Common Pitfalls During Product Development

Assuming that your ideas have passed the first two hurdles in the product development cycle (See **Six Steps to Great Technology Products**) you're ready to gear up your development team. Once the concepts have been evaluated and the go-ahead for detailed development has been obtained, you should have a schedule in hand and a document that establishes the requirements for your team. However, there are still pitfalls that can trip up the best-laid plans. These include:

Undefined Interfaces. If your technology product requires a team approach for development, the first steps must be to establish the interfaces between groups or team members. This means defining physical interfaces between parts designed by different team members or groups, setting the electrical interface between boards or subsystems, and a software interface to make sure all modules can operate together. Leave this step to the end, and you invite redesign because team members made assumptions that weren't shared by their co-workers.

Micromanaged Development. You have a highly skilled group, and your team takes pride in doing their work. Nothing kills enthusiasm faster than micromanaging the details of a design. After all, you hired the design team to do the development. So why do their job? Occasional guidance and review may be required, but if their idea solves the problem as well as yours, let it go. Concentrate on the end result, and make sure your customer will be happy.

Lack of Reviews. Once overall requirements are defined, use periodic reviews to make mid-course corrections that keep the development on track. Use these reviews to determine the status of the project and also to verify whether the original requirements are met, or if any assumptions have changed. Design reviews will also spot inconsistencies between groups or team members and ensure they can be corrected as early as possible.

The Wrong Resources. Because of the complex nature of products, a lot of development teams consist strictly of members with technical backgrounds. However, other specialties, including non-technical groups, must be included. These include marketing, manufacturing, quality, purchasing and customer support. Even if these groups are part of the development team, their tendency is to stay away because of the technical nature of the work. Make sure they're invited. These members can provide inputs with a different perspective, and their absence can cause significant problems if they must be addressed later.

Cutting The Design Cycle Short. Towards the end of the development cycle, it is tempting to move resources to the next project. While some overlap between projects is inevitable, it must be clear that members of a development team are responsible for completing their projects. Removing and replacing members before completion results in schedule delays since others must pick up the slack, and can result in missed tasks because the new members lack familiarity with the project.

Leaving a mess for others to clean up. Once developed, the product may be turned over to production, and the developer moves on to a different task. Make the developer responsible for fixing production problems. Even though this approach may be more expensive in the short run, the benefits are well worth it. Developers generally aren't fond of fixing problems in production, and the tendency is to blame production. But once developers see and learn from the problems their designs caused in production, a repeat of the same errors is avoided. If another person has to clean up the mess, the developers never learn to improve their skills, and the same production problems occur over and over.

Avoid these pitfalls and your development cycle will flow more smoothly. You'll end up with products that work better, and a staff better suited to solving the right problems.