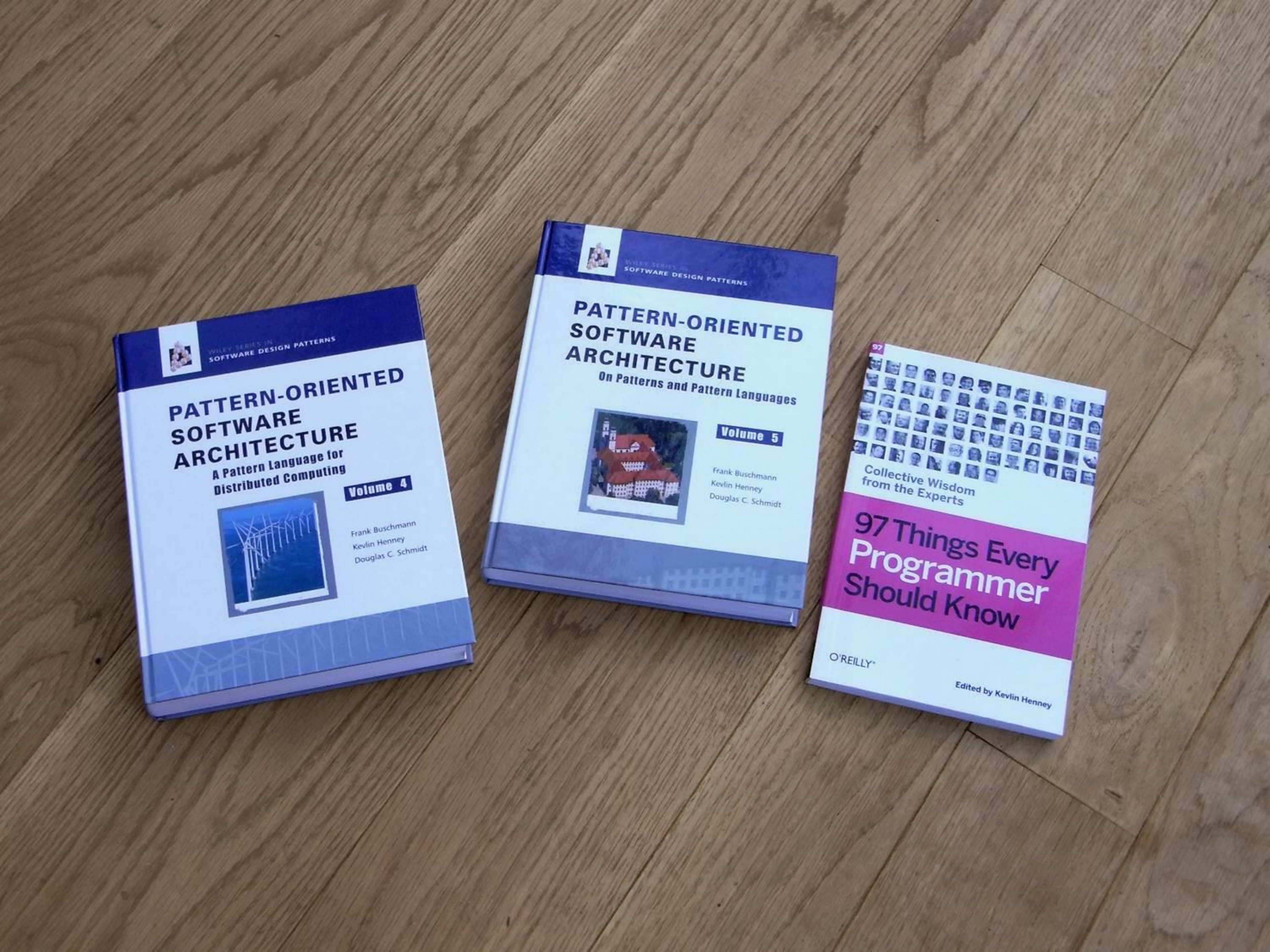
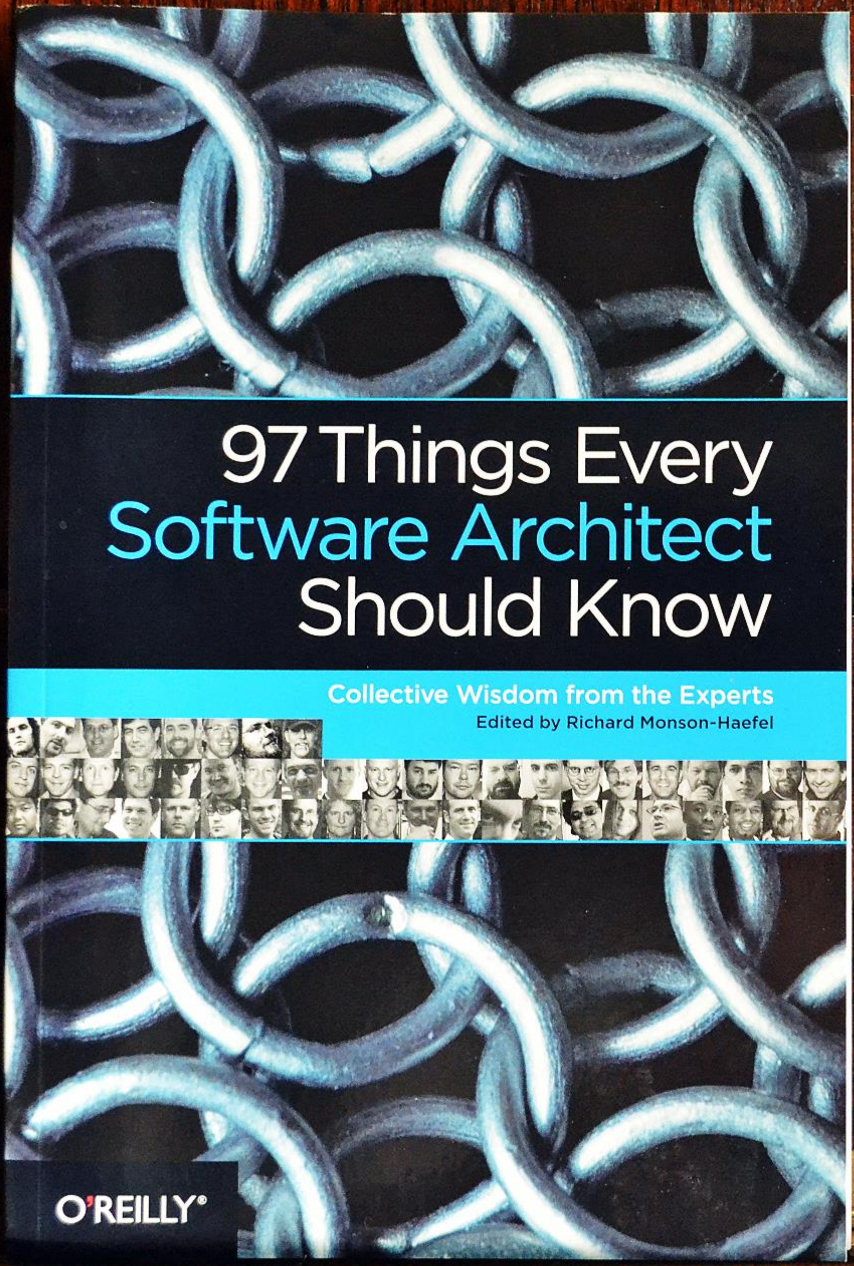
The Architecture of Uncertainty

