

OBJECT GUILD



Design Matters

ESUG 2019, Cologne, Germany

David West, Ph.D.

Object Guild

dwest@objectguild.com



What is Design

Engineering Design

All variables known, or in principle knowable, and quantified, or quantifiable.

All relationships known and amenable to formal specification.

System is Complicated. Problems are hard.

Design is an optimization problem.

Natural Design

Many variables unknown, and in principle, unknowable.

Almost all relationships are dynamic and situational.

Formal specification a myth.

System is Complex. Problems are “wicked.”

Design is a “Satisficing” Problem



Design for Software Developers

Code Design

Program Design

Framework Design

Architectural Design

Platform Design

Solution Design: Fit

System Design

Object Design

Code Design

```
const size = 80;
```

```
var
s: string[size];
i: integer;
c: char;
f: array[1..26] of integer;
k: integer;

begin
writeln ('enter line');
readln(s);

for i := 1 to 26 do f[i] := 0;

for i := 1 to size do
begin
c := asLowerCase (s[1]);

if isLetter(c) then
begin
k := ord(c) - ord('a') + 1;
f[k] := f[k] + 1
end
end;
end;
```

```
l s c f k l
```

```
f := Array new: 26.
```

```
s := Prompter
```

```
prompt: 'enter line' default: ' '.
```

```
1 to 26: do: [ i: l
```

```
c := (s at: i) asLowerCase.
```

```
c isLetter if True: [
```

```
k := c asciiValue + 1.
```

```
f at: k put: (f at: k) + 1
```

```
]
```

```
].
```

```
^f
```

```
l s f l
```

```
s := Prompter prompt: enterline'default: ' '.
```

```
f := Bag new.
```

```
s do: :c | c isLetter ifTrue: [f add: c asLowerCase] ].
```

```
^f
```

All Code is Arbitrary

All Code is Idiosyncratic

There is no Objective way to say “this code is good,” “this code is better than this code.”

“The sins of the fathers ...” — the “model” at the beginning remains the model forever.

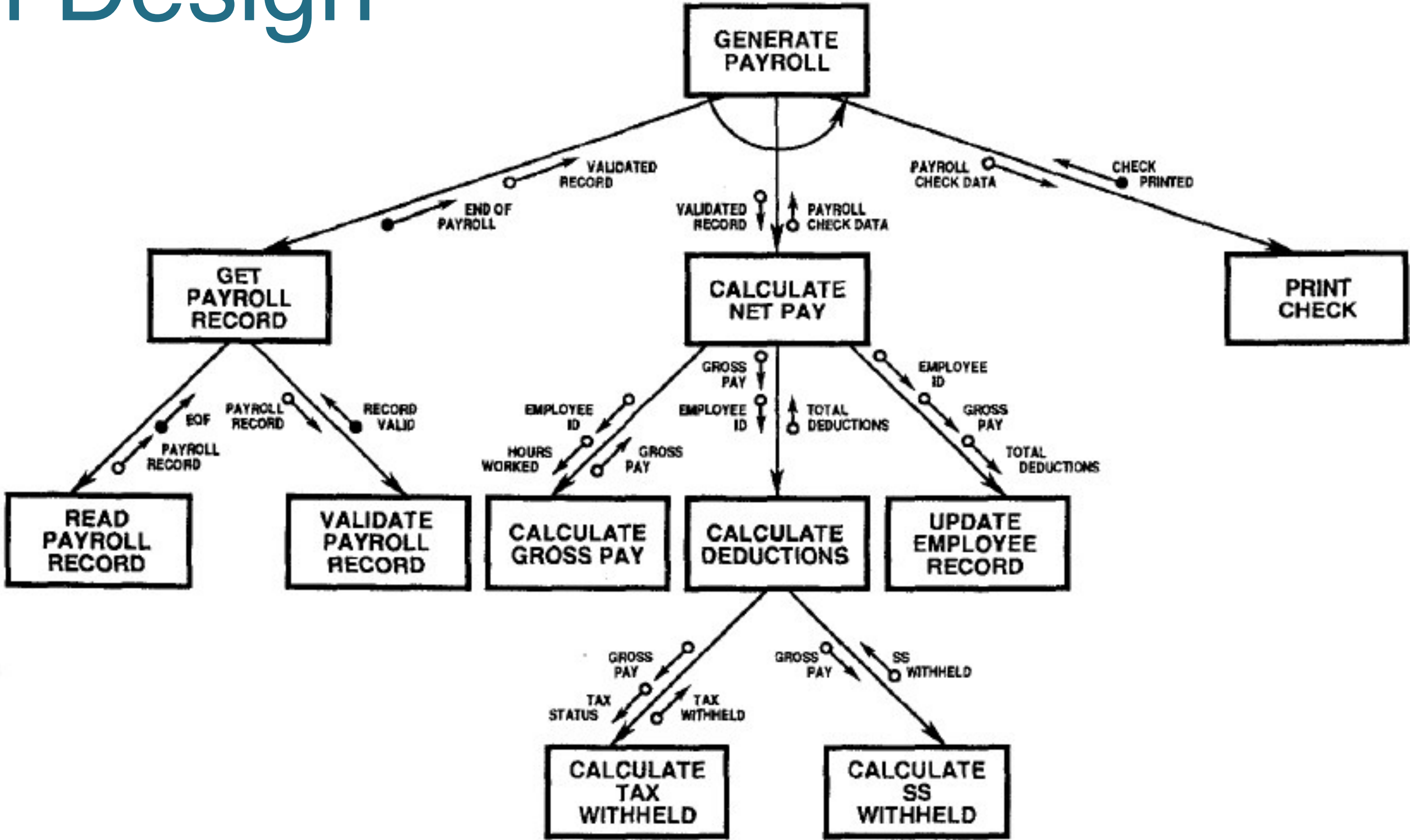
Microsoft commissioned an empirical study of how the tool was used, led by Chris Grainger. The findings: nearly 98 percent of the code was irrelevant, never used, not supporting any part of the code that was used, orthogonal to how people were thinking about the task of writing their own programs, developing their own software. Grainger observed:

