Designs change. Deal with it!

Changing a design in the middle of a project can be costly. But not changing it can cost a lot more.

Originally published in Machine Design *Volume 83, Issue 5, March 17, 2011, pp 38-41*

Authors: Preston G. Smith, Principal, New Product Dynamics, preston@newproductdynamics.com and John S. Farnbach, Principal, Silver Streak Partners, LLC, john@silverstreakpartners.com
Thanks to Stephen J. Mraz , Penton Media Inc. for editing assistance.

Key points:

- Change is not the exception; it's the rule.
- Software developers handle change using agile development. Other engineers can copy their methods.
- Anticipating change is cheaper than ignoring it.

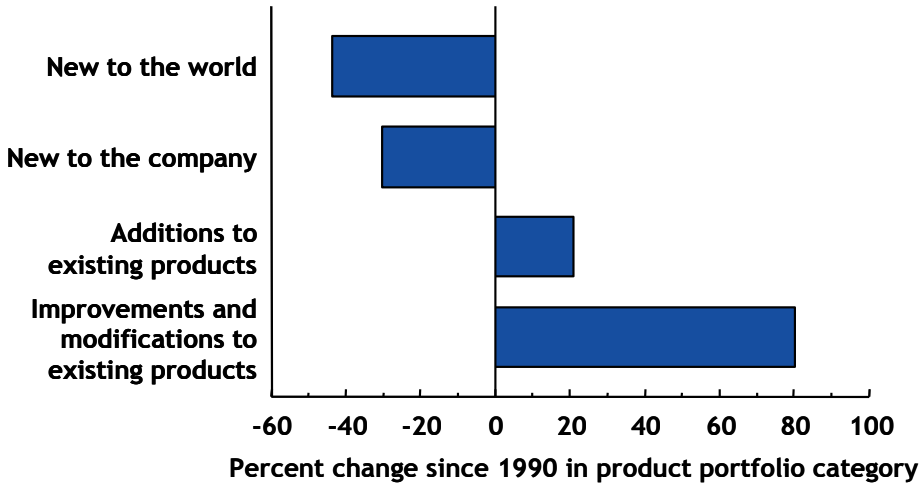
CEOs consistently name innovation as one of their companies' top priorities. But curiously, new-product innovation has been steadily declining over the past two decades, as the figure below indicates.

Management probably wants more innovation but gets less of it because innovation means changes. These changes often crop up in the middle of design projects as engineers learn more about the application and technologies they are using. Changes typically lead to a dilemma: If engineers truly innovate with new products, they should expect change as the project proceeds, but experience tells them that changes lead to expensive rework, blown budgets, and schedule overruns. It's time they and their managers discover that midproject changes need not be expensive, and you can innovate without paying a high price.

Change happens

Midproject change is actually more common than product engineers and managers like to admit. While interviewing product-development leaders, we asked them to identify specific examples of change during a development project. No one had any difficulty in recalling some. This confirms data from Donald Reinertsen, author of *The Principles of Product Development Flow*. He analyzed data from over a thousand design projects and not one had product requirements that remained stable over the course of the project.

This prevalence of change doesn't fit with how engineers carry out design projects. Most use a "plan-your-work, work-your-plan" approach. This mind-set may be driven by a corporate focus on process management to eliminate waste and reduce cost or a by a desire to replicate in R&D labs the advances won in the factory through lean manufacturing. For example, Marvin Patterson, author of *Leading Product Innovation*, says that without thorough planning, "This inevitably leads to scrapped engineering efforts, rework, and schedule delays." The roots of this plan-well, avoid-change approach are deeply embedded in management thinking and corporate culture. An analysis of 13 marketing textbooks backs this up by revealing that product developers are trained to plan thoroughly before they act.



Since 1990, there has been a large shift in companies' product-development portfolios. They've gone from being more innovative to being more mundane over a broad range of industries. (Figure reprinted by permission John Wiley & Sons, Inc., from *Flexible Product Development: Building Agility for Changing Markets* by Preston G. Smith)

The case of the bicycle-hub design project

Narrow bike hub flanges (left) provide enough torsional rigidity for riders, while wide flanges (right) require more material, thus they add weight and cost.

A company designs and manufactures mechanical bicycle components, and its engineering team is embarking on a project to design a new hub for a spoked wheel. The critical engineering work is in the bearings, seals, and a quick-release mechanism. But the hub flanges present a controversial style issue.

Spoked bike hubs can have narrow or wide flanges. Narrow flanges use less material and, thus, are lighter and have lower manufacturing cost. However, many bike aficionados believe wide flanges improve the hub's torsional stiffness. The design team discusses this but also knows narrow flanges are structurally adequate. Engineers on the team favor the narrow format for its technical merit, although marketing prefers the wide flange for its customer appeal.

The engineers are anxious to proceed, so they settle on narrow flanges. The table below summarizes their planned budget and schedule.

