McKinsey&Company
Operations Extranet

June 2016

The feature revolution

Feature-based design is transforming the way companies design their portfolios, evolve their offerings and delight their customers. Getting it right requires them to transform their processes as well as their products.

by Ondrej Burkacky, Eric Hannon, Sander Smits, and Florian Weig

Products are no longer static but dynamic: they improve their value to customers over time, not only in the quality of their manufacturing, but also in the functionality they provide. Take three examples: Tesla has added self-driving features to cars that customers already own. Apple has added features like a flashlight, voice recognition and music streaming to customer smartphones over time. French audio start-up Devialet has used its Phantom loudspeakers as a platform to introduce additional music services.

What do these examples have in common? The companies involved have a clear perspective that product development does not end when production starts, but continues to create new customer value throughout the product lifecycle. They rely on product architectures that allow for seamless integration of future features, from a hardware as well as from a software perspective. They use a clear understanding of feature roadmaps, based on customer needs and technology cycles. Features are disaggregated from the product development process, allowing their development to progress at different speeds.

This *feature-based* approach to product development opens up multiple sources of additional value:

- Customers appreciate, and increasingly expect, a product that remains “fresh” in an environment that is evolving rapidly. Products that meet these expectations see higher customer loyalty.
- Companies can capture additional margin through features released after the initial purchase. Like many aftersales businesses, this is associated with high margins and low cost of sales.
- Engineers and suppliers benefit, as basing the product around features and not specific components, allows for flexibility in product delivery and in the supply chain.
- Society and the environment profit from product platforms that improve over time and remain with customers for longer, a central proposition in a *circular economy.*[1]

Feature-based development has been dominant in the software-development process for many years. Now feature-driven design thinking is rapidly migrating into adjacent industries like telecoms equipment and semiconductors. As the approach emerges in other industrial goods and in consumer products, from cars to washing machines (see box: *Feature-based design moves into hardware*), we

[1] See http://www.mckinsey.com/client_service/sustainability/latest_thinking/growth_within_-_a_circular_economy_vision_for_a_competitive_europe

believe companies in these industries will be the next to move to feature-based development organizations.

While the end state is certainly attractive, the challenges for traditional product companies are real. Moving to the continuous development of features—disaggregated from product cycles—breaks with traditional R&D project management, incentives, structures, processes and systems, all of which are oriented toward a handover to supply chain and production, with evaluation shortly after launch.

Building on existing product development capabilities

Fortunately, companies seeking to create a feature-based product development organization do not need to start from scratch. Recent years have seen the growth of modular design approaches, in which discrete modules are designed to fulfill a specific purpose, have well-defined interfaces, and are assembled together in different configurations to build products for a multitude of customer applications.

We see feature-based development as the next iteration of flexible, modular design. Companies will not abandon the idea of products adapted to specific customer needs or the structures that enable modular design. Instead, feature-based development is a design methodology that allows companies to capture the additional value described above.

Most traditional R&D organizations do not need to look far to find many of the building blocks they need to assemble a feature-based approach to product development. Feature-based development has long been a mainstay of software engineering and is embedded in most applications of agile development. Many internet start-ups, as well as established software players, plan their development around new feature release cycles. The development of the feature is independent of any specific product or product module. Instead, the requirements that the developer must deliver are feature requirements, which may affect multiple products and several software modules. (See sidebar, “Feature-based design moves into hardware industry.”)

What will success look like?

How will feature-based development affect your R&D organization? Let’s take a look at some of the dimensions that make up product development excellence.

Steering and governance

Most R&D organizations have a stage-gate process that is set up for each product they develop. The maturity of the product, disaggregated into components and subcomponents, is assessed and tracked in these stage gates. Moving to a feature-based development approach would mean that the steering and governance process will track the progress of the features incorporated into the product, in addition to the separate components.

Core processes

As features will be developed on their own timeline, separate from the product’s development timeline, a project plan for a particular product should anticipate a different timing plan for specific features. Subfeature development will be tied to the timing of the feature to which they belong, rather than to the related components or subassemblies.

Most durability and performance testing today is done on a component level. There will be a transition to feature level testing. For example, power management is one of the features used in mobile phones. Testing this feature would include testing of antennae, the user interface and the battery itself, each being optimized for minimum power consumption. Testing in this way helps companies to focus solely on the delivery of features that matter to customers.

Features will also be developed continuously, well beyond the first sale of the product. Some features will continue to mature after the consumer starts using the product. Customers have become accustomed to their computers and mobile phones updating themselves seamlessly in the background. Now Tesla uses the same approach to provide updates to its customers, delivered over the air to the car.

Team work

Feature-based development will require intense and frequent interactions between the owners of features and the owners of the various physical modules providing the functionality to deliver the features. The interactions will become more complex, but with good planning and scheduling, these interactions help to define a product's value proposition, and the features required to deliver it, early in the development phase, thereby reducing rework.

IT and infrastructure

Most large R&D organizations use product data management (PDM) and product lifecycle management (PLM) software tools for storing and exchanging CAD data among R&D employees globally. Future PLM philosophies need to allow organizations to create and manage features, not just parts. The configuration of smart products is not fully reflected in a parts list or BoM.

With the core process of feature development extending from engineering through manufacturing to the end customer, the future PDM infrastructure should be able to capture and connect information beyond R&D to the whole life cycle of the product. These tools need to include more usage information, for example, as products are equipped to transmit real-time information.

Feature roadmaps

R&D organizations tend to track the latest technologies in their industries closely. A feature roadmap is a plan that matches short-term and long-term goals with solutions to help meet specific customer-experience requirements. This helps to prioritize R&D investment in an increasingly competitive environment. A feature roadmap will consist of a prioritized list of the features the organization has chosen to work on, with a timeline to achieve various levels of maturity, beyond the sale of the product to the customer. These roadmaps should also consider external partnerships and collaborations with potential contributors including suppliers, leading educational institutions, and technology startups.

