

ESCAPE FROM DESIGNERS' DILEMMA ON CREEPING FEATURISM

Dong-Seok Lee, David D. Woods, Daniel Kidwell
The Ohio State University
Columbus, Ohio

Creeping featurism is a serious problem that provides a surplus of complexity to users, and causes user dissatisfaction and eventual sales loss. The purpose of this study is to find the opposite of creeping featurism. In examining the situation of design practitioners, this paper discusses why it is hard to get out of the design traps of creeping featurism and why simplicity is not enough for achieving the opposite. It was suggested that we need new design management strategies to prevent and escape creeping featurism.

INTRODUCTION

A car that has 49 different buttons on the dashboard and a 700-page manual [1]; A cell phone that has 583 menu commands [2]; A portal webpage that has hundreds of links. Creeping featurism (or feature creep), now everywhere, is one of the most serious problems for products. Creeping featurism makes our products visually cluttered, causes them to respond more slowly, and requires more resources such as CPU and memory. All of this ultimately leads to poor task performance, increased workload, user dissatisfaction, getting lost while interacting with products, and eventual sales loss [3, 4]. Every feature or function is useful to someone, and it is expected that added features will increase products' fitness (a measure of the health of a product). But the increased number of features causes general complexity problems and leads to fitness saturation or fitness decrease (figure 1). For instance, 48% of people have put off buying digital cameras which have integrated the functions of camcorders and digital

imaging software because they see them as too complicated [5]. Creeping featurism is inadvertent cumulative complexity in use.

BACKGROUND

The negative consequences of feature creep have become visible to some managers and business people. *The Wall street Journal* reported, "Mercedes acknowledges removing hundreds of electronic functions from its cars" [6]. The *EE Time* reported that David Hytha, executive VP of T-Mobile International, said "the mad rush to add features to cell phones has got to stop" [7]. In 2005, Vodafone introduced 'Vodafone Simply,' which provides mobile phones that have a smaller set of features: those that are used most frequently [8]. This phenomenon seems quite strange, because other companies have tended to add new features and have advertised those new features. What motivates companies such as Mercedes, T-Mobile and Vodafone to move in that direction? The authors think that Mercedes has noticed that many of its products' features are annoying or are ignored by their customers, and T-mobile and Vodafone have seen users' need for a simpler phone after being frustrated by an abundance of unnecessary features. As a result, these companies decided to invest money to remove some of the very features that they had invested money to develop.

Though aware of the problem, designers and managers were puzzled as to how their products had arrived at that point of feature creep. They began to look for ways to escape the trap. As a result, they invested more in product usability and simplicity. This study looks at how we become trapped in complexity-producing processes, which makes us ask, "What is the opposite of creeping featurism?"

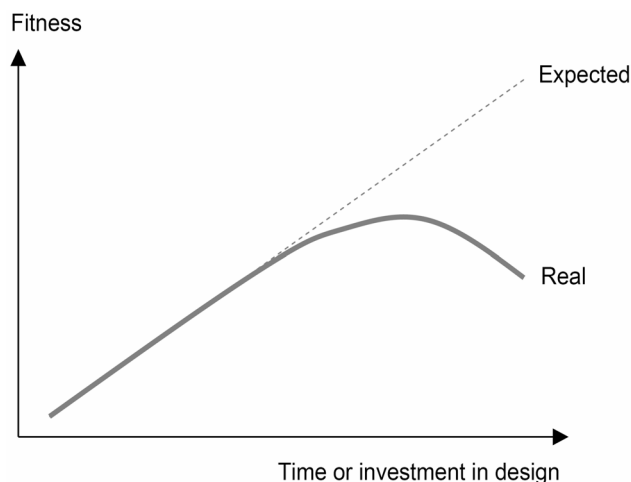


Figure 1. Fitness curve

DESIGNERS' DILEMMAS

The complexity caused by feature creep contributes to increased user complaints, high service costs, and high return rates. It would seem logical to blame the designers of a given product. But should they be blamed for feature creep? We've seen many user interface designers who have worked diligently, without the intention of making their products complex. In reality, feature creep plagues designers as well as users.

