Introduction

Mark Richards
Independent Consultant
Hands-on Enterprise / Integration Architect
Published Author / Conference Speaker

http://www.wmrichards.com
http://www.linkedin.com/pub/mark-richards/0/121/5b9

Published Books:
Java Message Service, 2nd Edition
97 Things Every Software Architect Should Know
Java Transaction Design Strategies

Neal Ford
Director / Software Architect /
Meme Wrangler

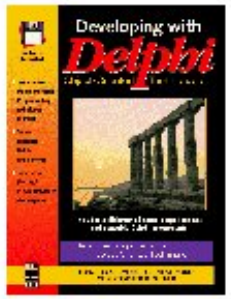
ThoughtWorks®

2002 Summit Blvd, Level 3, Atlanta, GA 30319, USA
T: +1 40 4242 9929 Twitter: @neal4d
E: nford@thoughtworks.com W: thoughtworks.com

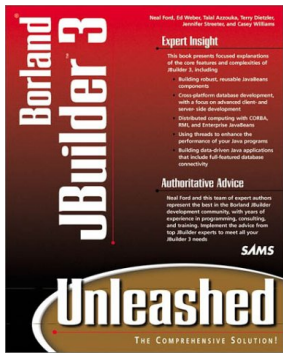
Professional (Long) Bio

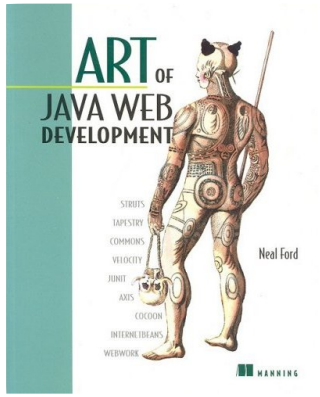
Neal is Director, Software Architect, and Meme Wrangler at **Thought**Works, a global IT consultancy with an exclusive focus on end-to-end software development and delivery. Before joining **Thought**Works, Neal was the Chief Technology Officer at The DSW Group, Ltd., a nationally recognized training and development firm.

Neal has a degree in Computer Science from Georgia State University specializing in languages and compilers and a minor in mathematics specializing in statistical analysis. He is also the designer and developer of applications, instructional materials, magazine articles, and video presentations. He is also the author of 5 books, including the most recent *Presentation Patterns*. Given his degree, Neal is a bit of a language geek, with affections including but not limited to Ruby, Clojure, Java, Groovy, JavaScript, Scala and C#/.NET. His primary consulting focus is the design and construction of large-scale enterprise applications. Neal is an internationally acclaimed speaker, having spoken at over 300 developer conferences worldwide, delivering more than 2000 presentations. If you have an insatiable curiosity about Neal, visit his web site at nealford.com. He welcomes feedback and can be reached at nford@thoughtworks.com.

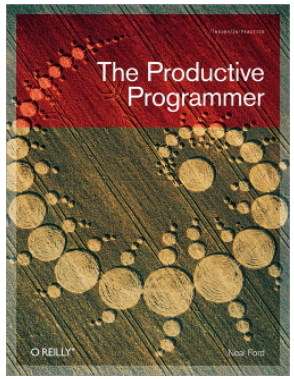


BorCon 97



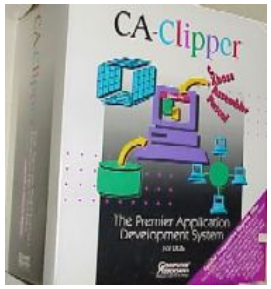


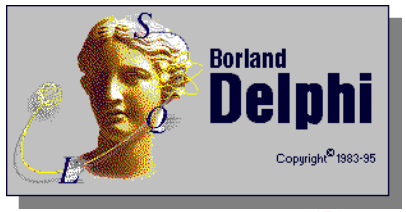
ThoughtWorks



1993 1995 1997 1999 2003 2005 2008 2012

1993 1995 1997 1999 2003 2005 2008 2012





client/server
3-tier
CORBA/COM

web
J2EE
EJB
app servers

functional

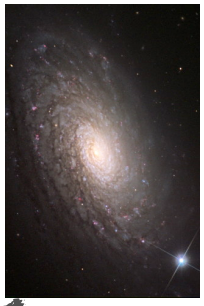
web 2.0
agile web
lighter-weight

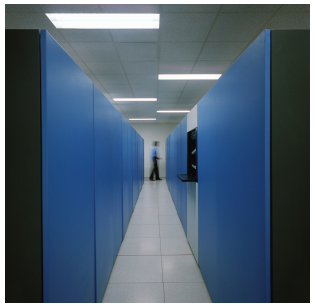
client
80x25
shared database file

professional bio

Mark Richards is an independent integration and enterprise architect involved in the architecture, design, and implementation of Service Oriented Architectures in J2EE and other technologies. He has been involved in the software industry since 1983, and has significant experience and expertise in J2EE architecture and development and systems integration. Mark has a masters degree in computer science from Boston University, and has served as the President of the New England Java Users Group from 1999 thru 2003. Mark is the author of several technical books, and has numerous architect and developer certifications from IBM, Sun, The Open Group, and BEA. He is a regular conference speaker at the No Fluff Just Stuff Symposium Series and speaks at other conferences and user groups around the country. When he is not working Mark can usually be found hiking in the White Mountains or along the Appalachian Trail.

CSC → IBM → independent consultant

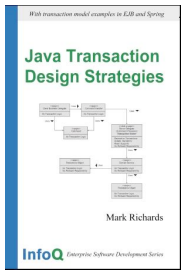


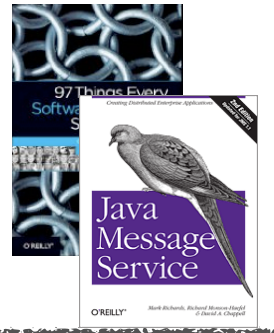


C++

application architect!

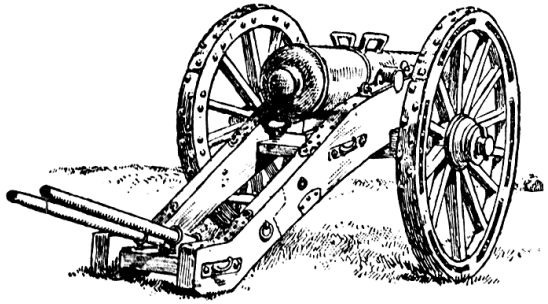
Java





1983 1985 1992 1995 1997 2006 2009 2014

what I learned along the way as an architect....



negotiation and political skills are often more important than technical skills

what I learned along the way as an architect....

service-oriented architecture

it's all about the data...

what I learned along the way as an architect....

you can make a lot of money as an architect!

the worlds best architectures are not the perfect ones, but rather the feasible ones

“what is software design?”

