

# Designing and Engineering Time: The Psychology of Time Perception in Software

시간 경험에 관련된 세 가지 요소(actual duration, perceived duration, tolerance)를 구분하고, 이중 주관적 요소(perceived duration and tolerance)를 조정하여 시간 경험에 영향을 미치는 방법을 소개한다. --AlanKang, Year2009

## **The objective of this book:**

The objective of this book is to ensure that the user experience is expeditious. This is done by not blindly applying what has worked for other people, but to go one step further to understand why it works. --loc.380

## **Every software requires users to expend time:**

Every interaction with the computer?from performing a search for information in a database that may last a few seconds to installing an operating system that can last for hours?requires users to expend time. --loc.395

\* Especially true for game software --AlanKang, Year2009

## **On subjective time:**

Two individuals may experience a three-hour documentary differently, depending on each individual's interests, context, or language, for instance. Although we can't roll back the clock to see how the Egyptians built the Great Pyramid of Giza or accelerate time to celebrate the birthdays of our great-great grandchildren, we can deal with the way we experience time. The key to engineering subjective time is to engineer the factors that affect the experience of time. --loc.459

## **Different questions to consider:**

Consider the following questions: How can we make the download faster? How can we make the download appear faster? How do we make users tolerate the download? Although these appear to be the same, they are really different questions that require different answers. --loc.681

### **On perceived duration:**

Perceived duration is only meaningful when it is compared to a tolerance threshold. If the perceived duration is shorter than the tolerance threshold, the user interprets that as fast. Conversely, if the duration is perceived as longer than the tolerance threshold, the user interprets the duration as slow. --loc.835

### **Weekly, Daily cycle affect time perception:**

Research has shown, for example, that many fast food restaurant customers are more impatient during lunch time than dinner time. In this instance, time constraints of lunch time (for example, having to return to work) obviously play a primary role. Day of the week can also play a significant role in tolerance, too, as well hinted by the phrase Monday blues. It is reasonable to speculate that many office workers will have certain priorities upon returning to the office at the beginning of the week. --loc.901

### **Emotional state also affects time perception:**

Research has repeatedly shown that under significant emotive or stressful circumstances, perception is highly susceptible to distortion. --loc.911

### **Summary:**

orrowing from Csikszentmihalyi's flow theory, this chapter discussed three ways by which you can ensure that user flow in your solution is optimal. The first concerns adjusting the challenge of using your solution to match the skill level of users. The second concerns providing clear goals and feedback by removing ambiguous communication in the UI. The last is to provide users with an environment in which they maintain their sense of potential control. --loc.2494

### **Two approaches - percetion management, tolerance management:**

The techniques in this chapter fall into one of two general approaches that must be considered to improve the user experience of your product in terms of time and timing. The first approach is to make users feel that the actual duration of a process is shorter than it really is. This is called perception management. The second approach is to make users more tolerant of the duration. This approach is called tolerance management. --loc. 2769

# Perception management techniques

---

In the following pages, we consider the following techniques that may be applied to your solution to make time "fly":

- \* Preemptive start
- \* Early completion
- \* Invisible deconstruction
- \* Descending durations
- \* Nonlinear progress indication
- \* Continuous durations
- \* Information
- \* Meaningful diversion
- \* Fire-and-forget

--loc.2777

## **Preemptive Start:**

Frequently, there is no need to wait for the user to provide all inputs before proceeding with a process. Whenever possible, start the process instead of waiting until all inputs are provided.

How and Why It Works:

People can define a duration only when there is a clear start time and a clear end time. When users finally get to a point where they explicitly start the process, this is what they will naturally and mentally assume as the start time.

--loc.2781

- \* File downloading process in Google Chrome, GMail attachment --AlanKang

## **Early Completion:**

Just as you can preemptively start an event, you can prematurely "end" an event by either performing some processes in the background or delaying these processes to give users the impression that the entire process has completed. This works well with processes that have substantial cleanup work. Users can get anxious to start using your product or stop using it to move on to another task. So, instead of making users wait for your solution to finish cleaning up, allow them to start consuming your solution while it completes the processes that users do not need to see or care about it.

How and Why It Works:

Like the previous technique, this works because the end time is defined without the cleanup processes.

--loc.2798

### **Invisible Deconstruction:**

Related to the preceding technique, this technique calls for you to unload and deconstruct components of your solution that users do not need to see.

