## A report on the background research and literature search

“Everyday tasks are not difficult because of their inherent complexity. They are difficult only because they require learning arbitrary relationships and arbitrary mappings, and because they sometimes require precision in their execution. The difficulties can be avoided through design that makes obvious what operations are necessary. Good design exploits constraints so that the user feels as if there is only one possible thing to do – the right thing of course… We do not have to experience confusion or suffer from undiscovered errors. Proper design can make a difference in our quality of life” – page 216

* Bad design can cause frustration, confusion to the user.
* Bad design will affect the usage of a product
* Repositories are things used on a daily basis therefore they have to be made simple and easy to use and understand – good design must be provided following the principles for such.
* Conscious and subconscious behavior have a good influence on this
* A bad design example – coins or ovens or switches (see folder problems – placement of switches, which switch control which function – page 92 to 98)
* This is due to….
  + User errs a lot
  + Designing troubles – designers aren’t users
  + Making things complex
* “It is not easy to develop effective and usable computer systems. For one thing, it is expensive. Consider the principles described in this book: visibility, constraints, affordances, natural mappings, feedback”. – page 181
* Solution – providing a good design
* Summary – at the end

**Design of everyday things:**

* **Problems**
  + **Conscious and Subconscious behavior (page 125)**
    - “Much human behavior is done subconsciously, without conscious awareness and not available to inspection.”
    - “Subconscious thought matches patterns….It proceeds rapidly and automatically, without effort.”
    - “Conscious thought is quite different. It is slow and labored.”
  + **Coins bad design example**
    - Confusions caused
    - Not clear design
    - Very similar to something that already exists
    - “The confusions probably occurred because the users of coins formed representations in their memory systems that were sufficiently precise only to distinguish among the coins that they actually had to use.”
    - Only partial descriptions were memorized (round, small, silver)
    - “The descriptions formed to distinguish among the old coins were not precise enough to distinguish between the new one and at least one of the old ones”

## Designing troubles – designers aren’t users

* + **The Foibles of Computer Systems (page 177)**
    - **“**Designers of computer systems seem particularly oblivious to the needs of users, particularly susceptible to all the pitfalls of design… No expertise in designing for people. **”**
  + **Designers are not typical users (page 155)**
    - **“**As human beings, we have access to our conscious thoughts and beliefs but not to our subconscious ones. We tend to project our own rationalizations and beliefs onto actions and beliefs of others.**"**
  + **Designing for special people (page 161)**
    - **“**There is no such thing as the average person. This poses a particular problem for the designer, who usually must come up with a single design for everyone**”.**
  + **How to do things wrong? (page 178)**
    - **“**The special powers of the computer can amplify all the usual problems to new levels of difficulty.**” – the tyranny of the blank screen**
    - e.g. Browsing for files through the categories imposed with no visual aid.
    - Make things invisible
    - Be arbitrary (bad naming)
    - Be inconsistent
    - Be impolite
    - Make operations unintelligible
    - Make operations dangerous (e.g. rm\*).
    - “Programs and systems do exist that have shown us the potential; they take the user into account, and they make it easier for us to do our tasks – pleasurable, even. This is how it ought to be”.

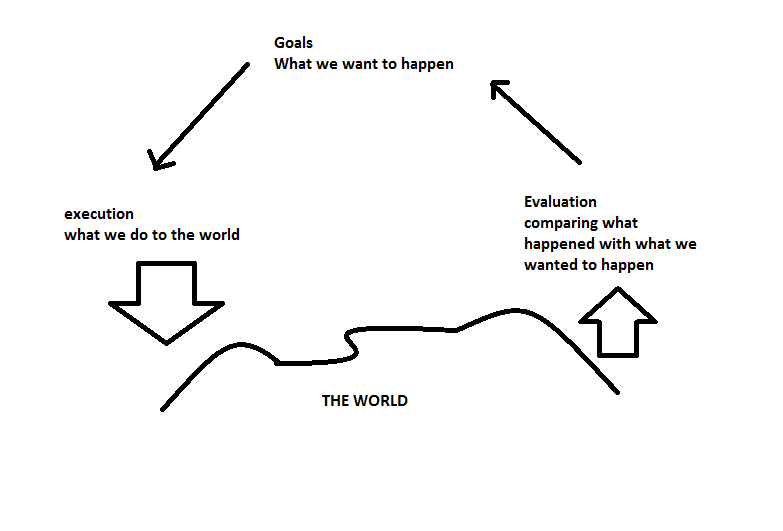
## Making things complex

* + **Forces that work against evolutionary design (page 142)**
    - “Modern designers are subject to many forces that do not allow for the slow, careful crafting of an object over decades and generations. Most of today’s items are too complex, with too many variables, for this slow sifting of improvements. But simple improvements ought to be possible”.
  + **Two Deadly Temptations (pages 172 to 174)**

Two deadly temptations that lead towards products that are overly complex, products that drive users to distraction.

