beyond Regulation and Liability

Some developers and project leaders consider only one aspect of risk management, failing to take advantage of the full suite of risk-related best practices and an available array of broadly capable design processes. Everyone already knows about managing regulatory and liability risks—but what else is available to product developers?

The design process requirements used within the medical device and diagnostics industry can be found in the quality system regulation (QSR; 21 CFR Part 820.30). These requirements help establish a framework for regulatory compliance. Of course, mere compliance with the regulation does not ensure success; however, failing to comply can create serious problems that may be fatal to an organization. Attention to design controls and other requirements is necessary, but not sufficient, for sustained business results. Regulated compliance must be managed.

The other commonly understood risk element is the traditional risk of liability. Managing such risks has an important place at the table but is neither automatically necessary nor sufficient in the big picture. All of

the well-understood techniques for managing liability (such as compliance with regulations, evidence of intent to do the right thing, etc.) are useful; however, more steps are necessary for sustained business success.

Although preventing liability is a large risk area to be managed, it's one area that can also have a partial safety net of insurance. While insurance does not prevent a problem, it may help mitigate the financial damage.

General Risk

Setting aside regulatory and liability risk management leaves a much larger group of risk-related domains. These areas can best be summarized as the risk of missed expectations. Successful development programs depend on management of countless factors, including the risks and unintended consequences associated with project details.

Risk management is important to a company's body of project knowledge, but successful risk management requires intersecting project management and technical skills with understanding, experience, and judgment. Addressing risk needs to be integrated into the core of both project management and design practices. This general risk management usually is one of the traditional roles of senior, functional, and program management, but is also done by individuals at all levels making product and project decisions.

Product development decisions work in concert to create the products and services a company offers. Sometimes a decision that seems inconsequential could have a major effect on a product. One medical device manufacturer had a massive product recall because of a seemingly innocuous decision about how to manage a tiny bit of product implementation. This sort of decision might barely get tested in a design review; it was such a routine-sounding linkage of conditions. To manage implementation, a design team member used one byte of memory in an embedded software system in a nonstandard way to create a shortcut. This programming was properly reviewed and accepted as part of detailed design and code reviews.

Years later, the problem surfaced as an unexpected and unintended consequence. In hindsight, the problem was clear: an inadequately reviewed higher-level architecture masked the shortcut. The first reviews were effective within their scope, but the scope was too narrow, while the higher-level reviews (at a broader scope) were too shallow. Ultimately, the problem was not caused just by design process limitations; a risk-exposing architecture was also at fault.

Small and often-hidden development decisions can lead to major consequences years later. In the short term, such decisions can cause annoying and disruptive issues, schedule delays, design changes, and unplanned expense.

Managing Risk

A commonly understood definition of risk is "the probability of an

undesirable event occurring and the impact of that event occurring.”¹ In medical device development programs, there are countless risks. Using frameworks to assess risk both aids and complicates the challenge of leadership. One existing framework, failure mode and effects analysis (FMEA) tends to focus on product risks rather than the broader risk of missed expectations.

When properly applied, FMEA is a good technique. However, for design projects, the majority of the risk-related opportunities cannot be realized by creating an FMEA late in the design cycle, when decisions about architecture have already been made. Some companies even do FMEAs primarily for compliance-related reasons, rather than for the power of these tools to catch and mitigate other risks. To get the full benefit of risk management, the middle or end of a project is far too late to use only an FMEA. The framework doesn't work well until there's a product design to analyze.

Other companies assign the production of the FMEA so that it becomes a chore for the design team or a task for a specific person. In the latter case, it is a part of the project that someone else owns, rather than a well-understood and well-tuned product of the entire team.

Inspection and Prevention

FMEA is also a technique that designers use as an inspection tool to reduce risk. It's often a better idea to use inherently risk-avoiding processes that help to design out risk. An FMEA looks at the risk of product failure, which leaves project or program failures that could lead to unmet expectations unaddressed. In general, it's always most effective to prevent conditions that create risks, rather than to attempt to count on detecting and correcting them.

Risk management requires both inspection and prevention, but prevention is better in many ways. The reality is that inspection is never 100% effective at catching problems. And companies need to address all kinds of issues, including those that are often hidden, such as misunderstandings among the project team. It is better to prevent a misunderstanding with good communication and processes than to try to catch an error several design phases later.

A full range of effective approaches is needed to optimize results and control risk. The highest leverage comes from early decisions and early action, since the cost of reactive change grows exponentially throughout the product life cycle, as noted in Figure 1. A \$100 change as a specification clarification can become a \$1 million change later in the project or after market launch.

