

## Designing for the Social Web: The Usage Lifecycle

by [Joshua Porter](#) | May 14th, 2008 | shortlink: <http://bokardo.com/p/768>

*The Usage Lifecycle describes how far a person has progressed in using your web application, helping to identify the hurdles someone needs to overcome to become regular, passionate users.*

Babycenter.com has a really [great newsletter](#). Once you tell the site when you're expecting, it sends you a weekly newsletter targeted at the specific stage of pregnancy you're in. At 4.5 months, [for example](#), it tells you that your baby weighs about 10.5 ounces and is 10 inches long. This information is timely and relevant...it knows exactly what stage you're in and helps you deal with the stresses and questions at that point.

The key to babycenter's ability to deliver a relevant newsletter is that they know your delivery date. Once they know that, they know \*a lot\*

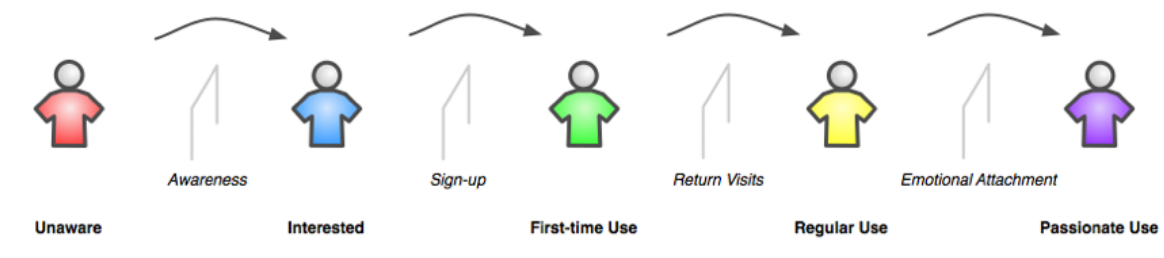
### About

Our world is made of interfaces! Interface and product design is my passion and that's what I write about here at Bokardo. My goal is to share practical knowledge with designers, product folks, copywriters, and entrepreneurs. Check out the [archives](#) for a list of popular posts.

Published by [Joshua Porter](#).

about what you're going through, as pregnancy is a well-defined process that is mostly the same for everyone. Nine month cycle. Kid. Simple.

Can people designing products of all sorts take advantage of this lifecycle process? Yes, I think they can. One of the primary ideas in my new book, Designing for the Social Web is a similar kind of lifecycle, what I call the "Usage Lifecycle". The usage lifecycle isn't as clear cut as pregnancy is, but it recognizes that *people go through a progression as they use software*. They go from not knowing much at all (like parents early on in pregnancy) to feeling comfortable with the product (like, say, when parents become grandparents 😊) to finally being passionate users.



## The Stages of the Usage Lifecycle

The stages of the lifecycle are straightforward and simple. You can dive into lots more depth as your application warrants, and you can add stages, but for the most part these five stages apply to almost all software.

- **Unaware** This isn't so much a stage as it is a starting point. Most

people are in this stage: completely unaware of your product.

- **Interested** These people are interested in your product, but are not yet users. They have lots of questions about how it works and what value it provides.
- **First-time Use** These people are using your software for the first time, a crucial moment in their progression.
- **Regular Use** These people are those who use your software regularly and perhaps pay for the privilege.
- **Passionate Use** These people are the ultimate goal: passionate users who spread their passion and build a community around your software

Note that each of these stages describes people, as opposed to a product or a market. It describes the different types of relationships people have with your software product. Have they used it yet? Have they even heard about it? What questions do they have?

Each of the stages are separated by hurdles. The hurdle between the “unaware” stage and the “interested” stage is “awareness”. At this stage what you need to do is make people aware of your product. How do you get people aware of what you’re doing? How do you get them interested and wanting to know more? How do you begin the conversation of what you do and carry that over into a meaningful

relationship?

The lifecycle is particularly relevant to web-based software because the product is inextricable from the service. The product *is* the service. If a person has a question about what your software does, for example, you can literally build that answer into the software itself. One of my favorite examples at the moment is [Tripit.com](http://Tripit.com). Tripit's design is great at moving people from the "interested" stage to the "first-time use" stage, getting people over the hurdle of "sign-up".

One of the ways that Tripit does this is by clearly explaining exactly what their service is and does. While this may seem like an easy thing to do, it's actually quite hard. To boil the essence of your software down into a handy 3-pane "how it works" graphic seems like child's-play. But only the resulting graphic is simple. Creating the simple thing is the difficult part.



Another way that Tripit helps people get over the hurdle of sign-up is to

make it super easy to sign up in the first place. They have a great feature that lets you simply forward them an email from a recent flight or hotel booking. They take that booking email and auto-create an account for you. No sign-up page to create an account. All you do is send an email.