How and Why It Works:

This technique works for the same reasons that make early completion work.

--loc.2817

\* Firefox process

### **Descending Durations:**

Many lengthy processes are made up of multiple discrete subprocesses that are meaningful to the user (Figure 10.4). An installation of an application, for example, may require the installation of supporting engines, technologies, or some other prerequisites. Whenever possible, arrange the order of the subprocesses such that the subprocess with the longest duration starts first.

--loc.2834

### **Nonlinear Progress Indication:**

For long running processes, instead of reporting progress indication in a linear way (50% means exactly half done), you can report progress nonlinearly, as shown in Figure 10.5.

How and Why It Works:

This technique works in the same way as the descending duration technique in that it takes advantage of the fact that people are more likely to view the tail end of long-running tasks.

--loc.2851

### **Continuous Durations:**

Ask for all user input up front, and have the task continue unattended all the way to completion instead of pausing midprocess to prompt users for more information. --loc.2868

\* Anti-pattern: Windows XP Installation process

### **Information:**

Give users the right amount, level, and type of information on how long a process or task takes (Figure 10.7). Research has shown that bank customers who were given information about a wait perceive the wait to be shorter than its actual duration. --loc.2886

### **Meaningful Diversion:**

This technique borrows from the adage "Time flies when you are having fun." During a lengthy task or process, present information to the user that will be engaging or valuable to the user. This is often done in lengthy installations. --loc.2902

### **Fire-and-Forget:**

Some processes and tasks, unlike installations or downloads, do not have any tangible value or purpose to users after they complete them. The value in these processes is that they complete successfully. Examples include system maintenance tasks, such as scheduled antivirus scanning. For such processes, inform users that the process completes in the background, as shown in the ABC backup example in Figure 10.9. Consider a notification of completion, too. --loc.2920

# On Tolerance Management

---

Tolerance Management In many instances, it is impossible to make users believe that actual time is shorter than it really is. Generally, these are during longer durations, such as when installing software or antivirus scanning. In such instances, consider using (or complementing perception management with) tolerance management. ...

- \* Underpromise, overdeliver
- \* The Priceline model
- \* Time anchors
- \* Worth the wait
- \* Buffer and offer
- \* First-time, one-time only
- \* Contextualized benchmarks
- \* End on time

--loc.2940

## **Underpromise, Overdeliver:**

When stating how much time a process will take, estimate slightly longer than the actual duration.

--loc.2946

## **The Priceline Model:**

During a long-running task, communicate the value of the task to users. If users understand that the system is performing a task that benefits them or brings them value that they cannot get through other means, they will be more tolerant of the wait. --loc.2962

## **Time Anchors:**

Use time anchors to express time in the UI, particularly when stating maximum time ("less than"), estimated remaining time, estimated range, or frequency. --loc.2980

## **Worth the Wait:**

Ensure users that the wait will be worth it and make sure that what users get at the end of a wait matches or surpasses what they expect. You want them to feel that it was "worth the wait." When users have to repeat the same process in the future, they will be more tolerant of the wait.

How and Why It Works:

The effectiveness of this technique hinges on building tolerance during the wait and securing satisfaction after the wait (Figure 10.13). Many restaurants, for example, proudly display rave reviews from magazines and newspapers in their waiting areas. Beyond fame and pride, these wall-mounted trophies work to build tolerance during a wait. Ensuring that service is indeed worth the wait is critical. Research has shown that a very positive value consumers receive at the end of a long wait can correct the negative perception that was formed during the long wait itself. That is, while standing in line, a consumer may first feel that the experience of waiting is absolutely terrible, but after receiving great service, the same consumer might feel the waiting experience wasn't as bad as originally thought. In such a case, the consumer's tolerance will be higher next time because she understands the restaurant is worth the wait.

--loc.2996

### **Buffer and Offer:**

Offer to users what they can begin to use or consume before the entire solution or product is fully available. The best example of this is streaming videos over the Internet?you do not have to wait for the download of the entire video file to start watching it. --loc.3013

\* Quake Live, Unity Engine

### **First-Time, One-Time Only:**

Some solutions will have delays that exist only when the solution is started up for the first time. In these instances, inform the user that this is a one-time delay that happens only when the solution is loaded or started for the very first time. --loc.3031

### **Contextualized Benchmarks:**

Provide benchmark information to users to help them understand the dependencies of your solution.

