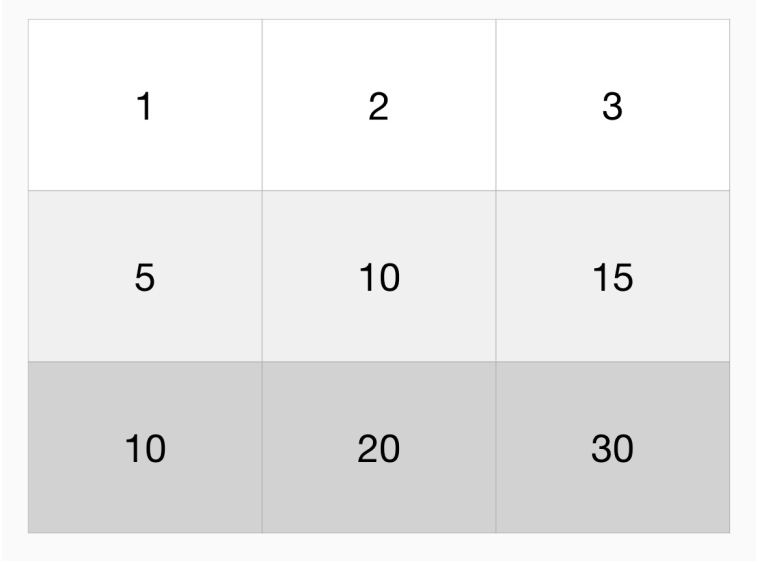
Don’t Make Me Wait! Cheat Sheet

Representing Time

Time anchors

Use Time Anchors when representing time in a system or interface. Time anchors are a method of representing time ranges and estimates in language that is more amenable to how humans think and represent time in language.

1. Use for time estimates under an hour
    a. 1, 2, 3, 5, 10, 15, 20, and 30 (can be minutes or seconds)
2. Time Anchor Matrix (see image below)
    a. Use this matrix to express ranges
    b. Never skip over a number when representing ranges
    c. Keep in mind: Ranges prevent users from holding us to exact numbers (or at least they are less likely to do this when given a range)
    d. Use anchors for countdown units in timers



Expressing Time Units

1. Singularize Single Units
    a. Not "1 minutes" but "1 minute"
    b. Not "Zero Minutes" but complete or finished
2. Zero means finished
    a. Do not move on to a new process such as unpacking files
        i. In such instances, inform user new process has begun
3. Express time units consistently
    a. one (written out) versus 1 (as a digit)
    b. Double digits are better expressed as numbers
4. Between X and Y
    a. This process will take between 3 *and* 5 minutes (see Time Anchors above)
5. Avoid Ambiguity
    a. Don't use *momentarily* or *awhile* as terms

Reporting Time

There are 3 essential ways to report time in a system.

1. Prospective Reporting (or how long will this take?). Use this method when:
    . Users need to decide if they can afford to start a process
    a. Users would like to attend to other tasks
    b. The process is very long or will hold them captive
2. Real Time (or how much longer will this take?). Use this method of reporting when:
    . The ongoing processes and operations are too technical or meaningless to the user
    a. There is a need for users to know and act on elapsed time
3. Retrospective (or how much time did that take?). Use this method of reporting when:
    . It is meaningful or valuable for users to know how long a process took
    a. Diagnostic measurement or performance assessment is necessary

Designing Progress Indications

There are 3 considerations when designing progress indicators:

1. The display mode: Textual or Visual
    a. Most processes will use both. Only when you are limited will you use one or the other (ex: LCD displays can only use textual)
    b. Visuals can be progress bars, flying documents/files, hourglasses, spinners etc.
    c. Use textual when:

        i. It is important meaningful or valuable to the user to know about on-going processes
        ii. Users are able and willing to read the textual representation
    d. Use a visual indicator when:
        i. Details are too technical or meaningless to the end-user
        ii. Users cannot or will not pay attention to progress indication
2. Time Units: Types and which type to use
    a. Two types
        i. Time elapsed
            1. Use only when a user can take action due to a certain amount of time passing
            2. Use when diagnostics or performance measures are of concern
        ii. Time Remaining
            1. Use in majority of situations
    b. Use time units when:
        i. You can confidently and accurately project completion time
        ii. Users are anxious to consume what comes upon completion
        iii. Users will likely work on other tasks while waiting for completion
        iv. Some communication with another live party is involved
    c. Two cautionary notes:
        i. Be careful how you anchor the time units. A second-by-second countdown can be suboptimal
        ii. Time units should never increase in increment and ***always*** decrease.
3. Work Units are a method of simply representing the amount of work being done. Norton Antivirus is a good example of this when the software indicates it has scanned X number of files. That is a work unit.
    . Use work units when:
        i. Remaining time is too variable or unpredictable
        ii. Users are involved in setting up the steps in the process
            1. ex: Downloading an album via iTunes (users want to know each song was downloaded correctly)
        iii. Mid-progress failures are meaningful and actionable to the user
            1. ex: The user can take some action on a failure in the process
4. Data type: Quantitative or Qualitative
    a. Quantitative has a numeric value and should be used when:
        i. Users need objective data to analyze performance, failure or other diagnostics
        ii. It will provide assurance that work unit have successfully completed
    b. Qualitative data simply expresses some state, phase or property of the progress, such as "Removing temporary files" and should be used when:
        i. Quantitative data is meaningless to the user
        ii. Users benefit more from simply reading a simple description of the work being done

