mbr63

Mangesh Raut

Assignment 3

**DOET-03 Summary of Chapters 5-7**

**Chapter 5: Human Error? No, Bad Design**

The author told in this chapter about the real world most of the problems or accidents are caused by human error estimates range between 75-90%. But have you thought about it How and why? They are not because of the human it’s design problem. They must be another underlying factor. When an accident is thought to be caused by people, we blame them and continue to do things just as we always done.

5a. What is the difference between Norman’s two main categories of error: slips vs. mistakes? Discuss the types of error that fall into each category and how/why these occur. How do you design differently, to avoid each of the two categories of error (read the whole chapter to find out – Norman keeps picking up the topic again)?

Ans: **DEFINITIONS: ERRORS, SLIPS, AND MISTAKES**

Norman Said they divided the human error into two major categories. Slips and Mistakes. Slips occur when the goal is correct, but the required actions are not done properly, the execution is flawed. Mistakes occur when the goal or plan is wrong. Slips and mistakes can be further divided based upon their underlying causes. Memory lapses can lead to either slips or mistakes, depending upon whether the memory failure was at the highest level of cognition (mistakes) and lower level at (subconscious) level (slips). Mistakes are errors in choosing an objective or specifying a method of achieving it whereas slips are errors in carrying out an intended method for reaching an objective.

For example, a mistake would be to buy a Microsoft Excel licence because you want to store data that should be made accessible to web clients through SQL-queries, as Microsoft Excel is not designed for that purpose. In other word you choose wrong method for achieving your objective. However, if you installed a MangoDB or MySQL Server for the same reason but in your haste forgot to give the programme privileges to go through your firewall, that would be a slip. You chose the right method of achieving your objective, but you made an error in carrying out the method.

**SLIPS**

Slips have two major classes: action-based and memory-lapse. In action-based slips, the wrong action is performed. In memory-lapse, memory fails, so the intended action is not done or its result not evaluated.

Example of an action-based slip. I add milk into my Tea and then put the pot on stove to boil it. This is the correct action applied to their causes.

Example of memory-lapse slip. I forgot to turn of the lights while leaving the home in morning.

**MISTAKES**

Mistakes have three major classes: rules based, knowledge-based, memory-lapse. In rule-based mistake, the person has appropriately diagnosed the situation, but then decided upon an erroneous course of action: the wrong rule is being followed.

In a knowledge-based mistake, the program is misdiagnosed because of erroneous or incomplete knowledge. Memory-lapse mistakes take place when there is forgetting at the stages of goals, plans, or evaluation.

Example of knowledge-based mistake. Temperature of weather calculated in Fahrenheit instead of Celsius.

Example of memory-lapse mistake. A software tester failed to complete the test cases because of distraction.

Slips are most likely to occur (a) when we must deviate from a routine, and automatic processes inappropriately override international, controlled processes; or (b) when automatic processes are interrupted – usually as a result of external events or data, but sometimes as a result of internal events, such as highly distracting thoughts.

Mistakes occur in multiple ways: (a) The situation is mistakenly interpreted, thereby invoking the wrong goal or plan, inappropriate rule. (b) The correct rule is invoked, but the outcome is incorrectly evaluated. This error in evaluation, usually rule- or knowledge based itself, can lead to further problems as the action cycle continues.

**DESIGN LESSONS FROM THE STUDY OF ERRORS**

**Remove Memory Burdens** whenever users need to keep a lot of information in their STM while working on a task, they are vulnerable to slips where in mistakes where users forget earlier decisions they’ve already made, and repeat the process with different outcomes. A good strategy for preventing both of these types of errors is to remove burdens on user’s memory. Whenever possible, remove conditions that require users to keep information in their own memory while they move from ne step to another in complex. Multistep procedures. A good approach is to imagine that your user could be interrupted by a phone call after every step in a multistep process. You want to show of the information users need to readily resume their tasks after having been interrupted for several minutes.

