The Design of Design, Part One

CSCI 5828: Foundations of Software Engineering
Lecture 20 — 10/30/2014

© Kenneth M. Anderson, 2014

Goals

- Cover material from Fred Brooks's The Design of Design, Chapters 1-3
  - The Design Question
  - How Engineers Think of Design — The Rational Model
    - What's Wrong with this Model?

The Design of Design

- A book by Fred Brooks, published in 2010
  - A series of “opinionated essays” about design
  - Brooks has designed “in five media across six decades”
    - computer architecture, software, houses, books, and organizations
  - The target audience of this book is
    - designers (weighted towards designers of software and computer hardware), design project managers, and design researchers

Brooks's Writing Style

- You may find that it takes a bit of work to get used to Brooks's style of writing
  - I find it delightful
- But it requires thought and careful reading
  - "Systematic designs excluding intuition yields pedestrian follow-ons and knock-offs; intuitive design without system yields flawed fancies."
  - "Partitioning a design task is itself an added task. The crisp and precise definition of the interfaces between subtasks is a lot of work, slighted at peril."
- I'll help you tease meaning and context and background out of this prose

The Design Question (I)

- The structure of Brooks's first essay is
  - What is the fundamental question about (thinking about) design?
  - What is Design?
  - What's Real about Design? The Design Concept
    - What's the value of having a design concept
  - Thinking about Design Process
  - Kinds of Design

© Kenneth M. Anderson, 2014

The Design Question (II)

- Chapter 1 begins with two interesting quotes; One from Francis Bacon, 1605
  - “[New ideas would come about] by a connexion and transferring of the observations of one Arte, to the uses of another, when the experience of several misteries shall fall under consideration of one mans minde.”
- He’s saying that we can learn a lot by taking lessons learned from one discipline and looking to see if we can apply them in another
- The second from Herbert Simon in The Sciences of the Artificial, 1969
  - “Few engineers and composers… can carry on a mutually rewarding conversation about the content of the other’s professional work. What I am suggesting is that they can carry on such a conversation about design, … [and then] begin to share their experiences of the creative professional design process”
- We might not be able to talk about the specifics of our fields, but we can talk about design and problem solving

© Kenneth M. Anderson, 2014

The Design Question (III)

- Why these quotes?
  - They set-up Brooks's design question
    - Are there invariant properties of the design process itself?
      - Properties that hold true across a wide range of media and fields?
  - If so, different designers will have different insights into these properties and we would be able to learn a lot about design itself by sharing experiences across domains

© Kenneth M. Anderson, 2014

What is Design?

- Calling on the Oxford English Dictionary, Brooks identifies these aspects
  - To form a plan in the mind for later execution
  - A design is a created object that is used to help create another object (!)
- He then cites Dorothy Sayers's book The Mind of the Maker
  - She talks about the creative process as including three aspects
    - **Idea**: the formulation of conceptual constructs <== ***the design***
    - **Energy**: Implementation in real media
    - **Interaction**: Interactivity with users in real time
  - When users finally interact with the product, they interact with the "mind of the maker"
- Mozart chaser: "Everything has been composed, just not yet written down"

© Kenneth M. Anderson, 2014

The Design Concept

- Similar to what I called “design themes” in Lecture 19, Fred Brooks talks about the difference between a design and its implementation(s)
  - He cites Rachael Luck of the University of Reading as giving name to a “shared, invisible entity” called the Design Concept
    - She identified this term after watching a series of design meetings by a team and analyzing the interactions of an architect and the client
      - They both would talk about the “design concept” of the thing being designed => what qualities it had, what they were trying to achieve
- In developing the System/360 at IBM in the early 60s, Brooks’s team developed a “reference architecture” and then several implementations (all at the same time) at different points along performance/price curves
  - To the team, the physical implementations were “platonic shadows” of the “real” System/360 that existed only in their minds and perhaps on paper

© Kenneth M. Anderson, 2014

What’s so important about Design Concepts?