Jack C. Reeves

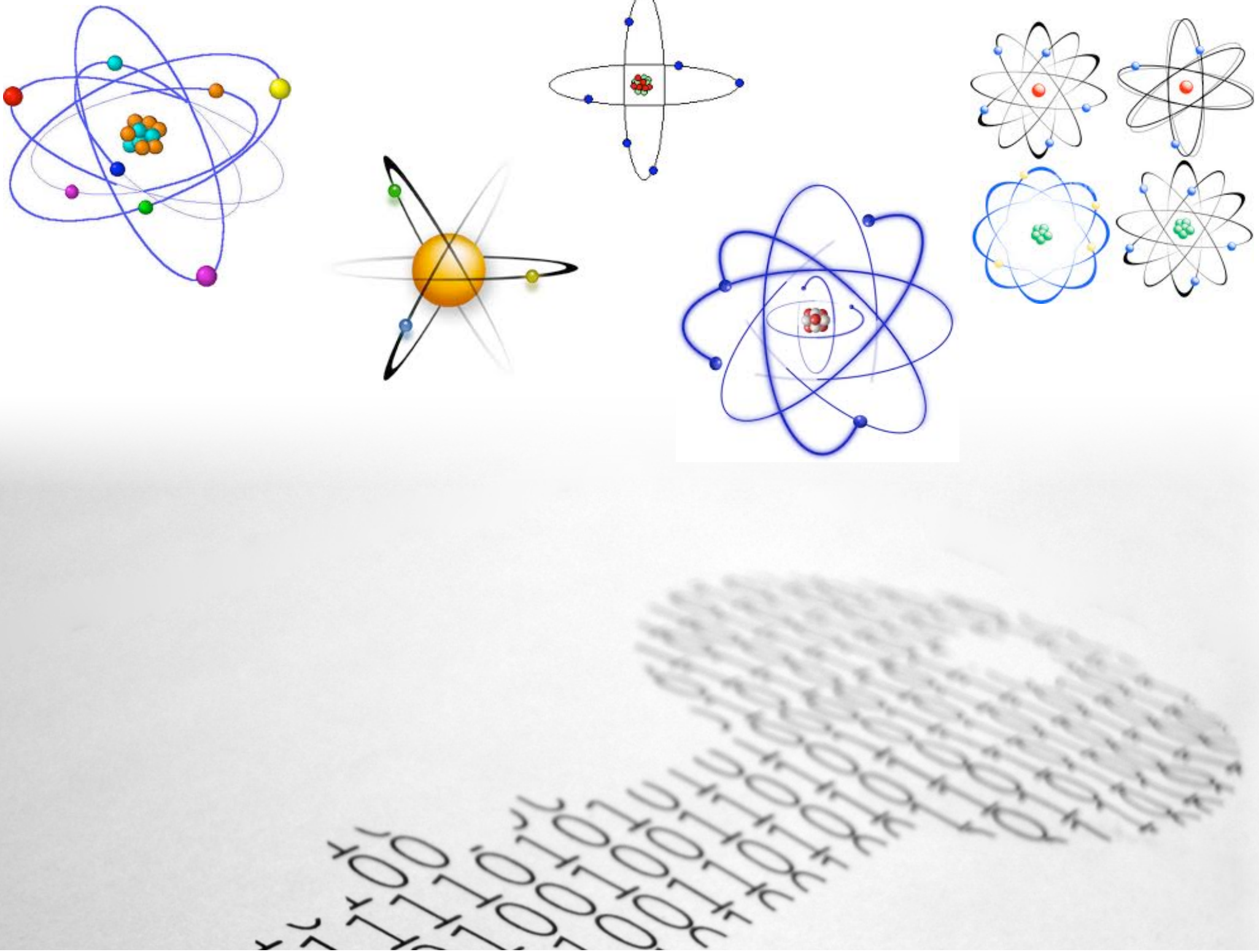
fall 1992, c++ journal

http://www.developerdotstar.com/mag/articles/reeves_design.html

"The final goal of any engineering activity is some type of *documentation*"

"When the design effort is complete, the design documentation is turned over to the manufacturing team."

manufacturing for physical things




software
manufacturing
compilation

```
if (doubleListValue != null) {
    if (altSyntax()) {
        // the same logic as with findValue(String)
        // if value start with %{ and end with }, just cut it off!
        if (doubleListValue.startsWith("%{") && doubleListValue.endsWith("}")) {
            doubleListValue = doubleListValue.substring(2, doubleListValue.length() - 1);
        }
    }

    addParameter("doubleListValue", doubleListValue);
}else if (tmpDoubleList instanceof Map) {
    addParameter("doubleListValue", "value");
}


if (formName != null) {
    addParameter("formName", findString(formName));
} else {
    // ok, let's look it up
    Component form = findAncestor(Form.class);
    if (form != null) {
        addParameter("formName", form.getParameters().get("name"));
    }
}

Class valueClazz = getValueClassType();

if (valueClazz != null) {
    if (doubleValue != null) {
        addParameter("doubleNameValue", findValue(doubleValue, valueClazz));
    } else if (doubleName != null) {
        addParameter("doubleNameValue", findValue(doubleName.toString(), valueClazz));
    }
} else {
    if (doubleValue != null) {
        addParameter("doubleNameValue", findValue(doubleValue));
    } else if (doubleName != null) {
        addParameter("doubleNameValue", findValue(doubleName.toString()));
    }
}
```

design ==

complete source code

```
if (doubleListValue != null) {
    if (altSyntax()) {
        // the same logic as with findValue(String)
        // if value start with %{ and end with }, just cut it off!
        if (doubleListValue.startsWith("%{") && doubleListValue.endsW
            doubleListValue = doubleListValue.substring(2, doubleLis
        }
    }

    addParameter("doubleListValue", doubleListValue);
}else if (tmpDoubleList instanceof Map) {
    addParameter("doubleListValue", "value");
}


if (formName != null) {
    addParameter("formName", findString(formName));
} else {
    // ok, let's look it up
    Component form = findAncestor(Form.class);
    if (form != null) {
        addParameter("formName", form.getParameters().get("name"));
    }
}

Class valueClazz = getValueClassType();

if (valueClazz != null) {
    if (doubleValue != null) {
        addParameter("doubleNameValue", findValue(doubleValue, valueC
    } else if (doubleName != null) {
        addParameter("doubleNameValue", findValue(doubleName.toString
    }
} else {
    if (doubleValue != null) {
        addParameter("doubleNameValue", findValue(doubleValue));
    } else if (doubleName != null) {
        addParameter("doubleNameValue", findValue(doubleName.toString
    }
}
```