@KevlinHenney


WILEY SERIES IN SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
A Pattern Language for Distributed Computing
Volume 4
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt
WILEY SERIES IN SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
On Patterns and Pattern Languages
Volume 5
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt
97
Collective Wisdom from the Experts
97 Things Every Programmer Should Know
O'REILLY
Edited by Kevlin Henney


97 Things Every
Software Architect
Should Know
Collective Wisdom from the Experts
Edited by Richard Monson-Haefel
O'REILLY

When a design decision can reasonably go one of two ways, an architect needs to take a step back. Instead of trying to decide between options A and B, the question becomes "How do I design so that the choice between A and B is less significant?" The most interesting thing is not actually the choice between A and B, but the fact that there is a choice between A and B.

Kevlin Henney
"Use Uncertainty As a Driver"

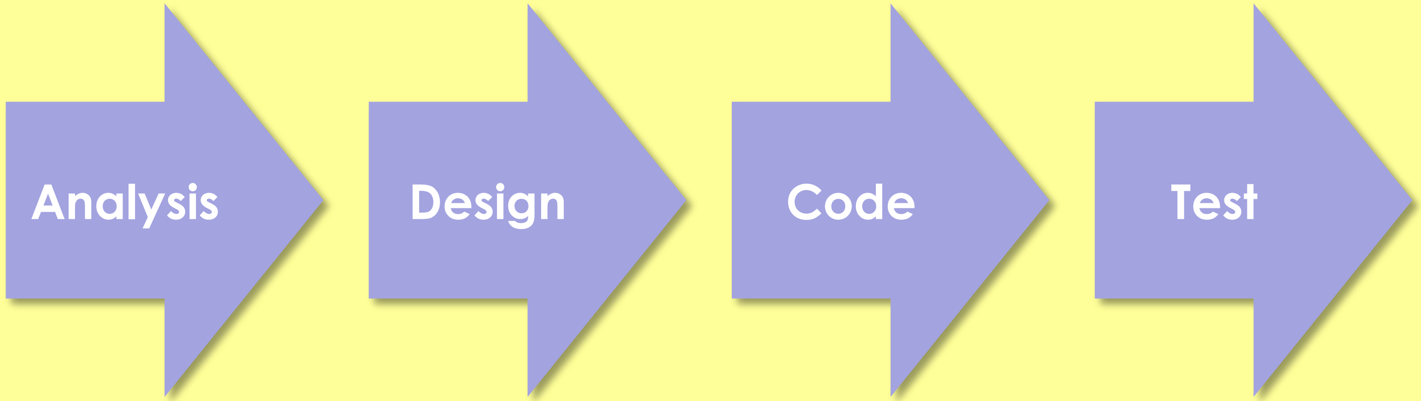
We propose [...] that one begins with a list of difficult design decisions or design decisions which are likely to change. Each module is then designed to hide such a decision from the others.

