

ABACUS

E.F. Schumacher
**Small
is Beautiful**



A STUDY OF ECONOMICS
AS IF PEOPLE MATTERED

ABACUS

E.F. Schumacher
**Small
is Beautiful**



A STUDY OF ECONOMICS
AS IF PEOPLE MATTERED

Small Is Beautiful

A talk on code as if
people mattered

A talk on code as if
economics mattered

@KevlinHenney

Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs.

The Report of the Brundtland Commission

ABACUS

E.F. Schumacher
**Small
is Beautiful**

**For every activity there is a
certain appropriate scale.**

A STUDY OF ECONOMICS
AS IF PEOPLE MATTERED

**To be, or not to be: that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them?**

William Shakespeare
Hamlet

Continuing existence or cessation of existence: those are the scenarios. Is it more empowering mentally to work towards an accommodation of the downsizings and negative outcomes of adversarial circumstance, or would it be a greater enhancement of the bottom line to move forwards to a challenge to our current difficulties, and, by making a commitment to opposition, to effect their demise?

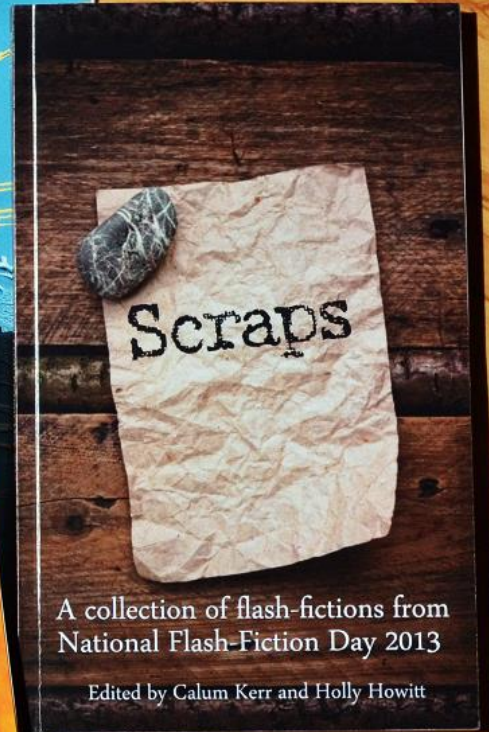
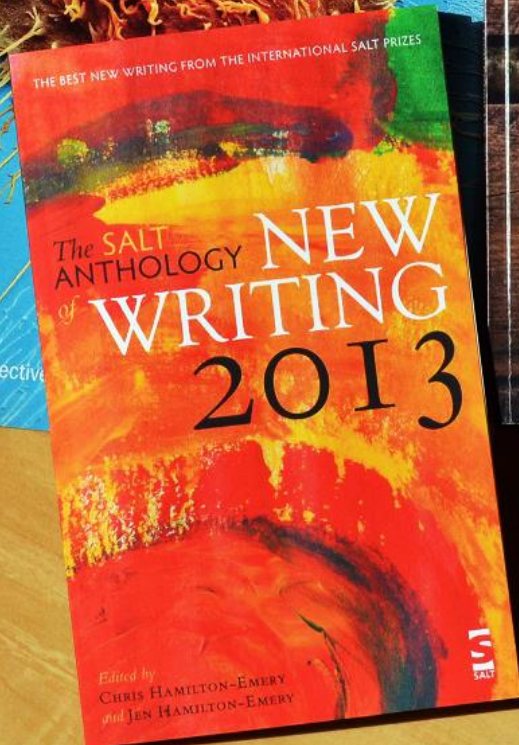
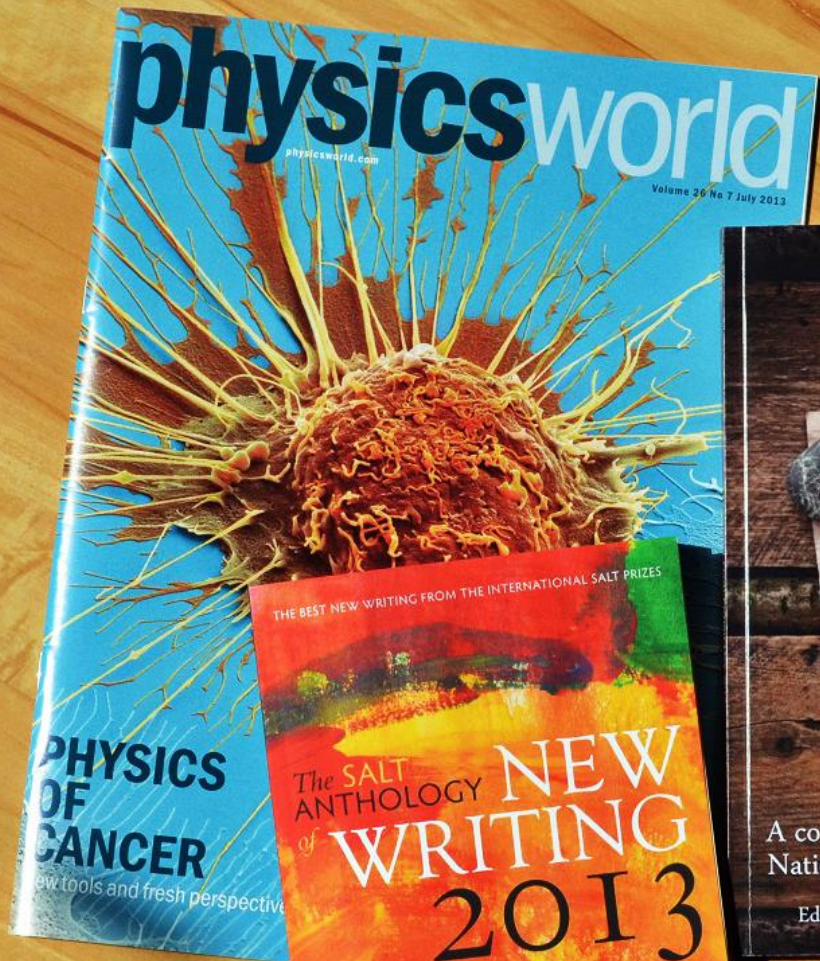
Tom Burton
Long Words Bother Me

Continuing existence or cessation of existence:

**those are the
more empow
to work towa
accommodat
downsizings
outcomes of
circumstance
a greater enh
the bottom li
forwards to a
our current d
by making a
opposition, t
demise?**

**This is the monstrosity in love, lady, that
the will is infinite, and the execution
confined; that the desire is boundless, and
the act a slave to limit.**

William Shakespeare
Troilus and Cressida





WILEY SERIES IN
SOFTWARE DESIGN PATTERNS

PATTERN-ORIENTED SOFTWARE ARCHITECTURE

On Patterns and Pattern Languages



Volume 5

Frank Buschmann
Kevlin Henney
Douglas C. Schmidt

**At some level
the style
becomes the
substance.**

/THEORY/IN/PRACTICE

Beautiful Code

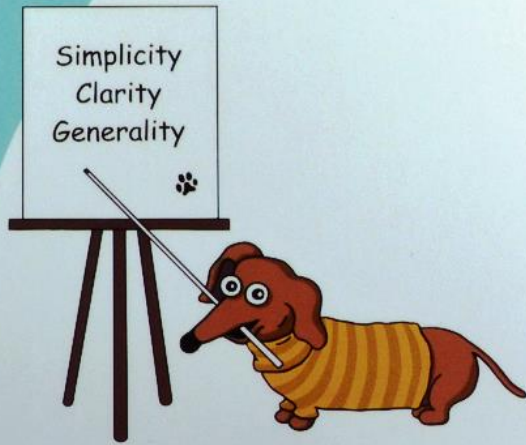
Leading Programmers Explain How They Think