software $$$ traditional

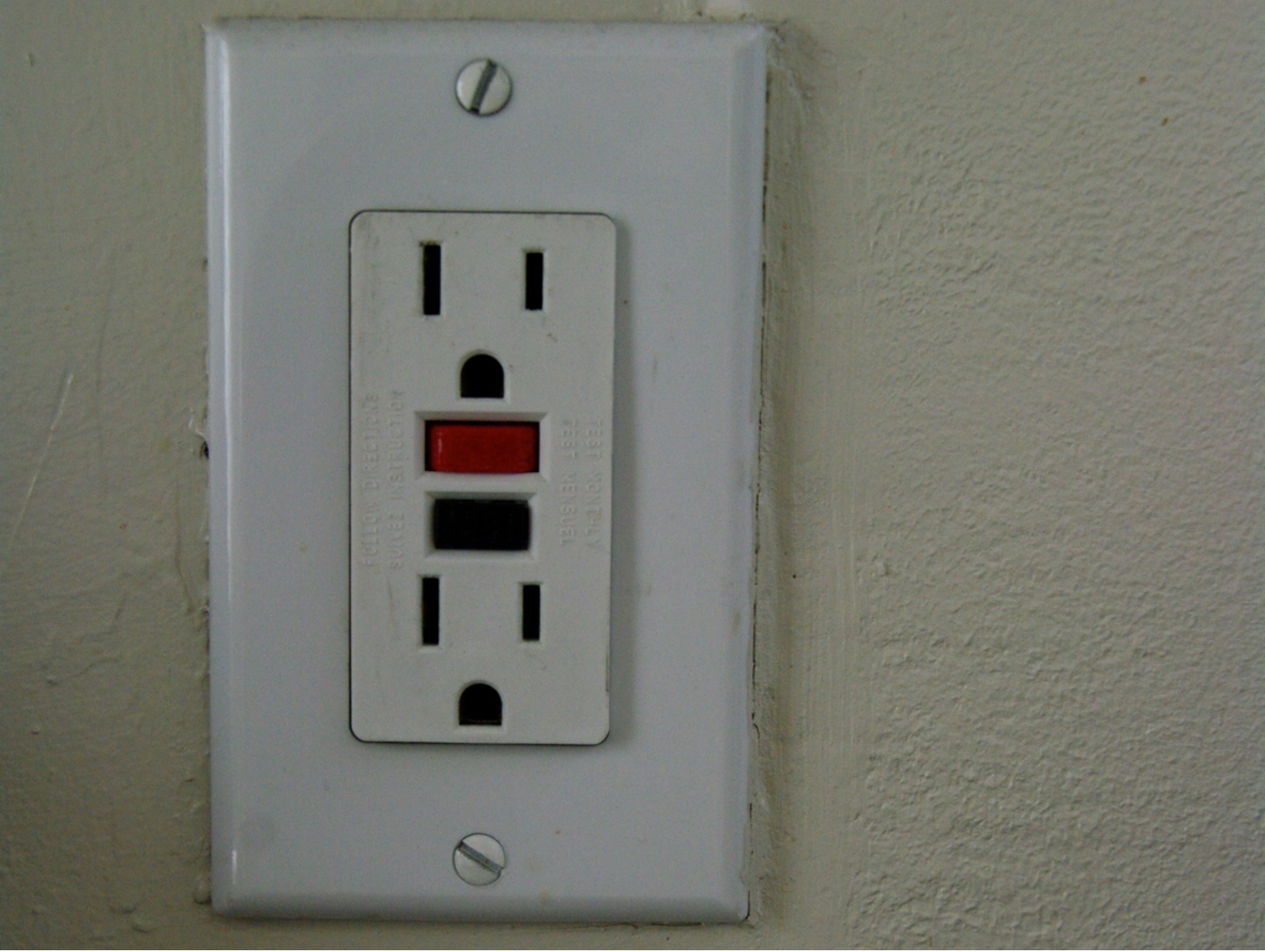
“Given that

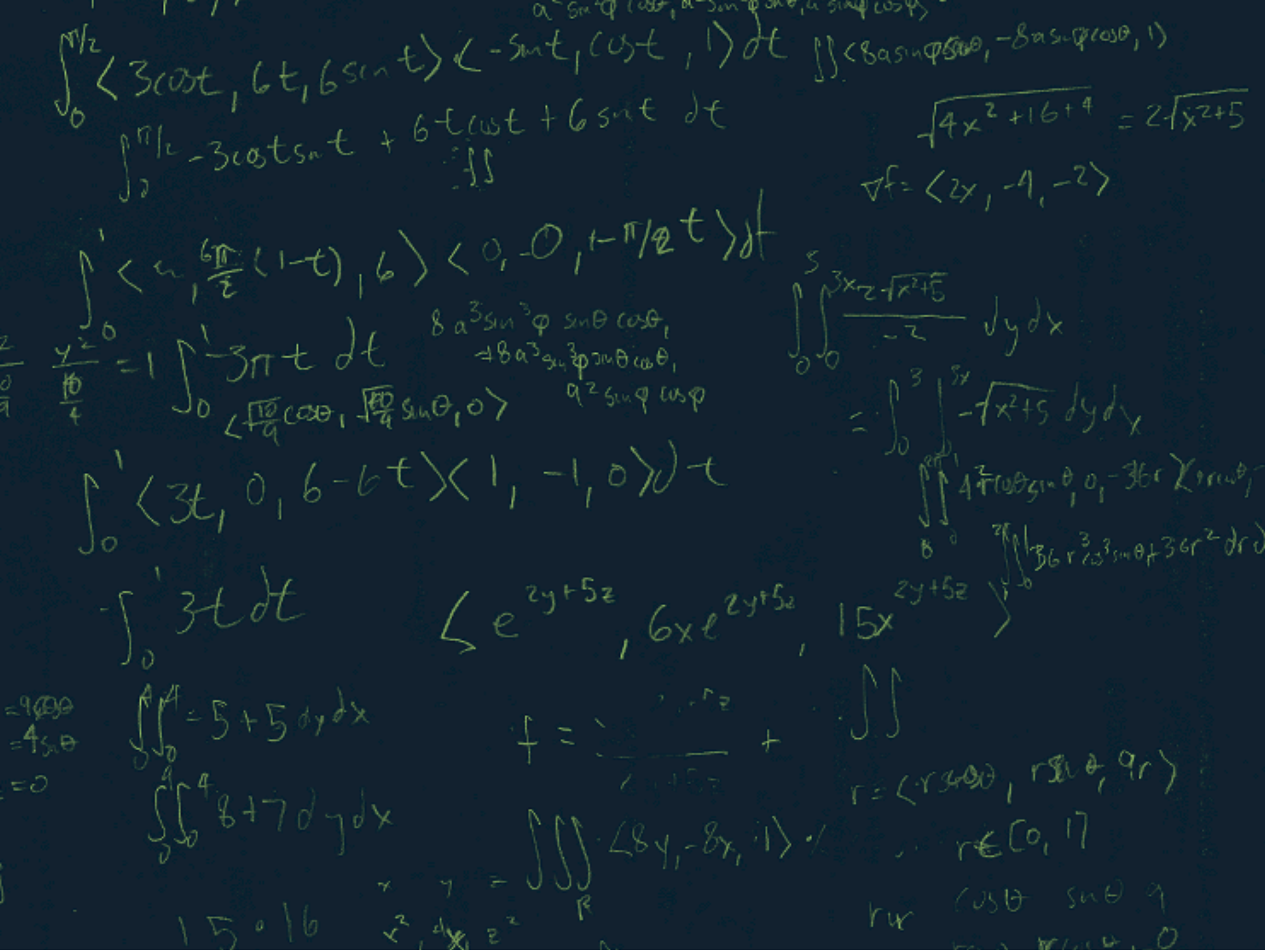
software designs are relatively easy to turn out

and essentially free to build,

an unsurprising revelation is that software designs tend to be incredibly large and complex.”