How and Why It Works:

Whereas many tolerance techniques persuade users to be tolerant, this technique essentially turns the table and gives the user an "it's not me, it's you" message. Figure 10.16 shows two examples. More accurately, it shifts the burden of performance from the solution itself to other dependencies, such as network bandwidth, system processor speed, memory, and so forth. --

loc.3049

### **End on Time:**

Give approximate remaining time in time anchors while a process is running, but when the process is actually complete, add a few seconds and show the process counting down to and ending exactly at zero (Figure 10.17).

Keeping in Mind:

This works for relatively longer durations that run for at least several minutes. Adding five seconds to a ten-minute process is fairly negligible. Adding five seconds to a three-second process would be gratuitous.

--loc.3069

### **Violations:**

If it is possible to make time fly, it is also possible to make time crawl. --loc.3130

# On Perceptual Violations

---

Many factors influence the perception of duration, and the following violations capture some of the factors that negatively affect the UserExperience:

- \* Watching the kettle
- \* Captive waits
- \* Negative appraisal
- \* Elapsed time
- \* Barnabus Effect
- \* Information overload
- \* Fragmented durations
- \* Anxiety

--loc.4088

## **Watching the Kettle:**

Paying attention to a long-running process might cause the actual duration of the process to be overestimated (Figure 11.1). --loc.3140

## **Captive Waits:**

Any running process that holds the user's computer, operating system, or application "hostage" likely causes the captive time to be perceived longer than it really is. --loc.3163

## **Negative Appraisal(평가, 견적, 감정):**

Any intended or unintended indication on the UI that explicitly suggests that a process takes a long time causes users to form a perception of the process as being too time-consuming. --loc.3190

## **Elapsed Time:**

Any indication on the UI that tells users how much time has elapsed without any other purpose only serves as a sore reminder of how much time has gone by. --loc.3210

**Barnabus Effect:**

The first experience of a new or unfamiliar process is prone to be perceived as longer than subsequent experiences. --loc.3231

**Information Overload:**

Overloading users with too much information about a process during the process itself can cause perception of duration to be inaccurate. --loc.3255

**Fragmented Durations:**

Interruptions break a single process into multiple parts, which are prone to be perceived longer than one that is remembered as a whole or made up of fewer parts (Figure 11.7). --loc.3275

**Anxiety:**

Under stressful or highly emotive states, perception of time is heavily distorted, and delays are typically magnified. --loc.3299

## On Tolerance Violations

---

Some violations do not cause actual time to be perceived longer than it really is. Rather, they cause users to lose patience and tolerance. ... This section focuses on what is within your control to identify and correct:

- \* Uncertainty
- \* Broken promises
- \* Cable company commitment
- \* Overprecision
- \* Loop confirmation
- \* Surprise supplement
- \* Deleyed consumption

--loc.3315

### **Uncertainty:**

Not giving users any indication or information about when a process will complete causes time to be perceived as passing more slowly. --loc.3322

### **Broken Promises:**

When a process goes beyond the duration as estimated or promised by the UI, user tolerance is negatively affected. --loc.3337

### **Cable Company Commitment:**

When the estimated range of duration is too wide, tolerance is reduced even before the process begins (Figure 11.11). --loc.3353

### **Overprecision:**

An overly precise projected time of completion invites users to put the precision to the test. --loc.3368

### **Loop Confirmation:**

A common practice to fill up time during long-running processes, especially installations, is to

display a finite series of preset information in the UI. However, when the series finishes and restarts at the beginning, the cycling of the series reduces user tolerance (Figure 11.13). --loc.3386

**Surprise Supplement:**

When users are not warned about a process that follows another, their tolerance for the unexpected process is greatly reduced (Figure 11.14). --loc.3406

**Delayed Consumption:**

When users feel they are forbidden to start using a solution because it has not loaded all its features, many of which the users do not need immediately, tolerance is reduced. --loc.3426

## See also

---

\* Geelhoed, E., P. Toft, S. Roberts, and P. Hyland (1995). To influence time perception. CHI'95 Mosaic of Creativity, 272-273.

\* Kellaris, J. J. and R. J. Kent (1992). The influence of music on consumers' temporal perceptions: Does time fly when you're having fun?

\* North, A. C., D. J. Hargreaves, and J. McKendrick (1999). Music and on-hold waiting time. British Journal of Psychology, 90, 161-164.

\* TimePerception