**Confirm Before Destructive Actions** Designers typically focus on user tasks related to creation. But deleting also has to be straightforward. Remember, when users delete an item, they destroy something that had taken work to create. Before you finish removing the object of that hard work, make absolutely sure that the user really meant t delete by showing users a confirmation dialog.

**Support Undo** Another primary principle of preventing users from making errors is to acknowledge that they will make mistakes and slips from time to time and provide a safety net that makes these errors less costly. Nearly everyone has experienced the moment of realizing that you just accidentally deleted an entire folder or directory or quit the websites move the file or important documents, when you really only meant to delete only one file or close the particular tab. Providing the ability to undo the most recent action can help users to feel more secure and more confident to experiment with unfamiliar features, since they are aware that a mistake is low cost and can be fixed easily.

**Include Helpful Constraints** while its not always a good idea to limit user’s choice, in cases where there are clear rules that define acceptable option, it can be a good strategy to constrain the types of input users can make.

For example, booking a flight typically involves selecting the dates of travel and there are a few rules that govern which dates are acceptable. One of the major rules is that a return flight cannot happen before a departure. If the user is not available in these dates which are limited, they are feel free to change the dates for their flight that don’t follow the rules. A helpful constraint here will force users to pick a date range that fits.

**Offer Suggestions** similarly to how constraints guide users toward the correct use of an interface, suggestions can pre-empt many lips before the user has the opportunity to make them. On websites that offers thousands of products, search is an effective way of helping users find their proverbial needle in a haystack. However, typing can be inaccurate especially on touchscreens where there isn’t any haptic feedback. While you cannot prevent a user from making typos, you can prevent typos from turning into problems by offering contextual suggestions while the user types in the search bar.

For example, searching shoes if we start typing s letter then h it starts suggesting about that letter then start suggestions for categories.

5b. Suggest ways in which you might design systems to avoid (i) rule-based mistakes, (ii) knowledge-based mistakes, (iii) memory-lapse mistakes, and (iv) the “swiss cheese” model of how errors lead to accidents.

Ans: **(i) RULE-BASED MISTAKES,** we’re aware of the “rules” of how to act. Rules-based mistakes are difficult to avoid and then difficult to detect. Once the situation has been classified, the selection of the appropriate rule is often straightforward. Provide as much guidance as possible to ensure that the current state of things is displayed in a coherent and easily interpreted format ideally graphical. This is a difficult problem. All major decision makers worry about the complexity of real-world events, where the problems are often too much information much of it contradictory. Often decisions must be made quickly. Sometimes it isn’t even clear that there is an incident or that a decision is actually being made. If every decision had to be questioned, nothing would ever get done. But if decisions are not questioned, there will be major mistakes rarely, but often of substantial penalty. The design challenges are to present the information about the state of the system (a device manual, vehicle brief information, plant life, or activities being monitored) in a way that is easy to assimilate and interpret, as well as to provide alternative explanations and interpretations. It is useful to question decisions, but impossible to do if every action or failure to act require close attention. This is difficult problem with no obvious solution.

For example, the common tendency to set thermostats to extreme temperature in the hopes that it will heat or cool the room faster is an example of the first type of rule-based mistake, based on a faulty conceptual model of the heating system. First you have to get learned the temperature of inside and outside so you can manage the thermostats.

**(ii) knowledge-based mistakes,** when we have incomplete or incorrect information to work with, leading us to misunderstand the situation and therefore respond inappropriately. Knowledge-based mistakes usually happen in novel situations where we have no applicable prior knowledge to pull from.

The best solution to knowledge-based situations is to be found in a good understanding of the situation, which in most cases also translates into an appropriate conceptual model. In complex cases, help is need, and here is where good cooperative problem-solving skills and tools are required. Sometimes, good procedural manuals (paper or electronic) will do the job, especially if critical observations can be used to arrive at the relevant procedures to follow. A more powerful approach is to develop intelligent computer systems, using good search and appropriate reasoning techniques (AI, decision making, and problem solving). The difficulties here are in establishing the interaction of the people with the automation: human teams and automated systems have to be thought of as collaborative, cooperative systems. Instead, they are often build by assignments the tasks that machines can do the machines and leaving the humans to do the rest. This usually means that machines do the parts that re easy for people, but when the problems become complex, which is precisely when people could use assistance, that is when the machines usually fails.