Jack Reeves


FOLLOW DIRECTIONS
SAVE INSTRUCTION
TEST MONTHLY
RESET BUTTON

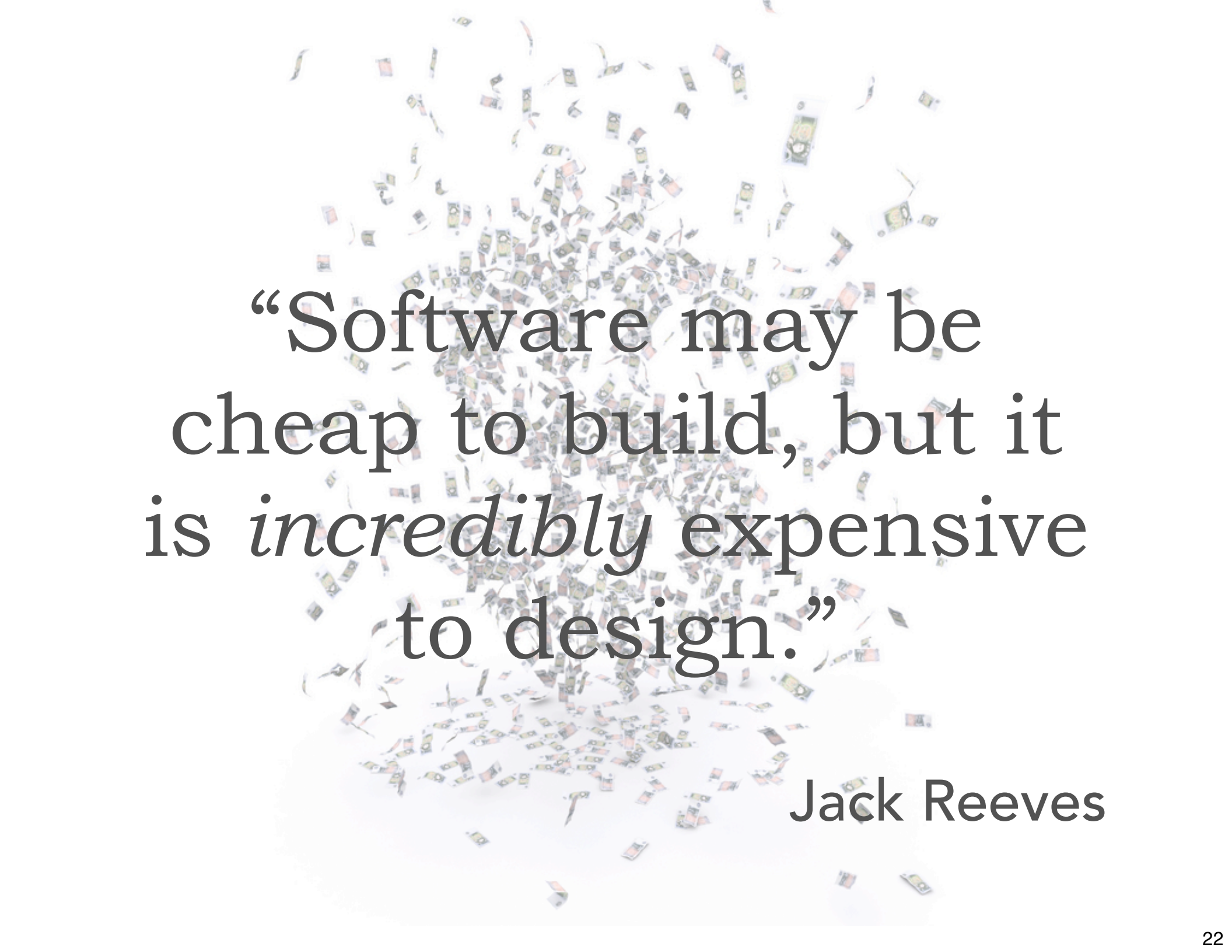


```
@Test public void test_a_bunch_of_numbers() {
    Set<Integer> expected = new HashSet<Integer>(
            Arrays.asList(PERFECT_NUMS));
    for (int i = 2; i < 33550340; i++) {
        if (expected.contains(i))
            assertTrue(classifierFor(i).isPerfect());
        else
            assertFalse(classifierFor(i).isPerfect());
    }
```