Let's look at the process of feature creep. The original design of a typical product does not start out complicated. As the product is "improved," it accumulates more features or functions as a result of input from users, designers, marketers, engineers, managers, and competitors (figure 2). After a few cycles of versioning up or redesign, the product becomes very different from its original form, lacking cohesion and possessing added design complexity. With regards to the process of feature creep and product evolution, this study found that designers are limited or even incapable of combating creeping featurism.

- **Hard to detect feature creep:** Creeping featurism is a gradual complexity-producing process. Complexity increases little by little, but suddenly, it dominates the overall fitness of a product. This characteristic makes it hard for designers to detect feature creep. Designers should pay attention to whether fitness is increasing, leveling off, or decreasing, and should recognize symptoms of fitness saturation and fitness decrease. The question is, "do we have measures or techniques to track product fitness?"
- **Externally paced by strong forces:** Designers are externally paced by other groups such as marketing, sales, engineering, management, and competitors.

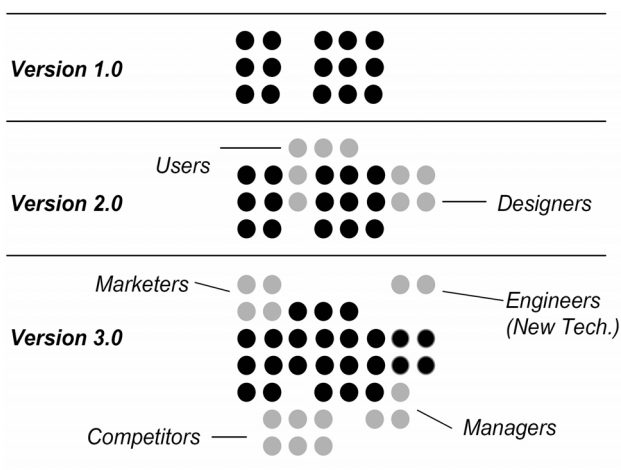


Figure 2. An illustration of process of creeping featurism

User interface designers are constantly busy responding to demands from external sources. Under workload pressures, designers add new features to current product designs by adding menu items or buttons. Suddenly, the fitness of the design is deteriorated by feature creep. To make matters worse, the current trends of product development such as convergence, connectivity, and fast-paced development accelerate the process of feature creep. The question is, "how can we balance research and development of a user interface?" and "how can we pre-explore design space so that we know where the most promising direction is?"

- **Over-reliance on incremental refinement:** We can't escape fitness saturation by incremental refinement. Once complexity begins to dominate, feature creep generates more serious problems (right-side of figure 1). This is why some products have become complicated and difficult to use, despite active usability engineering (UE) groups. UE appears to follow standard prescriptive approaches to technology development by solving short-term small repair problems on a local scale [9]. Relying on incremental refinement results in the search for solutions in a narrow design space, and doesn't lead us to design leaps or to innovation. We need to broaden our exploration in a design space, and to seek design leaps or innovations that are free from the current track of incremental refinement.
- **Accumulation of local user needs:** Designers conduct many activities in the name of user-centered design (UCD): participatory design, usability testing during development, user opinion surveys, field observations, and focus group interviews. Can these activities provide designers with what they need to escape creeping featurism? The authors don't think they can. Such activities address local user needs. They mark value, usability, and user opinions at one point in time. You're likely to find some interesting information each time you conduct these activities, but you're not likely to find trends or changes in fitness itself. Preventing and escaping feature creep requires us to have broader views such as awareness of where we are in the fitness curve (figure 1); otherwise the user-centered activity only serves to add more "wish list" items from users and stakeholders, and this is a major source of feature creep.
- **Mistakes in design advice:** Several studies have proposed solutions to creeping featurism, such as multiple interfaces [10] and providing a beginners' mode [11]. Such projected design solutions can easily cause mode problems and produce a keyhole

effect, which is a self-defeating mechanism. These recommendations for automation are brittle and create automation surprises [12]. We've seen that many previous attempts of reducing complexity by providing multiple interfaces and automation have not been welcomed by users.