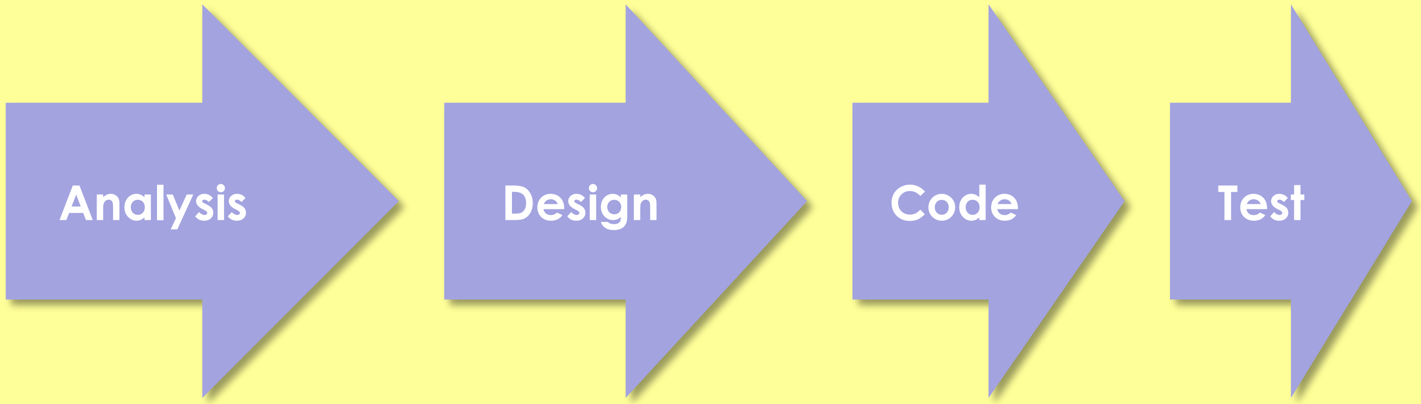
David L Parnas

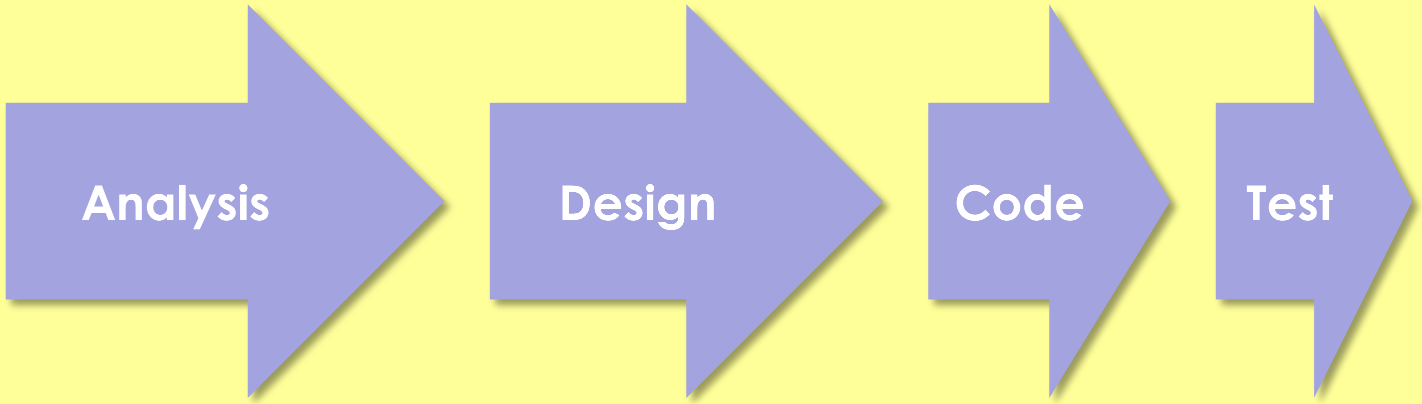
"On the Criteria to Be Used in Decomposing Systems into Modules"

All architecture is design but not all design is architecture. Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change.

Grady Booch


Analysis
Design
Code
Test


Analysis
Design
Code
Test


Analysis
Design
Code
Test

Walking on water and developing software from a specification are easy if both are frozen.