**(iii) memory-lapse mistakes,** The design cures for memory-lapse mistakes are the same as for memory-lapse slips: ensure that all the relevant information is continually available. Far too many designs eliminate all signs of these items once they have seen made or acted upon. Once again, the designer should assume that people will be interrupted during their activities and that they may need assistance in resuming their operations. A common cause of the lapse is an interruption that leads to forgetting the evaluation of the current state of the environment. These lead to mistakes, not slips, because the goals an plans become wrong. Forgetting earlier evaluations often means remaking the decision, sometimes their erroneously. It occurs by forgetting the evaluation of the current status of the environment by getting an interruption. environment: The goals, plans, system status

For example: You forget to finish the task in progress because you got distracted and carried away by an Instagram notification.

**(iv) the “swiss cheese” model of how errors lead to accidents,**

Add more slices of cheese, Reduce the number of holes (or make the existing holes smaller), Alert the human operators when several holes have lined up. Each of these has operational implications. More slices of cheese means more lines of defence, such as the requirement in aviation and other industries for checklist, where one person reads the items, another does the operation, and the first person checks the operation to confirm it was done appropriately. Reducing the number of critical safety points where error can occur is like reducing the number or size of the holes in the swiss cheese. Properly designed equipment will reduce the opportunity for slipes and mistakes, which is like reducing the number of holes and making the ones that remain smaller. This is precisely how the safety level of commercial aviation has been dramatically improved. Design redundancy and layers of defence: that’s swiss cheese. The metaphor illustrates the utility of trying to find the one underlying cause of an accident (usually some person) and punishing the culprit. Instead, we need to think about systems, about all the interacting factors that leads to human error and then to accidents and devices ways to make the systems as a whole, more reliable.

5c. Giving your own examples, explain what you might do to apply Norman’s three key design principles (at the end of Chapter 5).

Ans: There are 6 design principles of Don Norman’s for the system design.

Affordance is a term used to refer to an attribute of an object that allows how to use it. For example touch bar on apples mac laptop they or touch screen laptop first time its to touch see its working or not and we are not comfortable with the first time touching the screen. But we use touch screen smartphone which is easily we use daily routine. A touch bar on MacBook Pro is there for sliding selecting other options by the way its physically constrained in its display part. At a very simple level to afford means to give a clue. When the affordance of a physical object are good it is easy to know how to interact with it.

Consistency is the thing which is same we are doing for all the time like the designing interface to have similar operations and use similar elements for achieving similar tasks. In some time consistent interface is one that follow rules such as using the same operations select all objects. For example the power on button when use to start the computer or laptop we press it that is consistent for all the time. Consistent operation is using the same input action to highlight any graphical object at the interface.

Mapping is the relationship between controls and their effects in the world. In the world because its common for all over the world like everyone knows this thing. Like the volume up and down button which is used to for increasing volume we use up arrow and down for decreasing the volume respectively on a computer keyboard. This is a good mapping between control and effect. Same examples like brightness and for arrows up left right.

Constraints the design which is restricting the kind of user to interact with the system like the mouse arrow on screen it will not gone out of the screen it will move in the area of the screen only. This is example of physical constraint. Logical constraints are if there are two USB cables are there one for printer and one is for scanner so scanner USB will work for that only and printer USB should work for printer purpose only. Semantic constrains are if we use tablet it support touch right because tablets are big screen it attach the keyboard but the semantic are made only for touch purpose. Cultural constraints are if we press down arrow it goes down only and for up it goes up because in culture up is good and down is bad this things are happen in cultural constraints that follows in system design also.

Feedback is about the review of from the person who gives information after using that thing. What action is done and what has been accomplished allowing the person to continue with the activity. the presence of sound can serve a useful role in providing feedback about events, the absence of sound can lead to the same kinds of difficulties we have already encountered from a lack of feedback. The absence of sound can mean an absence of knowledge, and if feedback from an action is expected to come from sound, silence can lead to problems.

