

# Investigations in Systems Design: Structure, Context, Failure and Usability

**John S. Rhodes (john@webword.com)**  
President and Founder; WebWord.com  
Owego, NY 13827

## Abstract

The purpose of this paper is to briefly review and discuss three books related to systems design. The first book is *Design Paradigms: Case Histories of Error and Judgment in Engineering* (Petroski, 1994), the second book is *The Mythical Man-Month, Anniversary Edition: Essays on Software Engineering* (Brooks, 1995), and the third book is *Notes on the Synthesis of Form* (Alexander, 1964). In this paper, an emphasis is placed on describing the core ideas of the books. Brief discussions of structure, context, failure, and usability engineering are included to highlight several themes found throughout the trio of books.

## Background

The structure of the paper will be as follows. In Part I, a general overview of *Design Paradigms: Case Histories of Error and Judgment in Engineering* (Petroski, 1994) will be provided to the reader. Several interesting ideas from Petroski's book will be outlined following the overview. Brooks' *Mythical Man-Month* (1995) will be given the same treatment. An overview of the book will be provided first, followed by an outline of the key ideas. After Brooks, an overview of Alexander's *Notes on the Synthesis of Form* (1964) will follow. Again, key ideas from the book will be delivered. The books are rich and robust in their own right and Part I is meant to supply the reader with a broad range of interesting ideas from each of the three books.

In Part II, the core ideas discussed in Part I will be synthesized. The books are highly related in some key areas, such as their emphasis on structure, context, and failure. A brief discussion of usability engineering will also be included in Part II, which will further expand the concepts in the three

books. Usability will be used to glue together some loose ends.

## Part I

### *Brief Overview: Petroski*

Petroski's *Design Paradigms* is full of case histories of failures. While this might sound grim, this approach is powerful because it helps people understand several core principles in design and engineering.

Each case is a carefully detailed consideration of what causes failure. The roots of failure are explored in many ways, with each case driving home a new point. Interestingly, the cases Petroski discusses are deliberately not recent so that the reader does not get too caught up in any contemporary features of the disasters. Yet, while the cases are generally old, the core ideas shine through. They are timeless.

Petroski believes that by understanding engineering failures we can better understand some of the fundamental principles of engineering. Like a child

touching a hot stove, the reader is given a feel for the calamities, with the lessons burning brightly in the mind.

Failure is a central idea of design. When we understand failure, we can understand success, but only to a point. As we embrace and then extend technology, and when we stand on the shoulders of the design giants, we are likely to encounter error. But, Petroski understands the root of failure, which is human error. *Design Paradigms* makes it hard, if not impossible, to blame technology itself for catastrophic failures. Almost always, the failures we encounter are human failures. This is quite natural when you consider that design is a human endeavor.

As Petroski states, the purpose of the book is not just to provide a model for explaining how errors are introduced into the design process. It also provides designers with means by which they can avoid making similar errors in their own designs. In short, Petroski believes that by understanding history, we can understand and better control our future engineering designs.

### ***Core Ideas: Petroski***

*Idea One:* Errors deeply rooted in tools and techniques are hardest to discover and change. As a designer, you had better make sure that your tools are built on the right foundations. For this reason, it is important to challenge the dogmas of engineering and design. A related concept is that it is much better to capture problems early. If you wait to find the root cause of a problem, it becomes increasingly difficult and expensive to make changes.

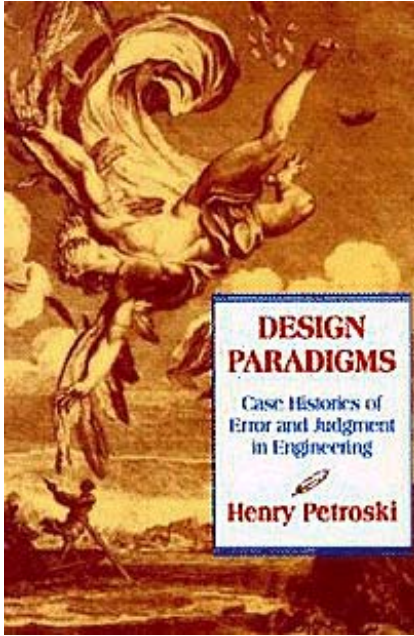
*Idea Two:* Not all failures are major. Specifically, not all failures kill people. However, Petroski makes it clear that small

errors and small flaws prevent products from reaching their full potential. The errors seen in many products, for example, could be prevented with an emphasis on failure prevention.