iii. There is a “middle of progress need" to call something to the user's attention

Progress Indication Guidelines

When do you need to use a progress indicator and what type should you use?

0-2 Seconds – generally there is no indication needed for this. However, a spinner is perfectly appropriate and certainly won’t hurt anything.

2-5 Seconds – a busy animation (i.e. spinner, hourglass etc.) is needed because this is just long enough to cause concern for the user.

5-X Seconds - progress indication needed (ideally a determinate progress indicator) so the user does not think the system is not responding.

10+ Seconds – progress indication is needed as well as a cancel button so the user can cancel out of the process.

Improving User Flow - 3 Methods

User flow is most concerned with not slowing the user down or fragmenting their process. It relies heavily on the user’s state of emotion, knowing emotions can distort time. A user in a state of flow will lose a sense of time passing.

1. Challenge-Skills Matching
    a. Match skill level of your user by building in varying skills levels
        i. Expert user vs. Beginner
            1. This could be used as a technique for segmented populations of users
2. Goals & Feedback
    a. Ensure feedback in your system is unambiguous and immediate
        i. Delayed feedback causes confusion and interrupts user flow
        ii. User has a goal when they enter the system. Slow or vague feedback causes them to question whether they will obtain that goal
    b. Trim information presented to user when at all possible.
        i. Long sentences and explanations are usually not read
        ii. If a paragraph looks chunky, studies indicate users will gloss over the information
        iii. Be descriptive in buttons so user clearly knows what action will occur when they click the button

3. Give Users a Sense of Control
    a. Undo, escapes hatches and back buttons give users a sense of control (i.e. forgiveness) where mistakes made in a system can be undone
    . This is all largely based on the research of Csikszentmihalyi

Techniques to Manage Perception

Perception Management

This involves making something *seem* faster and exploiting our subjective nature of time.

1. Preemptive Start
    a. Can we move the user into an "active wait" where they are occupied more quickly? Skeleton loading is an example where the user waits a short period of time, but then begins to consume content. It is most effective because it eliminates a clear start and end time. People have difficulty estimating a duration when there is no clear start or end time.
2. Early Completion
    a. Works just as the preemptive start does but ends a process early finishing minute details in the background and allowing user to start using the software or service. Buffering and streaming are good examples because the wait time is "occupied" (or an active wait) while the video loads in the background.
3. Invisible Deconstruction
    a. Unload and deconstruct (closing effects) in the background. Works in tandem with above solutions
4. Descending Durations for progress indication
    a. Place longer processes at the beginning of an event and descend them
        i. Use time anchors to estimate process up front
        ii. Users will most likely attend to other items when process begins and return as shorter processes complete (the tail-end of a process). When the tail-end is shorter, the user feels as though the system is more efficient.
5. Non-linear Progress Indication
    a. Mimics descending durations
        i. Once again, the user is more likely to watch the tail-end of a process. Therefore, stage the end processes (quicker processes) at the end so they will appear to complete more quickly.
        ii. Keep in mind this technique assumes the user does not need the exact timing of the process for a mission critical task or for some other reason.
6. Continuous Durations
    a. Get all user information (input) up front.
    b. Automate process once information is received.

    c. Multiple blocks of segmented time can lead to user perceiving processes as longer than they truly are.
7. Inform the User
    a. Give users a concrete sense of how long a process will take. Research shows that customers who were given a waiting time perceived the wait to be shorter than its actual duration.
    b. Uncertainty creates huge distortions in time perception.
8. Meaningful Diversion
    a. Give users a diversion that is valuable or engages them. Ideally, you could use this during a lengthy installation to distract the user from the time. However, the more engaging the better. Example, have them click on options to learn more about the product. Bad example: An automatic slideshow requiring no interaction. (The caveat with this approach is intensive diversions could slow down systems and processes.)
9. Fire and Forget
    a. Does the user need to be here for this process?
    b. Inform user of process, get their input and have it complete in the background.
        i. Example: Automatically shutting down the computer when updates install

Perceptual Violations

(or what not to do when managing perception)