Visibility is the view of the system you can say like the more visible functions are the more likely users will be able to know about it like the post on Instagram Facebook. Incontrast when functions are out of the window or that area it is difficult to see then know how to use that. Instrgram is the longest scrolling stream visual content because it is more visible to user.

This are the design principles Don Norman gives in his book these all are helpful to make system design more conceptual and interact with the user friendly with the system design.

Reference for chapter 5

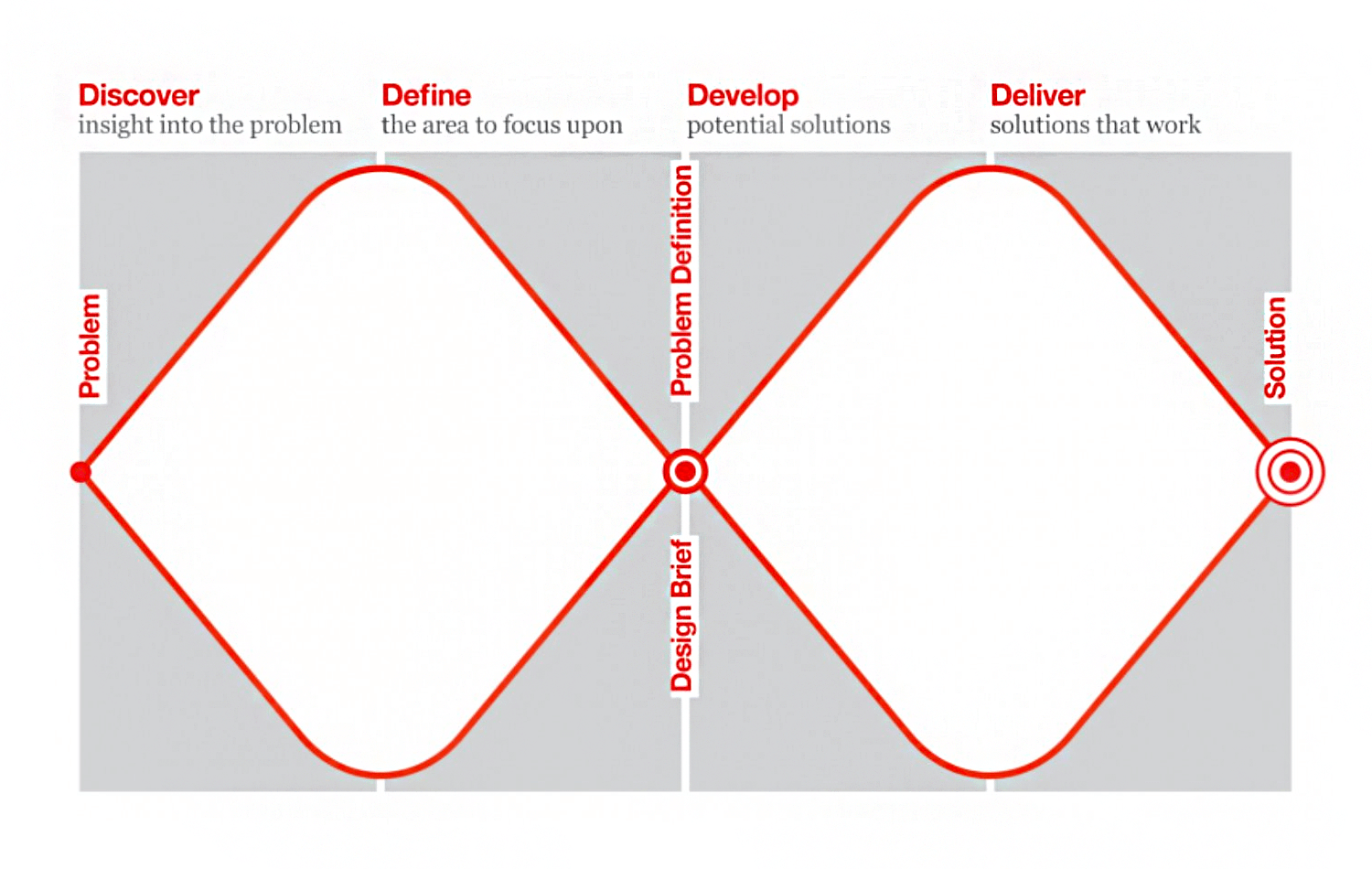
<https://www.nngroup.com/articles/user-mistakes/>