*Idea Three:* You cannot understand success without understanding failure. There is no guarantee that future successes can be built on past successes. In fact, if certain designs never fail, we can never know when failure will strike. When a failure occurs, it is actually good in that it provides tractable evidence of how good the design is. On the other hand, success proves nothing because failure can occur at any time.

*Idea Four:* Scale effects are pervasive. As things get larger, the modes of failure are likely to change. Petroski provides several ancient and modern examples of failures that happened because designers did simple extrapolation. Interestingly, as some structures get larger, proportionally smaller problems suddenly become extremely critical. These smaller problems are often more detrimental than larger, easy to identify problems.

*Idea Five:* Technology cannot solve problems if the technology in question was developed using methods and concepts that were flawed from the start. For example, if a designer used a computer aided design (CAD) program to create a blueprint, but the CAD program was programmed with flawed logic, then the design will be prone to failure. A tool is only as good as what it is based on, and that basis is generally not a matter of technology, but instead of human thinking and decision making.



Cover: Design Paradigms by Henry Petroski

**Paperback**

September, 1994 (209 pages)

**Dimensions** (in inches)

0.67 x 8.93 x 6.00

**Publisher**

Cambridge University Press

**ISBN**

0521466490

*Idea Six:* Failures can serve as metaphors. When designers view case histories this way, they can apply the lessons learned to many situations. Failure can be seen as a matter of the relations in designs.

*Idea Seven:* Too much of a focus on the major parts of a design project can lead people to ignore the smaller factors. Petroski makes it clear that sometimes you need to step back from a design and consider smaller factors. As stated previously, small design differences often lead to large disasters.

*Idea Eight:* To create a new design, a designer should not simply look at the success of another design and extend it. It is important to look back at the root failure analysis of the first design so that it can be understood in relation to why it was originally built the way it was built. How did the original design avoid failure? A design should never simply extend a success without this knowledge.

***Brief Overview: Brooks***

*The Mythical Man-Month* is Frederick Books' group of essays detailing some of his experiences managing large computer programming projects at IBM. While not exactly a series of case histories, the book is somewhat personal and several lessons are provided. There is definitely a human touch.

The book is valuable for people interested in computer programming and software development. Both managers and developers stand to benefit by reading it. However, because of the way Brooks writes, *The Mythical Man-Month* is also good for almost any person involved in the management of information technology.

Most of the content in the book is about 30 years old. However, the core ideas found in the book still apply to modern software development projects. Brooks added several newer essays to the 25<sup>th</sup> Anniversary Edition, along with some other useful material. There are many ideas and examples in the book but because of space limitations, only a few will be covered in this paper.

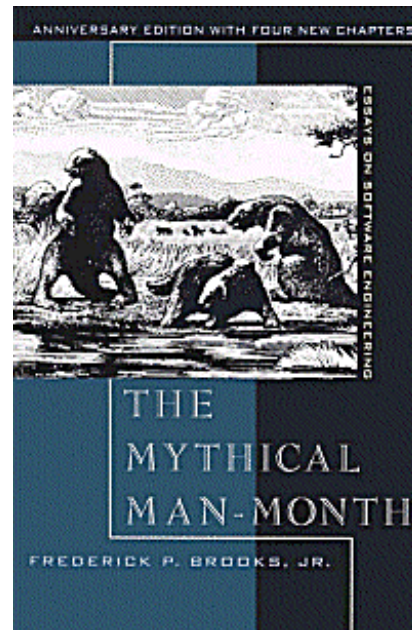
***Core Ideas: Brooks***

*Idea One:* Simply stated, while technology changes rapidly, people do not. Perhaps the central reason why *The Mythical Man-Month* has remained popular is that Brooks has identified several core principles that have stood the test of time. The book is rather old, especially in light of the flurry of changes that have occurred in the technology industry. Yet, many concepts identified by Brooks have lasted. Change might happen rapidly with technology, but managing humans is something that has been going on for a long time. In short, many of the principles of managing people identified by Brooks are still effective.