Edward V Berard

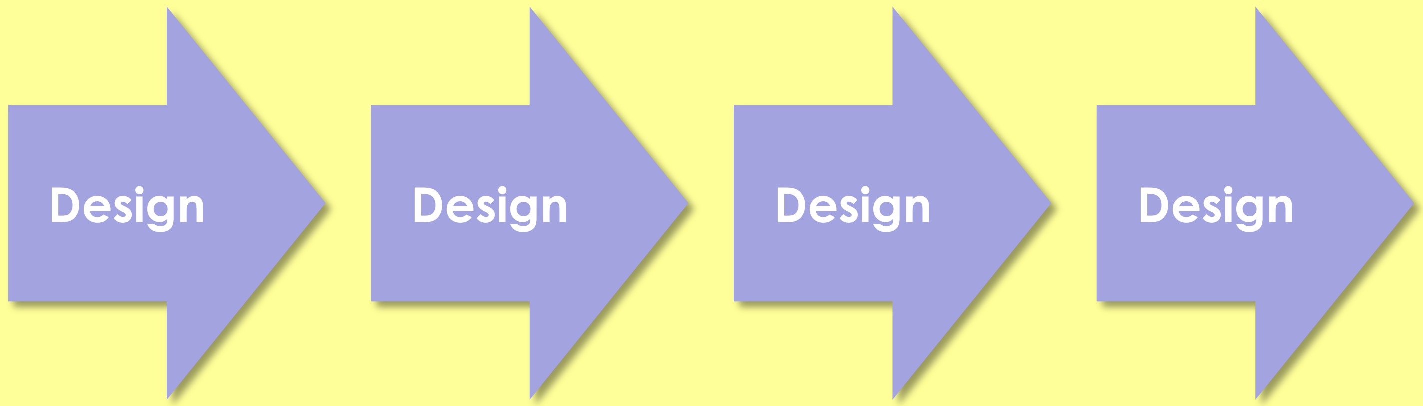
Expert

Proficient

Competent

Advanced Beginner

Novice


Design
Design
Design
Design

Design

Programming is a design activity.

Jack W Reeves
"What Is Software Design?"

Coding actually makes sense more often than believed. Often the process of rendering the design in code will reveal oversights and the need for additional design effort. The earlier this occurs, the better the design will be.

Jack W Reeves
"What Is Software Design?"

```
interface Iterator
{
    boolean set_to_first_element();
    boolean set_to_next_element();
    boolean set_to_next_nth_element(in unsigned long n) raises(…);
    boolean retrieve_element(out any element) raises(…);
    boolean retrieve_element_set_to_next(out any element, out boolean more) raises(…);
    boolean retrieve_next_n_elements(
        in unsigned long n, out AnySequence result, out boolean more) raises(…);
    boolean not_equal_retrieve_element_set_to_next(in Iterator test, out any element) raises(…);
    void remove_element() raises(…);
    boolean remove_element_set_to_next() raises(…);
    boolean remove_next_n_elements(in unsigned long n, out unsigned long actual_number) raises(…);
    boolean not_equal_remove_element_set_to_next(in Iterator test) raises(…);
    void replace_element(in any element) raises(…);
    boolean replace_element_set_to_next(in any element) raises(…);
    boolean replace_next_n_elements(
        in AnySequence elements, out unsigned long actual_number) raises(…);
    boolean not_equal_replace_element_set_to_next(in Iterator test, in any element) raises(…);
    boolean add_element_set_iterator(in any element) raises(…);
    boolean add_n_elements_set_iterator(
        in AnySequence elements, out unsigned long actual_number) raises(…);
    void invalidate();
    boolean is_valid();
    boolean is_in_between();
    boolean is_for(in Collection collector);
    boolean is_const();
    boolean is_equal(in Iterator test) raises(…);
    Iterator clone();
    void assign(in Iterator from_where) raises(…);
    void destroy();
};
```

```
interface BindingIterator
{
    boolean next_one(out Binding result);
    boolean next_n(in unsigned long how_many, out BindingList result);
    void destroy();
};
```

Speculative Generality

Brian Foote suggested this name for a smell to which we are very sensitive. You get it when people say, "Oh, I think we need the ability to do this kind of thing someday" and thus want all sorts of hooks and special cases to handle things that aren't required. The result often is harder to understand and maintain. If all this machinery were being used, it would be worth it. But if it isn't, it isn't. The machinery just gets in the way, so get rid of it.

Martin Fowler
Refactoring

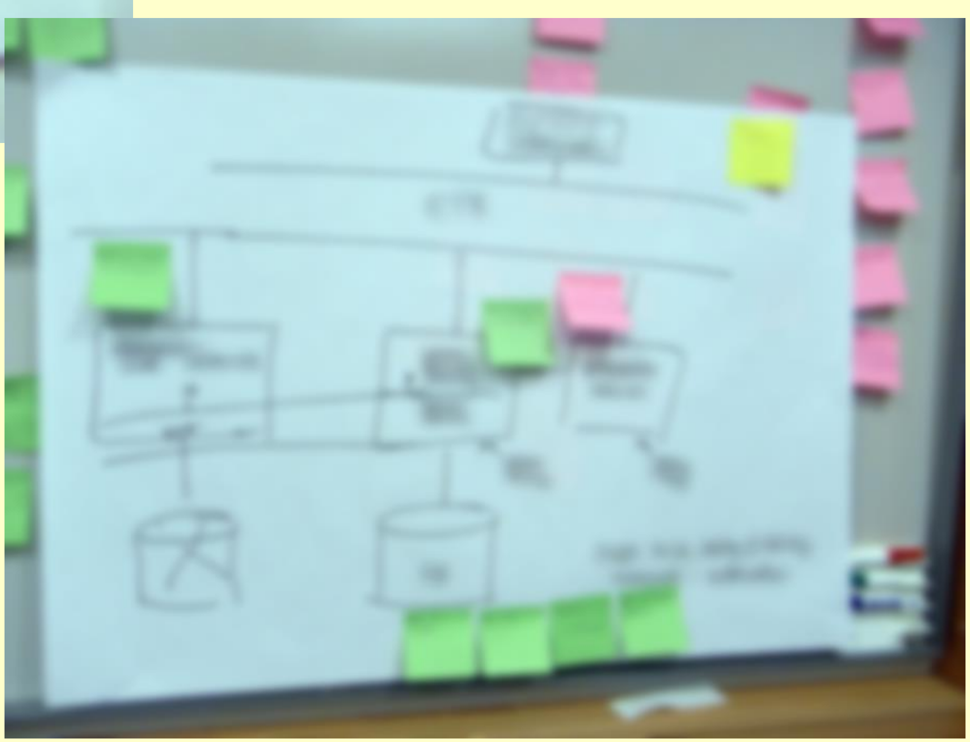
You have a problem. You decide to solve it with configuration. Now you have <%= $problems %> problems!