Many designers cannot actively respond to the process of feature creep. It's not because of their poor skills, or because they don't know design principles. It's because of gradual complexity-producing processes and technology- and marketing-driven development without consideration to assessing the fitness of products and users' complexity.

SIMPLICITY

The first thing that comes to mind as the opposite of creeping featurism is "simplicity." When managers and companies notice cumulative complexity in use, they try to achieve simplicity by removing features. The catch is that what is "simple" turns out to be "complicated."

- **Hard to select which features to remove:** Achieving simplicity by removing features is a kind of scale reduction finesse. The problem is that each feature has a reasonable basis and can be justified. Each user needs a different set of features, as well as the availability of features, even though they may not use the features. The incorporation of each user's needs easily becomes a full feature set.
- **Wasted design efforts and missed opportunities:** Simplicity by removing features assumes feature creep by definition. Simplification is only demanded after companies or organizations see the negative side-effects caused by significant fitness decrease. Figure 3 shows a change of fitness after achieving simplicity. Consider the design investment (or cost) involved in developing and removing features, and the missed opportunities to design leap or innovation.
- **No one responsible:** In many cases, to remove is much more difficult than to add. Most designers do not have the power to remove features that appear to be unnecessary. Other groups such as marketing and sales may insist that purchase decisions are affected by bells and whistles. What is worse is that we have found that, in many companies, no one benefits from removing features.

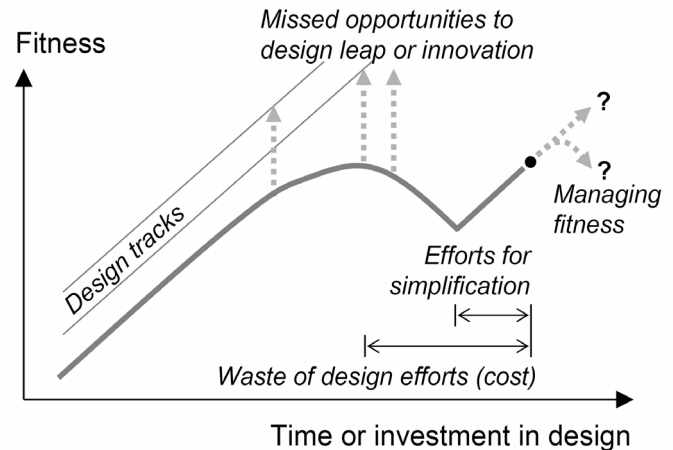


Figure 3. Fitness changes after simplification

DISCUSSION

Based on the above discussion, this study suggests that designers need to track fitness, lead design leaps or innovation, and implement new design management strategies, rather than relying on normal UE and UCD approaches.

First, designers should monitor fitness to track whether the fitness of their product is increasing, leveling off, or decreasing (figure 1). It was discussed that existing methods of UE and UCD do not focus on preventing creeping featurism. Methods, measures, or tools to assess product fitness are necessary. In this context, usability testing and evaluation needs to assess the real value of added features and the change of overall complexity of products, rather than just measuring performance after giving specific tasks and scoring users' satisfaction levels. For example, usual questions such as, "do you need this feature?", "is this feature helpful to you?", and "is it easy to perform this task?" could be replaced by "how much are you willing to pay for this feature?", "how would you feel if this feature were not included?", and "how much time do you think it took to perform this task?" [13]