**Chapter 6: Design Thinking**

The author Norman explains how designers are never really supposed to solve the problem the customers identifies. Instead, he urges one to take this problem merely as suggestion. The customer’s problem is only a symptom of the central issue at hand. Much like doctor, the designer has a responsibility to acquaint themselves with all symptoms that may exits, and ultimately make a diagnosis after close observation. In terms of design, the process only starts once the true desires needs and capabilities of the customer are given a proper diagnosis. Norman remarks how this is something that often takes a considerable amount of time.

6a. Explain the “Double Diamond” model of design, shown in Figure 6.1. You may wish to Google this model to see how it is implemented in practice, as it underlies the recent trend towards “design thinking.” What should designers do, to base their design practice on this model?

Ans: **The Double-Diamond Model of Design.** Start with an idea, and through the initial design research, expand the thinking to explore the fundamental issues. Only then is it time to converge upon the real, underlying problem. This model has four stages: Discovery, Definition, Development and Delivery. Together, these stages works as a map designers can use to organize their thoughts in order to improve the creative process.



Discover and Define for the divergence and convergence phases of finding the right problem and develop and deliver for the divergence and convergence phases of finding the right solution.

**Discovery**: gathering data. The very first stage of the Double Diamond model consists of learning more about the different variables that affects the problem and its possible solution. It’s common for companies to start this process by laying down their problem, presenting their hypothesis, and defining ways they can learn more. The objective of this stage within the Double Diamond model is to identify and contextualize the actual problem or opportunity. Think of it like an expedition to map out the playing field. The study found that keeping employee’s minds open to all possible solutions was crucial to ensure maximum creative problem solving.

**Definition**: filtering through data and sign-off. After gathering all that data in the first stage of the Double Diamond model, you come to the definition stage. The definition stage in this Double Diamond model consists of filtering through all the information you got from stage. One, and elaborating on it. This can mean identifying bottlenecks or resource waste, seeing hidden opportunities or setting a list of things the design team shouldn’t do. This model represents a filtering of the ideas and data taken from stage one. It also sets the context for product development, assesses the realism of what can be done and analyzes how this project agrees with the corporate brand.

**Development**: start to design. The development stage involves a lot multi-disciplinary work – putting the designer together with internal partners, such as engineers, developers, or other departments that have practice needed in this project. A big win from this aspect of the development stage is that by putting different departments together, you speed up problem-solving. This model represents that all companies engage in development methods as you might have guessed. Each of the companies in the study had their own preferent methods. Brainstorming, visualization, making of different scenarios, and so on. An interesting detail to note here: the study points out that continuous testing and feedback are common throughout the development stages in the Double Diamond model.

**Delivery**: test and release. The last stage of the Double Diamond model includes the final testing of the product, official sign-off to production and launching. The final testing is that is that one last look at the product to make sure there are no issues with it. This usually includes testing it against any regulation and legal standards, damage testing and/or compatibility testing. Companies also use this stage to assess the impact of the design on customer satisfaction, in order to quantify the value of good design for the brand.

Designers often start by questioning the problem given to them: they expand the scope of the problem, diverging to examine all the fundamentals issues that underline it. Then they converge upon a single problem statement. During the solution phase of their studies, they first expand the spaces of possible solutions, the divergence phase. The double diverse-converge process is quite effective at freeing designers from unnecessary restrictions to the problem and solution spaces. But you can sympathize with a product manage who, having given the designers a problem to solve, finds them questioning the assignment and insisting on travelling all over the world to seek deeper understanding. Even when the designers start focusing upon the problem, they do not seem to make progress, but instead develop a wide variety of ideas and thoughts, many only half-formed, many clearly impractical. All this can be rather unsettling to the product manager who, concerned about meeting the schedule, want to see immediate convergence.

