

# Building Support for Use-Based Design into Hardware Products

Tim Misner | Oracle | timothymisner@aya.yale.edu

*Editor's Note:*

*Use-based design is a new model of product development. It is a process of measuring user behavior and applying the resulting data to improve the next version of a product—creating a feedback loop between user and designer. (We might refer to the process as data-driven design, but that term has another meaning among software developers.)*

*Basing design decisions on customer behavior has roots in mail-order catalogs of the late 19th century, such as Montgomery Ward and Sears, Roebuck. Use-based design grew along with direct mail in the mid-20th century. More recently, the Web created opportunities for collecting data on user behavior. Google is famous for driving decisions with use data. But few designers have experience in basing decisions on data, and many are still uncomfortable with the practice.*

*Now the use-based design model is being applied to hardware. For consumer electronics and office products especially, connecting to networks has become almost standard, creating new opportunities to collect information about user behavior (and about device behavior). In a sense, networked hardware products are very much like Web services—and in many cases hardware products are being integrated with Web services. For a growing class of “monitoring” hardware and services (e.g., network management, security, sports, and health products), collecting user information may even be the primary mission.*

*Taking advantage of these capabilities requires a new model of design as well as a plan to integrate them into the first version of a product.*

—Hugh Dubberly

## All Products want to be Websites

Increasingly, hardware products (especially consumer electronics) include computers, sensors, and connections to the Internet. These capabilities enable changes in what products “know,” how they are used, and how we develop them. They are becoming more like websites. This simple fact became apparent to me during my work at Dash Navigation as their design director.

There, I worked on the Dash Express, a personal navigational device (PND) that included GSM networking and exploited the following networked-services principles.

Networked services differ from traditional hardware products in at least four important ways:

- 1 Networked services can recognize their users and respond uniquely.
- 2 Networked services collect information as a natural part of operation.
- 3 Networked services change continuously, largely based on user actions that feed product improvements.
- 4 Networked services may also enable field upgrades of software, continually improving the product.

At first, however, I didn't understand how designing an integrated system of hardware, software, and network applications requires a new way of thinking about a product and its development. On one hand, we design end-user applications. On the other, we design platforms for both services and user feedback on these services.

This allows us to adopt the essayist Clay Shirky's theme of building “systems where having good participants produces better results than having good planners.” [1] Internal discussions change from “what feature or quality status do we think our products need?” to “what data can we collect about our features and quality?”

## Networked-Service Principles

Networked services can recognize their users and respond uniquely.

Networked services collect information as a natural part of their operation.

Networked services change continuously, largely based on user actions that feed product improvements.

Networked services may also enable field upgrades of software, continually improving the product.

## The Product Development Experience

Getting from the standard 18-month hardware development cycle to a rapid cycle of data-driven verification requires that the product team commit itself to building the infrastructure to:

- send updated software to all, or just segments, of its user base
- obtain and analyze data from the users themselves
- provide explicit, periodic methods for the users to submit their opinions from the device.

## Examples in Practice

For Dash, all searches are, by default, tailored to the user's current location [2].

Availability of service depends on user's account status.

The Dash collects the user's current location and speed. It uses this information to build, at an aggregate metropolitan level, a real-time traffic model [3].

When the user sends a destination address from a personal computer to the Dash device, Dash adds it to the user's address book [4].

Google improved the performance of its Picasa download page by using A/B testing of live options with real users [5].

Microsoft asserts, quite startlingly, that "one percent of bugs cause half of all errors," thus enabling much more meaningful bug prioritization [6].