O'REILLY

Edited by Andy Oram & Greg Wilson

The Practice of Programming

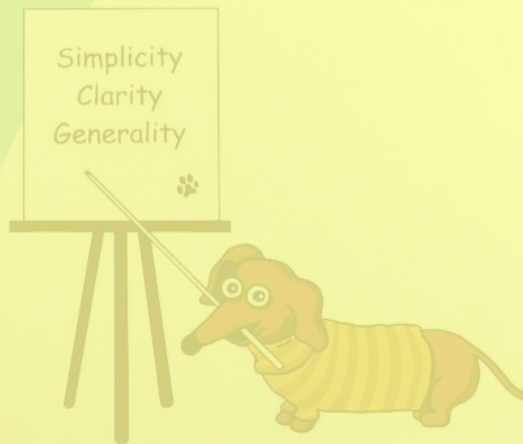
Brian W. Kernighan
Rob Pike



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

The Practice of Programming

Brian W. Kernighan
Rob Pike



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

```
/* grep: search for regexp in file */
int grep(char *regexp, FILE *f, char *name)
{
    int n, nmatch;
    char buf[BUFSIZ];

    nmatch = 0;
    while (fgets(buf, sizeof buf, f) != NULL) {
        n = strlen(buf);
        if (n > 0 && buf[n-1] == '\n')
            buf[n-1] = '\0';
        if (match(regexp, buf)) {
            nmatch++;
            if (name != NULL)
                printf("%s:", name);
            printf("%s\n", buf);
        }
    }
    return nmatch;
}

/* matchhere: search for regexp at beginning of text */
int matchhere(char *regexp, char *text)
{
    if (regexp[0] == '\0')
        return 1;
    if (regexp[1] == '*')
        return matchstar(regexp[0], regexp+2, text);
    if (regexp[0] == '$' && regexp[1] == '\0')
        return *text == '\0';
    if (*text != '\0' && (regexp[0] == '.' || regexp[0] == *text))
        return matchhere(regexp+1, text+1);
    return 0;
}

/* match: search for regexp anywhere in text */
int match(char *regexp, char *text)
{
    if (regexp[0] == '^')
        return matchhere(regexp+1, text);
    do {
        /* must look even if string is empty */
        if (matchhere(regexp, text))
            return 1;
    } while (*text++ != '\0');
    return 0;
}

/* matchstar: search for c*regexp at beginning of text */
int matchstar(int c, char *regexp, char *text)
{
    do {
        /* a * matches zero or more instances */
        if (matchhere(regexp, text))
            return 1;
    } while (*text != '\0' && (*text++ == c || c == '.'));
    return 0;
}
}
```

STR
16571 LD C A 79
(187) LD L A 195
(64) LD H A 38 67
16575 LD R (HL) 126
(131) LD B N 6 1
(64) AND N 230 127
CP H 254 8
JP Z DIS 40 28
INC B 4
CP N 254 110
JPDIS 40 15
CP H 254 39
JP C DIS 56 11
LD A (HL) 126 '0' DIFF
INC B 4
LD L N 46 25 '1' EMPTY
ADD (HL) 134
BIT 7, A 289 127 '2' HALL
JP Z DIS 40 2
LD B N 6 0 '3' SAME
LD A B 120
LD L C 195
RET 201

TABLES
16607 1 11 -1 -11 -10 -12 10 10
16615 13 -13 21 -21 23 -23 -9 9
16623 11 10 12
16626 54 55 39 51 53

PIECE
16631 XOR A 175
(247) LD (NN) A 50 70 64
(64) LD R (HL) 126
AND N 230 127
CP P 254 53
JP Z DIS 40 79
LD C H 14 1
LD B H 6 0
LD HL NN 33 231 64
CP N 254 51
JP Z DIS 40 22
LD L H 46 223
CP P 254 48
JP Z DIS 40 16
LD C B 72
CP 4 254 54
JP Z DIS 40 11
LD B H 5 4
CP R 254 55
JP Z DIS 40 5
LD L H 46 227
CP B 254 39
RET NZ 192

SHIFT
16882 LD HL NN 33 99 64
(242) LD DE NN 17 70 64
(65) LD BC NN 1 28 0
JP C DIS 56 1
EX DE HL 235
LDIR 237 176
RET 201

PSC
17162 AND N 230 127
(10) LD HL NN 33 242 64
(67) LD B N 6 5
CP (HL) 190
RET Z 200
INC HL 35
DJNZ DIS 16 251
LD A B 120
RET 201

NPSCAN
17046 XOR A 175
(150) LD (NN) A 50 65 64
(66) LD B H 6 86
LD HL NN 33 62 67
INC HL 35
PUSH HL 229
PUSH BC 197
LD E L 93
CALL STR2 205 191 64
CP N 254 3
JP NZ DIS 32 41
LD L E 107
LD (NN) HL 34 7 64
CALL MOVE 205 247 64
CALL TL 205 130 66
JP Z DIS 40 29
LD E A 95
LD D N 22 67
CALL MOVE 205 255 66
EXX 217
AND A 167

CALL SHIFT 205 242 65
CALL CHK 205 1 66
EXX 217
LD (HL) B 112
LD A C 121
LD (DE) A 18
JP C DIS 56 3
CALL SCORE 205 153 65
SCF 55
CALL SHIFT 205 242 65
JP DIS 24 222
POP BC 193
POP HL 225
DJNZ DIS 16 200
LD A (NN) 58 65 64
CP N 254 0
JP Z DIS 40 254
LD HL NN 33 69 64
LD A (HL) 126
DEC HL 43
LD E (HL) 94
LD D N 22 67
LD (DE) A 18
DEC HL 43
LD L (HL) 110
LD H D 98
BIT 0, L 209 69
(235) LD (HL) N 54 0
(66) JP Z DIS 40 2
LD (HL) N 54 120
CALL CHGMV 205 247 66
RET 201

INC
17176 LD A L 125
(24) EXX 217
(67) LD (NN) A 50 128 64
CALL SQ.AT 205 16 66
EXX 217
LD A C 121
RET 201

DRIVER
16909 LD B H 6 5
(63) LD A H 62 8
(66) LD HL NN 33 159 67
INC HL 35
LD (HL) A 119
DJNZ DIS 16 252
CALL KT 205 168 64
CP N 254 3
JP NZ DIS 32 238
LD (NN) HL 34 7 64
LD E L 93
CALL MOVE 205 247 64
LD HL NN 33 161 67
CALL KT 205 168 64
CP N 245 2
EX DE HL 235
JP NC DIS 48 220
CALL TL 205 130 66
JP Z DIS 40 215
CP C 185
JP NZ DIS 32 248
CALL MOVE 205 255 66
EXX 217
CALL CHK 205 1 66
EXX 217
JP C DIS 56 8
CALL CHG SQ 205 235 66
CALL NPSCAN 205 150 66
JP DIS 24 194
LD (HL) B 112
LD A C 121
LD (DE) A 18
JP DIS 24 249

TEST LIST
17026 LD HL NN 33 70 64
(130) DEC (HL) 53
(66) LD A (HL) 126
INC A 68
RET Z 200
ADD L 133
LD L A 111
LD A (HL) 126
RET 201

PANN

16721 LD A (HL) 126
(81) AND N 230 128
(65) LD HL NN 33 228 64
JP NZ DIS 32 2
LD L N 46 241
LD D N 22 3
LD A E 123
ADD (HL) 134
PUSH HL 229
PUSH AF 245
CP N 254 63
JP C DIS 56 32
CP N 254 148
JP NC DIS 48 28
CALL STR 205 187 64
CP N 254 0
JP Z DIS 40 20
CP N 254 1
JP NZ DIS 32 17
LD A D 122
CP N 254 1
JP NZ DIS 32 12
CALL ALIST 205 141 66
LD A E 123
CP N 254 82
JP C DIS 56 19
CP N 254 126
JP NC DIS 48 15
POP AF 241
POP HL 225
DEC INC HL 43
DEC D 21
JP NZ DIS 32 210
RET 201
LD A D 122
CP N 254 1
CALL NZ ALIST 196 141 66
JP DIS 24 241
POP AF 241
POP HL 225
LD E A 95
JP DIS 24 197

CHK
16897 LD A (NN) 58 55 67
(1) ADD N 198 48
(65) LD HL NN 33 62 67
LD B A 71
CP R 237 177
DEC HL 43
LD (NN) HL 34 120 64
LD B N 6 86
16912 LD HL NN 33 62 67
(16) INC HL 35
(66) PUSH HL 229
PUSH BC 197
LD E L 93
CALL STR2 205 191 64
CP 0 254 0
JP NZ DIS 32 25
CALL CHEMV 205 247 66
LD L E 107
CALL MOVE 205 247 64
CALL CHEMV 205 247 66
CALL TL 205 130 66
JP Z DIS 40 10
LD HL (NN) 42 128 64