*Idea Two:* Excellent small systems can be built rapidly. However, truly large systems take significantly more time, resources and coordination. There is not a simple line from small systems management to large systems management. It is often the case that different people with different skills are needed. The tools and techniques are certainly different.

*Idea Three:* Adding more humans to a project does not necessarily help that project. In fact, since new resources need to be added, and since training of the new people is required, and because tasks need to be reallocated, bringing people into a project after it started will often hurt it. The analogy is that just because one woman can have a child in nine months, nine women cannot have a child in one month. Software projects take time; you cannot easily exchange people for time.

*Idea Four:* Successful projects have excellent internal consistency driven by conceptual integrity. Brooks makes it clear that a few people should be responsible for



Cover: Mythical Man-Month by Frederick Brooks

**Paperback**

July, 1995 (336 pages)

**Dimensions** (in inches)

0.75 x 9.07 x 6.11

**Publisher**

Addison-Wesley Publication Company

**ISBN**

0201835959

the grand vision behind a project. In my mind this is similar to the saying that too many cooks in the kitchen will spoil the dinner. If there is a master chef, and people line up to follow, the kitchen will operate smoothly and the food will be great.

*Idea Five:* Structure in thinking and structure in action seem to be very important. Sometimes even an incorrect structure is useful because the lack of structure can cause projects to collapse. My impression is that Brooks thinks chaos is a major enemy of successful projects. He provides several methods for fostering and

capitalizing on structure, such as his recommendation for using a surgical team approach to project management. He also champions the idea that generating extensive documentation is a good thing.

*Idea Six:* Brooks makes many statements that seem obvious. However, while these common sense statements are obvious in retrospect, how many designers actually think about them? A common problem in communication is that what sounds obvious in retrospect is actually quite robust. Along a similar vein, Brooks notes that many projects fail because developers and designers do not fully understand the amount of effort put into the seemingly simple solutions, and therefore they deviate from the plan. They think that they understand the project better than their leaders. In effect, their own common sense gets in the way and harms the project.

*Idea Seven:* A project gets to be a year late, one day at a time. The small things matter; the details are important. People forget that the details form the whole, and to neglect the details is the neglect the whole. The idea is actually quite Zen, although Brooks does not characterize the idea that way.

### ***Brief Overview: Alexander***

Christopher Alexander's *Notes on the Synthesis of Form* is an attempt to explain the process of design. Although it was written in 1964, several of the core ideas in the book are still relevant. Unlike Brooks' book, Alexander does not really talk about any specific technology, so he is not burdened with the baggage of technological change. Yet at the same time, the book offers up several ideas that directly relate to contemporary design problems.

Alexander provides a way for one to understand how there is a structural correspondence between the pattern of a problem and its solution. When you understand the nature of a problem, you understand its structure. In turn, when you understand this structure, you can uncover the form that solves the problem.

To simplify, *Notes on the Synthesis of Form* gives us an appreciation of the design process, in contrast to the design itself. The end result of the design process is a pattern or structure that is the solution or the output of that process. This is obviously different than just studying the design of an end product.

I like to think about Alexander's book in this way: Rather than just thinking about problems and solutions, we should think about problems, solutions, *and* the design processes to connect them together.

Many people, when given a problem, simply start developing an answer. People have a tendency to hear a problem and then instantly develop a form that answers that problem. That approach is foolish when you consider that the process of design is significantly more difficult than stating a problem or stating a solution. The design process is the necessary glue.

### ***Core Ideas: Alexander***

*Idea One:* A diagram, or pattern, is probably just as important as the design process itself because the pattern holds the relationships together. The relationships, or forces, can then be related to other forces, and other diagrams. The design process generates patterns, and these patterns are the building blocks of still other patterns. This is important because it gives us a way to think about building relationships. This is in

contrast to the much more difficult task of comparing design processes to each other. For example, it is much easier to compare the number 10 to the number 10, versus comparing 5 times 2 to 50 minus 40. Comparing 10 to 10 is simple because it compares one mathematical output to another mathematical output, but comparing 5 times 2 to 50 minus 40 is not easy because it compares one mathematical process to another mathematical process. In short, design patterns can be easily compared with each other, whereas it is much more difficult to compare design processes to each other.