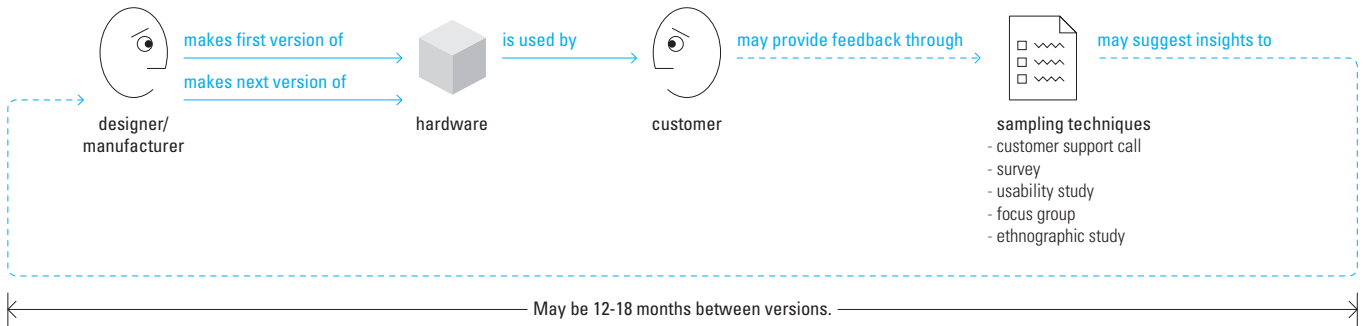
When permitted by its customers, Oracle monitors configuration information of database deployments [7]. This enables faster resolution of customer service requests.

As Apple has released new versions of its iPhone, it has also made software updates available to owners of older models [8].

Thematic features of this magnitude require strong executive and product management sponsorship to succeed. This evolution cannot happen without explicitly specifying these needs as a core product feature. Back-haul data, in particular, benefits from discussions between many departments (operations, support, QA, and development), as well as the user experience (UE) team. In fact, it's quite likely that the other departments will be the initial drivers of the feature. However, UE participation is required to create the kind of infrastructure that allows design iteration based on behavioral data.

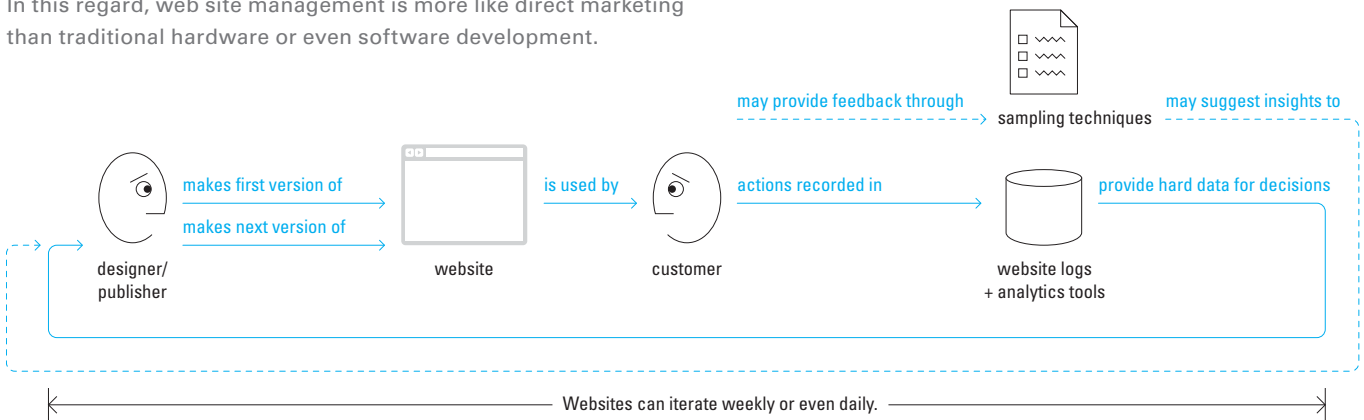
### Obtaining Customer Feedback for Traditional Hardware Products

With traditional hardware products, designers have limited knowledge of customer use patterns. Support calls can provide important data on trouble areas (if properly cataloged), but other information is only available from small samples and observers may bias results. Feedback is incomplete and lags actual use considerably.



