Design of Everyday Things Checklist

Focus: You can have a broad customers segment or a broad product feature/user experience but not both (Ex: Anyone can use Google but it has a very narrow product feature/simple user experience, Not everyone can use Microsoft Excel but it has a much broader product features/more advanced user experience)

Paper: Always draw it first on paper.

Random Objects: Think of random objects and try to find ways to apply them to your product.

Discoverability: Is it possible to figure out what actions are possible and how to perform them.

Understanding: How is the product supposed to be used?

Testing Environment: You don’t test a real shotgun playing buck hunter. Test your product in a fully developed environment that best matches the user’s situation.

Human Error Myth: When something fails, it means that it was not designed properly.

Human Centered Design: Designs to accommodate human needs, habits, capabilities, behaviors, and emotions.

Affordance (“is for”): Relationship between a physical object and a person. A chair affords (is for) sitting. If the chair weighs 300 lbs it does not afford lifting or moving. Affordance is a relationship, not a property.

Anti Affordance: The prevention of interaction.

Signifiers: Communicate where an action should take place.

Mapping: The relationship between the elements of two sets of things. Mapping helps mimic a product to help the user understand how to interact with the product. An example is the placing the switches for a stove top in the same order and position to mimic the burners that they operate.

CRAP: Contrast, repetition, alignment, and proximity are fundamental to good design.

Feedback: Some way of letting you know that the system is working on your request. Remember that too much feedback (dishwasher beeping at 3am) can be annoying.

Conceptual Model: A highly simplified model of how something works.

Mental Models: Conceptual models in people’s minds.

Constraints: Features that limit options to help use the product correctly.

Standards: Generally accepted terms and designs to help people carry knowledge and usability from one product to another.

Legacy Problem: Products are designed poorly because of outdated standards.

Drill Down: When people go to a bookstore to buy a drill, that is not their real goal. Perhaps they want to install shelves so why not discover their true goal and design shelves that don’t require a drill.

Subconscious v. Conscious: Subconscious is fast, automatic, multiple resources, conscious is slow, controlled, limited resources.

Expectations: Play and important role in our emotional lives.

Storytelling: Very persuasive medium. People are innately disposed to look for causes of events to form explanations.

Thermostat Myth: Most people think that if you turn a thermostat on high, it will heat up quicker, most thermostats only have two settings, full blast until temperature is reached, and off.

Learned Helplessness: People experience repeated failures at a task so they assume it cannot be done. Reason you should eliminate error message from programs and instead provide a process to help user.

Accepting Multiple Formats: Programs should be able to treat 1/5/15, Jan. 1, 2015 and 01-05-15 as the same.

Feedforward: Signifiers, constraints and mapping that help indicate what a user can do with the product.

Three Ways to Store Knowledge: In the head (brain), in the outside world (journals, hard drives), and via imposing constraints on the world (setting self up for future success). The most effective way of helping people remember something is to make it unnecessary.

Security Balance: Make something too secure and it becomes less secure. Example, having too many passwords that you must trust them in something insecure or securing the building to much so that people end up propping certain doors with bricks.

Short Term Memory: Limited to five to seven items.

Sleep Rehearsal: Sleep helps us rehearse things in our head.

Keep it Simple: Keep it simple.

Good Enough: Don’t strive for perfection in every single aspect of the design and be willing to settle for good enough.

Clues: Human mind with a few clues can discover how to do almost anything.

Purchaser v. User: Often the purchaser is not the user and you must design so that both parties understand.

Cultural Differences: In Arabic and Hebrew time flows from right to left, others use a left to right system.

Contrast: Like parts cause confusion. Contrast everything.

Semantics: Study of meaning.

Instructions Not Needed: Products should be simple and simple products do not need instructions.

Forcing Function: Physical constraint that forces the user to do something so that they cannot fail. An example is forcing a driver to push down the brake before you can shift gears.

Interlock: Forces an operation to take place in the correct sequence.

Lock-In: Prevents a a person from prematurely stopping an operation. An example is keeping the program running until it is finished saving.

Lockouts: Preventing and action until something has been done. Example password protected site.

Skeumorphism: Incorporating old fashioned and familiar ideas, images, machines, into new technologies.

Five Whys: When searching for the reason to a problem ask why it happened, then asked why that happened four more times. When do you stop the blame?

Design Thinking: Never solve the problem you are asked to solve. The problem you are asked to solve is never the root problem.

Shadowing: Follow customers into the shower to identify their real problems.

Time Stress: Most errors are caused by time stress.

Slip Error: Person intends to do one action but ends up doing something else.

Mistake Error: The wrong goal or wrong plan is established.

Combatting Errors: Minimize steps, contrast features, provide reminders, force functions (like removing card before receiving cash).

Checklists: Need to cover the essential but not too burdensome to perform.

Jidoka (Japanese): Automation with a human touch.

Poka-Yoke: Error proofing. Devices to constrain operations so they are done correctly.

Undo Command: Most powerful tool to prevent errors.

Swiss Cheese: Design redundancies to prevent the law of compounding errors. Requires you to identify all the possible mistakes that could be made and then install redundancies to make sure that they don’t all line up and compound.

Resilience engineering: How to help people cope with complexity under pressure.

Wizard of Oz: Set a customer up in a room to interact with your software. Have them ask it questions and instead of having the computer respond, have a person in the other room pretend to be the computer and respond. This will help teach you a lot about your user experience.

Number Five: Five is the ideal number of subjects to study for your test product.

Activity Centered Design: Focus on the greater activity, not the individual tasks a person is trying to accomplish.

Waterfall Method: Linear and sequential design that is hard to backtrack when decisions have been made.

Multiple Faculties: Make sure that people from all company divisions are involved in design process.

Complexity: Complexity is essential, confusion is undesirable.

Hidden: Hide all critical components that are not necessary for user. The ideal product should have one button.

Featuritis: Excessive features.

Different: The attempt to match competition is what makes all products the same. Focus on your users (“customer obsessed”), and throw off the competition.