“All this work had been put into this thing [Visual Studio], but it missed the fundamental problems that people faced. ... basically, [programmers] are playing computer inside their head. Programmers are like chess players trying to play with a blindfold on — so much of their mental energy is spent just trying to picture where the pieces are that there’s hardly any left over to think about the game itself.”

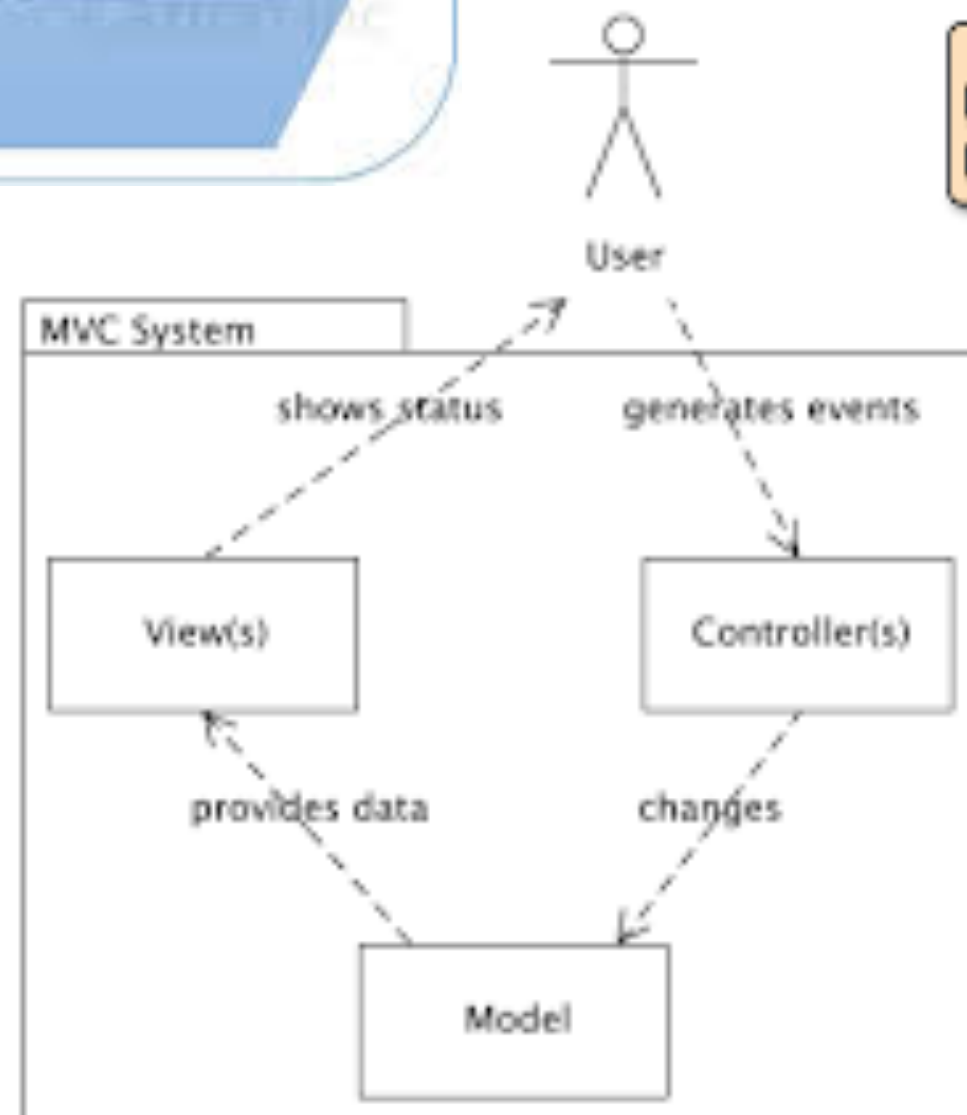
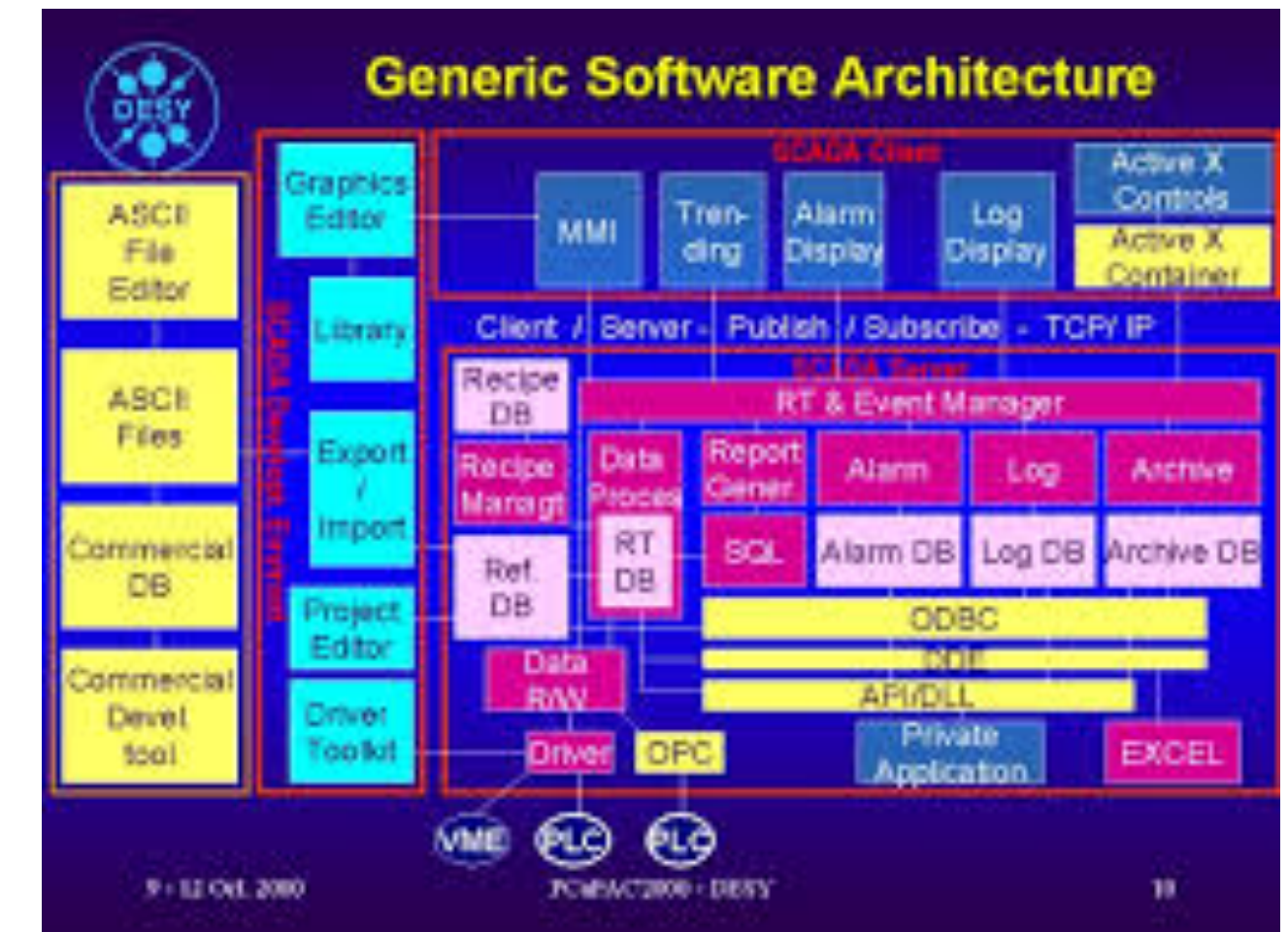
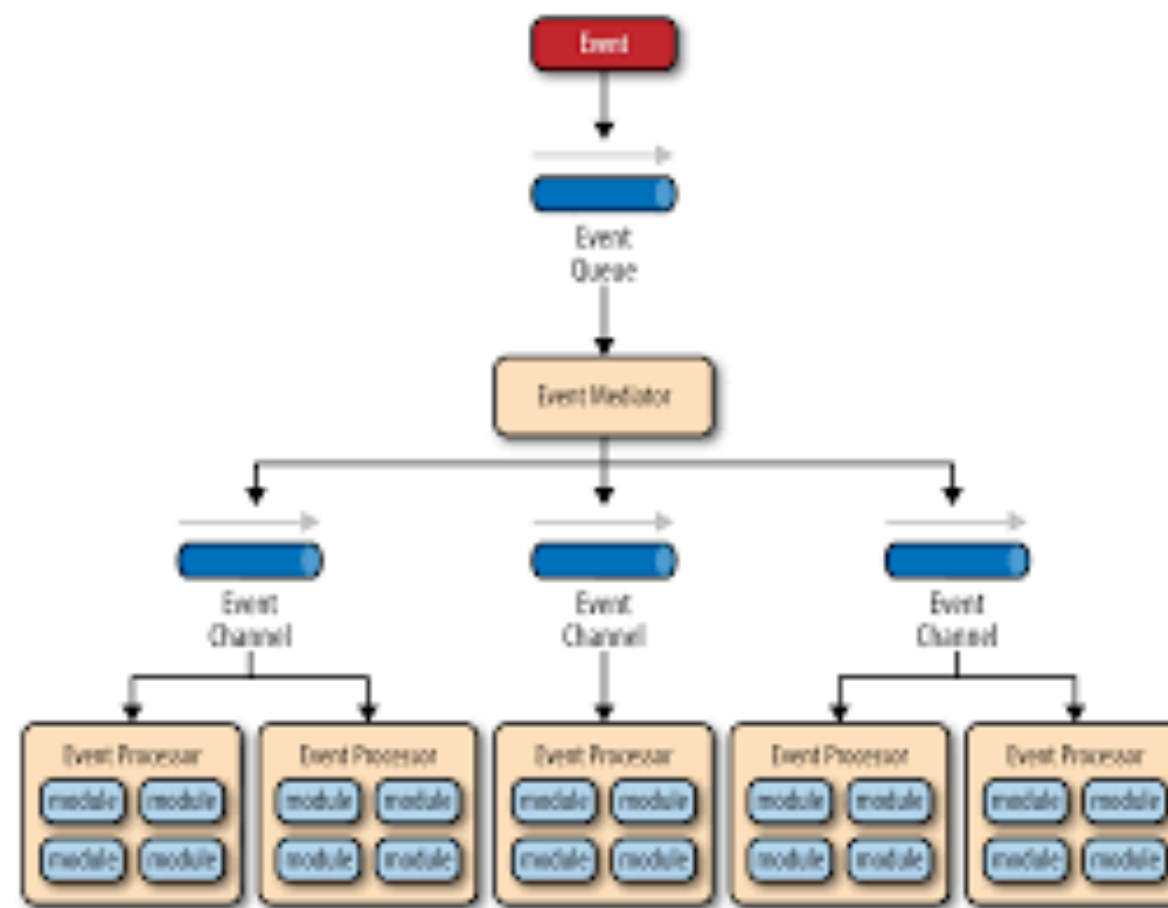
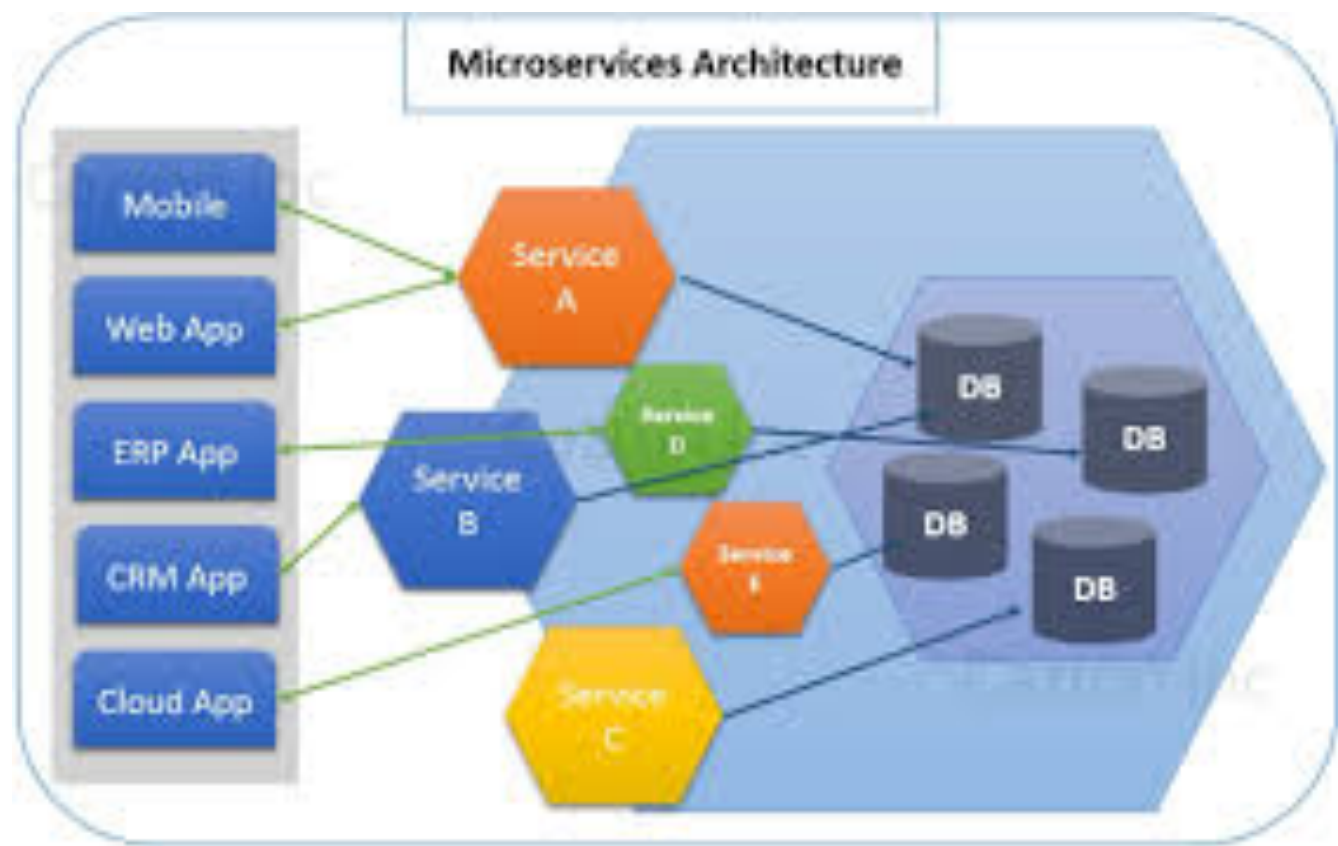
The only “Correctness” is Consistency /
Congruency With The DOMAIN.