For example, a mobile phone OEM developed its feature roadmap by first laying out customer expectations over time. Those expectations included increased connectivity, seamless call handovers, video conferencing, and the ability to connect to white goods at home, to name but a few. As a next step, the company linked those expectations to specific features that its R&D organization can develop and provide as an upgrade to software or to certain parts of the hardware.

Organization

Of course, engineers and designers will still be needed to build the product, but the way they are organized will change. A new role of “feature leader” will be required. This role will oversee the integration of various features that will ultimately result in the end product. This is a powerful role, since the feature leader will be responsible for balancing customer and technical requirements, by delivering features rather than specific components or assemblies.

Skills ecosystem

Features can involve multiple components within a product. R&D organizations will need more people who have the understanding of how components or modules interact with each other, with the ability to understand how a feature can be built to fulfill a certain customer desire.

With an increasing number of smart products with the ability to communicate, send and receive updates, and capture usage data, there is inevitably a need for software engineers capable of coding and analyzing the volumes of data that are generated during use. For example, autonomous driving requires systems that can combine and interpret the input from the sensors and cameras, then control the mechanical components in the vehicle to deliver an appropriate response.

First steps

Implementing a feature-based development process and organization means fundamentally changing today’s way of working. Making and sustaining this change will require a well thought-out implementation process.

An essential starting point in this process is a detailed understanding of the features needed to deliver on customer needs up to two product generations in the future. By definition, a feature will be developed across product platforms; hence, in-depth analysis and planning on where and when the feature affects the various product releases is required. Laying out the feature roll-out on a timeline and integrating it with the product roadmap gives an overall picture of the product portfolio.

Second, companies need to understand the implications of different features on their product architecture. In some cases, for example, two features may need the same module. Laying out these overlaps and defining the interfaces between the modules will influence the design of the product’s future architecture.

The third step is the design of an organizational structure that can support the changes in the development process and product architecture. Introducing the role of “feature leader” within the product integration team will enable the transition from the existing module-based organization to a feature-driven organization.

Faster, stronger, better

While the transition to feature-based development may be challenging, the payoff is significant: products that are developed more efficiently, and which are more likely to delight customers.

One high-tech player recently transformed its product development activities to a feature-based approach. Prior to the change, the company was struggling with spiraling product complexity. With several major platforms under development in parallel, customers asking for fundamental new features every year, over 100 product derivatives to be supported at any point in time, and a software versioning explosion, the organization's existing R&D process was not keeping up.

The introduction of feature-based development required several major changes to the company's development processes. All user requirements are now mapped to specific product features, ensuring that development focuses on what customers really value in the end product. A feature-driven architecture framework has been established, allowing individual features to be separated from the main product components and making it easy to integrate or remove features as required. A new "feature lead" role in the organization oversees each feature across the product lifecycle and signs off on relevant changes to individual components. Feature-development lines have been established, where individual features are developed and tested against user requirements, mainline products and in combination with other features. Some new features are developed as separate projects and then made available to various products in different versions.

The change process took almost three years, but the benefits to the organization have been profound. Product development efficiency has increased, with development cycle times reduced by roughly 40 percent, and productivity increased by a third. Agility has improved, with the fast compilation of features allowing the company to double the number of product derivatives it offers. Finally, build quality has risen by 50 percent, as the separation of features from product lines reduces complexity and allows the earlier detection of defects.

About the authors: Ondrej Burkacky is a partner in the Munich office, where Florian Weig is a senior partner. Eric Hannon is a partner in the Frankfurt office, and Sander Smits is a partner in the Amsterdam office.

The authors wish to acknowledge Ruth Heuss and Swarna Ramanathan for their contributions to the development of this article.

Sidebar: Feature-based design moves into hardware industry

In sectors like the telecommunications and semiconductor industries, we are already seeing the application of feature-based development to the full scope of hardware and software systems. For example, in networking equipment, features are driven independently of the underlying hardware development, disaggregating the long hardware product cycles from the continuous improvement of the feature set. A simple feature such as “SMS Delivery” can be continually updated, though it spans different hardware and software modules, because its development is not tied to any specific hardware release.

In the automotive sector, the AUTOSAR specification for software modules used in vehicle electronic control units shows how feature-driven design thinking can work in mature, hardware-focused product development environments. AUTOSAR provides a specification, adopted by most major automotive OEMs and suppliers, which separates basic hardware-level software modules from higher-level feature modules through an application interface. Theoretically, this allows a feature to be placed in any electronic control unit, as long as the underlying hardware can support it.

Using this approach, developers are now focused on designing the software for a vehicle feature, and are detached from the constraints of the design, development, and release cycles of the underlying hardware. Developers are free to continuously update, refine, and improve their feature, and thanks to the disaggregation provided by the application layer, their feature development can be largely independent of the typical 4 to 5 year vehicle development cycle.

Some suppliers are now extending the software features of AUTOSAR to include the design of the supporting underlying hardware. This means that these suppliers can plug a feature into any hardware module that an OEM desires. For example, the feature windshield wiper control along with its input and output circuitry could be placed in either a Front Control Module or a Body Control module—of any microprocessor generation—based on the OEM requirements.

The concept of feature-based development is equally applicable to larger and more comprehensive features. In the automotive world, the advent of autonomous driving technology is a far broader feature, touching many more vehicle systems than windshield wiper control. A feature like this will involve a wide array of vehicle modules—from cameras and LIDAR, to brake systems and electronic steering—that are often developed largely independently.