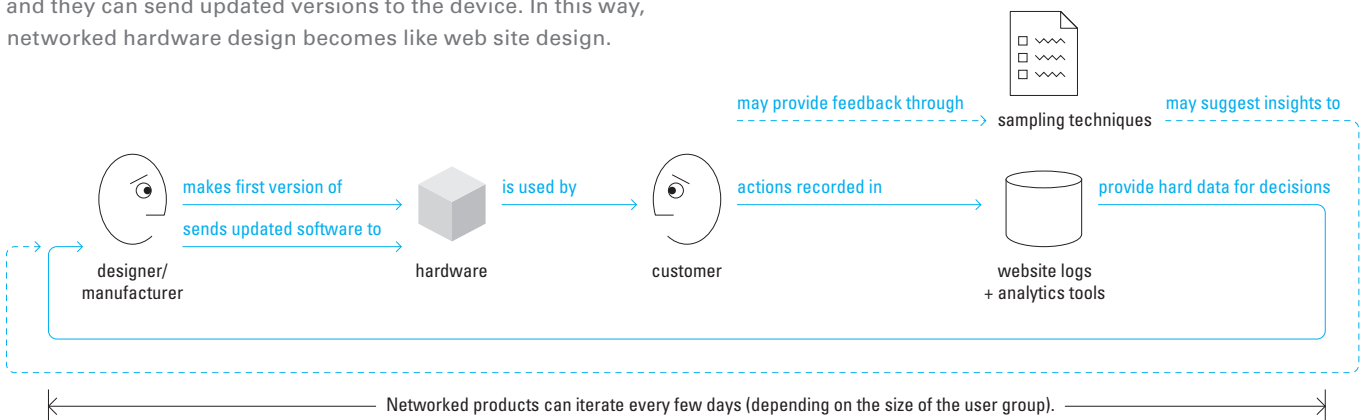
### Obtaining Customer Feedback for Websites

With web sites, designers can have almost complete knowledge of how customers use a service: Which links receive the most traffic? Where in a process do customers drop out? Web sites also allow A/B testing: Which of several examples “performs” better? With realtime feedback, use-based design becomes possible. In this regard, web site management is more like direct marketing than traditional hardware or even software development.



### Obtaining Customer Feedback for Networked Hardware Products

Consumer electronics (and office products) are increasingly connected to the Internet. That means use patterns can be recorded and sent to an online database, giving designers almost complete knowledge of how customers use a device: Which features are used most often? What’s not used at all? With this knowledge, designers can refine software and content—and they can send updated versions to the device. In this way, networked hardware design becomes like web site design.



## Uses of Back-haul Data

In contrast to data submitted by users through emails, phone calls, forum posts, and surveys, back-haul data enables direct observation of user and system behavior. (“Back-haul” refers to moving data from a remote site, e.g., a networked device, to a central site, e.g., an application server.)

One great place to start prioritizing your needs is by brainstorming with a broad range of stakeholders about what questions they’d like to answer. Some of the most obvious will be questions that you can’t reasonably or reliably ask users to answer themselves. With behavioral data, we can collect answers from the total user population, not just those who respond to survey requests. Another good place to start is with the statistics that measure, from a product management perspective, whether the feature was successful against business goals.

Secondarily, questions asked in user surveys can be crosschecked against behavioral data to assess the degree to which users accurately answer survey questions. For example, when asked how many times a month they use a device, the estimate from survey respondents is often larger than the observed degree of device usage. While this is a well-known bias of surveys, unsophisticated product teams often believe that what a user says is the unvarnished truth. They have either forgotten or not learned the principle that says, “Don’t trust users; observe them.” [9]

## Common Networked Functions in Connected Devices

Many connected devices will use the following networking functions:

- Transmission of sensing data from the device back to a server. For Dash, the sensing data is the driver’s current road and speed; for a medical device, the data might be the monitored patient’s heart rate, blood oxygen level, or glucose level.
- Automated distribution of software updates to the device. Dash advertises its auto-update feature as an important advantage over competitors.
- Ability for the manufacturer to configure and diagnose the device. At Dash, the networking-derived functions of the device (real-time traffic, Internet search) are disabled if users let their service plan lapse [10].

- Ability for the user to send data from the Web to the device, thus eliminating more tedious device side entry. For the Dash Express, a well-loved function is entering an address by sending it from the web to the car [11].

All of these connected functions can potentially be exploited by UE for research needs. For example, variants of the software can be sent to target user populations for either their subjective feedback or to generate “A/B” test cases. A population’s configuration settings can be analyzed to determine the most and least popular setting changes.

Most important, the software development organization can rapidly update the product during usability testing and then observe users for their reactions to the device and its new software.

## Many Groups Use Back-haul Data

User experience is hardly the only group motivated to scoop up back-haul data. Back-haul data collection is essential for other functions to fulfill their missions.