*Idea Two:* There seems to be a tradeoff between simplicity and rigor. If you want to fully specify a design problem, design process, or pattern, you will need to use more language and more math. On the other hand, if you want to paint a simple picture, you will need to use less language and less math. This is not always true, of course, but it is generally true. Simple tools yield simple results, and therefore cannot be used in complex situations. Fortunately, the somewhat complex tools simplify extremely complex situations. They are worth the cost. To understand the process of design, sophisticated tools are generally required. For Alexander, logic is one such tool, whereas set theory is another. Both augment human mental capacity.

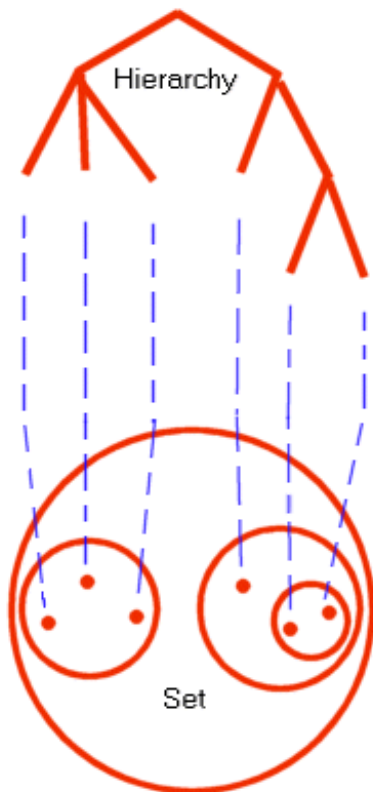
*Idea Three:* Alexander makes the point that goodness of fit is a function of the relationship between a form and its context. The form is the solution to the problem whereas the context defines the problem. They form a functional dyad; they need each other. In fact, form and context are so tightly coupled that you cannot effectively talk about design or the design process without talking about both form and context. Given that you understand this relationship, you can then begin to discuss misfits. A misfit

occurs when the form does not meet the context. Furthermore, good fit occurs when there is an absence of negative qualities. In many ways, this is similar to Petroski's desire to focus on, and eliminate, failures. Alexander states that the task of design is not to create a form that meets certain conditions, but to create a form in a context that does not include misfits.

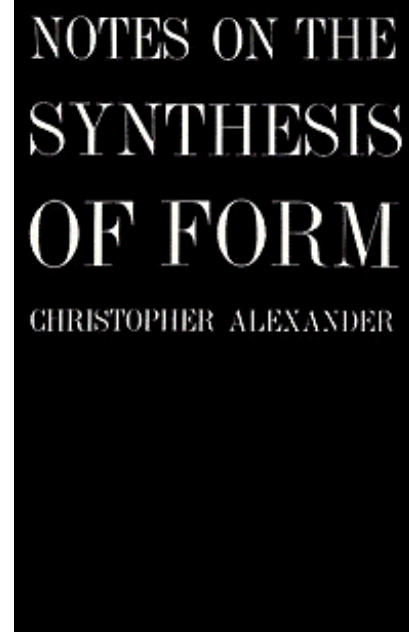
*Idea Four:* It is only through communication that design information can be passed along to other people. However, the method of communication is quite important. If the method of communication is purely verbal, from father to son for example, then there is little room for using logical and structural techniques. Tradition rules the design process. On the other hand, if the communication consists of mathematical notation, printed records, and similar information, the design can be significantly more robust. The rules of design are more explicit and concrete. These forms of communication transcend time and space, to some degree. More importantly, as design continues to become more complex, self-conscious rigor is necessary. Following tradition will yield misfits, and failure is more likely. Structural methods are superior.

*Idea Five:* One of the most useful things for a designer to do is construct a hierarchy of concepts. Alexander talks at length of lists, sets, and hierarchies. Following Brooks and Petroski, the idea is to improve design via structure. At its heart, success results from structured methods. Finding patterns is similarly helpful. Alexander also makes an excellent point about structure: When you classify and structure design problems, you are free to handle them at a later time, using various formal methods. Structure begets structure.