To add to the frustration of the product manager, as the designers start to converge upon a solution, they may realize that they have inappropriately formulated the problem, so the entire process must be repeated (although it can go more quickly this time).

6b. What should designers do at each stage of the Iterative Cycle of Human-Centered Design, shown in Figure 6-2? How does this relate to the double-diamond model - and why would anyone care about using either model of design?

Ans: **The Iterative Cycle of Human-Centered Design**. Make observations on the intended target population, generate ideas, produce prototypes and test them. Repeat until satisfied. This is often called the spiral method (rather than the circle depicted here), to emphasize that each iteration through the stages makes progress.

1. Observation

2. Idea generation (ideation)

3. Prototyping

4. Testing

These four activities are iterated; that is, they are repeated over and over, with each cycle yielding more insights and getting closer to the desired solution. Now let us examine each activity separately.

**OBSERVATION:** The first phase is all about observing the end-user, learning, and being open to creative possibilities. Your goal is to understand the people you’re designing for. Identify patterns of behaviour, pain points, and places where users have a difficult time doing something—these all lend to tremendous opportunity. If you can, put yourself in their situation so you can see what their experience is, and feel what they feel. The design researcher will go to the potential customers, observing their activities, attempting to understand their interests, motives, and true needs. The problem definition for the product design will come from this deep understanding of the goals the people are trying to accomplish and the impediments they experience.

**IDEA GENERATION:** In this phase, you start brainstorming ideas with your team based on what you learned from your observations and experiences in Observation. Your goal is to come up with as many ideas as you can. As you’re coming up with ideas, stay focused on the needs and desires of the people you’re designing for. If you do this, your group’s ideas will eventually evolve into the right solution. This is the fun part of design: it is where creativity is critical. There are many ways of generating ideas: many of these methods fall under the heading of “brainstorming.” Whatever the method used, two major rules are usually followed: Generate numerous ideas, Be creative without regard for constraints, Question everything.

**PROTOTYPING:** In this phase, you’re going to quickly build a simple prototype of your idea. This makes it tangible and gives you something to test with the end-user. Don’t try to build a fancy high-fidelity prototype right now. IDEO is notorious for creating simple prototypes made out of cardboard. Ask yourself this: What can I spend the minimum amount of time building that will allow me to get user feedback as quickly as possible? The purpose of this phase isn’t to create the perfect solution; it’s to make sure your solution is on target.

Prototyping during the problem specification phase is done mainly to ensure that the problem is well understood. If the target population is already using something related to the new product, that can be considered a prototype. During the problem solution phase of design, then real prototypes of the proposed solution are invoked.

**TESTING:** In this phase, gather a small group of people who correspond as closely as possible to the target population—those for whom the product is intended. Have them use the prototypes as nearly as possible to the way they would actually use them. If the device is normally used by one person, test one person at a time. If it is normally used by a group, test a group. The only exception is that even if the normal usage is by a single person, it is useful to ask a pair of people to use it together, one person operating the prototype, the other guiding the actions and interpreting the results (aloud). Using pairs in this way causes them to discuss their ideas, hypotheses, and frustrations openly and naturally. The research team should be observing, either by sitting behind those being tested (so as not to distract them) or by watching through video in another room (but having the video camera visible and after describing the procedure). Video recordings of the tests are often quite valuable, both for later showings to team members who could not be present and for review.

**ITERATION**

The role of iteration in human-centered design is to enable continual refinement and enhancement. The goal is rapid prototyping and testing, or in the words of David Kelly, Stanford professor and cofounder of the design firm IDEO, “Fail frequently, fail fast.”

The double diamond model clearly conveys a design process to designers and non-designers alike. The two diamonds represents a process of exploring an issue more widely or deeply. Same in human cenetered designers use a mixture of investigate methods and tools and generative ones to develop an understanding of user needs these things are relate to each other.