- Is there value in creating a design concept for a system?
  - Brooks argues “yes”
  - First, great designs have conceptual integrity
    - unity, economy, and clarity
    - You can develop a mental model about the design that lets you make predictions about how it will operate; if they predictions come true, we often take delight in the use of the design
  - Second, a design concept gives a design team something to talk about
    - Unity of Concept is essential and it can only be achieved though lots of conversation; what’s in scope, what’s out? what concepts does this design make use of? how do they relate?
- Imagine if Unix’s “everything is a file” didn’t apply to network sockets
  - The jarring effect that would have, would mar the beauty of the design

Design Process (I)

- As I said in Lecture 19: Design is Ancient
  - People have been thinking about designs (i.e. a particular design) for a long time
    - Brooks mentions
      - Vitruvius: lived ~2050 years ago (!)
      - Leonardo da Vinci and Andrea Palladio, i.e. ~600 to ~450 years ago
  - People thinking about the design process itself, however, is much more recent: last 150 years or so
    - Of particular note for computer science: Herbert Simon's The Sciences of the Artificial in 1969: a foundational book on artificial intelligence and the science of design

© Kenneth M. Anderson, 2014

Design Process (II)

- Fred Brooks spent time in The Mythical Man-Month reflecting on the design process of the IBM Operating System/360 project
    - In that book, he spends time developing his concept of conceptual integrity
        - One example from that book is that of Cathedrals
            - One “problem” with cathedrals is that they often took 100s of years to be built. During that time, the person in charge of the design of the building would change (the previous person having died)
                - With these changes would come a change in style and the resulting building would have “conflicting concepts” that would often not mesh

Design Process (III)

- As counterpoint, the Reims Cathedral
  - Eight generations of architects who stuck with the original plan
- As a result, the concepts behind the design of the building are uniform throughout
- The major lesson from Brooks is that “too many cooks, spoils the broth”: the design of a system should be controlled by “a few minds”, preferably one.

Kinds of Design

- Brooks briefly mentions a few types of design
  - System Design: Design of complex systems
  - Artistic Design: Design of objects meant to delight and convey meaning
  - Routine Design: Design of well-understood objects (i.e. bridges)
  - Adaptive Design: Changing an existing design for a new context or purpose
  - Original Design: blank slate design of a new object
- This book focuses on original, system design from the standpoint of an engineer

© Kenneth M. Anderson, 2014

The Rational Design Model (I)

- In Chapter 2, Brooks presents the “rational model of design”
    - He quotes Herb Simon: “The theory of design is that general theory of search… through large combinatorial spaces”
- The basic idea is that **design can be thought of as a systematic process**
    - We have **goals** for the design: primary and secondary
    - a **utility function**: given an instantiation of the design, rate how well it satisfies all primary and secondary goals, allowing us to compare
        - *“most designers imagine the terms linearly summed but conceive of each [variable as not linear but curved asymptotically to saturation]”*
    - **constraints**: some simple and some terrifying (“satisfy all building codes”)
    - **resources**: people, materials, and budgets

© Kenneth M. Anderson, 2014

The Rational Design Model (II)

- We then start a search process with some combination of the goals having been achieved and use the utility function to rate it
  - We also make sure that it satisfies the constraints and doesn’t overwhelm our resources
- If it has a “low rating”, we instantiate a new design with one of the goals achieved in another way
  - We then assign a value and compare
- We continue “searching through a combinatorial space” until we have “satisficed” our goals; that is we have found a design that is “good enough”
  - satisfice is a term developed by Herb Simon that is a combination of “satisfy” and “suffice”
- That is, we acknowledge that achieving all of our goals while also meeting all of our constraints (including resources) is typically impossible

© Kenneth M. Anderson, 2014

The Rational Design Model (III)

- The rational model was primarily developed by Herb Simon
  - Brooks points out that Simon was optimistic that AI could be used to perform design
    - As a result, Simon NEEDED a rational model because such a model was required if we ever hoped to automate it
    - Those of you who have taken AI courses should recognize the rational model as a search strategy used in a lot of AI algorithms
      - e.g. early attempts at automating the playing of chess
- Brooks himself used this model in a book he wrote about the design of computer architectures (processor design): he developed 83 linked decision trees (!) that could be used to identify various points in the CPU design space

© Kenneth M. Anderson, 2014

Benefits of the Rational Model

- Brooks points out several benefits of the rational model
    - Any systematization of the design process is a great step forward when compared to “let’s hack it”
        - Such a process forces you to think before you build
            - What ARE my goals, resources, and constraints?
            - Helps to avoid wasted effort searching parts of the design space that are untenable
        - It provides a structure to organize your work and aids communication
        - It is readily taught to new project members
        - It helps teach designers to broaden their horizons beyond their own personal experience