*Idea six:* Formalized information can be more easily manipulated than non-formalized information. For example, it is relatively easy to move from a hierarchy to a set, and back again (see Figure 1). This kind of information manipulation is not nearly as easy if the information lacks structure. In short, if we think of design as being an exercise in developing structure, we are open to many more possible design solutions. Furthermore, those designs are very likely to be more rich and robust; form fitting the context, with fewer misfits. When formal structure is available, one can test multiple, symbolically generated alternatives.



**Figure 1.** Diagram of a hierarchy to set relationship



Cover: Notes on Synthesis of Form by Christopher Alexander

**Paperback**

June, 1964 (216 pages)

**Dimensions** (in inches)

0.62 x 8.22 x 5.34

**Publisher**

Harvard University Press

**ISBN**

0674627512

*Idea Seven:* Like Petroski, Alexander does not believe that technology (i.e., mechanically computed decisions) can automatically eliminate design problems. If the foundation of the technology is flawed, the results of the computation will be flawed. In computer science, this is described with the phrase: Garbage In, Garbage Out. For Petroski, the solution is to understand the design process. That in turn means that both the form and the context need to be structurally defined. Their patterns must be made obvious. Once this happens, then the patterns can be compared

using set theoretical devices. Eliminating misfits becomes significantly easier once both the form and context are similarly defined, and their fundamental elements are brought into alignment.

*Idea Eight:* Alexander is definitely enamored with diagrams. A diagram can capture both the form and the context of a design situation. Excellent diagrams allow designers to switch back and forth between the context and the form. Diagrams also help to uncover various design extensions, or new answers, that are not apparent without them.

## **Part II**

### ***Theme: Structure***

There are several themes that run through the three books. All of the authors seem to be concerned with patterns and structure. For example, Petroski literally cares about structure because bridges, by definition are structures. But he also seems to believe that structure is a natural part of the design process.

Brooks is concerned with structure in that he talks about utilizing hierarchical project teams. His rules of project development are based on structure, and his good words about generating documentation are also aimed at structure.

A huge fraction of Alexander's book is about structure. You cannot design without understanding and developing structure. Furthermore, you cannot match a design problem to a design solution unless both are properly structured.

All three books place a premium on decreasing chaos. The chaos in each book is

described differently, but ultimately, each book shows how structure and logic help the designer. In every case, there is a certain amount of thinking and rigor that are necessary to generate a successful design.

Good design does not just happen; it is a matter of careful planning. Petroski, Brooks, and Alexander care about this planning and provide people with tools to understand design. All of the tools depend on designers acting and thinking in structured ways. Structure yields success.

### ***Theme: Context***

All of the authors seem to feel that design can only happen in relation to some context. In effect, you cannot design in a vacuum. Further, a design will not inhabit empty space, because ultimately it will be embedded in some context.

Petroski makes this clear when he describes the various bridge failures. A bridge will fail when you do not consider how it will be used. In one part of his book, Petroski talks about suspension bridge failures caused by the marching feet of soldiers, the sudden movement of people at boat races, and cattle stampedes. These bridges were designed without these uses in mind, and as such, the context of use was ignored. Failure ensued.

Brooks also talks of context. In many places in his book, his concern is with time, which is a key element of any context. He discusses project time, budget time, and even human time. Context is critical.

Alexander's notion of context is more abstract, yet it is just as important. The core idea is that good design is a matter of a form fitting an appropriate context. The context, in this sense, is the environment. Alexander is very strict about this definition because he

claims that you cannot talk about a form without simultaneously talking about the context. The form and the context are inextricably linked.

So, all three authors are concerned with context. But, this should not be surprising because it is intuitive that great design work cannot happen in a vacuum. Stated in yet another way, the environment defines the very products that are being designed. Knowing the backdrop of a design is therefore a key factor in developing a successful product.

### ***Theme: Eliminating Failure***

It is interesting that all three books discuss failure, in one manner or another. The authors make it clear that failure is a part of design. In the case of Petroski, studying failure is directly related to generating excellent designs. Similarly, for Brooks and Alexander, failure is seen as symptomatic of poor design processes.