This is Software “Fit.”

Program Design




Architectural Design





01

Solution Design: Fit

The background features a teal color with a pattern of diagonal stripes in a darker shade of teal, creating a sense of depth and movement. In the top-left corner, there are two vertical bars of the same teal color, one slightly taller than the other.

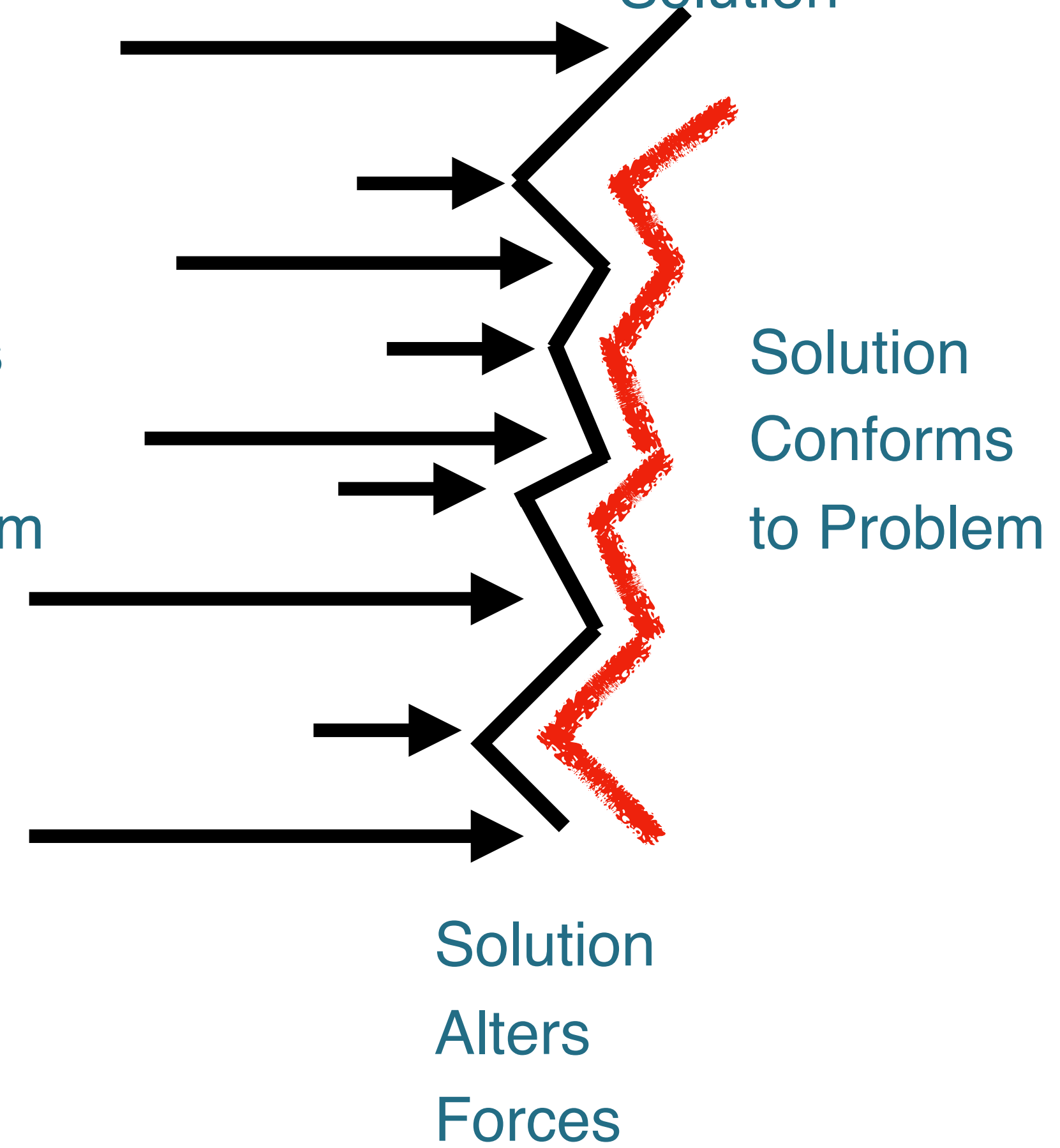
Every design problem begins with an effort to achieve fitness between two entities: the form in question and its context. The form is the solution to the problem; the context defines the problem.

The Block is the Context
colored inserts the Solution



Problem
Shapes
Solution

Forces
Shape
Problem



Definition of "Wicked Problem"