* + - Creeping futurism
      * “.. . the tendency to add to the number of features that a device can do, often extending the number beyond all reason.”
      * Making it too complicated and complex for no reason.
    - The worshiping of false images
      * Complex and difficult people worship complex systems
      * “The designer – and the user – may further be tempted to worship complexity”.
  + **The paradox of technology (page 31)**
    - “Whenever the number of functions and required operations exceeds the number of controls, the design becomes arbitrary, unnatural, and complicated. The same technology that simplifies life by providing more functions in each device also complicates life by making the device harder to learn, harder to use. This is called the paradox of technology.”
    - “The paradox of technology should never be used as an excuse for poor design… the principles of good design can make complexity manageable”.
  + **Grouping design problem (page 92)**
    - **“**Basic switches and controls should be relatively simple to design well. But there are two fundamental difficulties. The first is the grouping problem, how to determine which switch goes with which function.**”**
  + **Pity the poor designer (page 28)**
    - Thankfully this work is for free done by university students like myself- one more reason why this project is a good idea.
    - “The manufacturer wants something that can be produced economically. The store wants something that will be attractive to customers. The purchaser has several demands… Nonetheless, the designer may be able to satisfy everyone”.

## User errs a lot

* + **User execution stages (page 45-49)**
    - **Goal**
      * Forming the goal
    - **Execution**
      * Forming the intention
      * Specifying an action
      * Executing the action
    - **Evaluation**
      * Perceiving the state of the world
      * Interpreting the state of the world
      * Evaluating the outcome
  + **Probability to err (page 36)**
    - **“**If an error is possible, someone will make it. The designer must assume that all possible errors will occur and design so as to minimize the chance of the error in the first place, or its effects once it gets made**”.**
  + **Shallow and Narrow structures (page 121)**
    - **Shallow Structures (e.g. chess – page 119 – wide and deep structures)**
      * “There are many alternative actions, but each is simple; there are few decisions to make after the single top-level choice. The major problem is to decide which action to do. Difficulties arise from competing alternatives, not from any prolonged search, problem solving, or trial and error. ”
    - **Narrow Structures**
      * “A narrow structure arises when there are only a small number of alternatives, perhaps one or two. If each possibility leads to only one or two further choices, then the resulting tree structure can be said to be narrow and deep”.
  + **The gulfs (page 51)**
    - **The gulf of execution**
      * “The difference between the intentions and the allowable actions is the Gulf of execution. One measure of this gulf is how well the system allows a person to do the intended actions directly, without extra effort..” – there is a lot of effort to browse for files in EdShare.
    - **The gulf of evaluation**
      * “The gulf of evaluation reflects the amount of effort that the person must exert to interpret the physical state of the system and to determine how well the expectations and intentions have been met. The gulf is small when the system provides information about its state in a form that is easy to get, is easy to interpret, and matches the way the person thinks of the system”.
  + **Description errors (page 108)**
    - Occur most frequently when the wrong and right objects are physically near each other (share and search).
    - Likely to happen under stress/no full paying attention.
    - Mistakes involving misinterpretation
    - Can take long time to be discovered

## Solution – providing a good design for understandability and usability

* + **Principles of Design for Understandability and Usability (page 13)**

Provide a good conceptual model

* + - **Conceptual models (page 12)**
      * “How do people cope? Part of the answer lies in the way the mind works – in the psychology of human though and cognition. And part comes from the ability of the designer to make the operation clear, to project a good image of the operation, and to take advantage of other things people might be expected to know.”
      * “You can do the simulation because the parts are visible and the implications clear. Other clues to how things work come from their visible structure – in particular from affordances, constraints, and mappings”.
      * “A good conceptual model allows us to predict the effects of our actions. Without a good model we operate by rote, blindly; we do operations as we were told to do them.” (page 179)
      * “…mental models, the models people have of themselves, others, the environment, and the things with which they interact. People form mental models through experience, training, and instruction. The mental model of a device is formed largely by interpreting its perceived actions and visible structure. I call the visible part of the device the system image. When the system image is incoherent or inappropriate, as in the case of the refrigerator, then the user cannot easily use the device. If it is incomplete or contradictory, there will be trouble”. – page 17
      * Three aspects of mental models – page 190