CP L 189
JP NZ DIS 32 245
POP BC 193
POP HL 225
SCF 55
RET 201
POP BC 193
POP HL 225
DJNZ DIS 16 216
AND A 167
RET 201

SCORE
16793 PUSH HL 229
(153) PUSH BC 197
(65) PUSH DE 213
PUSH HL 229
PUSH BC 197
LD D L 85
LD HL NN 33 64 64
CALL NN 205 36 7
CALL PSC 205 10 67
LD A B 120
ADD A H 132
LD C A 79
POP AF 241
CALL PSC 205 10 67
POP HL 225
CALL INC 205 24 67
JP NC DIS 48 1
ADD A B 128
LD C A 79
POP HL 225
POP DE 209
LD E (HL) 94
LD (HL) D 114
PUSH HL 229
PUSH DE 213
CALL INC 205 24 67
JP NC DIS 48 1
SUB B 144
PUSH AF 245
CALL CHEMV 205 247 66
CALL CHK 205 1 66
POP BC 193
JP NC DIS 48 2
INC B 4
INC B 4
POP DE 209
POP HL 225
LD (HL) E 115
POP HL 225
CALL CHG 205 250 66
CALL INC 205 24 67
JP NC DIS 48 1
DEC B 5
CALL CHG 205 250 66
CALL CHEMV 205 247 66
LD A B 120
LD HL NN 33 60 64
LD (HL) A 119
EX DE HL 235
LD HL NN 33 65 64
CP (HL) 190
RET C 216
LD BC NN 1 5 0
JP DIS 24 11

begin 644 FLOPPY.7Z

M-WJ\KR<<` `/ \$P0TX"0, ``````! = ``````````):A<QL`=0%+!`0%F, \C) .-Z
M;\$=*5]JZ\`-6, ?XTW!KSPJ'1-6G\$ARFF9!HS3, #)')1N7*D;8SS;"#0\60AE
M-[CNTB+D/0Z9#=:_9P2KN#\MYXC�QEJ4BH1SX>/1=Y.`CHNM%WS\8U`\$`F]A
ME2F#R_9K`29FA1V18\#-B:U'D*!T7:#C@1E\F9V(:ADLP`IM:_8Y*V'DZ8&R
M^3E%4FY*)6'@1\[, #&`C8,] ([!@(HMOQ"W\G<F[88BP>-@<]WC*%UH*N^X%-
M[WY\$\$I90UY2)N@1, XML;SK@NNA3&38?K<%CX1W4"L5!5W2S395P18V+\0=V"
MJWNAR%@(*3_ . :<P- :G1ZO [%@A5**6J&Y3)!<5HFF0B, HLD, J\S:U. / [EM: -X
MYP8KC)<=B(;IT":8((FP%_*K904!T?\7\4K, +.C040HG9SC[0["\!W"M%_L^
M?%\$0.4L.X3.G&\+FC?Y;9(`J+U". :ZCS>H, 60^&21__MUPNQBLD&>EBP2%S<
M%:(#KYU5?5E?R;S>). `S)AMFCK^>;\$X&3\C4H)>2V_I+I3G29!?CM^@P'A1\
M;>P"I59*Q3Y71<^834CZT/<14KJ.U*`42>PDP%N^." :ED\/)96K[*YB5#=NW
M;.JWHJ'C-GI!UCI1&7"#DB;Z4VE'/\$W7/X^=\$<H_+>Z1MR9-NU":DB8Q:5=>
M1\VA)\TS&7OZZ:IKAN+KC:P=4?76V>:H:24^AT4S;JOY)@JWNMC>`AUI-K)+
M,Y2F890+C'`)T1\$B258[[S\F6D>&=F>2@"56SM+N&<\B+\Z_G-H`/JTFF\$``
MC[496E4YX;VB/E9E?DCTT'16@YP\$;LR]V>B*K9<M_R[KFAVZSK\$G&%Q9ANEF
M4EB^Z7:L6>3E6P4(^<?:K?S[4BMTS1Y;(\$+YW5,]^"ED"3N`RRILW[4[\,2]
M+E^J#SY+9D*0\$P[_\$)/X<7A9^`O-_Y4H1@^I_'S>^YHP+E;`CX7S@X'`9<0E
M4_CUD38Q!:6P[F_!<\$U'#-&1\$:JM8!VZSK\$G&%Q9ANEF4EB^Z7:L*"W<Q``!
M!`8` `0F#"0`'"P\$``2, #`0\$%70``&``, U@"```@*`3?;3\$L```4!\$1T`0@!0
M`&\`= `!#`&@`90!S`', `+@!I`&T`9P```!0*`0!.7V=GM#G0`14&`0`@(```
"````

end

"After 20 years, this is still the best exposition of the workings of a 'real' operating system."
Ken Thompson

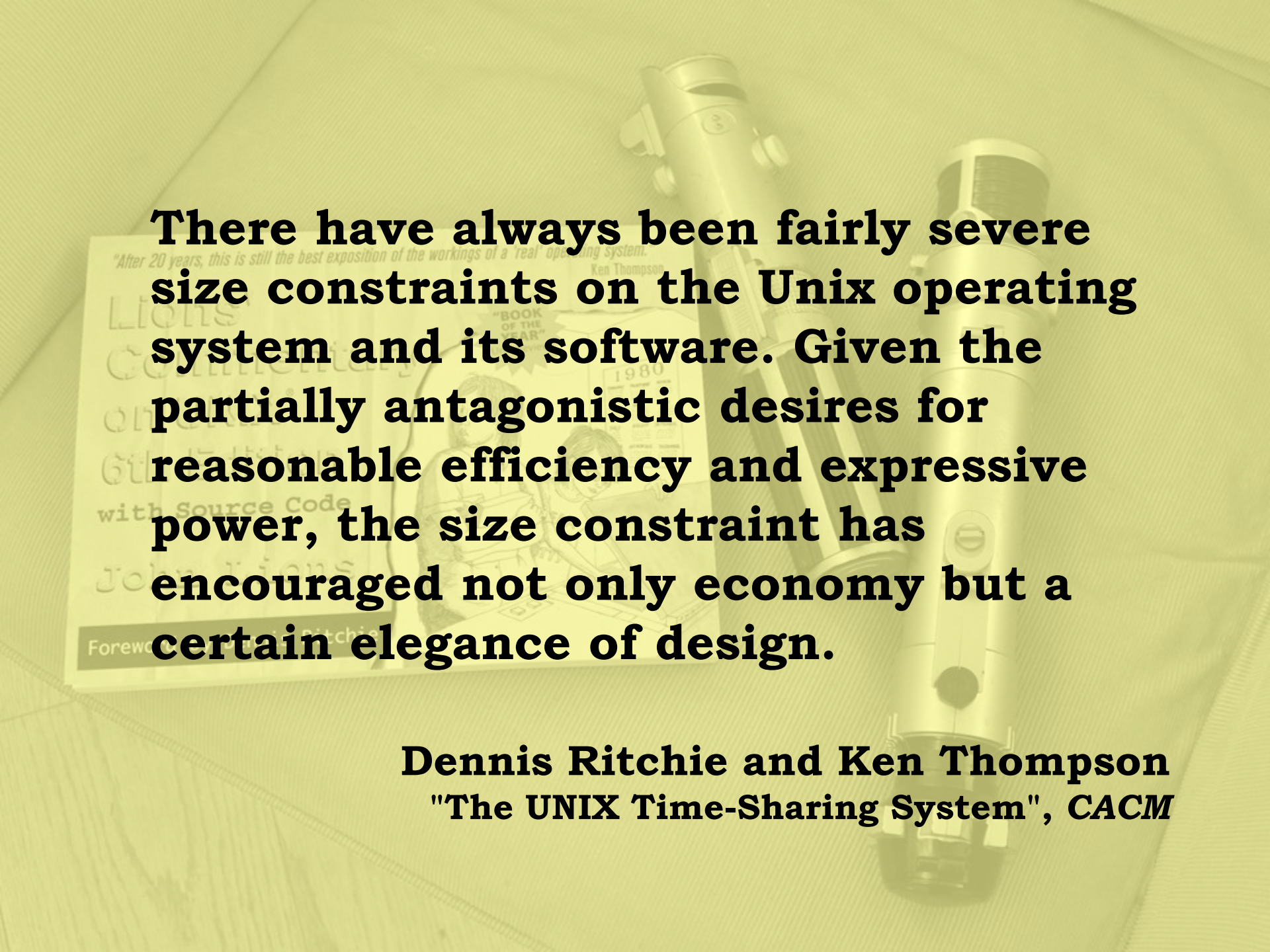
Lions' Commentary on UNIX®

6th Edition
with Source Code

John Lions

Foreword by Dennis Ritchie





There have always been fairly severe size constraints on the Unix operating system and its software. Given the partially antagonistic desires for reasonable efficiency and expressive power, the size constraint has encouraged not only economy but a certain elegance of design.