1. Watching the Kettle
    a. Painstakingly visualizing a process to the user with no clear indication of the ending time.
        i. This generally refers to long processes.
    b. Give clear indication of end time.
    c. Don't show details of the process unless it is necessary.
2. Captive Waits
    a. Holding the user captive to a process
        i. This cedes user control of the system.
    b. Any process lasting 10 seconds or longer should include a way for the user to cancel or stop the process.
    c. For processes that take several minutes, consider minimizing the interface, encouraging users to attend to other tasks or present a notification when the process is complete.
3. Negative Appraisal
    a. Any intended or unintended indication in the UI explicitly suggesting a process may take a long time will cause users to form a perception the process is too time consuming.

    b. Consider the level of information the user needs to see - only show what the user needs to see.
4. Elapsed Time
    a. Unless there is an express purpose for showing elapsed time, don't do it.
        i. Akin to watching the kettle above.
        ii. Consider reporting work units completed (or remaining) instead.
5. Barnabus Effect
    a. A new or unfamiliar process is likely to be deemed to be longer.
        i. Barnabus is the name of a lab rat at Brown University who performed tricks for food. A short video clip of his performance was shown twice to subjects and they overwhelmingly assessed the second video was shorter even though it was not.
        ii. The brain is more actively involved in observing a new process and thus the process will seem longer than the second time when the brain is now familiar with the event.
6. Information Overload
    a. Showing a flow of information can cause time to be perceived as longer.
        i. Based on Attenuation Hypothesis.
    b. This is especially true if the information is meaningless.
        i. Parse information and ensure the user truly can use it or make sense of it.
7. Fragmented Durations
    a. The sum of the parts is greater than the whole.
        i. This principle occurs when a process is broken into smaller parts and are prone to be perceived as a longer process than the whole.
    b. Tasks should be structured so user input occurs at the beginning of a process and then the process executes.
        i. Keep the parts tight - especially when user interaction is minimal or rote.
8. Anxiety/Anger
    a. Emotive states increase the length of perceived time.

Techniques to Manage Tolerance

Tolerance Management

With tolerance management, we are acknowledging their will be a wait and attempting to change the user's perspective by leveraging their tolerance for the situation.

1. Underpromise & Overdeliver (or Exceed Expectations)
    a. Restaurants do this when they estimate a longer wait time than reality.
    b. Do not underestimate
        i. Use next highest time anchor as a rule of thumb
2. Justify the Wait

    a. Communicate value of task to users
        i. Users are more tolerant of a task when they know the wait is necessary for good quality, etc.
        ii. Users are also more tolerant of a task when they know how hard a system or person is working to honor their request.
        iii. Think Priceline and how it tells the user it is scanning through millions of sites to find them the best deal.
        iv. Value must be valuable to user and not to us!

0. Worth the Wait
    a. Underscore the worth of the wait or ensure the wait is worth it
        i. Think of the restaurant that displays its awards or winning reviews on the wall
        ii. If the experience is truly better, the wait will be forgotten since it is in the beginning and the experience comes later

0. First Time & One Time Only
    a. Ensure users know if a delay is a one-time delay
        i. This is akin to building tolerance like when explaining you were late because you were not familiar with a certain location.

1. Contextualized Solution
    a. Turns the tables on the user and informs them of the performance of their machine, internet connection, etc.
    b. Give user a benchmark visual if possible.
        i. Ensure this is done with finesse.
        ii. Give specifics on which aspect of their system is inadequate.

2. End on Time
    a. Add a few seconds to counter at end of process to show it is done (i.e. a small 3,2,1, countdown will make the system appear more efficient)

Tolerance Violations

(or what not to do when managing tolerance)

1. Uncertainty
    a. Not indicating how long a process will take increases length of perceived time through uncertainty.
    b. Not indicating how long is left in a process is also a violation.
2. Broken Promises
    a. Set realistic expectations.
        i. breaking an estimated time length will reduce tolerance considerably.
3. Cable Company Commitment
    a. Do not include too wide a range when estimating.
        i. Use time anchor table.
4. Overprecision

    a. Do not be too precise in time projections.
        i. Overly precise projections invite users to test the system.
        ii. Avoid unless there is some need to show the user precise times.
5. Loop Confirmation
    a. Repeating information such as a slideshow on installation.
        i. Once the slideshow begins to loop, tolerance decreases.
6. Surprise Supplement
    a. Additional wait times
        i. Think doctor office where they bring you back for a second wait.
        ii. Software installation that adds another process into the flow.
7. Delayed Consumption
    a. When users feel they are forbidden to start using a solution because it has not loaded all of its features, tolerance is reduced.

Source Material: Seow, S. C. (2008). Designing and engineering time: The psychology of time perception in software. Boston, Mass: Addison-Wesley.