The ability for UE to understand, leverage, and champion these other needs will maximize its influence over the form that back-haul infrastructure takes. In particular, UE needs to realize its stake in the design of these infrastructure components when initial implementations are sketched. A common mistake for typical small UE teams is not seeing the potential at early stages.

For example, the operations group will require usage data for authentication and billing. IT needs aggregate usage metrics that trigger alarms when usage levels drop, whether due to instability in its own infrastructure or to service-provider outages. Support needs some form of back-haul data to understand customer issues more efficiently, such as versioning, configuration settings, stack traces [12], and prior usage. QA values back-haul data on common performance metrics, such as “time until first GPS fix” and “time to first network connection” so that it can assess release readiness. QA also values back-haul data to help track metrics of system performance and stability. With this data, the QA team can better evaluate release readiness. Finally, everyone wants some method to “file a problem report” from the device to reduce the time to file, reproduce, and fix bugs.

## Issues Unique to UE Needs

The UE group has needs distinct from the other functional groups. Both QA and operations, by and large, are more concerned with aggregate numbers than an individual's experience. Their numbers may say the product is performing as specified, but they don't indicate if the users are actually happy with that level of performance.

For UE design, it is beneficial to triage use cases into buckets such as "frequent for all," "frequent for some," or "infrequent for all." In order to do this, the data source must maintain a marker of individuality along with the data, so the data analyst can slice and dice the data to discover such relationships.

In addition, the UE group desires the ability to run longitudinal queries, particularly the ability to see that new user features lead to perceptible user benefits. This requires the UE group itself to create and maintain over time a database of high-level user events in a normalized format.

## Infrastructure Needs

In order to exploit the back-haul data, central sampling issues such as who, what, and when (how frequently) need addressing.

Behavioral logging benefits from the ability to understand and filter the results through the lens of various user-grouping mechanisms. For example, stakeholders will question unwelcome results because various outlier groups may have skewed the results.

Invariably, both business and operational groups have an interest in creating user segmentation; this functionality is thus almost certainly available to the UE group. However, the grouping mechanisms will be driven by customer purchasing segments, and this in turn can constrain an experiment's design.

**Problem reports.** Since users have problems, they need a mechanism to explain what those problems are: the problem report. There are two different design approaches:

- Mostly passive, where the problem report is created on behalf of the users. They just need to submit the report.
- Mostly active, where the user initiates the problem report.

Both are, of course, useful. But for devices, the active report is particularly useful, as it is often very difficult to create bug reports independent of a complex environment that only the user fully understands.

The work required is not just the data transmission and recording. In implementing the problem report feature, I recommend budgeting a fair amount of time for implementing tools that facilitate analysis of the reports. Tools that facilitate categorization and visualization of data pay for themselves in short order. Otherwise, time will be spent telling the bug reporter, "I can't reproduce your problem."

**Server-side logging.** The simplest and most cost-effective technique is to execute server-side system logging. Then scripts can be run to collect and analyze the logged data.

The main advantage of this approach is that little planning is required to generate data because there are no changes to the device code. In the simplest case, an IT operator or engineer adds some logging code (using the existing logging framework) to a server-side component. These changes are usually low risk to any release. This modification gets deployed to the main user population at the next server-side software release.

However, back-end logging alone doesn't provide a rich picture of the user experience because many behavioral issues can be resolved only via data from the device side.

In addition, tools construction is required to "extract knowledge from data." Mike Kuniavasky gives an overview of both the benefits and issues with server-side logging [13].

**Device Logging.** Logging from the client side is more complex than logging from the server side, because device logging has to handle the transmission of state back to the mother ship. In addition, the organization has to be willing to pay the bandwidth cost required to send back the data.

The simplest code approach is to just add a logging function to each "user event" (such as a key-press). This is analogous to a website "click-stream." The UI framework may have a very small number of places where this logging code could occur. So the implementation can be both simple and broad. The downside, of course, is that an enormous amount of data is generated and transmitted and most of the data is not germane to the user experience questions at hand.

Fundamentally, this functionality is only useful for debugging purposes when focused on a small number of users.

A more complex approach is to log, for the purposes of a targeted experiment, only the data germane to the researcher's question. This requires custom code to map between various system states and the user state. This code can be more complex because it is integrated into the application logic as opposed to targeted at a base-system level. In addition, you need to remember to turn it off after the experiment is complete.