**Dennis Ritchie and Ken Thompson
"The UNIX Time-Sharing System", CACM**

Almost 19 years ago, when I was Byte magazine's executive editor and Web developer, I received the following email:

From: Ward Cunningham
Subject: wiki
Date: May 22, 1996 at 11:23:10 AM PDT
To: Jon Udell

Jon -- So what do you think of wiki? I put it up a year and a half ago when HTML authoring tools were in short supply. I think it has held up pretty well, though I suppose its days are numbered. Still, there is something about it we call WikiNature that is in short supply on the net and in computers in general. Regards. -- Ward

```

#!/usr/bin/perl
# ----- PerlInterpreter
# PerlInterpreter must be the first line of the file.
#
# Copyright (c) 1995, Cunningham & Cunningham, Inc.
#
# This program has been generated by the HyperPerl
# generator. The source hypertext can be found
# at http://c2.com/cgi/wikibase. This program belongs
# to Cunningham & Cunningham, Inc., is to be used
# only by agreement with the owner, and then only
# with the understanding that the owner cannot be
# responsible for any behaviour of the program or
# any damages that it may cause.
# ----- InitialComments

```

```

# InitialComments
print "Content-type: text/html\n\n";
$DBM = "/usr/ward/$ScriptName";
dbmopen(%db, $DBM , 0666) | &AbortScript("can't open $DBM");
$CookedInput{browse} && &HandleBrowse;
$CookedInput{edit} && &HandleEdit;
$CookedInput{copy} && &HandleEdit;
$CookedInput{links} && &HandleLinks;
$CookedInput{search} && &HandleSearch;
dbmclose (%db);
if ($ENV{REQUEST_METHOD} eq POST) {
$CookedInput{post} && &HandlePost;
}
# &DumpBinding(*CookedInput);
# &DumpBinding(*old);
# &DumpBinding(*ENV);
# ----- WikiInHyperPerl

```

As it turns out that wiki was less fragile than I feared -- and its days were more numerous than Ward had expected. But on Feb. 1, 2015, [this notice](#) appeared:

Wiki Wiki System Notice

After twenty years of service I'm pleased to announce a complete rewrite of wiki as a single page application with a distributed database which will last us for at least 20 years, maybe 200.

see <http://c2.fed.wiki.org>

ABACUS

E.F. Schumacher

Small
Beautiful

Today, we suffer from an almost universal idolatry of giantism. It is therefore necessary to insist on the virtues of smallness — where this applies.

A STUDY OF ECONOMICS
AS IF PEOPLE MATTERED

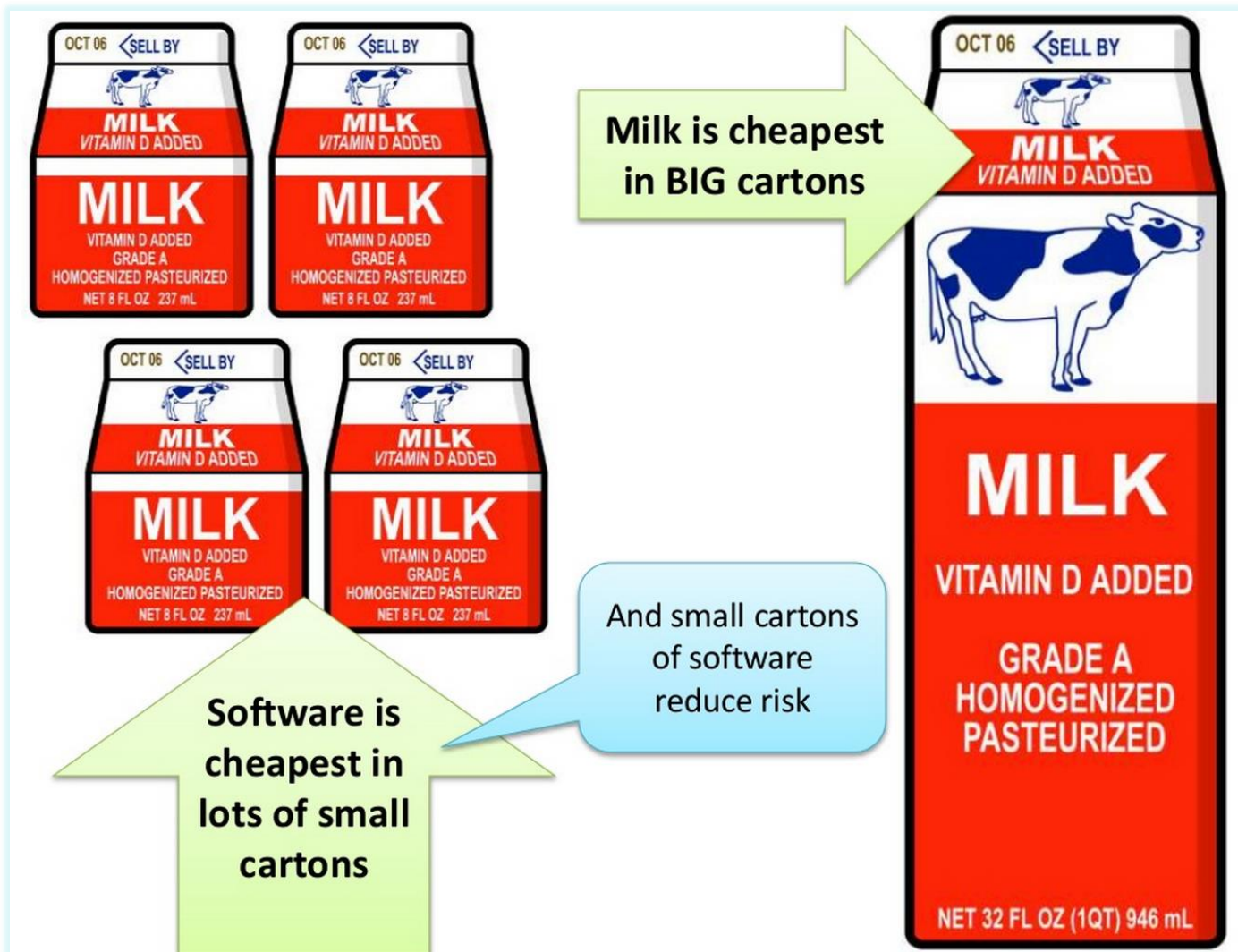
Software development does *not* have economies of scale.

Development has *diseconomies* of scale.

Allan Kelly

Beyond Projects

<http://www.slideshare.net/allankellynet/no-procects-beyond-projects-refreshed-version>



Allan Kelly
Beyond Projects

ABACUS

E.F. Schumacher

Small
is Beautiful

I say, therefore, that we think *with*
or *through* ideas and that what we
call thinking is generally the
application of pre-existing ideas to
a given situation or set of facts.