Second, designers should escape creeping featurism by shifting to better design tracks (figure 3). Because of time pressure, designers tend to increase menu items or buttons in order to add new features to ancestors, and it is difficult for usability engineers to change the design even if they find serious usability problems. *Design track* refers to the idea that once a product is designed, its specifications frame and limit future design decisions. Thus the range of possible successor products can become narrowed around the dimensions and structure of the previous design. Designers become "stuck" in this frame or channel of creating a design track. As the track matures (right-hand side of figure 1 and 3), design changes produce complexity, resulting in less value for

users. The success of the Apple iPod is a good example of achieving innovation through escaping current design tracks. While other MP3 player manufacturers were trapped in the design of buttons and joysticks, Apple introduced the 'Wheel' and 'Click Wheel' interface to iPod, which provides users with great convenience for manipulating a large number of menu items and music files [14, 15]. Moving to a new design track is a conceptual shift which requires reintegration, abstraction, and reorganization. The key challenge is to develop participatory methods for pre-exploring design space and assessing the "promisingness" of new directions in order to know where and when to leap to better design tracks. Remember, feature creep suddenly deteriorates your products' fitness. It is too late to explore design space to find better directions after the side effects of feature creep are visible (figure 3).

Last, the opposite of creeping featurism cannot be achieved simply through designers' efforts. Managing fitness and leading design innovation requires managerial and organizational consideration of the

development process; role definition of design-related groups; new information to detect feature creep; and balanced R & D. Remember that Mercedes' decision to remove features and launch 'Vodafone Simply' occurred after high-level management noticed the consequences of feature creep. We call the new management strategy "fitness management," where we explore design space, envision new possibilities, and scout promising directions, as described earlier. Figure 4 shows how fitness management is different from normal UE and UCD. Fitness management strategies track product fitness in fitness increase stages to detect feature creep, while UE and UCD rely on user research to find more hidden user needs. Once the side effects of feature creep are detected, fitness management strategies begin a Klondike search [16] for figuring out promising directions in design space, while the other strategies begin to invest in usability in order to solve emerging usability problems. The purpose of fitness management is to prevent products from reaching the fitness decrease stage, where products' value will have significantly

	<i>Fitness increase stage</i>	<i>Fitness saturation stage</i>	<i>Fitness decrease stage</i>
Users' reaction	<ul style="list-style-type: none"> ▪ Welcoming addition ▪ Requesting more features ▪ Usability problems don't reduce product's fitness 	<ul style="list-style-type: none"> ▪ Serious navigational usability problems ▪ Many features used infrequently or not used ▪ Users no longer welcoming new features ▪ Reluctant to upgrade 	<ul style="list-style-type: none"> ▪ Many features annoying ▪ New features doesn't make sense to users ▪ Users look for simpler products ▪ High user complaints, service cost, and return rates, and dissatisfaction ▪ Warning on product reviews
Examples	<ul style="list-style-type: none"> ▪ iPod ▪ Google homepage 	<ul style="list-style-type: none"> ▪ Cell phones ▪ MP3 players ▪ Microsoft office 2003 ▪ eBay, Amazon 	<ul style="list-style-type: none"> ▪ Yahoo! homepage ▪ MSN messenger 7.0 ▪ Acrobat reader 6.0 ▪ Car Dashboards
Return on Design Investment	ROI >> 0 (Sales increases)	Lower ROI, ROI = 0 eventually (Sales stagnates)	ROI < 0 (Sales erodes)
Normal UE and UCD Strategies	<ul style="list-style-type: none"> ▪ User research to find user needs 	<ul style="list-style-type: none"> ▪ User research to find user needs ▪ Starts investment on usability 	<ul style="list-style-type: none"> ▪ More investment on usability ▪ Reduces features ▪ Innovation requested
Fitness Management Strategies	<ul style="list-style-type: none"> ▪ Tracking fitness ▪ Usability studies and user researches 	<ul style="list-style-type: none"> ▪ Starts Klondike search for innovation ▪ Reintegration, abstraction, and reorganization ▪ Looking for simplicity ▪ Raise the hurdle rate of adding features 	<ul style="list-style-type: none"> ▪ Moving to the most promising design tracks ASAP

Figure 4. Comparing normal usability engineering and user-centered design strategies vs. fitness management strategies to creeping featurism.

decreased and return on design investment become negative [17].

