Tools Available to Reduce Risk During Drug Development

Mark Tunkel, Insight Product Development

The market for combination products continues to grow ever increasingly competitive, especially as innovative therapies and high-volume generic alternatives crowd the marketplace. To achieve success within this new paradigm, combination product manufacturers must get their products to the market quickly and efficiently. At the same time, manufacturers also face substantial lead time required to meet U.S. FDA approval, resulting in intense pressure.

Yet companies that shortcut their design and development process in an attempt to accelerate speed to market, run the very distinct risk of discovering core device functionality issues at a later stage. This not only adds months to the development cycle but can also cause substantial cost overruns. Many drug developers erroneously assume that core-to-function design issues can be mitigated as part of the commercialization process, resulting in a fast-tracked development process that leads to poor outcomes. So how can developers avoid falling into this trap? By leveraging appropriate tools at every step of the development process—from components and subsystems to fully integrated systems design—developers can uncover issues well before commercialization.

Factoring Risk Early on in the Process

As the saying goes, prevention is cheaper than a cure. For this reason, successful companies consider risk mitigation when setting the functional requirements of a product, holistically integrating it into the process early on. This is accomplished by focusing on the most critical, high-risk elements of a potential product and ensuring full operability prior to commercialization. Iterative development—a cyclical process of prototyping, testing and analyzing every technical and user experience element—is a developer's best tool in this area. This approach reveals core issues early on in development, allowing for the time needed to refine product design from concept to initial tooling and production.

By its very nature, iterative development is a nonlinear process that leverages both *theoretical* and *physical* models to vet concepts in tandem with one another throughout the entire design and development cycle. Successful developers leverage each method to continually refine their assumptions instead of relying on one tool to the detriment of the other.

Several tools and methods exist within both models to help companies effectively push risk upstream. It is important to note, however, that no single tool on its own represents a universal solution for every given scenario, When used appropriately for the specific functional requirements of a development program, these tools can effectively confine core functionality issues to the design phase of development.

Just as none of these tools on their own represent a magic bullet to mitigating risk, neither should theoretical analysis or physical prototyping be undertaken independent of each other.

Theory Comes Before Application

After defining functional requirements for a product, the next step toward preventing late-stage core functionality issues involves leveraging theoretical analysis tools. These enable the development team to review assumptions about the initial product concept as well as consider the physical design of the product.

Theoretical Models

In the earliest stage of product development, theoretical models provide developers with a means to prove their design assumptions using analytical sources. These models leverage software simulation, companion engineering math and heuristic usability product design modeling. Multiple optimized iterations ➤

Photo courtesy of Insight Product Development

A CNC machine can be a valuable tool for replicating components during development

at the theoretical level can confirm or disprove the expected product performance results of higher fidelity prototypes in later stage development.

Successful developers leverage each method to continually refine

User Experience Prototyping

User experience prototyping is another tool deployed early in development. This tool ensures devices meet users' physical, cognitive and emotional needs, and includes workflow projection using storyboards, ergonomic models, and mock device instructions at low fidelity. User interviews and in-field observation vets outputs from this prototyping.

Breadboards

Utilized in tandem with user experience prototyping, breadboards are the earliest physical iteration of product development and design. At the lowest level of design fidelity, using off-the-shelf products or improvised components from string to Legos, breadboards demonstrate high-level functional principles of device components and subsystems, and inform higher fidelity CAD renderings that will ultimately be realized in physical form.

Moving from Theory to Practice

After vetting theoretical design iterations, a number of physical prototyping tools offer developers a means to confirm core product functionality during the design phase of development. Featuring a wide range of fidelity levels, these tools enable developers to work concurrently with theoretical tools.

3-D printing

The fastest growing tool in the developer's prototyping toolbox is 3-D printing. Advancements in material properties have been substantial, resulting in 3-D prototypes that not only represent the intended geometry, but also the intended engineering function of a product, allowing for functional testing. Given the wide range of fidelity available using this tool, prototypes at this stage can also prove user workflow assumptions through observation of in-context usage.

CNC (Computer Numerical Control) Machining

A highly automated tool capable of replicating components at the highest level of fidelity, CNC machine prototyping is a valuable tool in any developer's risk mitigation arsenal. At this stage, proving key functionality through an integrated proof of concept is central to confidently defining design inputs for the final commercialization phase—ultimately producing a final product without core functionality issues. Not only do prototypes at this level of resolution enable the evaluation of core device performance (mechanical, technical and functional) through models that closely mirror full scale production outputs, they also enable manufacturing variability range testing to ensure that any postproduction device performance issues can be confined to issues within the manufacturing process itself—without worrying whether the design will work or not.

So how does the value of this approach translate to a delivery device development program exactly? Well, consider the following example: a hypothetical company enters the market with a novel device; the development team defines the preliminary functional requirements of the product. In an attempt to accelerate the product's entry to the market, the team opts to move quickly from an early design concept that has not undergone the rigors of iterative prototyping using these tools, and proceeds straight to commercialization. Not only is there the distinct chance the company will emerge with substantial volumes of product that does not work reliably, but worse yet, no one can initially attribute the core functionality issues to either product design or the product manufacturing process. Now, the company is in the unfortunate position of incurring significant cost overruns and additional time to rectify the issue. The required remediation to uncover core underlying issues also adds to the product launch timeline.

By addressing the entire range of development considerations with an appropriate combination of tools, companies can avert costly and time-consuming mistakes as quickly as possible, and emerge from the process with devices poised to effectively compete in the market—on time, and on budget.

About the Author

Mark Tunkel is a partner and director of business development at Insight Product Development. His company will be exhibiting at this year's *Universe of Prefilled Syringes and Injection Devices conference.*