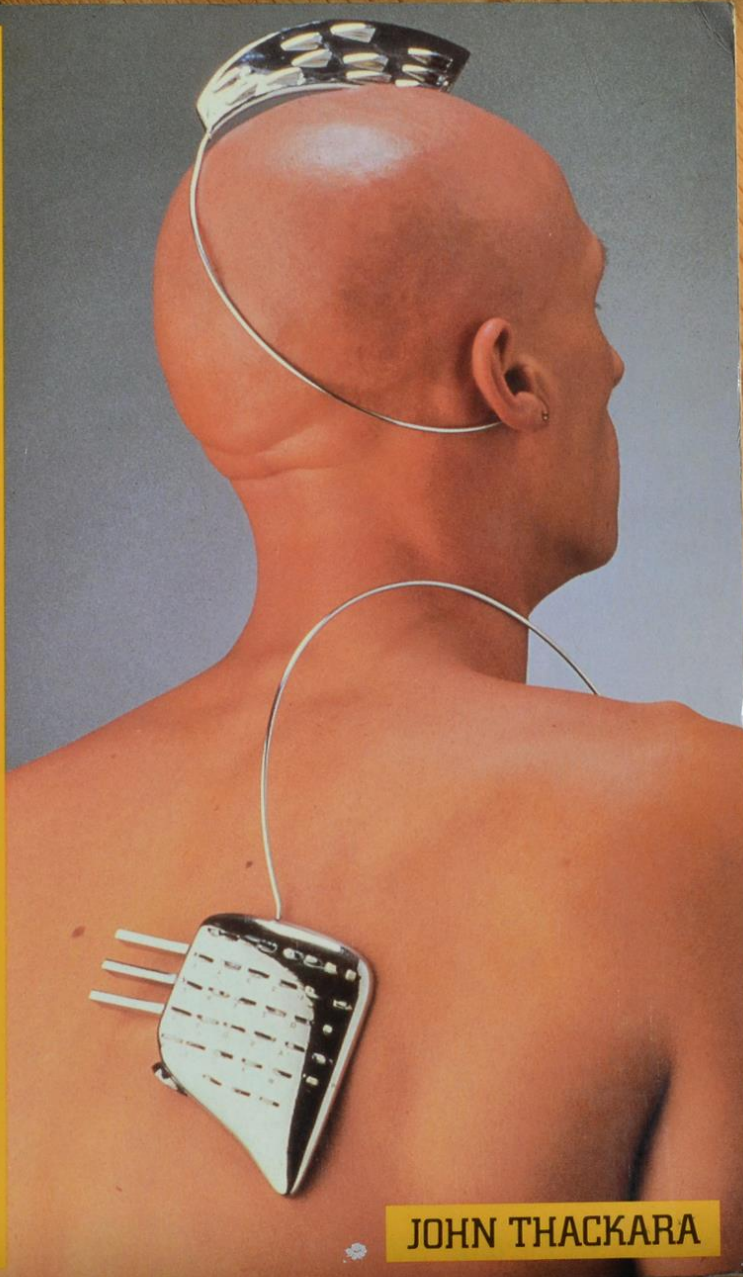
A STUDY OF ECONOMICS
AS IF PEOPLE MATTERED