**Just forward your travel confirmation emails to:**

**plans@tripit.com**

---

One of the problems I've seen over and over (and I've been guilty of this myself) is to recognize the stages while talking to people face to face, answering their questions, but then failing to bake that knowledge into the interface itself. By formalizing this conversation with the usage lifecycle, you can begin to set up a process of describing each stage in-depth, and then creating screens with that exact same information placed right on your web site. Just like Tripit does.

The usage lifecycle isn't a new idea. It's very similar to what a good salesman does when they target customers. They find out where the person is in the purchase lifecycle, and then tailor their message to get people moving along toward purchase. They answer the same questions over and over, point out the same features and benefits over and over. The lifecycle for any particular product or service is remarkably stable...it's only a matter of identifying the lifecycle and

designing for it. What babycenter has done with pregnancy, we should all be able to do with the usage lifecycle of our software.

So that's an introduction to the usage lifecycle. I'll be blogging more about the lifecycle as I work through the sections of my book:

[Designing for the Social Web.](#)



**Make them Care!** - Struggling to communicate the value of your product or service? I'm writing a new book that shows you how to make people care about your product or service by clearly communicating the most important bits. For designers and marketers creating product web sites. [Find out more.](#)

PREVIOUS

[Douglas Adams on Interactivity](#)

NEXT

[More on the Usage Lifecycle: Lifecycle Messaging](#)

## Links to this Post

- [Basic Thinking Blog | Buch: Designing for Social Web](#)
- [Where is the Love? Thoughts on Facebook and Work Life. « Gen Y PR Prescriptions](#)
- [More on the Usage Lifecycle: Lifecycle Messaging – Bokardo](#)
- [one man writes » Recently Read](#)

- [Identity Agnosticism « Jason Kinner](#)
- [links for 2008-05-19 at Ip's.](#)
- [Experience is Everything » Designing for the Social Web: The Usage Lifecycle](#)
- [saucybeige blog » Blog Archive » links for 2008-05-20](#)
- [Designing for the Social Web: The Usage Lifecycle | White Sands Digital](#)
- [Bookmarks de May 20th a May 22nd — rodapÃ© & marcadores](#)
- [Bookmarks for May 20th through May 22nd – 2.0 Webmania](#)
- [Weekend Reader – friendfeed, blogging, technology, programming, wordpress « // Internet Duct Tape](#)
- [Bullets of “Designing for the Social Web” by Joshua Porter « My Perspective](#)
- [association comments-o-matic » Blog Archive » Comment on "Quick clicks: Member service and unspoken truths"](#)
- [JISC-IRET » Blog Archive » A blip on the institutional radar](#)
- [Societyne » Blog Archive » Keeping Users Interested](#)
- [Webstock 09 : Joshua Porter | Ask Kalena](#)
- [Louise Ventris » links for 2009-03-04](#)
- [Understanding Audience and Purpose « Peg Mulligan's Blog](#)
- [Developing Audience-Centered Content: A Usage Lifecycle for the Social Web « Peg Mulligan's Blog](#)
- [Designing for Social Traction \(slide deck\) – Bokardo](#)

# Comments

1. [Stephen](#) 9:44am, Wed 14th, 2008

After *Passionate Usage*, there's *Boredom*, then *Notices Superior Competing Product*, then *Dumps Your Product*, and so on. Or, the alternate branch, *Hooked Forever*, which is what monopolies are looking for. Monopolies are good, if you're the monopoly.



2. Ben 3:00pm, Wed 14th, 2008

Interesting about "boredom." Lifecycle does imply death, and I wonder if that's inevitable — that people will move on. Do we need to reinvent the service so people can return to step 1, or is it sensible to aim for immortality in the last stage?



3. [Daniel](#) 9:26pm, Wed 14th, 2008

Great article Joshua! I really need to pick up a copy of your book.



@Ben: Interesting point. As Joshua points out, "Emotional Attachment" is really the ultimate stage you wish your users/customers to reach. At that point, you will have formed a real relationship with them.

Just like with marriage, you are very emotionally attached (or should be) to your spouse. Yet just because you reach this strong love/attachment stage, that doesn't mean that's it! You both still have to work hard to

maintain your relationship or else the emotional attachment and passion will fade. You must do the same with all of your users no matter what stage they are at (but especially those with emotional attachment).

As for whether the death of your software (people stop using it) is inevitable... I think that's up to you, the software developer. Do you want to create something short term? Somewhat fad-like? Or do you want to build and maintain something that creates value for the long haul? I don't think you need to completely reinvent your service to escape death, but you definitely need to be able to adapt to your environment and what your users want and need.

---

**4. Charles** 1:52am, Thu 15th, 2008

Great Post, I order your book immediately !

As I understand it most user drop out of the circle at the *Interested* stage.