**Design: Developing Technology for People.** Everyone care for this design because this Design is a marvellous discipline, bringing together technology and people, business and politics, culture and commerce. The different pressures on design are severe, presenting huge challenges to the designer. At the same time, the designers must always keep foremost in mind that the products are to be used by people. This is what makes design such a rewarding discipline: On the one hand, woefully complex constraints to overcome; on the other hand, the opportunity to develop things that assist and enrich the lives of people, that bring benefits and enjoyment.

Reference for chapter 6

<https://www.justinmind.com/blog/double-diamond-model-what-is-should-you-use/>

<https://www.usertesting.com/blog/how-ideo-uses-customer-insights-to-design-innovative-products-users-love>

**Chapter 7: Design in the World of Business**

In this chapter Norman talks about the world of business with the design like the realities of the world imposes severe constraints upon the design of products. He have described the ideal case, assuming that human-centered design principles could be followed in a vacuum; that is, without attention to the real world of competition, costs, and schedules. Conflicting requirements will come from different sources, all of which are legitimate, all of which need to be resolved He conclude with reflections about the history and future prospects of this book.

7. Don Norman spends a lot of words attacking “featuritis,” by which he means adding features (and complexity) to product designs without considering the impact on usability, ease-of-learning, or error-avoidance. Explain how you would go about designing “things that make us smart” rather than systems that get in the way of us doing things seamlessly, easily, and without errors. Think about what you have learned from reading this book – use ideas from other chapters to supplement what Norman discusses in chapter 7.

Ans: In Things That Make Us Smart, Donald A. Norman explores the complex interaction between human thought and the technology it creates, arguing for the development of machines that fit our minds, rather than minds that must conform to the machine. Humans have always worked with objects to extend our cognitive powers, from counting on our fingers to designing massive supercomputers. But advanced technology does more than merely assist with thought and memory—the machines we create begin to shape how we think and, at times, even what we value. Norman, in exploring this complex relationship between humans and machines, gives us the first steps towards demanding a person-centered redesign of the machines that surround our lives.

Featuritis (aka “feature creep”) is a phenomenon whereby a successful product keeps adding new features and capabilities to the point that it overcomplicates and obfuscates what was once a simple, elegant, straight-forward solution. Reasons for this could be; 1) Existing customers want more features, functionality and capabilities. 2) Competing companies add new features that create pressures to match those offerings. 3) Market is saturated or stagnant. Adding new enhancements will boost the upgrade cycle.

By referring book, and Normans idea, as people read books, but they cannot learn maximum but by watching videos and using internet tools make them smarter as they can use there brain to establish or design anything. Here many people make their own design for their work. In ancient times, when there were no computers, people used to do their work by own, they try to think hard to achieve the goal. But now all this growing technology stop human thinking much, as they enlarge in the computers and electronics tools to make their work easier and smarter.

As per my understanding of the relationship between technology and cognitive ability will be instrumental in future attempts to improve learning. In a more general and indirect way, Norman's book inspires many questions related to learning and technology.

If calculators can compute more efficiently and effectively that can a human, do children need to spend as much time learning these skills or can we simply do away with teaching long division, and multiplication tables? So I do agree with that when there were no calculators kids go to academy to learn fast calculations skills but now children’s are not taking that much efforts to do calculations by their own.

Technology will affect how and what we learn. And some of this new learning will, in turn, affect the evolution of technology. As the two interact, they will adapt to and change each other in complex and likely unpredictable ways.

One more point to notice, by writing notes/letter by notebook instead to type in computers is harder, because system is fast and by keywords it’s easy to make an electronic mail and easy to share with respective person. Designing videophone like virtual conference call in companies and universities, facetime, video camera all took 40 years to reach here. So, computers makes us smart in all things, it makes projects, studies, etc things easy in their fast paced life. Design is successful only if the final product is successful — if people buy it, use it, and enjoy it, thus spreading the word. A design that people do not purchase is a failed design, no matter how great the design team might consider it.