The
Timeless Way of
Building



Christopher Alexander

DESIGN AFTER MODERNISM

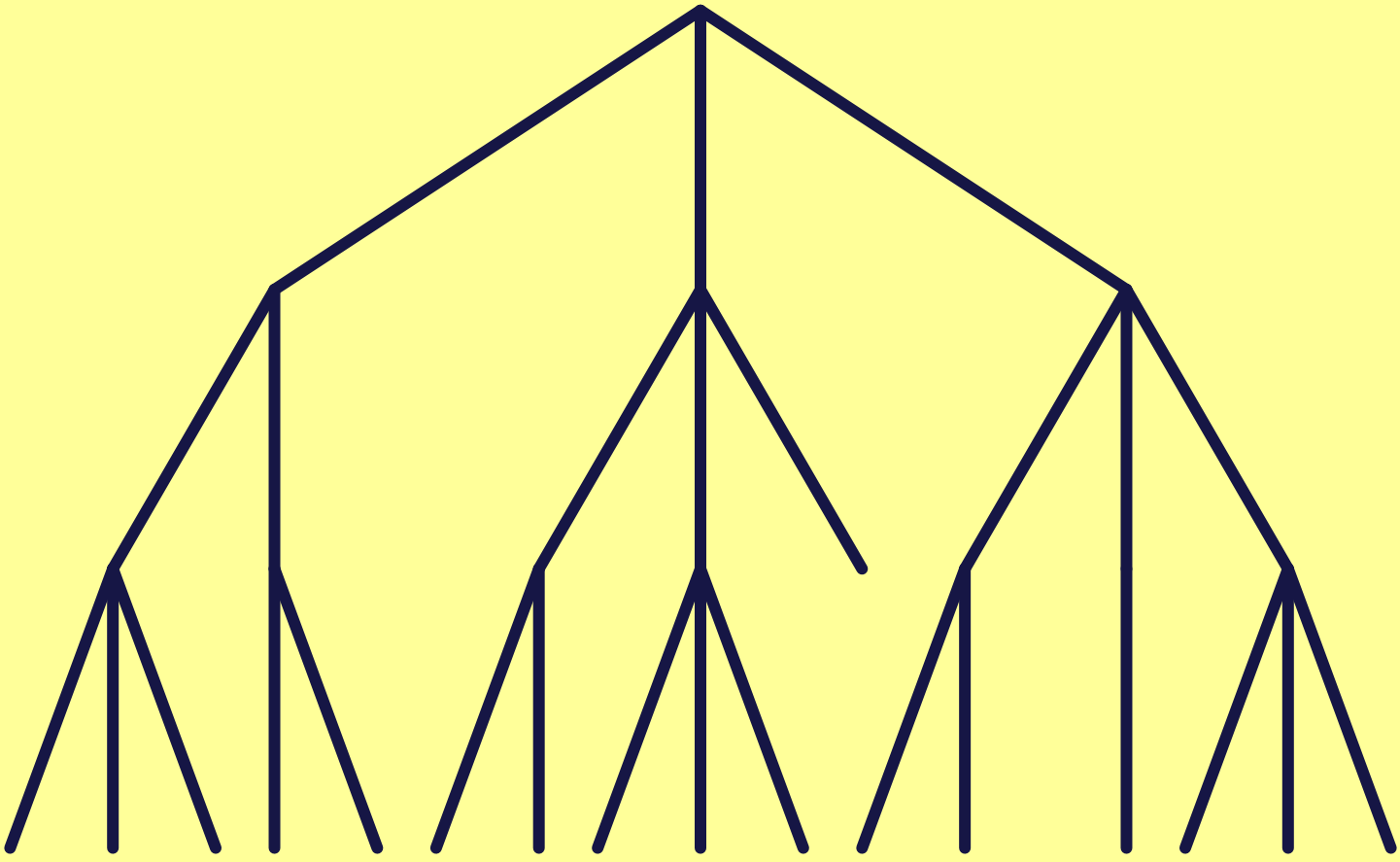


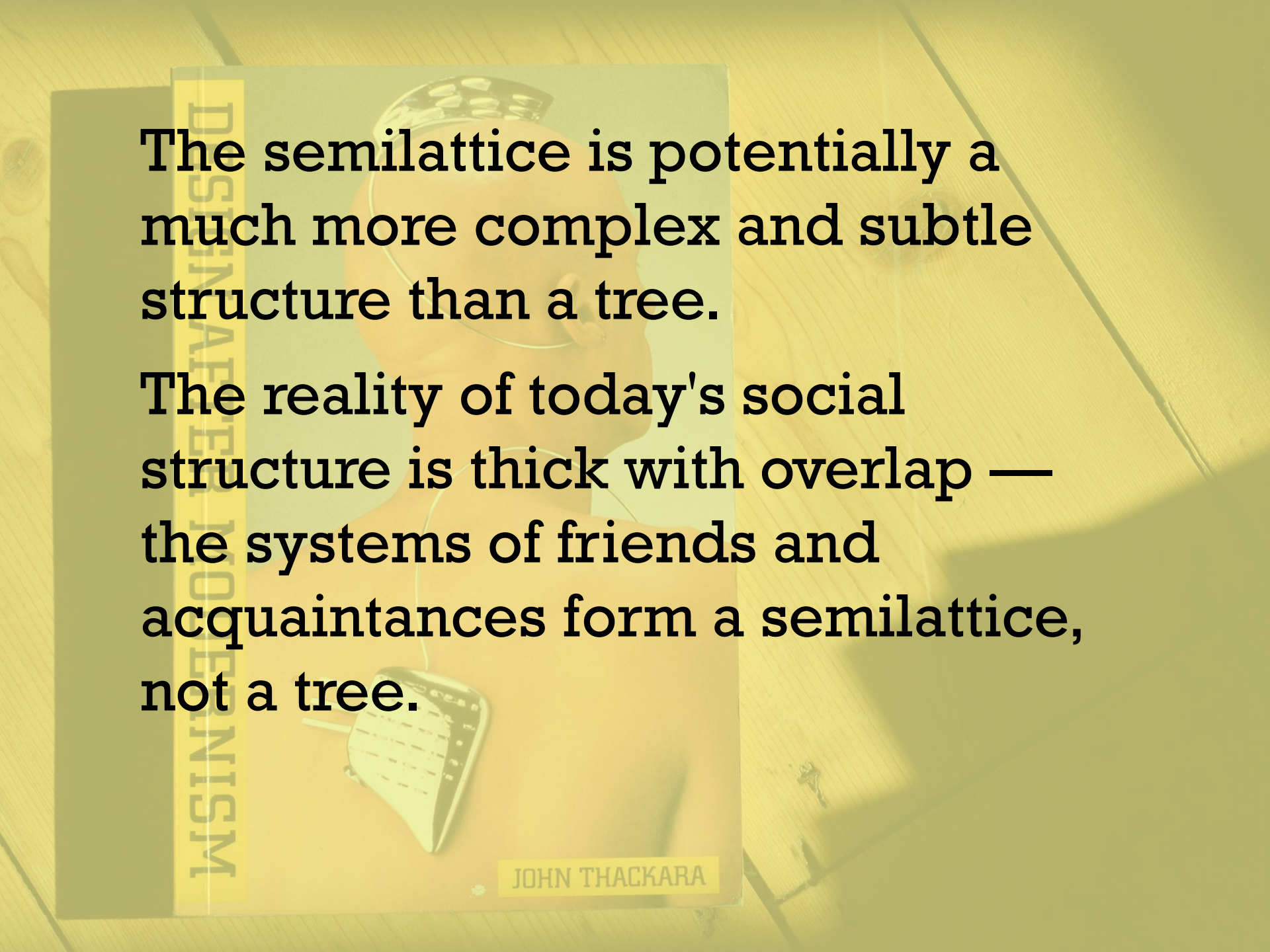
JOHN THACKARA

A city is not a tree
Christopher Alexander

The tree of my title is not a green tree with leaves. It is the name of an abstract structure.

Both the tree and the semilattice are ways of thinking about how a large collection of many small systems goes to make up a large and complex system.





The semilattice is potentially a much more complex and subtle structure than a tree.

The reality of today's social structure is thick with overlap — the systems of friends and acquaintances form a semilattice, not a tree.

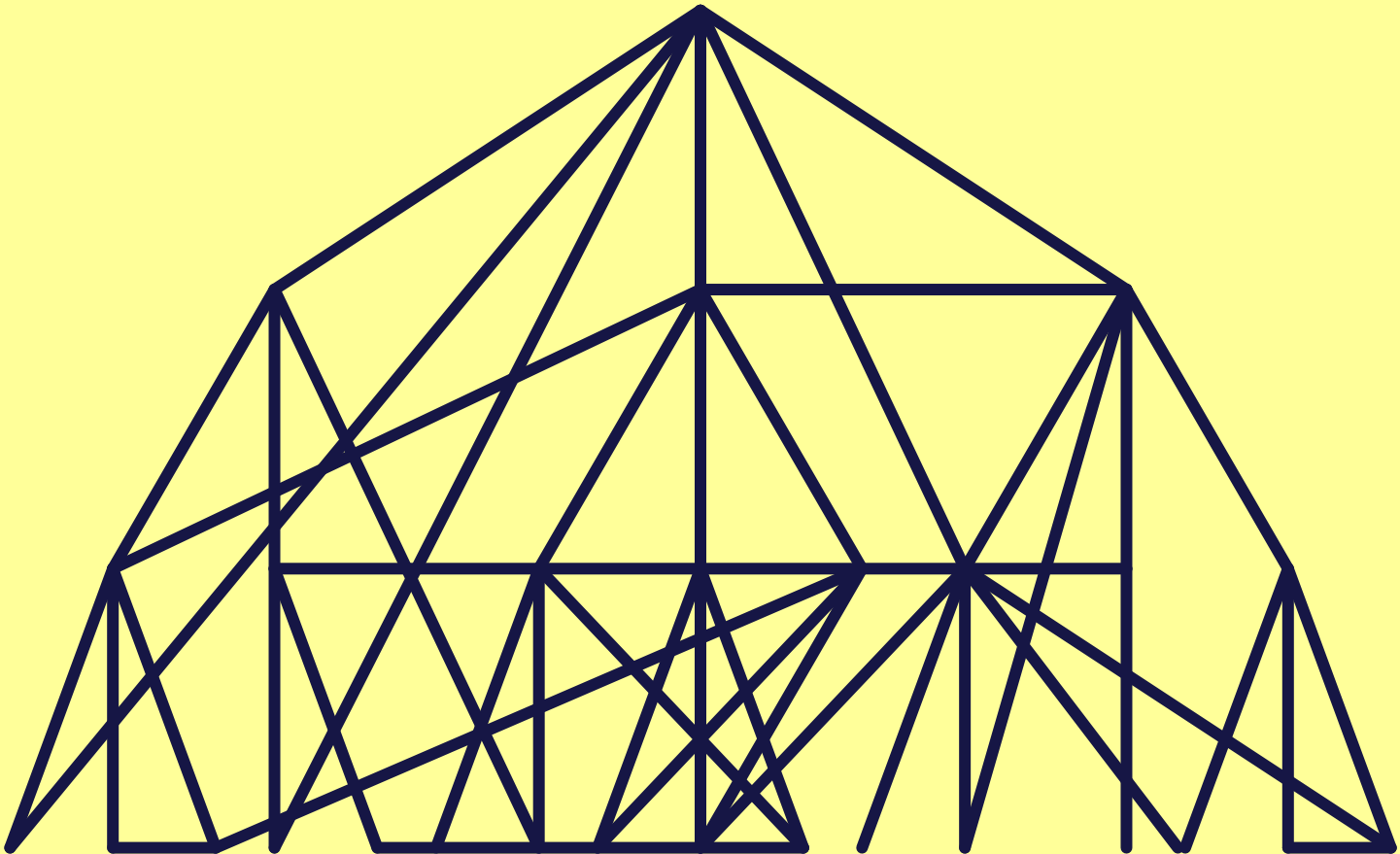


Figure 1
 The system, shown at the top, communicates with the outside world through the three interfaces 1, 2, and 3. The middle figure shows the major subsystems, two of which are shown in detail at the bottom.