Figure 1 can be generalized to apply to any cost of change, but in this article, it relates to changes made in response to the earlier definition of the consequence of risk: the effect of a risky event occurring. Therefore, all types of risk management, including the risk of unmet expectations, benefit a company and industry in general.

Effective and Early Action

To control the risk of unmet expectations, device development

companies must first master the transitions from invention and research through phases including product development and launch. These transitions, while also important for process compliance, are perhaps the most important for controlling the risk of unmet expectations.

Simply said, development companies must answer certain questions. Will this specific design approach and implementation meet the performance and market expectations? Are the requirements properly and completely defined, and will this project team be able to deliver? Many companies set expectations too early. They advance the development process for an ultimately incomplete design concept, start an underresourced effort, or have some other hidden program flaws. By contrast, many companies use an effective and essential phase-review process with a senior product approval committee.² However, some companies have mature versions of such processes, yet still have significant problems. Why?

Unclear communication, oversummarization, and a failure to properly communicate status—for many reasons, including bad communication, real unawareness, or fear of consequences—can all lead to problems.³ Some corporate cultures discourage the communication of bad news, while others with technically capable design teams adopt a bravado, based perhaps on nothing more than the naiveté of youth or inexperience. In some cases, leaders are just not connected enough with their domains of responsibility.

The first change companies can make is to improve the preparation and content of their phase reviews. All device development companies should use a phase-review process. If a company already has a phase-review process, it is important to review the process with an eye toward ensuring that it addresses the risk of unmet expectations. Sometimes the purpose for the phase review is biased toward compliance, market success, or finance, rather than a balanced mix of factors leading toward success.

Awareness and Leadership

Because a lot of time and investment can pass between phase reviews, expensive risks that arise between reviews can be missed. Some organizations consciously pass a phase review with exceptions; but improperly addressed risks will resurface later in the cycle. Additional changes are needed on the part of both leadership and developers. Leaders should ask (and encourage others to ask), "What could go wrong?" often enough to add value by optimizing paths. Eventually, most everyone reaches a state where decisions are based on experience and good judgment, and the design processes are mature.

The leaders in improvement discussions should be a bigger group than organizations usually consider. Anyone making design and product decisions, from senior, program, and technical management down through any engineer or marketer should be included. Those defining

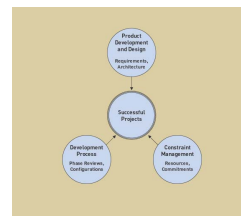


Figure 2. Risk management that addresses these categories and more leads to successful projects (click to enlarge).

requirements, making design choices, establishing processes, creating plans, testing, etc., often encompass almost everyone in the product development process.

It is critical for companies to encourage everyone to seek and inoculate against risks. Well-set expectations and good decisions, at multiple product and organizational levels, will lead to better results.

Trim Tabs and Rudders—What Else Can You Control?

Risk management involves clarifying actionable areas and then deploying actions specific for each area. Controllable factors for risk management fall into the following three categories (see Figure 2):

- Product definition and design, including requirements and architecture.
- Development processes, including systems, processes, and methods.
- Constraint management, including resource allocation and commitments.

Each category has an almost limitless array of best practices. However, the following are some important practices to keep in mind during risk management.

Assess and Improve Product Definition and Design

More than any single area, proper requirements are crucial to predictable success. Importantly, a higher standard for requirements is needed to manage project expectations and risk than just to manage process compliance and liability. Review development processes for their value to the entire company, not just for their adequacy to fulfill regulatory and legal needs.

The following are a few of the characteristics typical of projects having good requirements:

- Requirements are in place prior to significant development and are reviewed and updated in parallel with ongoing development activities.
- Requirements are organized in groupings to guide development across and within technical disciplines and project subteams.
- Requirements and the resulting implementations are aligned to facilitate understanding and checking.
- Requirements use unambiguous language.
- Requirements are both complete and nonconflicting; there are neither overlaps nor gaps.

As requirements are developed, risk is reduced by well-structured investments in development, which should lead toward a specific plan to reduce risks. Each design activity should address the question, "What do we need to do to reduce the risk of surprises?" Companies sometimes forget that invention does not stop until all development uncertainty is removed, which often takes longer than the proof of concept activity.

Development activities must also include a design phase that focuses on a capable architecture. It should be structured to ensure robustness and have goals of high levels of team performance and

product implementation predictability.

Finally, the development team should identify specific testing activities—beyond those required for design control—and develop testing plans that overlap development plans. Such plans are a concurrent approach for testing. Ideally, these plans would also apply to the transfer to manufacturing and into the hands of customers.