Project	Project expense	Schedule	Cost per hub
Hub development (narrow flange)	$100,000	3 months	$7

Two months into the project, marketing is working with distributors on the product launch and discovers they strongly prefer wide flanges and insist the narrow-flange hub will not be competitive. So the project must change, and it will be costly. (This is exactly why designers fear change.) The project now looks like this:

Project	Project expense	Schedule	Cost per hub
Sunk project costs	$70,000	2 months	none
Redesign with Wide flange	$100,000	3 months	$10
Project outcome	$170,000	5 months	$10
Variance from plan	$70,000	2 months	$3

Normally, this would be the end of this sad but all-too-common story. But there is an alternative. The team could have realized at the outset that the hub design was an item of uncertainty. Then the next step would've been to explore the issue in an initial iteration. Specifically, they could have experimented by building mock-ups of the two styles or simply procuring competitive samples of each, and showing them to some customers and distributors to gauge their reactions. This early experiment would have resolved the uncertainty before the company had sunk significant amounts of time and cash into the project. With this approach, their project now looks like this:

Project	Project expense	Schedule	Cost per hub
Prototype both hubs	$5,000	none	0
Develop chosen hub	$100,000	3 months	$7 to $10
Variance from original plan	$5,000	0 months	$3 maximum

The new result differs significantly from that of the original plan. Although the flexible plan cost $5,000 more than the original, it saved $70,000 of unexpected cost overruns and two months of schedule slip. This is typical of flexible product development: The price for flexibility ($5,000) was far less than the cost of sticking with the wrong design. If the choice of flange had been more certain, the $5,000 for flexibility would not have been justified. However, the decision wasn't certain, so it was well worth the extra cost to mitigate the chance of expensive surprises later in the project.

In short, midstream change is common and natural in innovation projects, but engineers and their managers have learned that such changes often have costly consequences. So they plan carefully and stick to the plan to avoid these consequences. Unfortunately, this path does not lead to successful innovation. But there is a middle path: Anticipate changes so you are ready for them when they happen. This makes changes less costly and fosters an environment where innovation can flourish.

A cue from software developers

Software developers have pioneered new methods of dealing with midstream change. Formerly, they used a so-called waterfall method, which was based on heavy up-front planning and strictly sticking to the plan to avoid rework. As the software world became more turbulent, however, this approach began to conflict with reality. Thus, over the past decade, software developers have embraced agile software development (see box), which uses an iterative approach to deal with change by relying on a make-a-little, try-a-little strategy. By continually testing design features on real customers, they avoid many of the big surprises that typically crop up toward the end of projects.

Unfortunately, agile development can only be applied directly to software because it depends on some unique characteristics of software, such as object technologies and the ability to automate testing. But agile provides a wonderful list of tips for other engineers on how to accommodate change:

- Identify uncertainties early and work to resolve them.
- Adopt a more iterative style.
- Try things out—experiment a little.
- Defer decisions in uncertain areas.
- Create and maintain options.
- Use product architecture to "fence off" areas likely to change.

Agile software development

Agile development typically works against a backlog or list of product features gathered from the marketplace. This backlog, rather than a detailed plan, comprises the product requirements. Instead of tackling the entire backlog at once, the agile approach uses short iterations (also dubbed sprints in the popular methodology called Scrum). Iterations typically last two weeks, a duration that remains constant throughout the project. Detailed plans are not formulated until the beginning of each iteration as marketing prioritizes the backlog. Then the development team, knowing how much they can accomplish in an iteration, draws a line that far down on the prioritized backlog, and this becomes their plan for that iteration. At the end of each iteration, the team demonstrates to marketing the working software they've completed, and often to customers as well, to ensure the code meets customer needs. Then the team plans and proceeds with the next iteration.

Three fallacies about change

Designers' apparent rigidity about careful planning and sticking to the plan seems to stem from fears of what might happen if they stray from the plan.

Here are three fallacies that lead engineers to fear change.

Fallacy #1: Flexibility is expensive.

Engineers and their managers believe midproject change is expensive because they have seen changes lead to project delays and budget overruns. They attribute those delays and overruns to the cost of being flexible and open to change. But you can reduce the cost of being flexible by preparing for changes rather than treating them as surprises and dealing with them as they arise. For example, the cost of flexibility in the bike-hub example (see The Case of the Bicycle-hub Design Project) was only $5,000 for prototypes. The $70,000 squandered on the wrong design was actually cost of inflexibility, of ignoring uncertainty and failing to seek better information.

Insurance provides a useful analogy. When it comes to fire insurance on your house, you have two options. One is to be careful with fire, assume the odds of a fire are negligible, and be willing to assume the full cost of replacing your house if it burns down. The other option is to pay an insurance premium up front each year. Then, if your house burns down, your cost is much lower. Similarly, if you take the time and pay for exploring uncertainties up front during a design project, the project won't come off the rails when it's bushwhacked by an "unforeseen" change, which leads to major schedule slips and budget overruns.