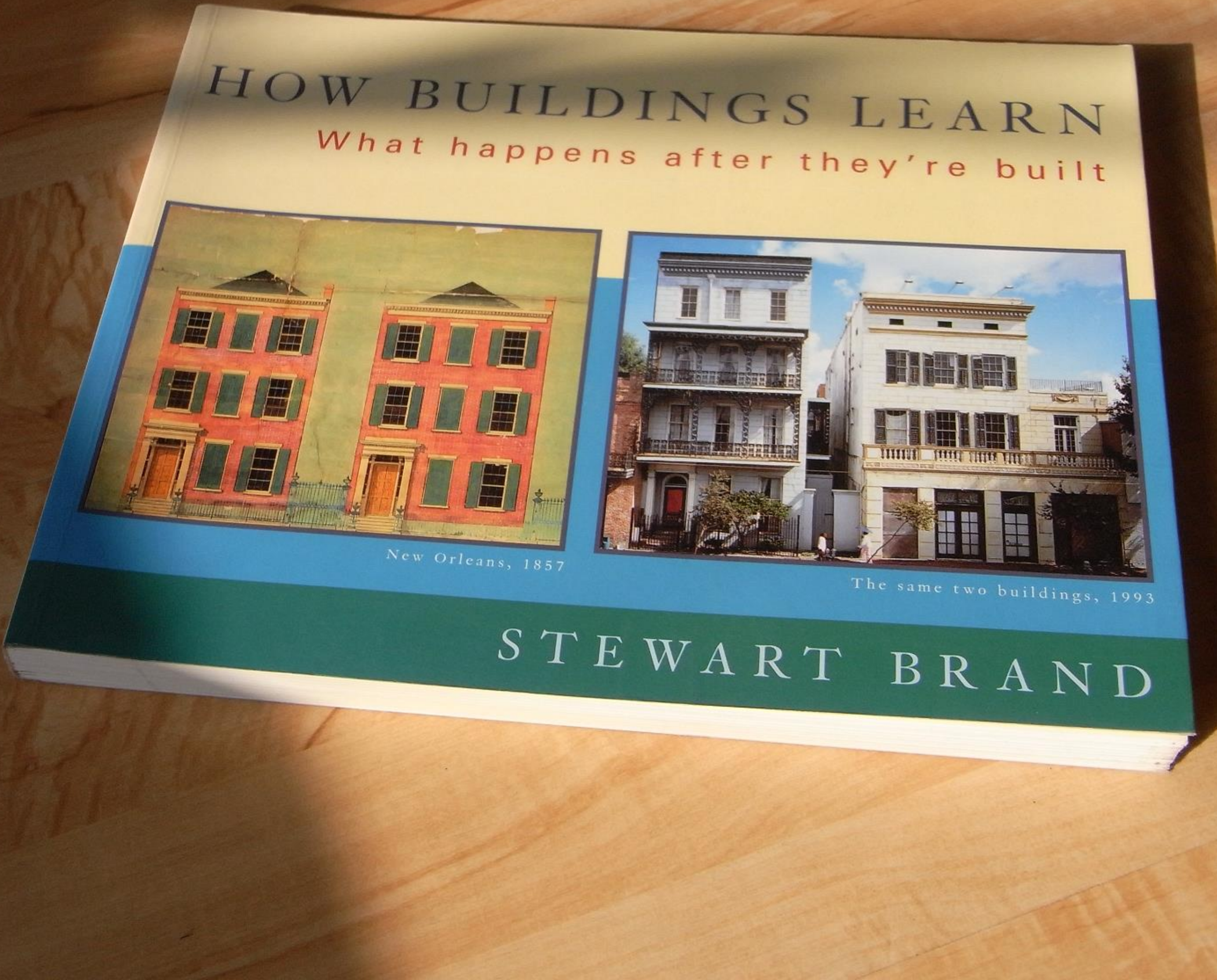
Dan North

https://twitter.com/tastapod/status/342935892207497219

Prediction is very difficult, especially about the future.