“What's the opposite of creeping featurism?” is what every design organization needs to answer. This study proffers that the answer is outside normal UE, UCD, and simplicity solutions. We suggest that considering products' fitness points us toward the answer: New design management strategies that help designers to detect feature creep, enable organizational changes, and lead design leaps or innovation by reintegration, abstraction, and reorganization.

What is “fitness” anyway? This is the question that designers and usability engineers must continue asking themselves. You have to develop a strategy to prevent feature creep and to bring innovations, whether it involves measurement of some sort, or whether it consists of watching symptoms. Precision doesn't matter. What is important is that you know whether the fitness of your product is increasing, stagnating, or decreasing. Using the fitness measure, you then need to develop a design strategy using the fitness measure.

ACKNOWLEDGEMENTS

Prepared, in part, through collaborative participation in the Advanced Decision Architectures Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009, and IT Scholarship of IITA, the Ministry of Information and Communication of the Republic of Korea.

REFERENCES

1. Seo, J.-Y. Alvin Toffler Warns of Korea's Overdependence on Chaebol. The Korea Times, SEP 6, 2005.
2. Pogue, D. The Cellphone That Does Everything Imaginable, at Least Sort Of. The NY Times, May 12, 2005.
3. Gussow, D. Unraveling technology: Manufacturers worry that if a product is complicated, consumers won't buy it. St. Petersburg Times, April 11, 2005.
4. Kapica, J. My Cellphone, My Enemy: Feature Creep and the Cellphone Dilemma. Digital Journal, January 3, 2006.
5. Philips. Philips. Why simplicity? Available from: <http://www.philips.com/about/brand/whysimplicity/>
6. Power, S. There's No Place Like... Cars? More room, more wrinkles -- and other efforts to make automobiles comfortable. The Wall Street Journal, July 25, 2005.
7. Yoshida, J. T-Mobile exec calls for restraint. EE Times, Jun 27, 2005.
8. Vodafone. Introducing Vodafone Simply (2005). Available from: <http://shop.vodafone.co.uk/simply>
9. Roesler, A., Woods, D., and Feil, M. Inventing the Future of Cognitive Work: Navigating the 'Northwest Passage' of Design. In Proceedings of 6th international conference of the European Academy of Design (EAD), 29-31 March, 2005, Bremen, Germany.
10. McGrenere, J. The Design and Evaluation of Multiple Interfaces: A Solution for Complex Software. Unpublished Ph.D. Dissertation. Department of Computer Science, University of Toronto, Toronto, ON (2002).
11. Soegaard, M. Featuritis (or creeping featurism). interaction-design.org (2003). Available from: http://www.interaction-design.org/encyclopedia/featuritis_and_creeping_featurism.html
12. Woods, D.D. and Hollnagel, E. Joint Cognitive Systems: Patterns in Cognitive Systems Engineering. CRC Press (2006).
13. Czerwinski, M., Horvitz, E. & Cutrell, E. (2001). Subjective Duration Assessment: An Implicit Probe for Software Usability. In Proceedings of IHM-HCI 2001 Conference, Volume 2, (September, 2001, Lille, France), p. 167-170.
14. Team Interface. MP3 Touch Navigation Expert Review (2005). Available from: http://www.uidesign.co.kr/case/team_report/team_report_upload_pdf/MP3%20Touch%20Navigation%20Expert%20Review_uidesign.pdf
15. Samsung electronics. Corporate internal reports: Comparative usability evaluation of MP3 players (2004).
16. Perkins, D. The Eureka Effect: The Art and Logic of Breakthrough Thinking. NY: W. W. Norton and Company (2000). pp. 67-87.
17. Lee, D.-S., Woods, D.D., & Kidwell, D. (2006). Escaping the design traps of creeping featurism: Introducing a fitness management strategy, In Proceedings of Usability Professionals' Association (UPA) 2006 Conference, 13-16 June, 2006. Broomfield CO, USA.