02

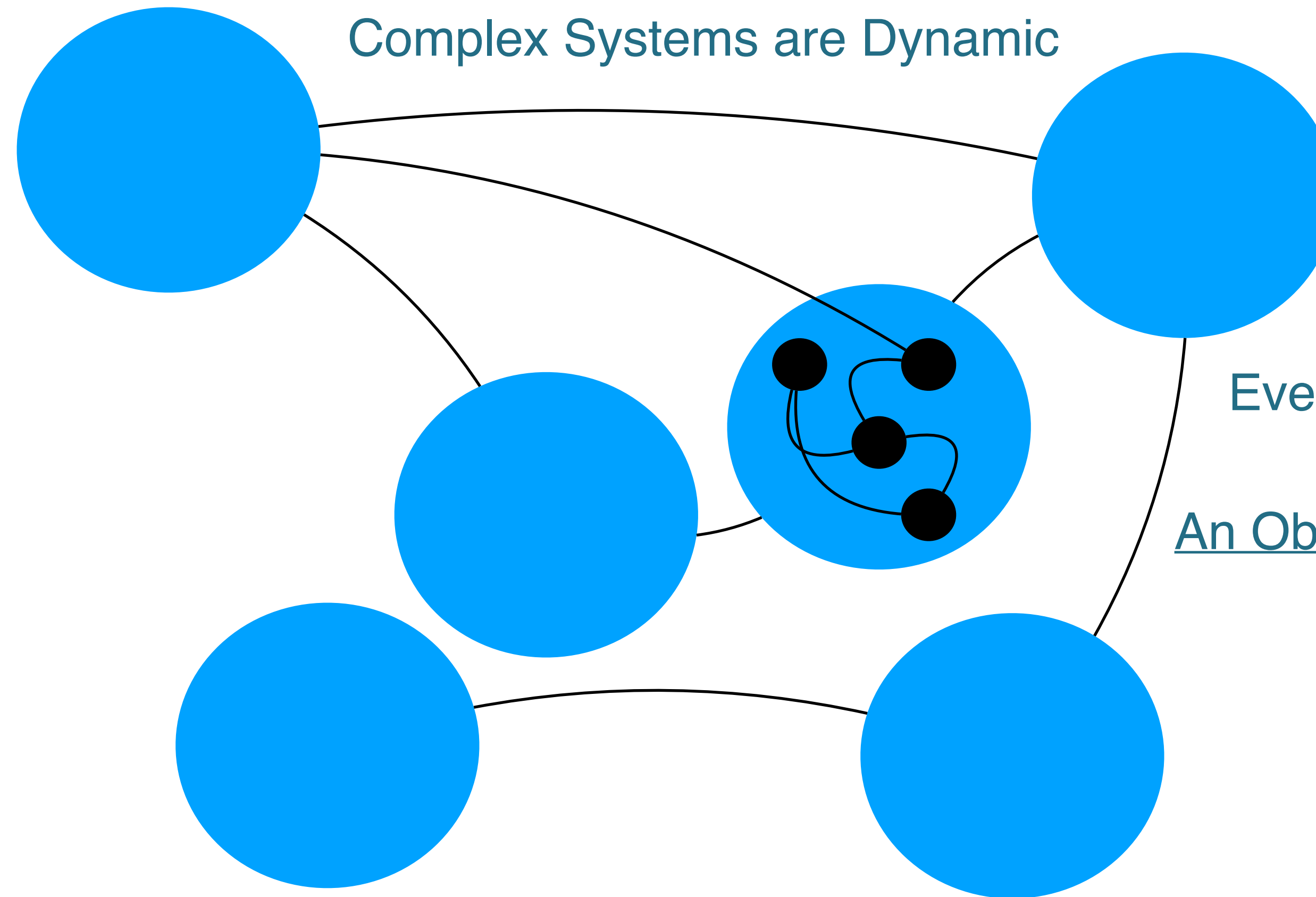
System Design

System Design

Elements and Relationships are differentiated from each other on the basis of their “contribution” (aka behavior towards) to other elements/relationships in the system.

A system is a set of elements and the relationships among them.