testing = engineering rigor in software

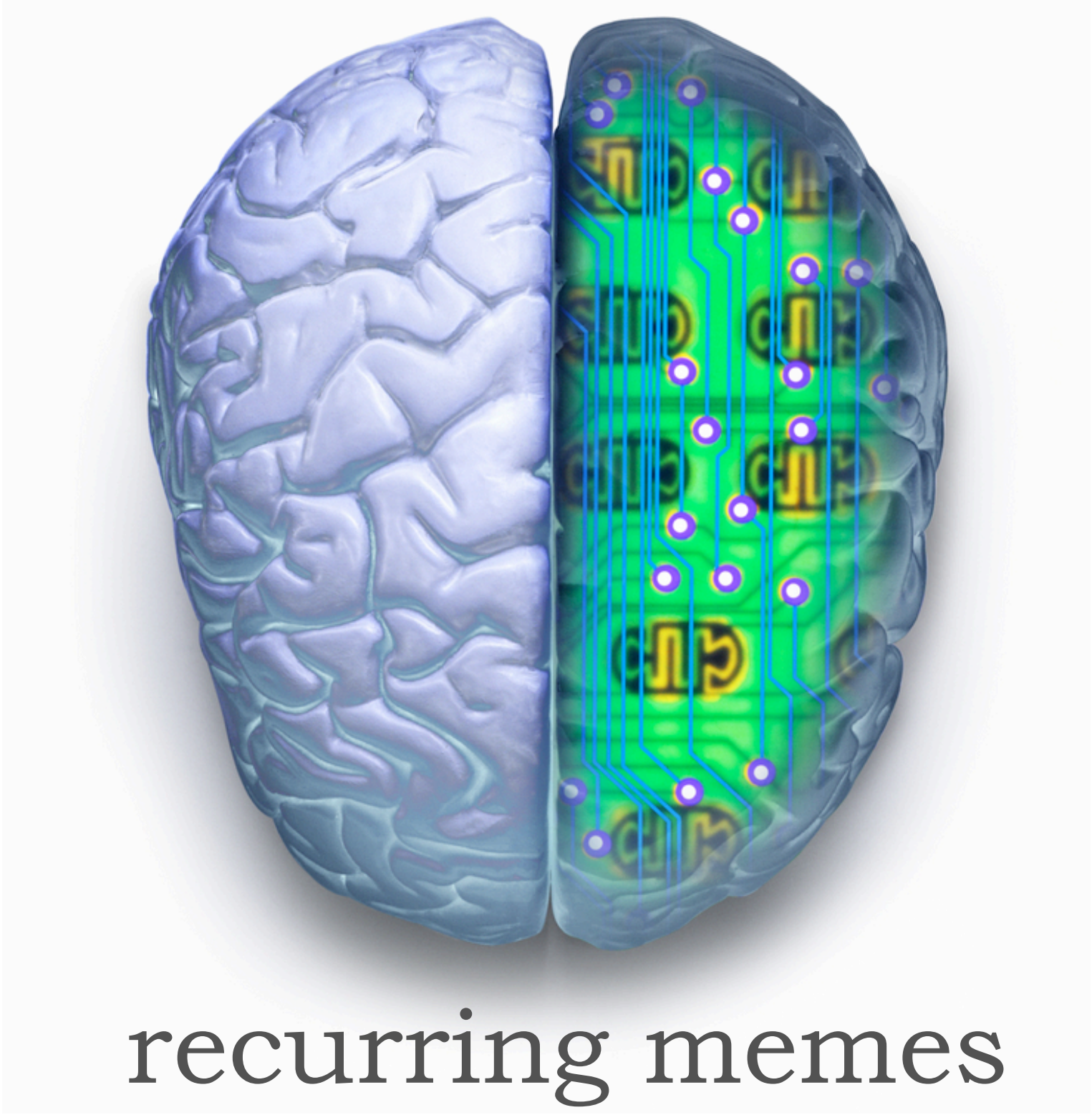
```
@Test(expected = InvalidNumberException.class)
public void cannot_classify_negative_numbers() {
    new Classifier6(-20);
}

@Test public void sum() {
    Classifier6 c = new Classifier6(20);
    calculateFactors(c);
    int expected = 1 + 2 + 4 + 5 + 10 + 20;
    assertThat(sumOfFactors(c), is(expected));
}
```



"Software may be cheap to build, but it is *incredibly* expensive to design."