Refine and Align Processes

Strictly put, a development process cannot ensure success any more than an expensive car will make you a better driver. Yet organizations sometimes spend too much time on having a process rather than on using processes effectively, with smart people making strong judgments within the process framework. More overall attention should be shifted to predictability and effectiveness, where alignment of processes to other factors is essential.

For example, software development teams have mastered the art of configuration management. They use processes and tools to ensure reproducible product configurations and, therefore, ensure the team's ability to master change. All development organizations should learn how to apply this rigor to all parts of their design processes, to ensure responsible configuration management and prototype revision control.

Some companies test down-revision configurations without proper control of the implications. Although this is a violation of QSR requirements, it is also—and possibly even more painfully—a recipe for unmet expectations. With that testing process, the team may draw incomplete or invalid conclusions.

In one specific case, an error was traced to a junior staff person who had not been trained to be careful about using consistent development prototypes. The project's success was at risk because invalid data were being collected during testing. The lack of a development-phase hardware configuration management process allowed invalid data to be collected. Most organizations have a formal engineering change order system for product configuration after it is released to manufacturing and the field. A similar management process was needed in this case; each development prototype should have been identified to its revision level when it was changed and while being tested. Configuration management is just one example of many development processes that lead to success when tuned for risk management, not just regulatory compliance.

Practice Constraint Management

Project teams need to be structured to support success. Eliyahu Goldratt defines the "theory of constraints" as a tool leading to awareness and change in the impediments to reaching a goal.⁴ Changes in constraints lead to changes in the ability to perform. Judiciously removing constraints improves capacity.

Constraints include factors such as internal resource allocation and commitments to external entities. For example, consistent with Goldratt's theory, the sheer number of development prototypes could

be a limiting factor in project success. Although it is more work for staff to keep double the number of prototype systems running for testing, the additional capacity can easily lead to more data more quickly, which can lead to faster conclusions about stability and repeatability.

External constraints, such as commitments to customers, regulatory submittals, etc., also require active management. In extreme cases, project teams have met a goal by ultimately creating something off the mainstream path of development; a "special" prototype used only for data collection or marketing, but not leading to release to manufacturing, for example. Constraints that are properly managed should lead to an ever-decreasing risk profile. Resources, once used, should lead to effective project completion. The decision to detour off a main path of development and risk management should be assessed for intended and unintended consequences.

Constraint management may also be as simple as changing the constraints or the goal. Underresourced or overconstrained projects, or both, bear the risk of unmet expectations. Better to act with integrity early than suffer the consequences of product failure later. This does not mean that properly structured stretch goals are wrong—just that the context must be appropriately defined for the team.

Conclusion

Risk management is critical to developing a successful medical device. However, it must take into account all aspects of risk and must pervade all stages of product development. Applying the following risk management principles during product development enables companies to achieve better products and more-predictable results:

- View risk management as a comprehensive development activity, not just compliance, liability, or FMEA.
- Make the phase-review process richer and deeper; encourage a focus on expectations management beyond the limited phase definitions.
- Change the culture in the organization, broadly and deeply, to include a "what could go wrong?" mind-set with inoculation against risk as a goal.
- Seek to improve product definition and design, development processes, and constraint management.

Product development leaders must act defensively to optimize results for their companies. The view of risk management needs to be expanded beyond the required minimum into the rich array of factors and techniques that lead to predictable results, so as to reap the rewards.

References

1. "Risk Management," in Guidelines for Successful Acquisition and Management of Software Intensive Systems, Version 3.0 [on-line] (Software Technology Support Center, 2000); available from Internet: www.stsc.hill.af.mil/resources/tech_docs/.
2. M McGrath, "The Phase Review Process and Effective Decision

Making," in Setting the PACE in Product Development (Boston: Butterworth-Heinemann, 1996).

3. E Tufte, "ET on Columbia Evidence—Analysis of Key Slide" [on-line] (2003); available from Internet: www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0000Rs&topic_id=1.

4. E Goldratt, Theory of Constraints (Great Barrington, MA: North River Press, 1990).

Bibliography

Design Control Guidance for Medical Device Manufacturers [on-line] (Rockville, MD: FDA, CDRH, 1997); available from Internet: www.fda.gov/cdrh/comp/designgd.html.

Guide to the Project Management Body of Knowledge (Newtown Square, PA: Project Management Institute, 2004).

Taxonomy Based Risk Identification, CMU/SEI-93-TR-6 (Pittsburgh: Software Engineering Institute, 1993).

*Stan Telson is president of **QSPP Group** (Lancaster, PA). He advises senior management and development teams.*

Copyright ©2005 **Medical Device & Diagnostic Industry**

Comments about this article?

Post them in our [Members' Discussion Forums](#).

© Canon Communications LLC 2001