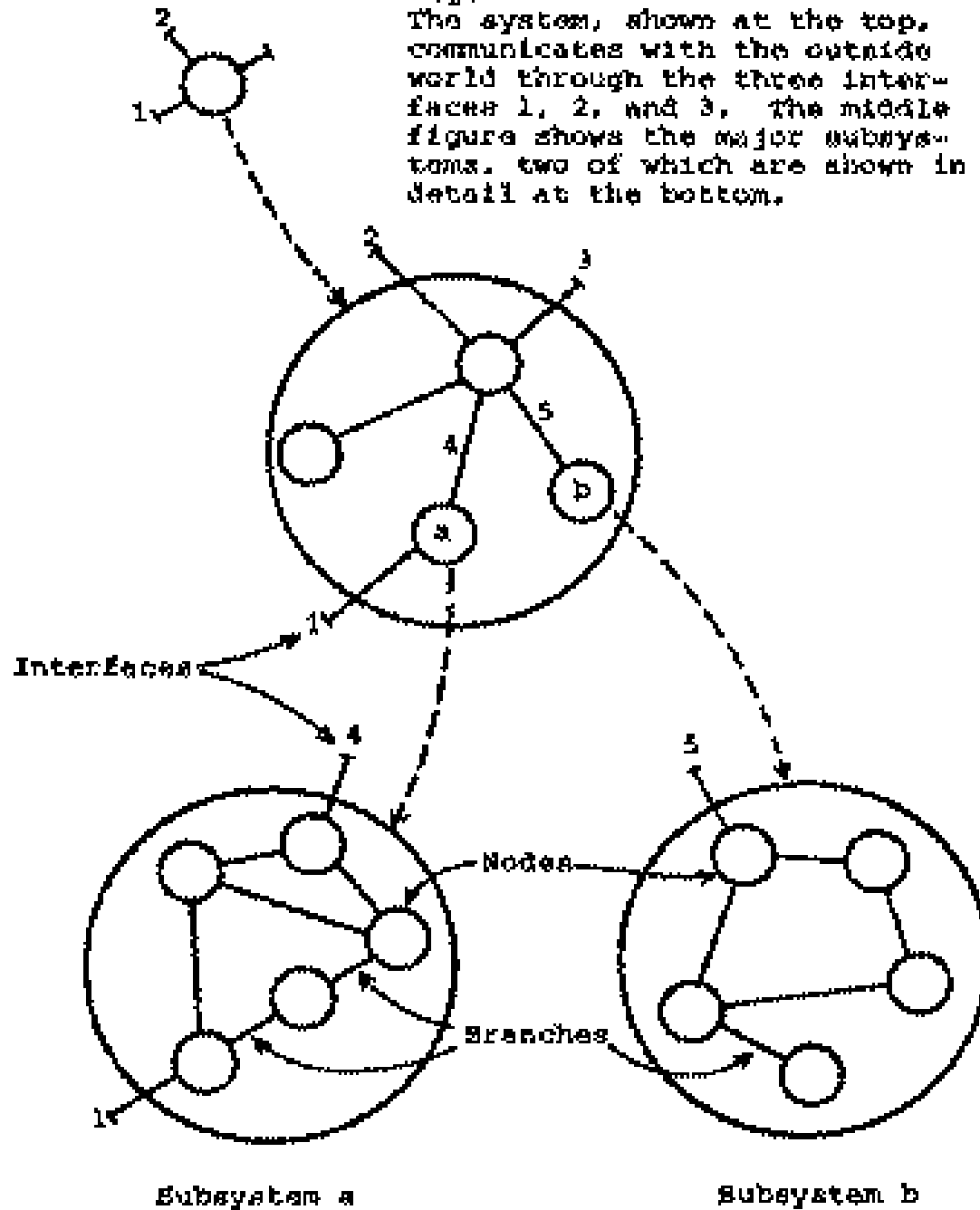
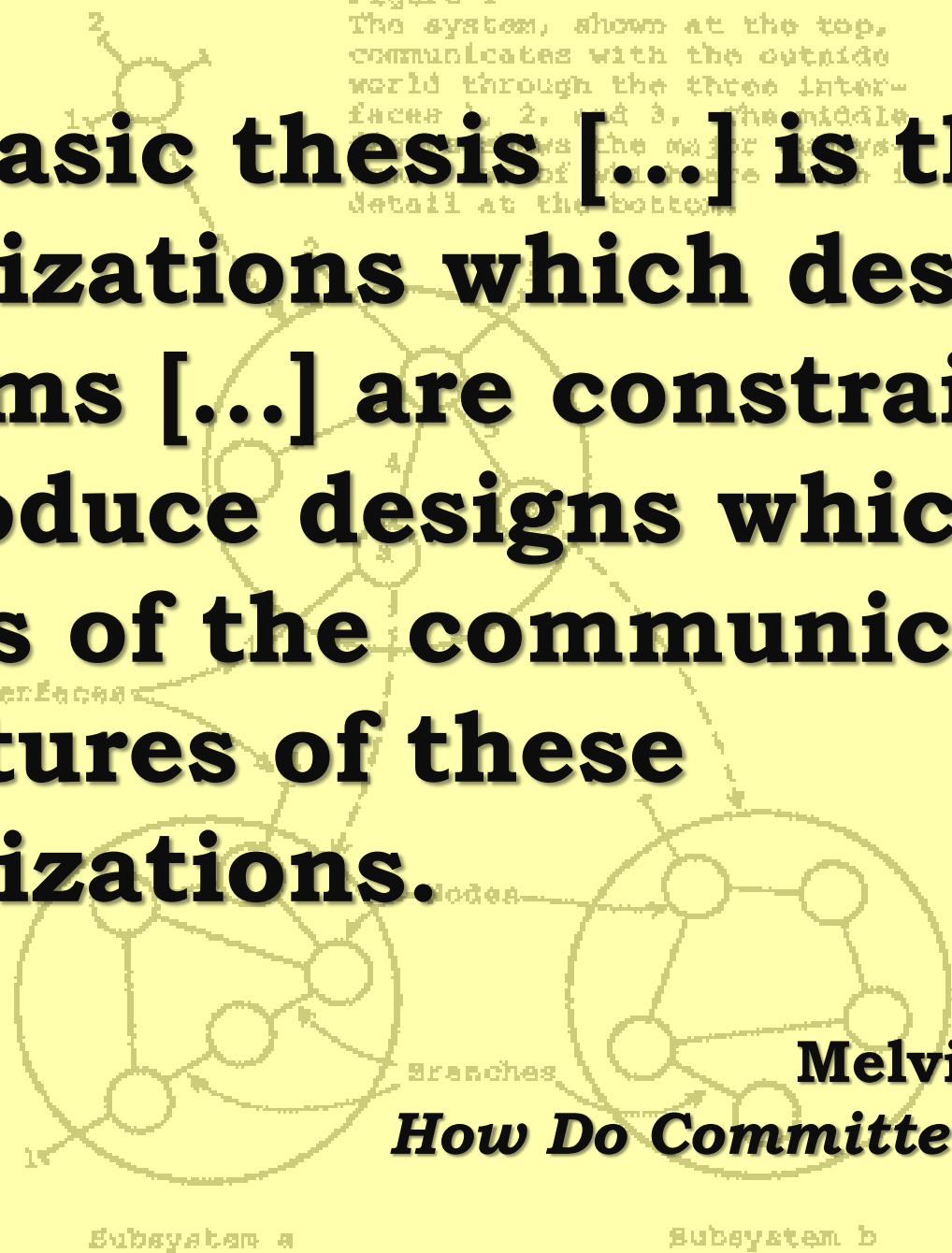


Figure 1
The system, shown at the top, communicates with the outside world through the three interfaces 1, 2, and 3. The middle part shows the major nodes of the system in detail at the bottom.

The basic thesis [...] is that organizations which design systems [...] are constrained to produce designs which are copies of the communication structures of these organizations.



Melvin Conway
How Do Committees Invent?

Subsystem a

Subsystem b

View Definition: Conway's Law

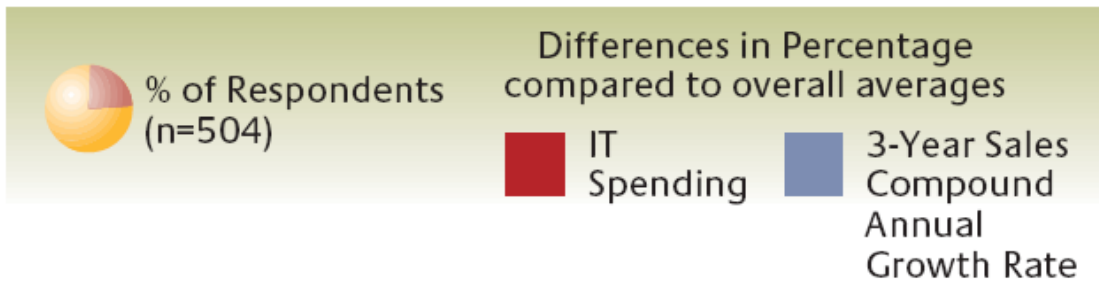
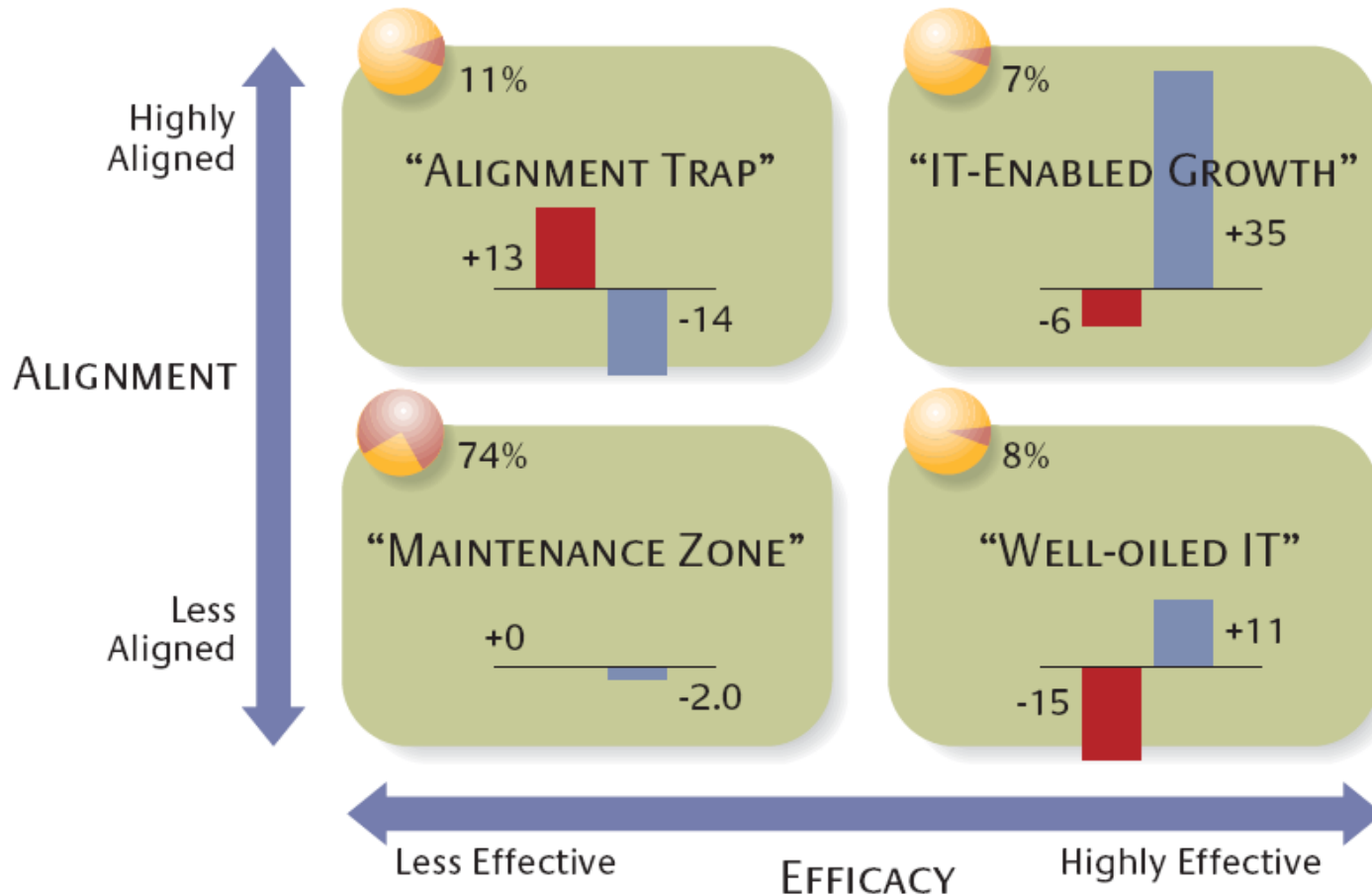
Conway's Law