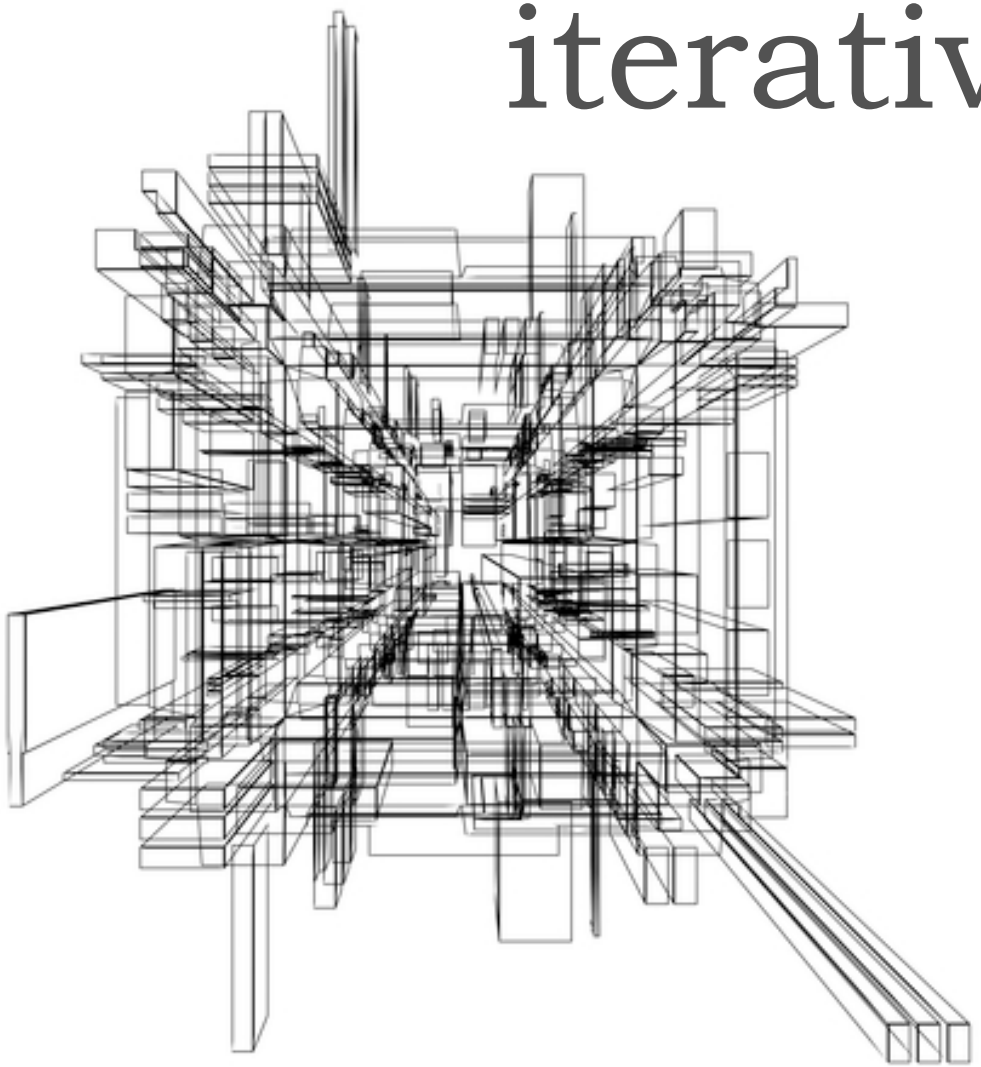
Jack Reeves



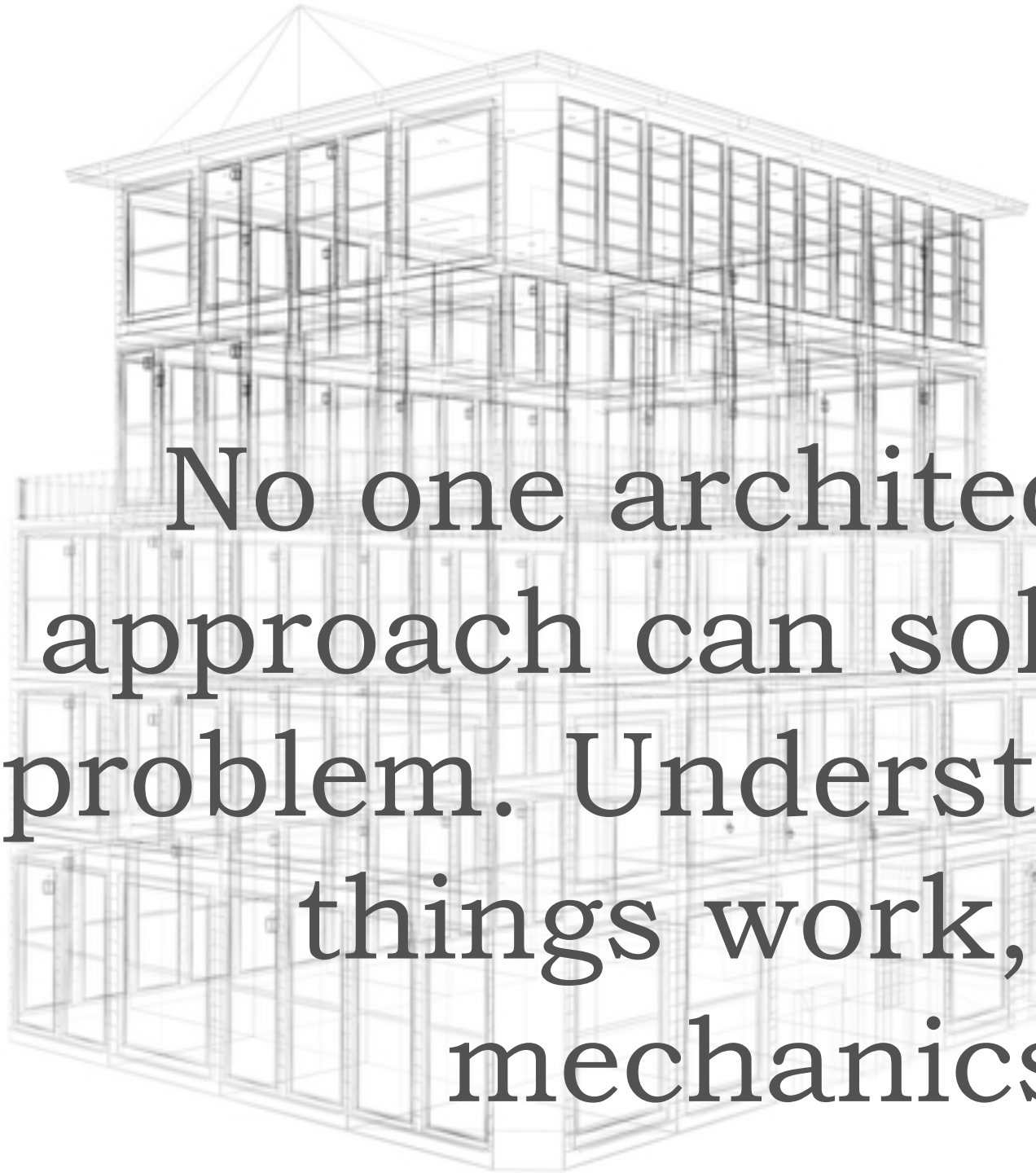
recurring memes

architect for change

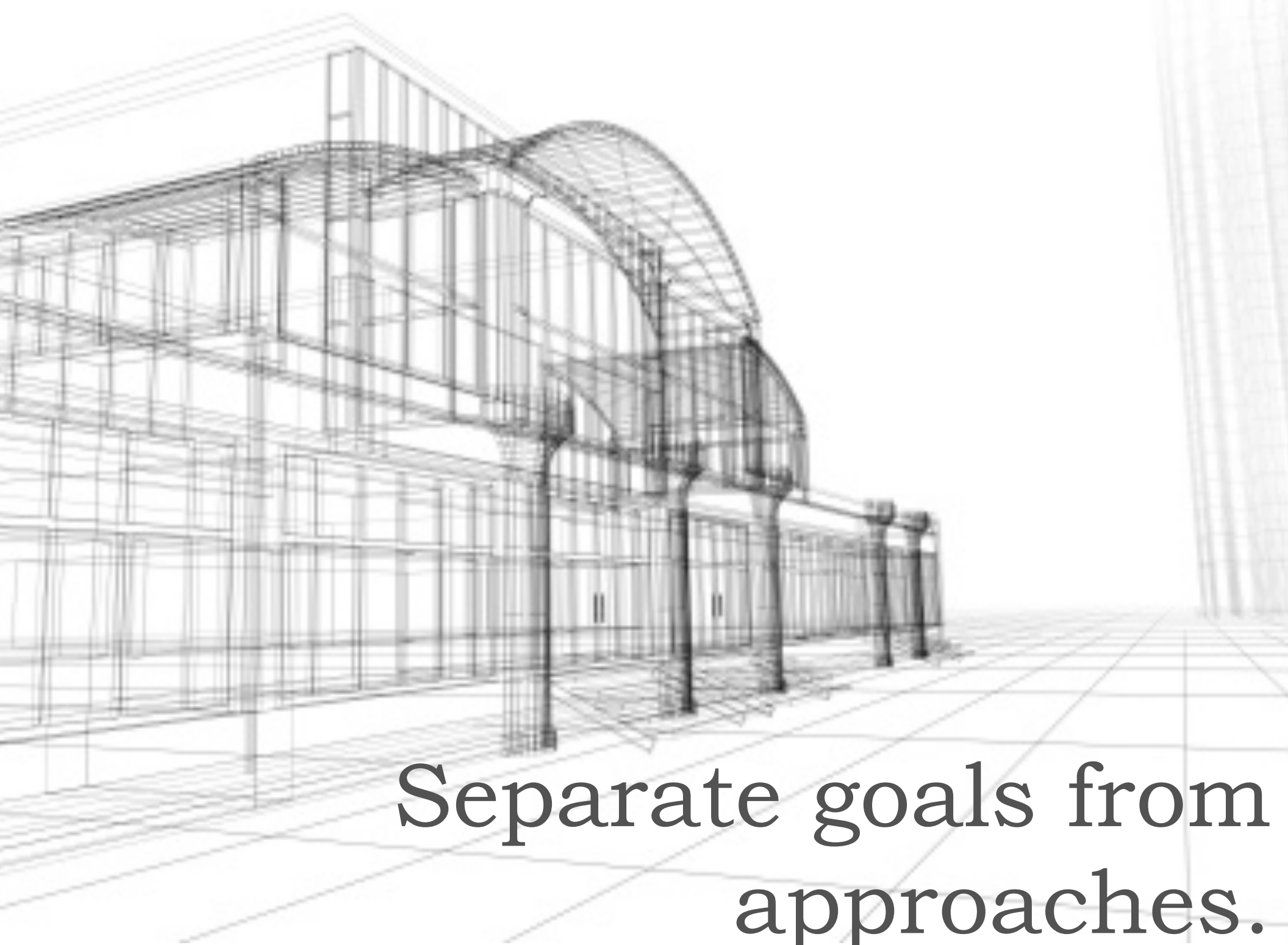
software always becomes iterative



(Agile just does it sooner).