Actually it seems like a big climb upwards to get your users *passionate* 😊



---

**5. James Creare** 10:59am, Thu 15th, 2008

Lifecycle indeed, however I have noticed that all good processes like your examples, especially in the commercial world, are picked up by your competitors.

For about the last 2 years, TGI Fridays email me vouchers when it is my birthday for a free drink/meal etc, and the same sort of things when it is close to christmas etc.



Now all of a sudden, another similar restaurant chain are doing the same thing, cashing in on the same lifecycle process that TGI's use, at emailing their customers and bribing them, at the right times.

---

**6.** [K.D](#) 2:43am, Tue 20th, 2008

- 1, Sometimes you aren't even aware of what yourself is doing, cuz you are doing beta.
- 2, Sometimes people feel interested with certain kind of service first, then they come across your product.



---

**7.** [Austin Govella](#) 1:19pm, Fri 23rd, 2008

Your cycle echoes the marketing discipline's "AIDA" concept. Might be fuel for more thought.



Also, at Comcast, we've noticed every experience has two distinct cycles.

1. The acquisition cycle (roughly parallels your cycle above) is about how users come into contact with and begin interacting with the experience.
2. The consumption lifecycle is about how users interact with the service space your experience impacts. (Not very clear, but bear with me.)

Our example is fancast.com. We have a very clear acquisition cycle designed to make users aware of the site and then move them gently from first-time to comes every day users.

The consumption cycle, though is about for Fancast talks about how people consume entertainment. There's a dovetail at the end of the two cycles: people will come all the time, if we help them consume appropriately.

I've found thinking about the two cycles individually helps focus specific interactions on the site.

There's a corollary thought here as well. Design is always about affecting user behavior. With Fancast, we have an image of an ideal user being someone who watches lots of entertainment and talks about and shares lots of entertainment with their friends.

Part of our design focus is based on moving users from one behavior group into our desired behavior group. (Of course the assumption is the desired behavior group would make us lots of money).

So maybe that's three cycles to consider when designing an experience?

1. How users become aware and then develop a relationship with your service.
2. How users consume the service your experience provides.
3. How you affect user behavior to benefit both you and them.

---

**8. [Jessica Enders](#)** 7:35am, Thu 29th, 2008

Hi Joshua

Love your thoughts and writing – always full of gems.



Reading this article I was struck by similarities with a behaviour change model put out by Alan R Andreasen in his 1995 book

“Marketing Social Change” (search on Amazon). Andreasen’s model describes four basic stages people go through when adopting a new behaviour: pre-contemplation; contemplation; action and maintenance.

I think you’ve nicely captured the ideas underlying Andreasen’s model and applied them to the social web context. Maybe you can do a similar adaptation to take his ideas about how to move people from one stage to the next in a social *marketing* context into the social *web* space.

Cheers  
Jessica

---

**9. Jeff Hock** 2:47pm, Wed 11th, 2008

I’m not buying into this whole concept of “usage lifecycle”, it’s a half baked idea. Geoffrey Moore’s “crossing the chasm” hits the nail on the head regarding this topic. You can try to spin it your own way as much as you’d like, but that dog don’t hunt.



---

**10. Bruce P. Henry** 1:15pm, Fri 11th, 2008

There seem to be two kinds of social interactions that each have different challenges. I don’t know what to call them but I’ll try:

- 1) Communicative social software
- 2) Collaborative social software

Communicative social software is where a single user controls a thing (like their profile) and then interacts with other users via some



mechanism (e.g. Twitter, Facebook, LinkedIn).

Collaborative social software is where multiple users at different lifecycle stages need to collaboratively interact (e.g. Wikipedia, SourceForge).

Our company builds online project management software. I don't think that project management software is often thought of as "social" but since (ideally) it's designed to facilitate interactions between people, I think it should be social. That's why I read this blog; we want health social interactions built into our product. Our product is a good example of "collaborative social software".

The thing that vexes me is having different users at different stages in their lifecycle. This seems to be easier to cope with in the communicative model than in the collaborative one.

How the heck do you design for these different stages simultaneously? It feels like trying to build a single chair that is perfect for infants, teens, adults, and seniors all at once.

---

**11. Firebubble Design** 5:48am, Thu 9th, 2008

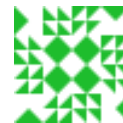
@Jeff Hock to say the usage lifecycle concept is a 'half baked idea' is a bit of an understatement in my opinion.



---

**12. paydayadvances** 4:49am, Wed 18th, 2009

I'm interesting in designing a social web of my own. May be it will usefull for me. Thank you.



---

**13. [web designer](#)** 6:35am, Thu 26th, 2009

The diagram of the different stages shows the psychology of why social networks do so well. When a user gets to passionate use stage they are emotionally attached