My experience is that adding this instrumentation is unnecessarily costly when done after the design and initial implementation phase of a project. Don't shoehorn it in during the beta cycle.

## Conclusion

From my inefficient, but valuable, initial experience, a central theme emerges for UE professionals:

Think through the data you want to collect as a fundamental part of sketching and specifying your design. This facilitates a cheaper engineering implementation, but it also allows you to drive design directions from an initial, limited implementation published to a specific user population.

In addition, all functional groups can feel enriched by the quality of user data derived from an Internet-connected device. Customarily, these user research and debugging techniques were available only to Web services. But they are now possible with today's connected, physical devices, provided your product team invests in the key infrastructure items such as:

- User segmentation
- Over-the-air software updates
- User and system state logging
- Data analysis tools for the user and system logs
- User research resources devoted to designing, maintaining, and analyzing back-haul data

With such infrastructure, you can implement a truly iterative, use-based (or data-driven) design approach for your device, just as website designers do today. But to get this infrastructure built in the first place and to exploit it fully, you'll also need to convince key stakeholders that your complex, sophisticated, stand-alone product can in fact be conceived by your product managers, your designers, and your users, as just another very profitable website.

## Sidenotes:

- [1] Shirky, C., "Clay Shirky's Writings About the Internet Blog," "Themes," <http://www.shirky.com>
- [2] DashTruTraffic product description <http://www.dash.net/product/traffic-ddn.php>
- [3] MyDash product description <http://www.dash.net/product/mydash-saved-searches.php>
- [4] Cheng, S., "Make sense of your site: tips for webpage design," March 16, 2009, <http://googleblog.blogspot.com/2009/03/make-sense-of-your-site-tips-for.html>
- [5] Ballmer, S., "Connecting with Customers," June 26, 2006, <http://www.microsoft.com/mscorp/execmail/2002/10-02customers.msp>
- [6] Oracle Usage Reporting FAQ, <http://www.oracle.com/support/premier/software-configuration-manager.html>
- [7] Apple iPhone 3.0 Software Update description <http://www.apple.com/iphone/softwareupdate/>
- [8] Dash Express product description <http://www.dash.net/product/internet.php>
- [9] Nielson, J., "Jakob Nielsen's Alertbox," "First Rule of Usability? Don't Listen to Users," August 5, 2001, <http://www.useit.com/alertbox/20010805.html>
- [10] Dash Service product description <http://www.dash.net/product/service.php>
- [11] MyDash, Send2Car, product description <http://www.dash.net/product/mydash-send2car.php>
- [12] Wikipedia, "Windows Error Reporting (WER)," [http://en.wikipedia.org/wiki/Windows\\_Error\\_Reporting](http://en.wikipedia.org/wiki/Windows_Error_Reporting)
- [13] Kuniavasky, M., *Observing the User Experience: A Practitioner's Guide to User Research*. San Francisco: Morgan Kaufmann, 2003.

## About the Author

Tim Misner is senior director of software engineering at Oracle. Before joining Oracle, he was director of software engineering at Dash Navigation, makers of Dash Express, a two-way, Internet-connected GPS-based navigation system. Misner also served a stint as director of user experience engineering at Sun. He has a background in product management, engineering, and mathematics.

## Customer-Data-Driven Business: 10 Enabling Trends

Google and Amazon have built big businesses by collecting and analyzing previously unheard of amounts of data. Their businesses are not accidents. They are not corner cases. They are signals of an emerging future.

Several trends point to the same thing: the value of large amounts of data and the ability of that data to support tailoring, learning, and decision making—to enable new categories of business, or perhaps a new model for all businesses.

Today, these trends may still appear separate:

### 1 Big Data

Big data refers to the assembly of very large databases, perhaps first by research in the physical sciences and intelligence gathering by the NSA and the military; followed closely by telephone companies, securities exchanges, credit card companies, and consumer list consolidators (e.g., Acxiom, ChoicePoint, Equifax, Experian, and TransUnion); more recently, Web-based services have begun to generate huge amounts of data.