Complex Systems are Dynamic



Everything an Object

An Object Is What It Does



Object Design

03



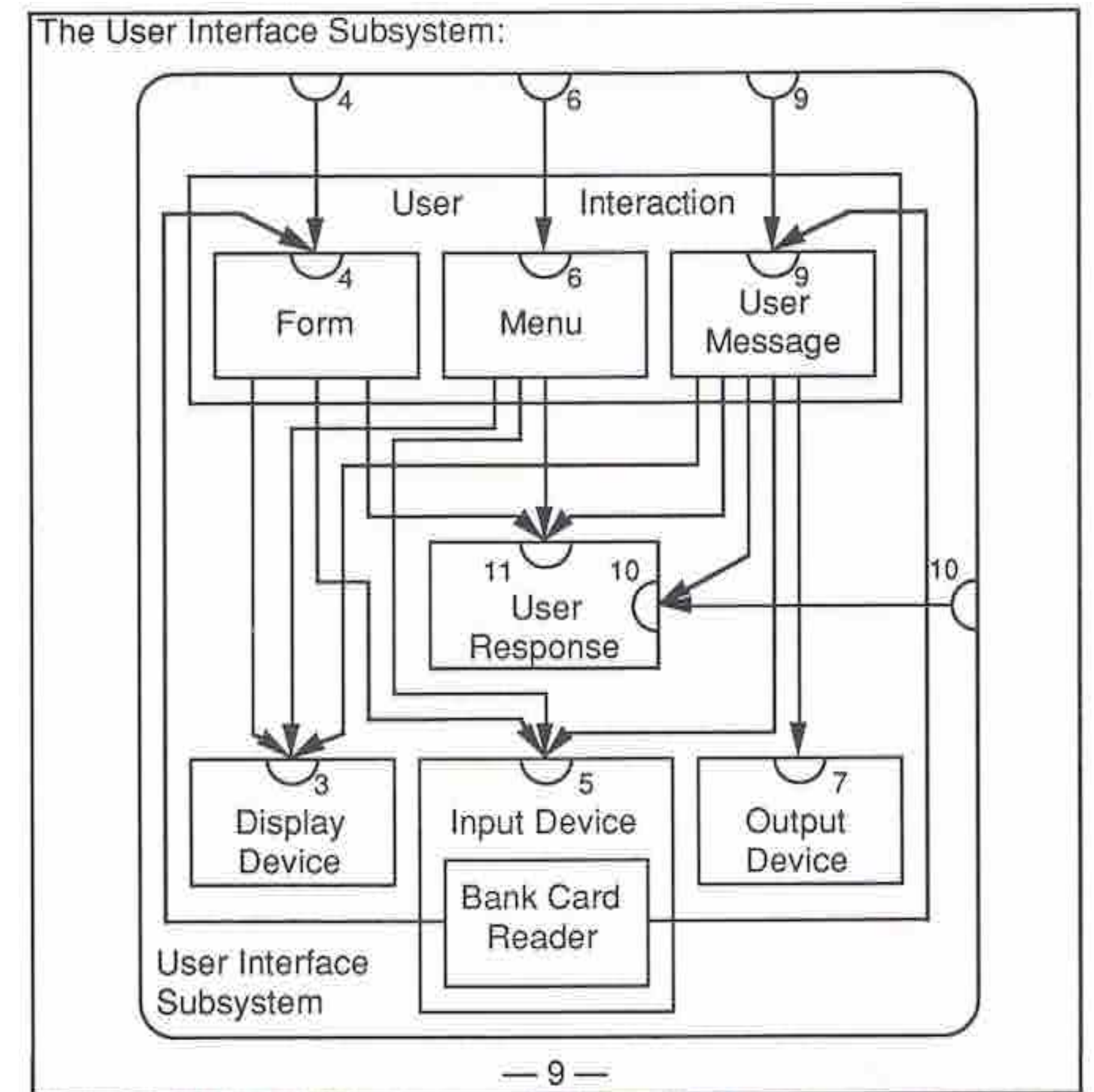
Object Design

historical

Class Name	
Responsibilities	Collaborators

overlooked

(foreshadows message oriented programming)

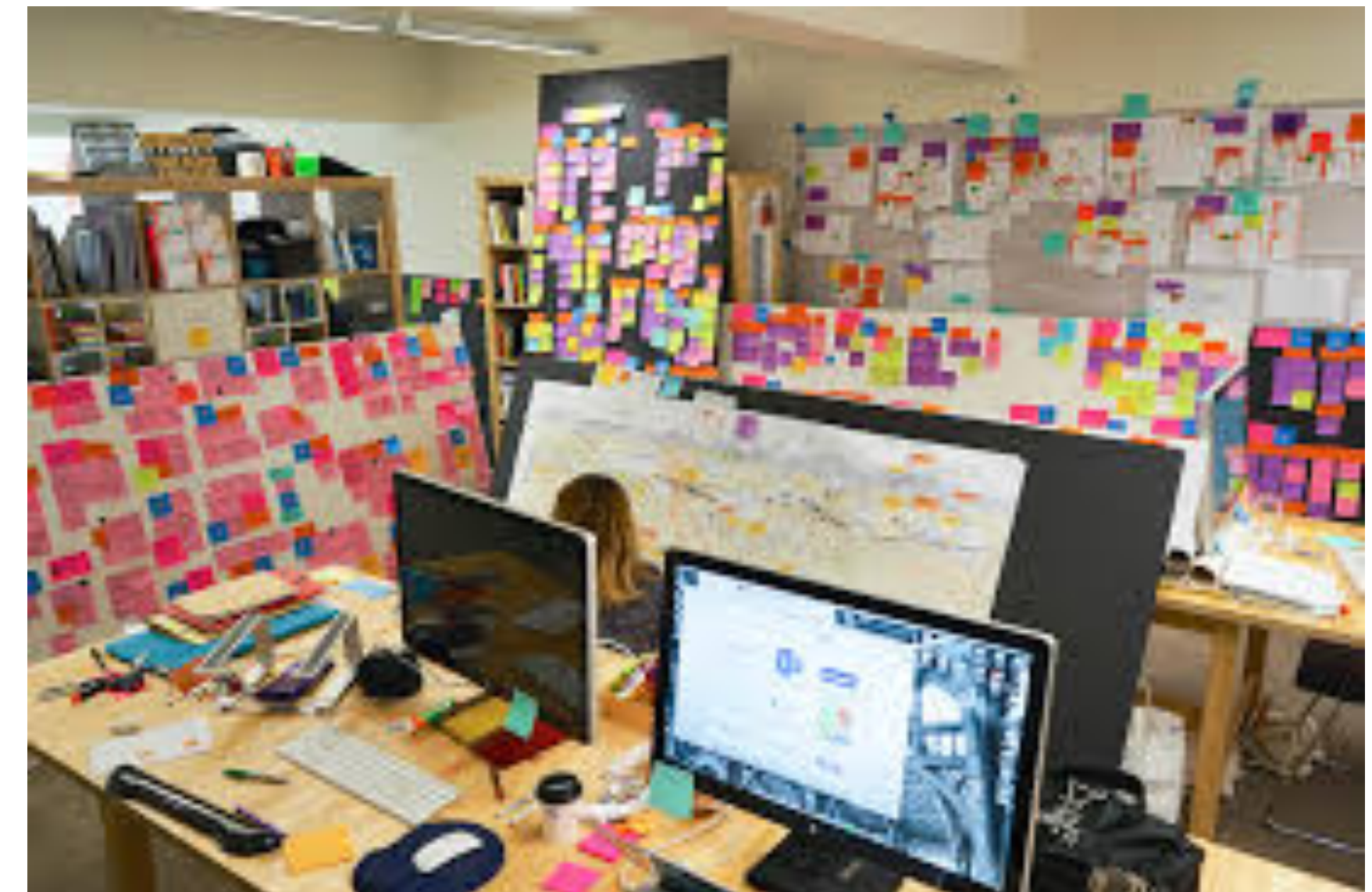



per Dave West, Object Thinking

Name / Description	Responsibilities	Knowledge Required	Protocol	Contracts	Events
(class) Name	specific behaviors one per line max seven	name how obtained [V I A I M I G] embodied class	invoking method signatures object returned	frequent message pathways	changes in state willing to share others interested in