Make things visible

* + - **Visibility**
      * **Using aesthetics first (page 151)**
        + **“**If everyday design were rules by aesthetics, life might be more pleasing to the eye but less comfortable; if rules by usability, it might be more comfortable but uglier. If cost or ease of manufacture dominated, products might not be attractive, functional, or durable. Clearly, each consideration has its place. Trouble occurs when one dominates all the others**”.**
      * **Macintosh example (page 182-183)**
        + “The deign emphasizes visibility and feedback. Its “human interface guidelines” and its internal “toolbox” provide standards for the many programmers who design it. It has emphasized consideration for the user”. **– one kind of software, all follows a template so user has experience on it when using a new software.**
      * **Making visible the invisible (page 100)**
        + “In numerous designs crucial parts are carefully hidden away.”
      * “Visibility indicates the mapping between intended actions and actual operations. Visibility indicates crucial distinctions…” – page 8
    - **Mappings**
      * **The principle of mapping (page 23)**
        + “Mapping is a technical term meaning the relationship between two things, in this case between the controls and their movements and the results in the world.”
        + “The mapping is easily learned and always remembered” – visibility + feedback.
        + “Natural mapping, by which I mean taking advantage of physical analogies and cultural standards, leads to immediate understanding”.
        + “Mapping problems are abundant, one of the fundamental causes of difficulties”.
        + E.g. burners example.
    - **Feedback**
      * **The principle of feedback (page 27)**
        + “Feedback – sending back to the user information about what action has actually been done, what result has been accomplished”.
        + Supply a good visual display.
      * **Visibility and feedback (page 99)**
        + “Visibility. Make relevant parts visible. Feedback. Give each action an immediate and obvious effect.” – Same questions are used for both.
    - **Constraints**

“Design is the successive application of constraints until only a unique product is left” – page 158

Constrain possible operations.

* + - * **Affordances (page 9)**
        + “…affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used.”
        + “Affordances provide strong clues to the operations of things.”
        + E.g. a chair affords support, and therefore, affords sitting.
      * **A classification of everyday constraints**
        + “The thoughtful use of affordances and constraints together in design lets a user determine readily the proper course of action, even in a novel situation.”
        + E.g. Lego motorcycle built.
      * **Physical constraints (page 84)**
        + “The value of physical constraints is that they rely upon properties of the physical world for their operation; no special training is necessary” – limit the number of possible actions.
        + They must be easy to interpret and easy to see in order to be useful.
        + **Forcing functions (page 132)**

“Forcing functions are a form of physical constraint: situations in which the actions are constrained so that failure at one stage prevents the next step from happening.”

E.g. Starting a car.

* + - * **Semantic constraints(page 85)**
        + “Semantic constraints rely upon the meaning of the situation to control the set of possible actions…. Semantic constraints rely upon our knowledge of the situation and of the world. Such knowledge can be powerful and important clue.”
        + E.g. Sitting in a motorbike
      * **Cultural constraints (page 85)**
        + **“**Some constraints rely upon accepted cultural conventions, even if they do not affect the physical or semantic operation of the device**”.**
        + “Each culture has a set of allowable actions for social situations”.
        + E.g. signs are meant to be read.
      * **Logical constraints (page 86)**
        + “Natural mappings work by providing logical constraints. There are no physical or cultural principles here; rather there is a logical relationship between the spatial or functional layout of components and the things that they affect or are affected by”.
    - **The seven stages of Action as design aid (page 52)**
      * “The seven-stage structure can be a valuable design aid, for it provides a basic checklist of questions to ask to ensure that the Gulfs of Evaluation and Execution are bridged”.
        + Visibility: by looking, the user can tell the state of the device and the alternatives for action.
        + A good conceptual model: The designer provides a good conceptual model for the user, with consistency in the presentation of operations and results and a coherent, consistent system image. – three aspects of mental models (page 190)
        + Good mappings: It is possible to determine the relationships between actions and results, between the controls and their effects, and between the system state and what is visible.
        + Feedback: The user receives full and continuous feedback about the results of actions.

## Design philosophy – summary at end – page 140

“But the philosophy of user-centered system design still holds. Think of the user’s point of view. Assume that every possible mishap will happen so protect against it. Make actions reversible. Try to make them less costly.”