Niels Bohr



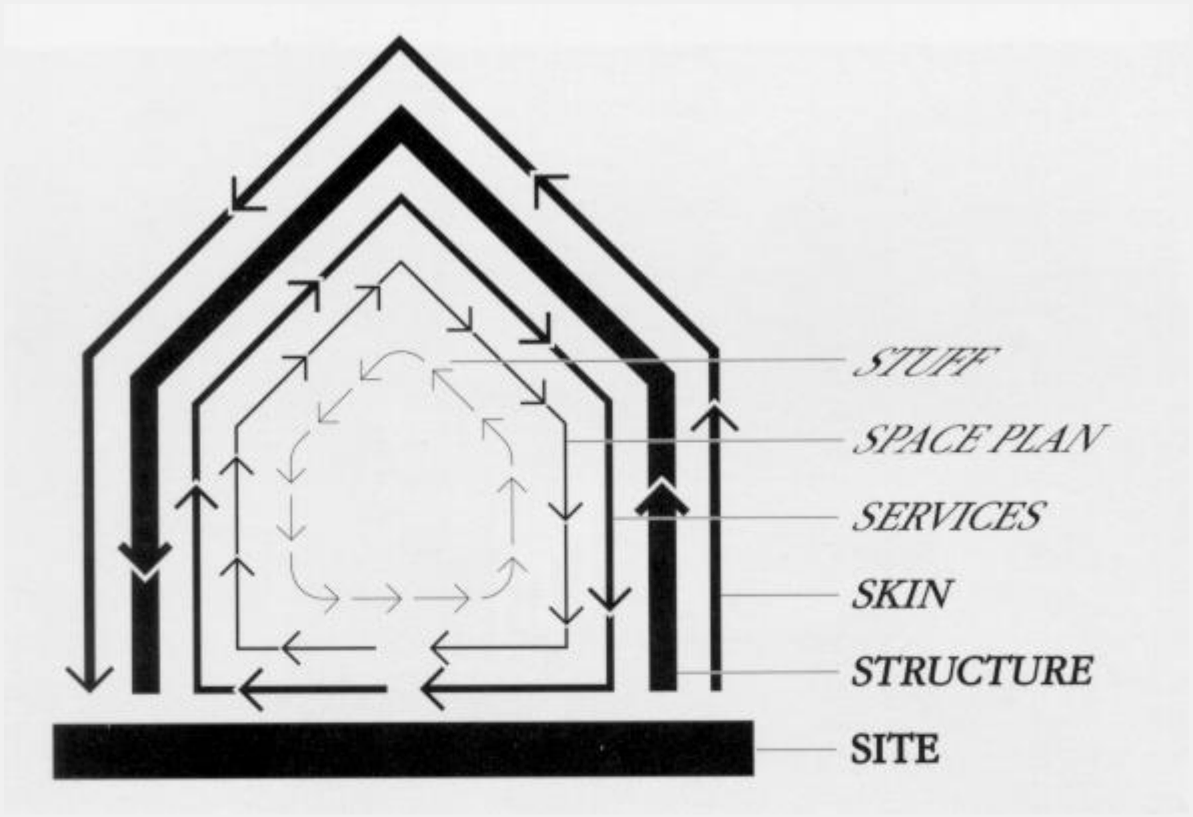

HOW BUILDINGS LEARN

What happens after they're built

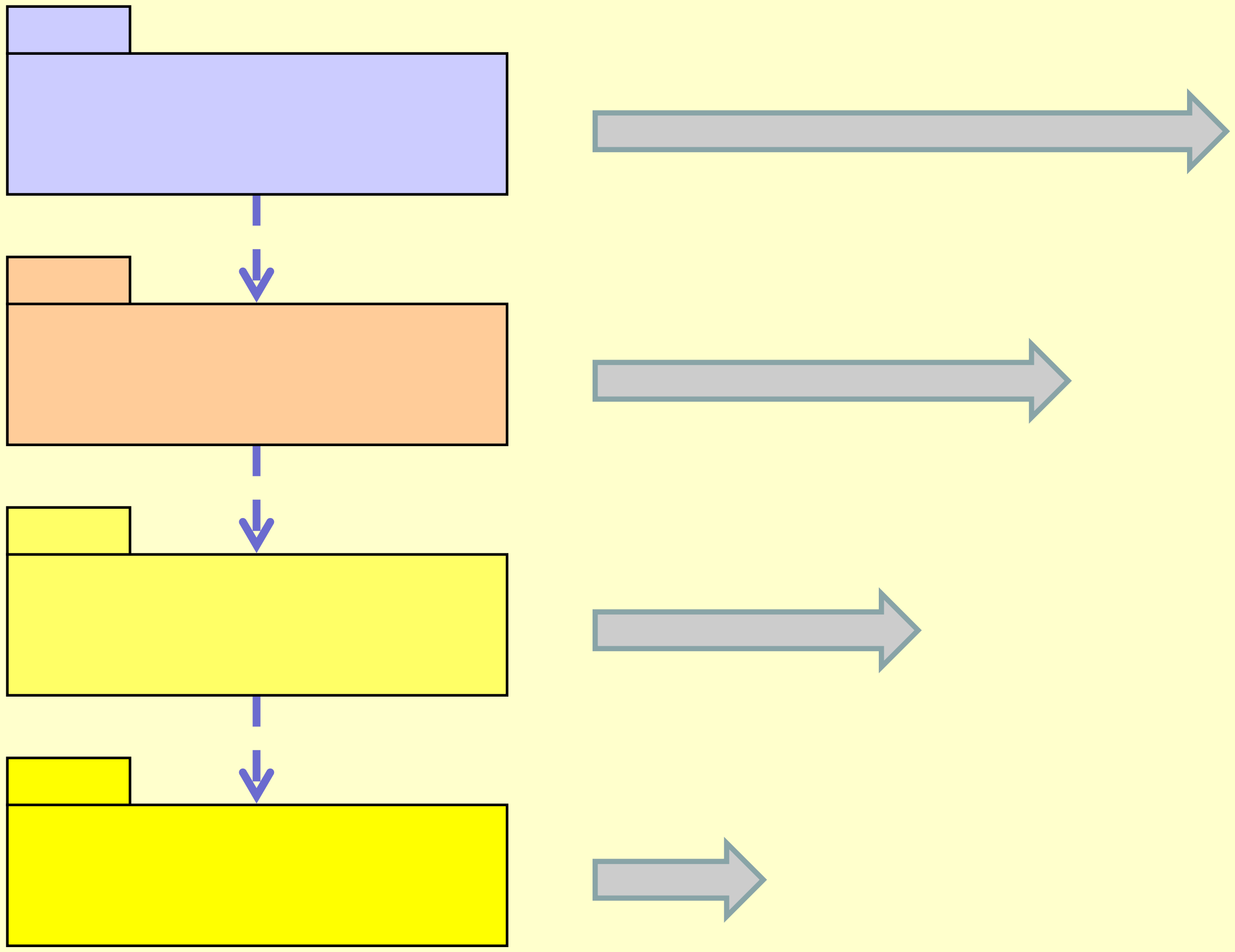
New Orleans, 1857

The same two buildings, 1993

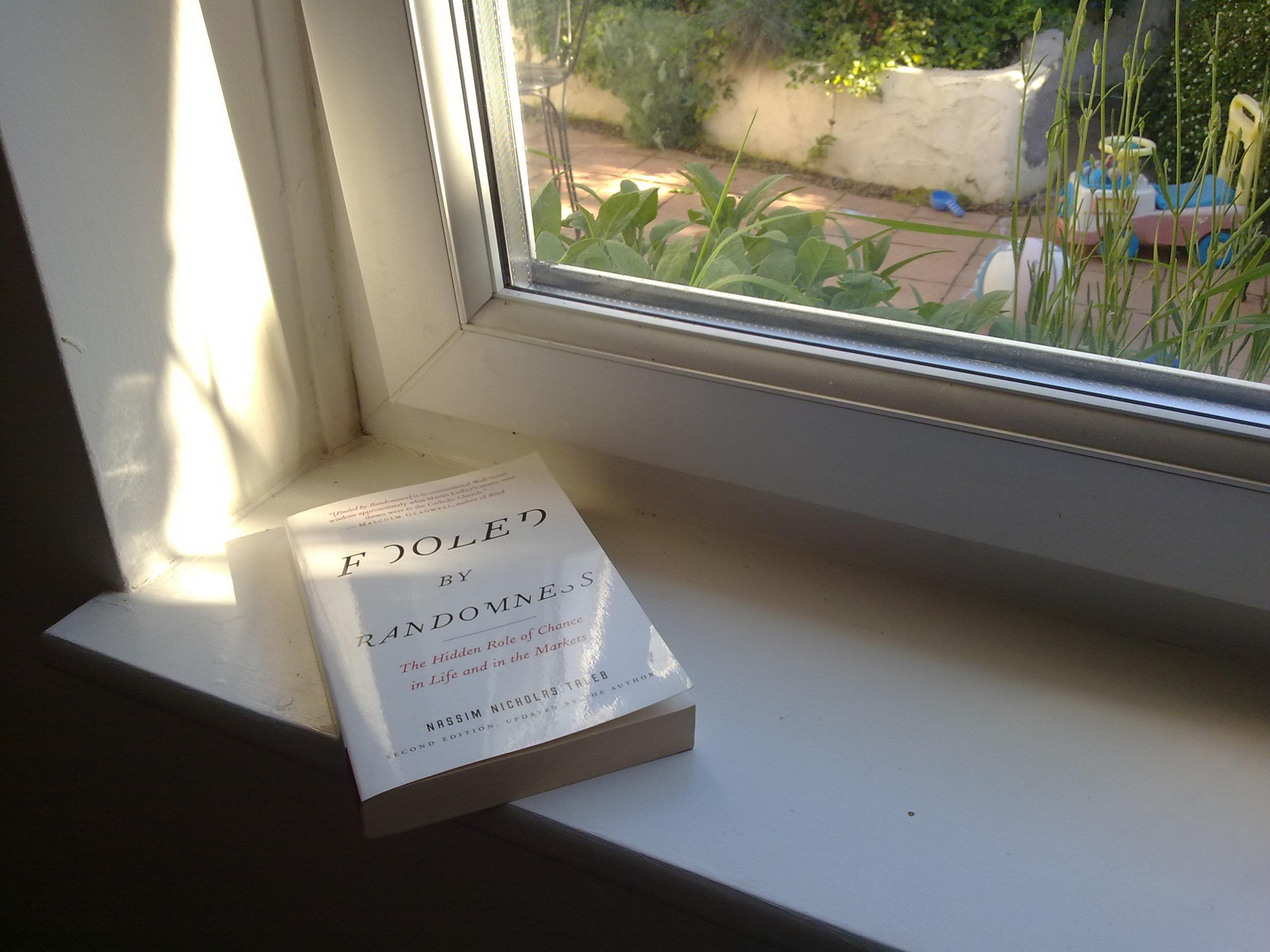
STEWART BRAND



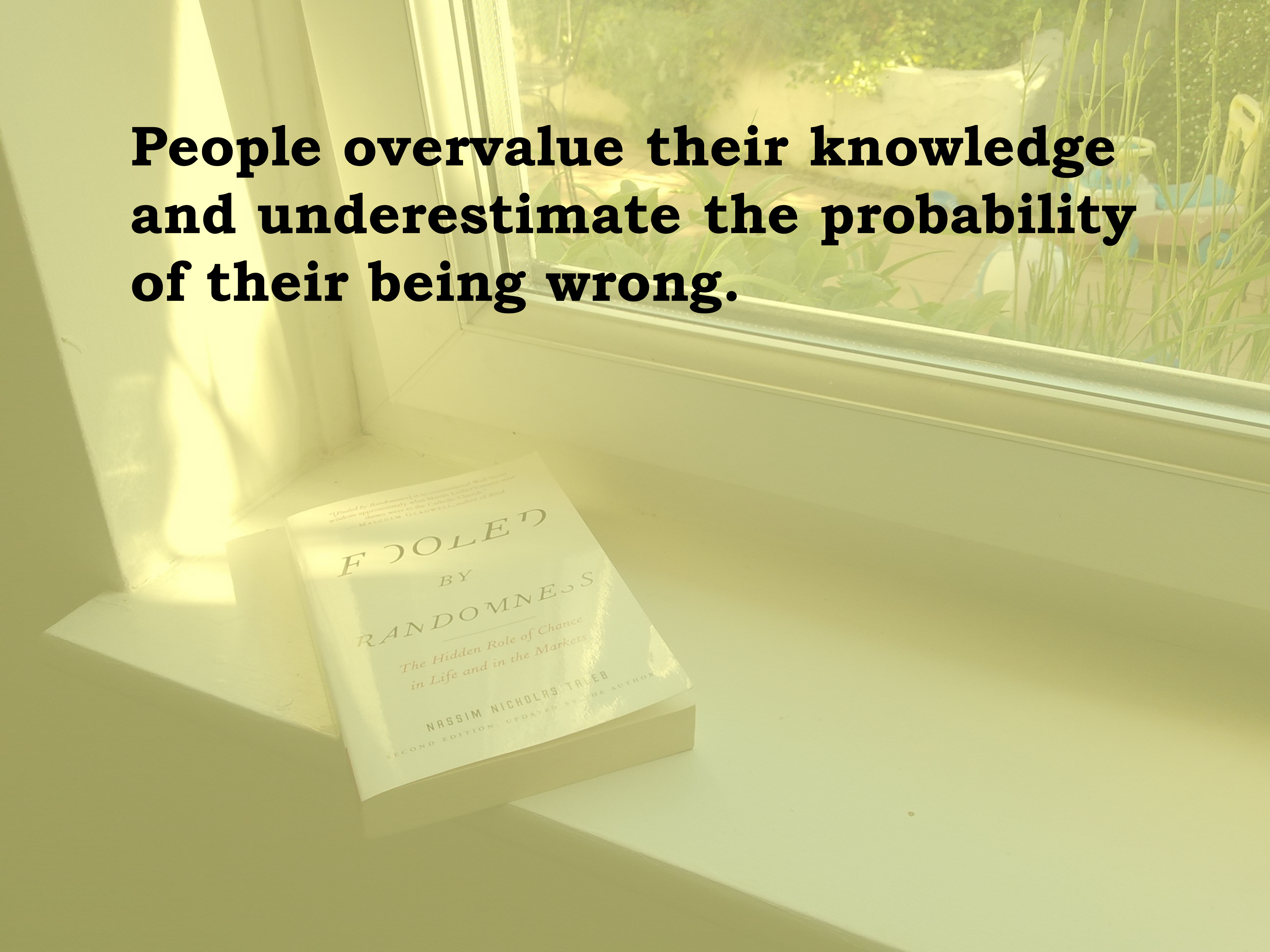
Stewart Brand, *How Buildings Learn*
See also *http://www.laputan.org/mud/*



Rate of change


"[Fooled by Randomness] is to conventional Wall Street wisdom approximately what Martin Luther's ninety-nine theses were to the Catholic Church."
—MALCOLM GLADWELL, author of Blink
FOOLED BY RANDOMNESS
The Hidden Role of Chance in Life and in the Markets
NASSIM NICHOLAS TALEB
SECOND EDITION, UPDATED BY THE AUTHOR

People overvalue their knowledge and underestimate the probability of their being wrong.


FOOLED BY RANDOMNESS
The Hidden Role of Chance in Life and in the Markets
NASSIM NICHOLAS TALEB
SECOND EDITION, UPDATED BY THE AUTHOR

Education is learning what you didn't even know you didn't know.

Daniel J Boorstin