All of the authors indicate that eliminating errors is important. Specifically, they provide evidence that eliminating errors early in project development will save enormous resources later in the project. Basically, going back in time costs a lot of money. This concept is important because many projects are fully delineated before they even start. This can hamper changes in development. If the path is too well defined, and changes are not allowed early, then the project moves along to a point where changes are very expensive and time consuming.

Eliminating failure is not just a matter of eliminating potential problems in an actual product or output. Instead, the authors indicate that problems should be eliminated from the design process itself. In other

words, there is a drive for quality and efficiency in both the product and the design process itself.

In some sense, this is the most consistent theme found through the three books. For example, Brooks talks about scheduling failures, Petroski illustrates structural failures, and Alexander discusses misfits. In all of these cases, failure is a key element. In terms of lessons we can learn, the core idea is to identify and eliminate errors, as early as possible. Success comes from understanding and preventing failure, not from simply trying to perpetuate success.

### ***Usability***

The unseen glue that binds these books together is usability. For purposes of this paper, the Institute of Electrical and Electronics Engineers (1990) definition of usability will be employed:

*Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.*

But usability is much more than this definition implies. For example, usability is concerned with continuous improvement, product quality, and problem solving.

Petroski's focus on failure, Brooks' emphasis on documentation, and Alexander's emphasis on the process of design, are all related to continuous improvement. To successfully practice usability, one should continuously strive to improve a product and to meet the needs of users.

Usability for a product is never perfect. Therefore, as each author indirectly shows, a product is never perfect and can always be improved.

Usability is also concerned with product quality. Alexander, Brooks, and certainly Petroski, all understand that quality is a critical element of any product.

Interestingly, while quality is important, it is understood in much different terms by each author. For example, for Brooks, a software product could be released that had some bugs. Quality might still be considered to be very high if, for example, support calls were minimal. On the other hand, Petroski's book is full of disasters caused by seemingly small design flaws. The point is that quality is relative to the users of the system, and therefore quality is related to usability.

One interesting thing about each book is that they are really about problem solving. Each author is trying to explain how people should solve design problems, and minimize the pain that people feel. This focus on problem solving is very similar to the problem solving found in usability books such as Don Norman's *Psychology of Everyday Things* and *Things That Makes Us Smart*, as well as Jakob Nielsen's *Usability Engineering*.

There are other ways that the three books reviewed in this paper relate to usability and human factors. For example, as discussed above, all three books discuss failure. This is exactly the approach taken by Vincent Flanders and Michael Willis in *Web Pages That Suck* and the somewhat popular Bad Designs web site. Some people might even say that the central purpose of usability is to prevent failure, and to address the poor designs that lead to failure.

### ***Summary***

The three books reviewed in this paper all make one thing blatantly clear: Design is not easy. Brooks helps us understand why projects fail, Petroski shows us failures in

design, and Alexander outlines a complex set of tools to deal with design issues.

In all of these ways, it becomes obvious that design is not easy. Furthermore, it is often the case that design is so bad that products are unsatisfactory. In some cases, people die. Again, we see that design is not trivial.

In the future, even more emphasis should be placed on usability, particularly as it relates to design. Usability helps designers understand how their designs impact real people. Ultimately, that is the real test of success.

### ***References***

- Alexander, C., *Notes on the Synthesis of Form*, Cambridge: Harvard University Press, 1964.
- Bad Designs web site (<http://baddesigns.com>)
- Brooks, F.P., *The Mythical Man-Month, Anniversary Edition: Essays on Software Engineering*, Boston: Addison-Wesley Longman Publication Company, 1995.
- Nielsen, J., *Usability Engineering*, San Diego: AP Professional, 1993.
- Flanders, V and Willis, M., *Web Pages That Suck*, Alameda: Sybex Inc.
- Norman, D., *The Psychology of Everyday Things*, New York: Basic Books, 1988.
- Norman, D., *Things That Make Us Smart*, Reading: Perseus Books, 1993.
- Petroski, P., *Design Paradigms: Case Histories of Error and Judgment in Engineering*, Cambridge: Cambridge University Press, 1994.