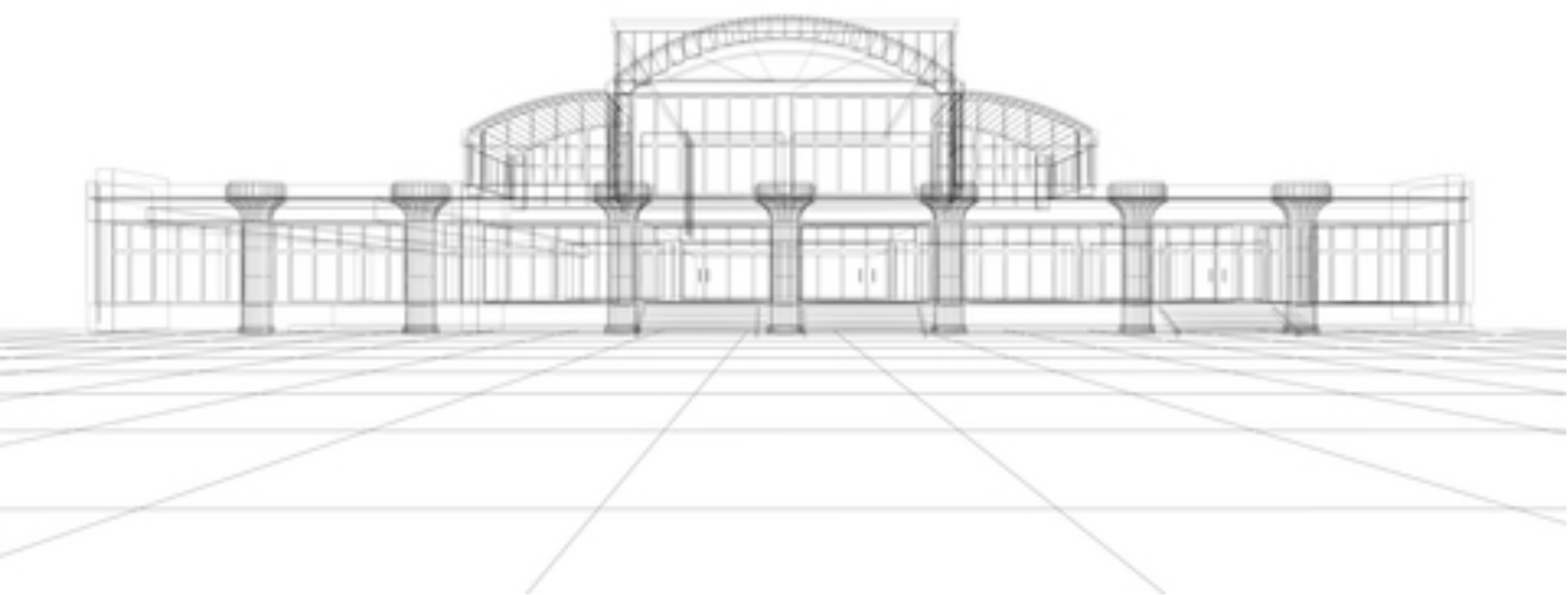


No one architecture/ approach can solve every problem. Understand ***why*** things work, not mechanics.

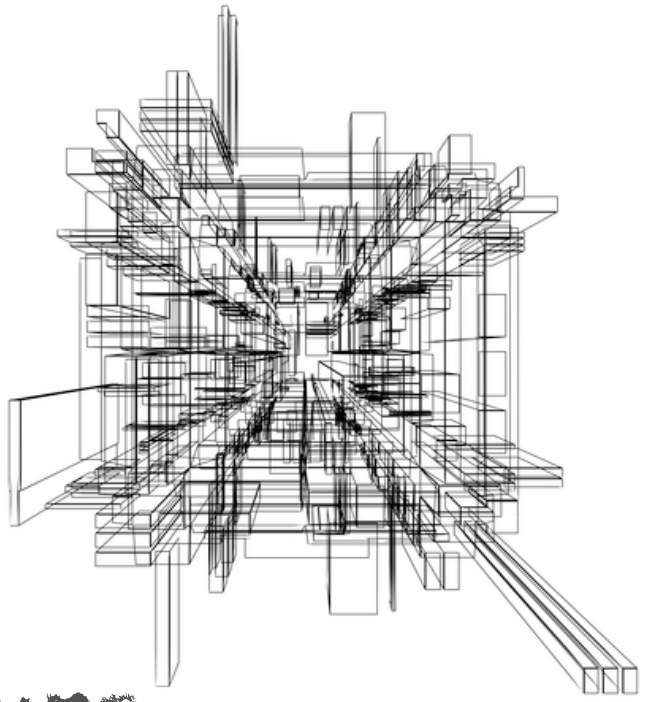


Separate goals from approaches.

Architecture isn’t an equation to be solved; it’s a snapshot of a process.