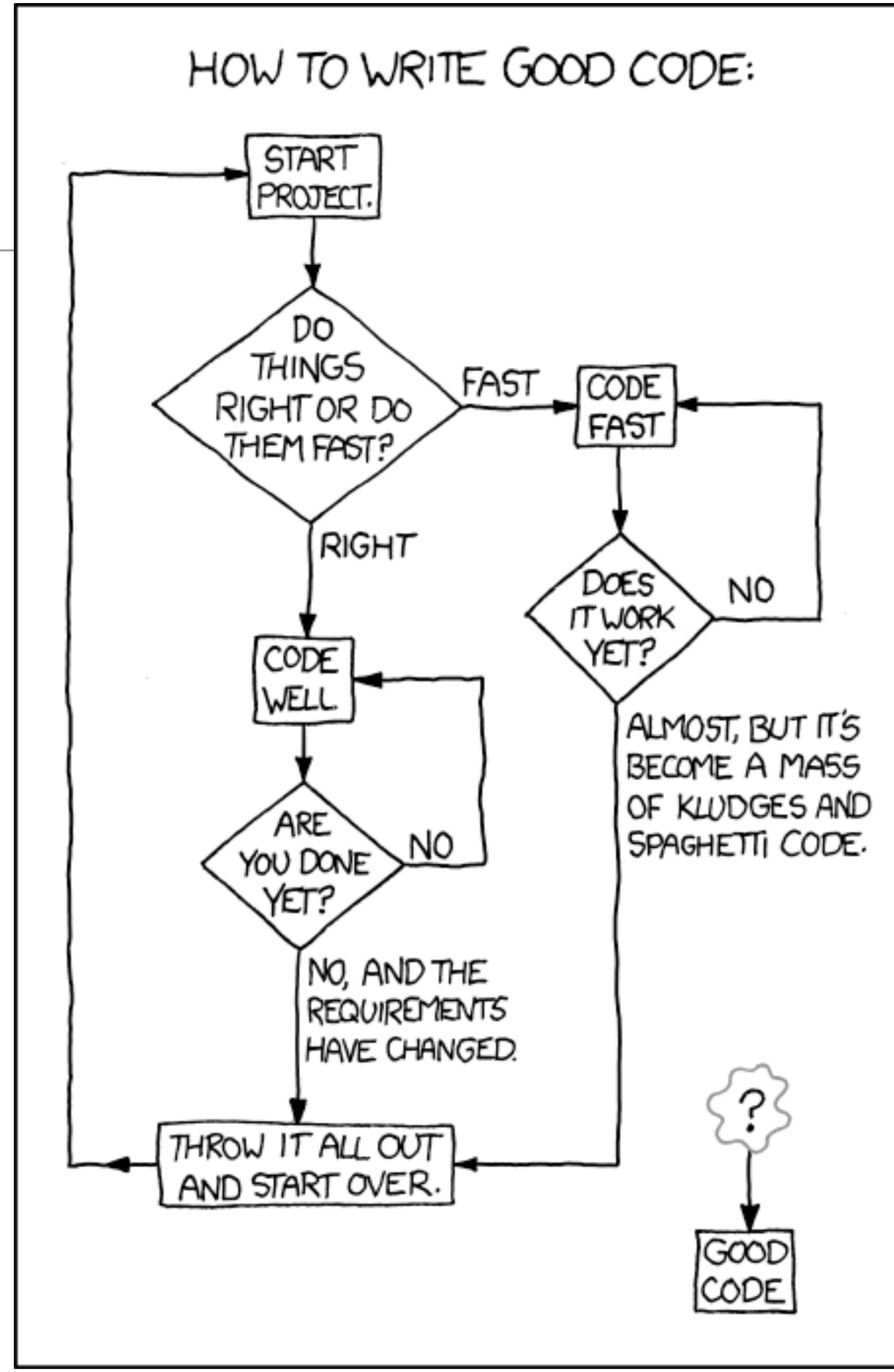
BUT (!)

- One of my favorite foundational papers in software engineering research is
  - "A Rational Design Process: How and Why to Fake It"
    - by David Parnas and Paul Clements
    - IEEE Transactions on Software Engineering, 12(2), 251-257, 1986
  - Basic Lesson: Design is Messy
    - There are too many variables; two many wrong paths; inspiration strikes and you jump to a new part of the design space for no logical reason
  - BUT: once you find a good design, there is worth in documenting it as IF you followed a rational process to find it
    - what are the final design's parameters, why is it good, etc.