prov. The rule that the organization of the software and the organization of the software team will be congruent; originally stated as "If you have four groups working on a compiler, you'll get a 4-pass compiler".

This was originally promulgated by Melvin Conway, an early proto-hacker who wrote an assembler for the Burroughs 220 called SAVE. The name `SAVE' didn't stand for anything; it was just that you lost fewer card decks and listings because they all had SAVE written on them.

IT's effort to satisfy its various (and sometimes conflicting) business constituencies created a set of Byzantine, overlapping systems that might satisfy individual units for a while but did not advance the company's business as a whole.

David Schpilberg, Steve Berez, Rudy Puryear and Sachin Shah
"Avoiding the Alignment Trap in Information Technology"
MIT Sloan Management Review
<http://sloanreview.mit.edu/article/avoiding-the-alignment-trap-in-it/>



Aligning a poorly performing IT organization to the right business objectives still won't get the objectives accomplished.

Richard F Connell

David Schpilberg, Steve Berez, Rudy Puryear and Sachin Shah
"Avoiding the Alignment Trap in Information Technology"
MIT Sloan Management Review
<http://sloanreview.mit.edu/article/avoiding-the-alignment-trap-in-it/>

Cargo cult programming is a style of computer programming characterized by the ritual inclusion of code or program structures that serve no real purpose.

Cargo cult programming can also refer to the results of applying a design pattern or coding style blindly without understanding the reasons behind that design principle.

http://en.wikipedia.org/wiki/Cargo_cult_programming

FizzBuzzEnterpriseEdition

build passing

Enterprise software marks a special high-grade class of software that makes careful use of relevant software architecture design principles to build particularly customizable and extensible solutions to real problems. This project is an example of how the popular FizzBuzz game might be built were it subject to the high quality standards of enterprise software.












FizzBuzz

FizzBuzz is a game that has gained in popularity as a programming assignment to weed out non-programmers during job interviews. The object of the assignment is less about solving it correctly according to the below rules and more about showing the programmer understands basic, necessary tools such as `if` `-/` `else` -statements and loops. The rules of FizzBuzz are as follows:

For numbers 1 through 100,

- if the number is divisible by 3 print Fizz;
- if the number is divisible by 5 print Buzz;
- if the number is divisible by 3 and 5 (15) print FizzBuzz;
- else, print the number.

[FizzBuzzEnterpriseEdition](#) / [src](#) / [main](#) / [java](#) / [com](#) / [seriouscompany](#) / [business](#) / [java](#) / [fizzbuzz](#) / [packagenamingpackage](#) / [interfaces](#) / **factories** / +

Merge #127		2 comments 
 Dmitry-Me authored on 22 Jul 2014	latest commit aad80defc4 	
..		
 FizzBuzzOutputStrategyFactory.java	Adding a factory to create the output strategy so that we never get I...	a year ago
 FizzBuzzSolutionStrategyFactory.j...	Problem solution should be instantiated through factory class to make...	2 years ago
 IntegerPrinterFactory.java	Move sources	2 years ago
 IntegerStringRetumerFactory.java	Move sources	2 years ago
 IsEvenlyDivisibleStrategyFactory.java	Move sources	2 years ago
 OutputGenerationContextVisitorFac...	Merge #127	8 months ago
 StringPrinterFactory.java	Move sources	2 years ago
 StringStringRetumerFactory.java	Move sources	2 years ago

**I have yet to see any problem,
however complicated, which,
when you looked at it in the
right way, did not become still
more complicated.**

Anderson's Law



Paul McMahon

@pwim

 Follow

All code is technical debt; some code just has a higher interest rate.

7:13 AM - 29 Dec 2014

139 RETWEETS **97** FAVORITES



<https://twitter.com/pwim/status/549463179084853248>





sum
1986

```

end if;
L_M_DON_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_DON) *
                                G_M_INFO_DERIVE(T_ALG.E_DON))
if L_M_DON_32 > 32767 then
  P_M_DERIVE(T_ALG.E_DON) := 16#7FFF#;
elsif L_M_DON_32 < -32768 then
  P_M_DERIVE(T_ALG.E_DON) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_DON) := UC_16S_EN_16NS(
    TDB.T_ENTIER_16S(L_M_DON_32));
end if;

P_M_DERIVE(T_ALG.E_DOE) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
((1.0/C_M_LSB_DOE) *
G_M_INFO_DERIVE(T_ALG.E_DOE))

L_M_BV_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
                                G_M_INFO_DERIVE(T_ALG.E_BV));
if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS (TDB.T_ENTIER_16S(L_M
end if;

P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
((1.0/C_M_LSB_BH) *
G_M_INFO_DERIVE(T_ALG.E_BH)))

```

501

```

end LIRE_DERIVE;
--$finprocedure

```

```

--(
procedure LIRE_SEUIL (P_M_SEUIL : out TDB.T_ENTIER_16NS) is
--)

```

RUD, *noun*

- Rapid Unscheduled Disassembly
- Rocket science and amateur rocketry jargon that's acronymous, euphemistic and explosively self-explanatory

The failure resulted in a loss of more than US\$370 million.

**Knight Capital Group realized a
\$460 million loss in 45 minutes.**

Doug Seven

<http://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/>

The update to SMARS was intended to replace old, unused code referred to as “Power Peg” — functionality that Knight hadn’t used in 8-years.

Doug Seven

<http://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/>

Why code that had been dead for 8 years was still present in the code base is a mystery, but that's not the point.

Doug Seven

<http://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/>

In the first 45 minutes the market was open the Power Peg code received and processed 212 parent orders. As a result SMARS sent millions of child orders into the market resulting in 4 million transactions against 154 stocks for more than 397 million shares.

Doug Seven

<http://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/>

Um. What's the name of the word for things not being the same always. You know, I'm sure there is one. Isn't there?

There's must be a word for it... the thing that lets you know time is happening. Is there a word?

Change.

Oh. I was afraid of that.

Neil Gaiman
The Sandman

Our task is not to find the maximum amount of content in a work of art.

Our task is to cut back content so that we can see the thing at all.

Susan Sontag



Kevlin Henney

@KevlinHenney



We keep talking about incremental development; we don't talk enough about decremental development.

11:39 AM - 28 Nov 2014

10 RETWEETS **4** FAVORITES



empirical, *adjective*

- based on, concerned with, or verifiable by observation or experience rather than theory or pure logic
- pertaining to, or derived from, experience
- capable of being verified or disproved by observation or experiment

ABACUS

E.F. Schumacher

Small
is beautiful

There is wisdom in smallness if only on account of the smallness and patchiness of human knowledge, which relies on experiment far more than on understanding.

A STUDY OF ECONOMICS
AS IF PEOPLE MATTERED