Fallacy #2: Flexibility makes projects unpredictable.

Product designers seem to believe that flexibility is a wild card that injects unpredictably into their projects. Following this logic, they try to improve predictability by planning out all the details and then slavishly following the plan. This works well if the market and technology are stable and little is apt to change. But if change is likely, the plan becomes brittle, and trying to follow it leads directly to unpredictability.

The bicycle-hub example shows that maintaining options, a form of flexibility, actually improves project predictability. While the original plan led to a $70,000 budget overrun and a schedule slip of two months, the flexible plan caused no schedule slip and a small $5,000 bump to the baseline budget.

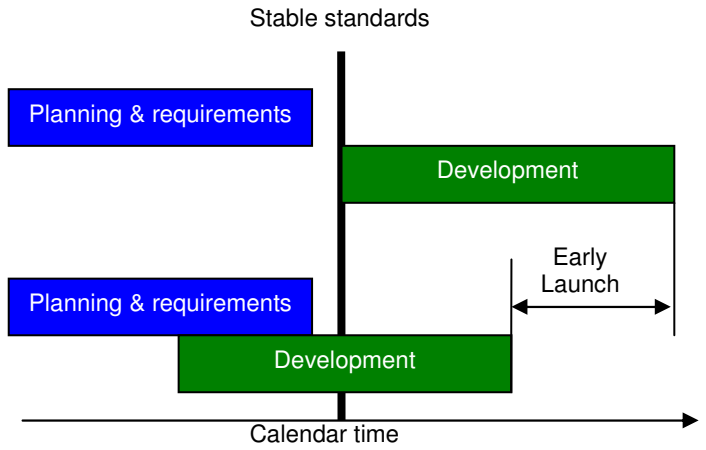
Think of it this way. If you have a single design path and follow it rigidly, you will likely be surprised greatly by any deviation. You may be faced with either starting

over or turning out a design that doesn't satisfy the market. With flexibility, you open alternative intermediate paths. Then, if the original one fails, you have palatable backups. This broader choice of options improves predictability rather than decreasing it.

Fallacy #3: Flexibility delays market launch.

Another common belief is that starting a project before all details are pinned down leads to wasted effort when the design changes. Here's a real-world example of this fallacy at work.

A design team was developing a communication device in an industry where compatibility standards were still evolving. After wasting time chasing standards that proved to be moving targets, management put the project on hold. However, a competitor beat the company to market by following the evolution of the standards more wisely. The key, in this case, was to use product architecture to isolate, or "fence off," the small portion of the design affected by the standards and proceed with the rest. Then, when standards were finalized, the competitor already had most of the design completed and could finish it in time for a quick market launch.



A company can shorten a product's time to market by focusing on its stable features if design standards on other features are not firm yet.

In short, flux in industry standards delayed the competitor's project but not as thoroughly as it did the inflexible project. Furthermore, the competitor did not waste design resources by continually revising the part of the design subject to evolving standards.

The technique used by the competitor was to identify the gating event (stable standards in this case) and work around it by proceeding with activities that were certain. In this case, the critical time-to-market clock started when the standards were finalized, not when the team started its development.

Focusing on stable features before design standards or other features are firm can accelerate product launch.

What you can do

The first step, of course, is to assume something will change. Then take these steps to make projects more change-friendly:

- Move beyond fixed schedules and frozen requirements. Early on, identify areas likely to change and explore options.
- Keep in touch with customers. They are always receiving new information and might change their minds regarding your design.
- If a project relies on emerging technology, keep up to date with changes in that field.
- Encourage, don't just tolerate, experimentation to explore alternatives both up front and during the project.
- Let development teams make more decisions themselves. They have the freshest information, and waiting for gate meetings is simply too slow.
- Align performance metrics with the chaotic world. Reward people for the quality of their midproject decisions, not for merely sticking to a plan.

Postproject assessments are also critical for reinforcing the idea that changes can be good and don't always stem from poor planning. For example, to reduce reliance on a frozen plan, assess whether a surprise was due to poor planning or to new information that arose after the planning phase. And outline what could have been done to obtain this information before it became a surprise. Finally, assess the opportunity cost of changes not made. Those who steadfastly follow their plans ignore what it cost to avoid a change. The cost of a lost market can far exceed the cost of making a change.

Resources

New Product Dynamics, www.newproductdynamics.com

Silver Streak Partners, silverstreakpartners.com/FPD.htm

Flexible development Web site, flexibledevelopment.com

Preston G. Smith, Flexible Product Development, Jossey Bass, 2007

Preston G. Smith and John S. Farnbach, "Avoid Costly 11th-Hour Project Dilemmas by Preparing for Change" Visions XXIV No. 4, December 2010.