© Kenneth M. Anderson, 2014

xkcd strikes again

<http://xkcd.com/844/>



What’s Wrong with this Model? (I)

- In Chapter 3, Brooks starts to unpack the problems with the rational model
  - “Sometimes the problem is to discover what the problem is”
    - —Gordon Clegg, 1969 (in another book called “The design of design”!)
- The key problem with the rational design model is **we don’t really know the goal when we start**
  - This should sound very familiar to you given our coverage of agile methods
- Brooks presents an example of a highly iterative design process that he engaged in as an intern at an aerospace company
  - He was given a project to build a database for tracking the status of 10,000 drawings related to a radar subsystem
    - He implemented it and was told “that’s what I asked for but that’s not what I want” (!); he then iterated many times until they found something that worked;
- The chief service Brooks played was helping ***his client decide what he really wanted***

© Kenneth M. Anderson, 2014

Decision Trees

- The Rational Model assumes we have a set number of parameters and that we can create decision trees that allow us to (rationally) explore a set of alternatives for that parameter
  - Brooks presents an example of such a tree for designing an alarm clock in
    - Clock
      - Visibility
        - Luminous Dial
        - Plain Dial
      - Alarm
        - Sound
        - Setting
        - Control
  - The problem: **We don't know these trees in advance, we discover them!**

Nodes in the Search Space

- The nodes of a search space represent “tentative designs” not “design decisions”
  - In the alarm clock model, some branches are “parallel attributes” while others are “alternative attributes”
    - “The choices in one branch are linked to those in others—by exclusion, affinity, or trade-off”
- As a result, we are make MULTIPLE decisions as we go from one node to the next to create a separate, tentative design for the object being created
  - In addition, this decision space is combinatorial
    - The possible combinations quickly expand to infinity

© Kenneth M. Anderson, 2014

The “magic” utility function

- We make the assumption that we can look at each node (i.e. design) and apply a (magic) utility function that will tell us how to compare one design with another
    - It is, in fact, impossible to do this
        - In some cases, you’re comparing apples with oranges
        - In other cases, you would need to explore each alternative “down to the leaves” to understand some parameters/goals/constraints such as performance or cost
- While the utility function is theoretically possible, it too encounters a combinatorial explosion of alternatives in practice
    - What to do in response? Estimate (hard to do) and prune (declare LOTS of paths as out-of-scope) allowing us to focus on the alternatives that matter

© Kenneth M. Anderson, 2014

The Goals and their Weights keep Changing

- As you create a design, the design “talks back” and you must respond to that feedback
  - such a response typically changes the entire design
- In other words, as you design, you encounter limits to your original understanding of the problem space
  - your new understanding will change how you create the design in the first place
  - in addition, what you value about the project might change, again changing the space of “designs that matter”
- Constant iteration is required in order to keep up with these changes
  - Agile anyone?

© Kenneth M. Anderson, 2014

The Constraints Keep Changing

- Even if we had clear, unchanging goals and resources, design would still be an iterative process because your external constraints will evolve and change
    - The world will change around you; new regulations come on-line; old regulations expire; your client's environment changes requiring you to change
- One complexity with this change in design is that you might have previously pruned a portion of the design space based on an original constraint
    - If that constraint changes or goes away, you may not remember that you had pruned a portion of the design space away and thus not take advantage of revisiting that space now that circumstances are different
- Sometime a solution requires ignoring the constraints: Brooks set-back example
- One approach that can help balance this is to always list constraints explicitly
    - and keep that list up-to-date as things change

Designers Don’t Think This Way

- Brooks points out that one of the more devastating critiques of the rational model is simply pointing out that most designers
  - do not think about design in this way
  - do not work in a way that matches the model
- In practice, intuition and flexibility appear to play a greater role
  - along with finding the right starting point based on experience
    - many designers avoid bad designs simply because they used them in the past; their experience now lets them prune that space early
- Our challenge then is to find a way to work that allows design to be iterative, leverages the experience of the designer, and allows for non-rational jumps to new approaches and new solutions => a BIG challenge for engineering

© Kenneth M. Anderson, 2014

Summary

- We reviewed material from the first three chapters of The Design of Design
  - What is Design: the formulation of conceptual concepts
  - The importance of having a design concept (a design theme)
    - instantiations of this concept are an attempt to achieve this ideal
      - they inevitably fall short of that ideal but the closer we get the better the implementation
    - conceptual integrity is key
      - a design with uniform concepts conceived by a few minds
  - A reminder of how long people have been thinking about designs (thousands of years) and the design process (150 years)
  - The Rational Design Model: its benefits and limitations

© Kenneth M. Anderson, 2014

Coming Up Next

- Lecture 21: User Stories, Chapters 12-16
- Lecture 22: User Stories, Chapters 12-16

© Kenneth M. Anderson, 2014