* Seven principles for transforming difficult tasks into simple ones (page 188-189)
  1. Use both knowledge in the world and knowledge in the head
  2. Simplify the structure of tasks (page 191 -208)
     + Keep the task much the same, but provide mental aids.
     + Use technology to make visible what would otherwise be invisible, thus improving feedback and the ability to keep control.
     + Automate, but keep the task much the same
     + Change the nature of the task (when the task seems inherently complex because of the manual skill required)
     + Don’t take away control (automation has its virtues, but can be dangerous. No over automation.)
     + Make things visible: bridge the gulfs of Execution and Evaluation.
     + Exploit the power of constraints: both natural and artificial.
     + Design for error
     + When all else fails, standardize
       - The timing of standardization : Standardize and you simplify lives: everyone learns the system only once.
     + Note that easy looking is not necessarily easy to use – aesthetic first problem.
  3. Make things visible: bridge the gulfs of Execution and evaluation
  4. Get the mappings right
  5. Exploit the power of constraints
  6. Design for error
  7. When all else fails, standardize
* Put the required knowledge in the world. Don’t require all the knowledge to be in the head. – Knowledge with experience and guidelines for usage of software.
* Use the power of natural and artificial constraints: physical, logical, semantic, and cultural. Use forcing and natural mappings.
* Narrow the gulfs of execution and evaluation. Make things visible, both for execution and evaluation. On the executions side, make the options readily available. On the evaluation side, make the results of each action apparent. Make it possible to determine the system state readily, easily, and accurately, and in a form consistent with the person’s goals, intentions, and expectations.
* Hence, Make the system explorable – invite for experimentation! Encourage the user to experiment and learn the possibilities through active exploration. – page 183.
* The answers should be given by the design, without the need for trial and error. – page 3

## Rocket surgery made easy

Page 13 - Usability testing definition: “Watching people try to use what you’re creating/designing/building (or something you’ve already created/designed/built), with the intention of (a) making it easier for people to use or (b) providing that it is easy to use”.

* It is not a type of quantitative testing “… you’re interested in proving something… and you do this by measuring things like success rate and time-on-task”.
* It is a type of qualitative testing “The purpose isn’t to prove anything; it’s to get insights that enable you to improve what you’re building”.
* Different from surveys, interviews etc in the sense that in these you are asking people for their opinions about things, or their past experiences using things.

**Why does it work? – page 16 and 17**

* All sites have problems
* Most serious problems tend to be easy to find
* Watching users makes you a better designer

**Why so little of it gets done? – page 18 & 32**

* Lack of time
* A lot of work (sometimes we are already swamped with other work)
* Reluctance to show our work before it is finished
* We don’t have enough done yet
* It’s too rough
* Why waste people’s time looking at something we know we’re going to change.

**Demo – see video and script online (**[**www.rocketsurgerymadeeasy.com**](http://www.rocketsurgerymadeeasy.com)**)**

**Plan (page 23-29):**

* “A morning a month, that’s all we ask.” – three users
  + One morning per month
  + Continuous testing
  + Three participants
  + Recruit loosely – frequent testing more important than testing actual users
  + Held on-site
  + 1-2 page email summarizes decisions
  + Decisions can me discusses in the same day
  + Cheaper
* “the big Honkin’ test”
  + Made nearly the end
  + 1-2 days
  + 10 participants
  + One or two per project
  + Target audience recruited
  + **Held off-site**
  + **At least a week to prepare it**
  + **Person running the tests analyses results and recommends changes**

“Ah, yes. Agile. Given the short cycles in an Agile environment, if you wait a month the world will have passed you by. Perhaps it’s more like “A morning a sprint, that’s all we ask””. – less users, more often, use incentives

**What and when? – page 31-37**

* “Start earlier than you think makes sense.”
* Test sketches, wireframes, prototypes, everything!
* Test your site and other people’s sites.
* Test sketch on napkin – page 35 (now)
* Test wireframes – page 36 (soon)
* Test page designs – page 37 (Jan)
* Test working prototypes and beyond – page 37

**Who to recruit? – page 40-49**

* Real users, target audience, users of the system, representative users.
* Recruit loosely – beginners etc. they might give you insights on navigation, page layout, visual hierarchy etc. Diverse audience you want!
* “Recruit loosely and grade on a curve” – try to find users who reflect your audience but don’t get hung up about it.
* Put out an invitation :
  + Bulletin boards
  + Message boards
  + Email
  + Link on home page
  + Pop-up invitation
  + Craigslist
* Don’t use the same participants in later rounds – they know too much.

**Forming the test (page 51-55)**

* List of tasks
* Expand tasks to scenarios

**Check lists (page 58-61) – to not lose track of everything that has to be done prior the usability tests**

**Fixing things based on the tests (page 111 - 199)**

* “when fixing problems, try to do the least you can do”
* Tweak, don’t redesign
* Take something away