### 2 Conversation-based Marketing

Conversation-based marketing refers to a broad shift from one-size-fits-all broadcast advertising to real one-on-one conversations with customers. The shift began with direct mail that broadened into direct response and direct marketing. Direct mail was one of the first areas to apply big data to drive design decisions and improve customer service.

### 3 CRM Systems

Customer relationship management systems collect and store information on a business's customers (and potentially other constituents in service delivery). CRM systems are one part of the complex community of systems that sophisticated businesses need to manage conversations with constituents. CRM systems and practices are related to, but typically separate from, DM/DR activities. A business will install a CRM system internally, but serious players in DM/DR outsource customer database management, because it's a specialized skill not available from in-house IT groups. (For example, your IT folks are not likely to know where to begin to "de-dupe" your customer list or perform other forms of regular "data cleansing.")

### 4 Crowd Sourcing

Crowd sourcing is a system (often electronic) that uses large groups of people to collect and organize data. We might propose crowd sourcing as an umbrella term for a range of activities involving

large numbers of people: Open source projects that rely on volunteers (e.g., the OED, Wikipedia, Linux); collaborative filtering, the process of making recommendations based on the actions of people with shared traits; Google-style search, which uses links as one of the predictors of relevance; and flash mobs, which quickly form, take collective action, and disperse.

#### 5 **Data Visualization**

Data visualization refers to the application of graphic design principles to making large volumes of data easier to understand—that is, making pictures out of lots of numbers. Today data visualization is a subspecialty at the intersection of the hard sciences, computing, and information design, drawing on the disciplines of statistics, animation, and filmmaking. It will become increasingly important to communications design, interaction design, and service design. Closely related are design of service dashboards and augmented realities (virtual overlays).

#### 6 **Use-based Design**

Use-based design refers to a process by which the actions of users are recorded, analyzed, and used to make decisions on changes in the next version of a product or service. Use-based design has roots in the world of direct marketing. Websites created a new level of opportunity. Sites can log every view and click, and enable the testing of alternatives. Google is famous for its data-driven decision making. And now, as more products are connected to the Internet, consumer electronics (and office products) can also log every action users take, providing data to drive decisions for the next version of a product or for interim software upgrades.

#### 7 **Massive Cloud Computing**

Cloud computing involves Internet-based services that provide large amounts of computing power on an as needed or lease basis, much as utilities provide electric power. Massive cloud computing refers to recent initiatives by IBM+Google and HP+Yahoo supporting research and development efforts to increase the power and speed of cloud computing systems.

#### 8 **Sensors**

London is under surveillance by 400,000 closed-circuit video cameras. Wal-Mart has mandated that all its suppliers build RFID chips into packages that go through its distribution system. Every iPhone includes at least six types of sensors. We are on the cusp of a huge wave of sensor technology. Sensors will be embedded in everything, and they will pour out a continuous flood of data.

#### 9 **Service Design / Service Science**

Service design and service science refer to the process of developing and managing services. As hardware becomes increasingly commoditized, services offer opportunity for differentiation. A customer's experience with a brand may extend across a family of services, each with a collection of touch points. These touch points are increasingly networked, and thus customer behavior may be logged, analyzed, and used to drive improvements.

#### 10 **Social Media**

Social media refers to media (communications channels, most often enabled by the Internet) in which users create most (or all) of the content. Users engage in conversations with each other—sometimes about and with businesses that serve them. Participating in social media and its attendant conversations is a growing part of managing relations with customers. Social media are forms of crowd sourcing—or crowd sourcing from another perspective.

The next generation of computing will assemble vast stores of data—from a growing array of physical and virtual sensors. These technical and social changes will create opportunities for a wide range of companies. Consumer electronics makers like Apple will monitor their hardware and supporting services. Network-tools makers like Cisco will instrument their customers' networks. Google (and other Web-services providers) will continue to instrument search and everything you do with their tools. Health care device makers like Johnson & Johnson will increasingly offer services to complement products that continuously monitor your vital signs. Sports and apparel makers like Nike will also build biometric sensors into the soles of their shoes and the fabrics of their clothes, supporting another form of continuous monitoring.

Apple, Cisco, Google, Johnson & Johnson, Nike, and others like them will find themselves in essentially the same business, certainly dealing with the same customer management, design management, and information technology management questions. They will have entered the era of customer-data-driven business.

—Hugh Dubberly