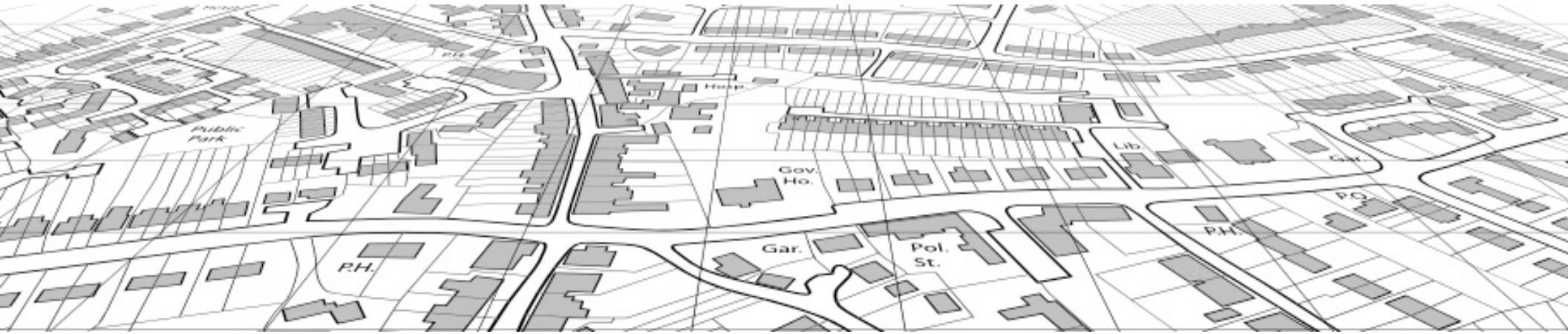


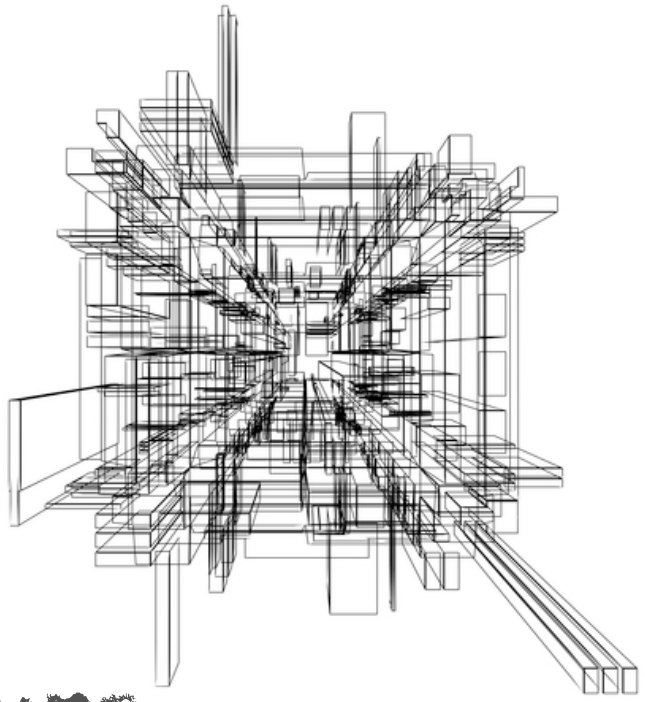

architecture is coupled to
process (especially
continuous delivery)



Software Architecture Fundamentals

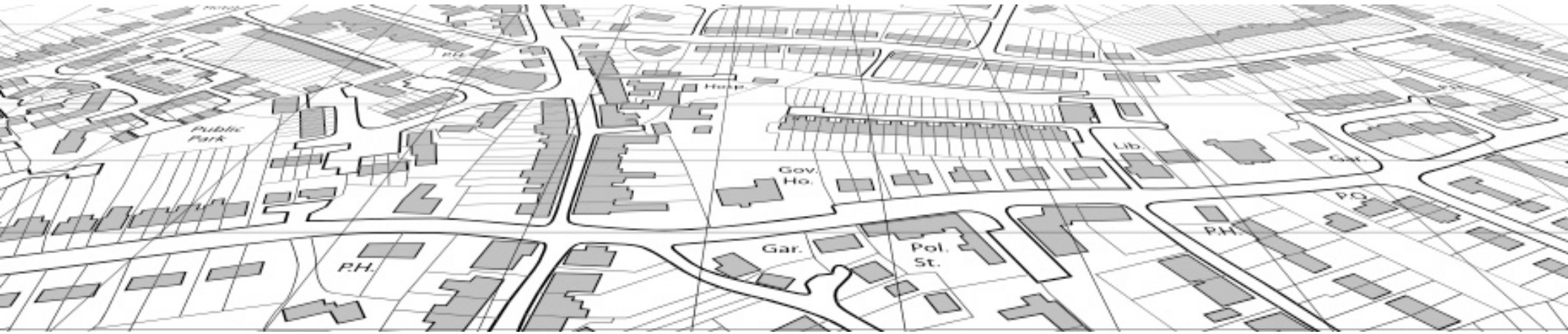
From Developer to Architect



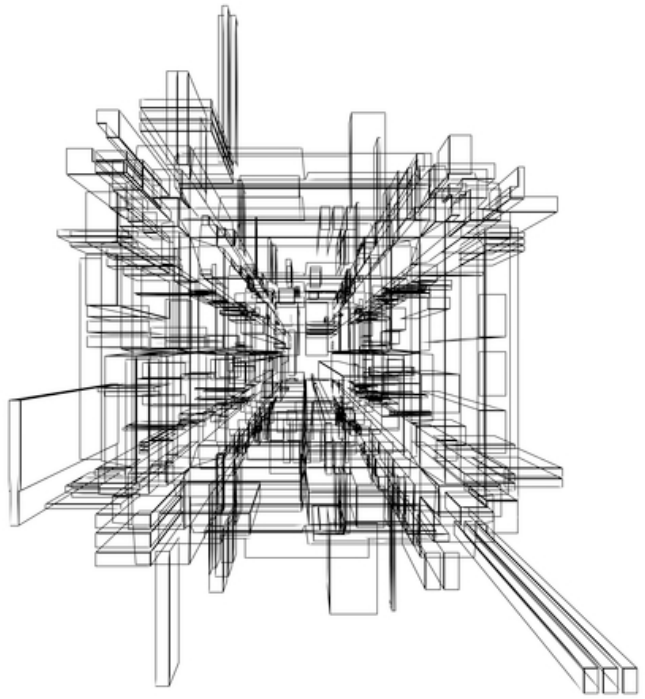


Software Architecture Fundamentals

A Deeper Dive



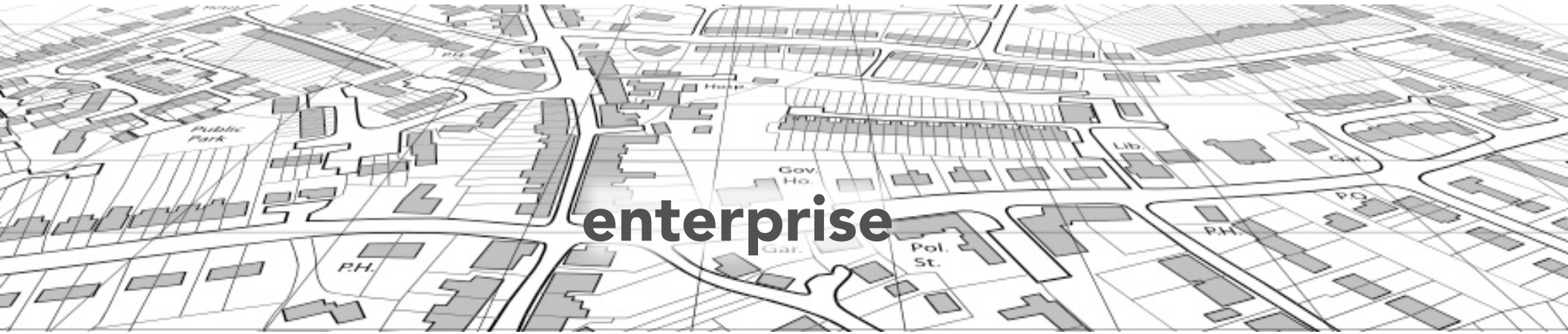
Software Architecture Fundamentals



application

integration

From Developer to Architect



enterprise

?’S

Mark Richards
Independent Consultant
Hands-on Enterprise / Integration Architect
Published Author / Conference Speaker

http://www.wmrichards.com
mark@wmrichards.com

Books:
Java Message Service, 2nd Edition
97 Things Every Software Architect Should Know

Neal Ford
Director / Software Architect /
Meme Wrangler

ThoughtWorks®

2002 Summit Blvd, Level 3, Atlanta, GA 30319, USA
T: +1 40 4242 9929 Twitter: @neal4d
E: nford@thoughtworks.com W: thoughtworks.com