Better Designers





Great designs come from great designers.
Software construction is a creative process.
Sound methodology can empower and liberate
the creative mind; it cannot enflame or inspire
the drudge.

Fred Brooks, "No Silver Bullet"

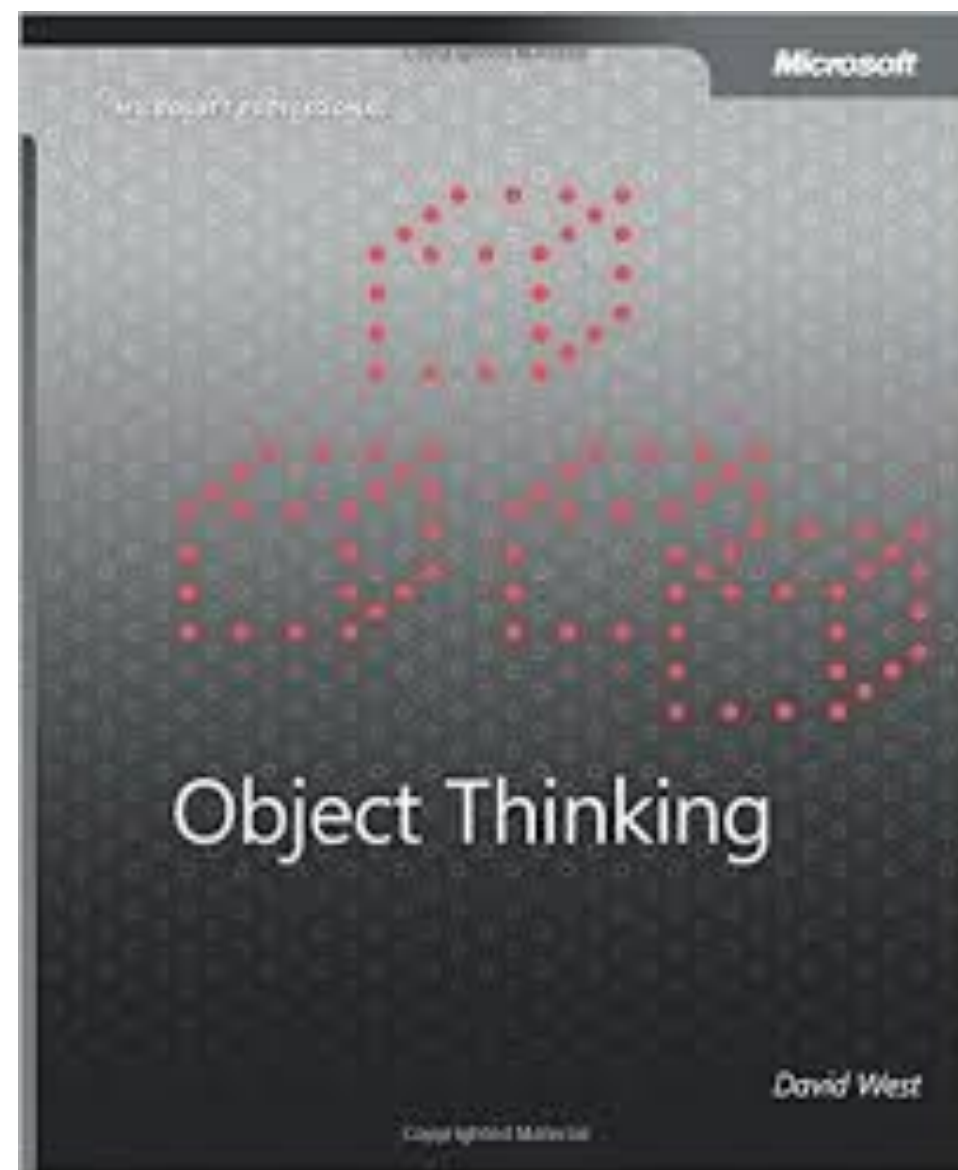
*Thirdly; ... the power not only of throwing my
whole energy & existence into whatever I choose,
but also bring to bear on any one subject or idea,
a vast apparatus from all sorts of apparently
irrelevant & extraneous sources. I can throw **rays**
from every quarter of the universe into **one** vast
focus."*

Ada Lovelace

"The ideal architect should be a man of letters, a skillful
draftsman, a mathematician, familiar with historical studies, a
diligent student of philosophy, acquainted with music; not
ignorant of medicine, learned in the responses of
jurisconsults, familiar with astronomy, and astronomical
calculations."

Vitruvius 25 B.C.

Questions?



David West, Ph.D.

OBJECT GUILD



Amsterdam, NL

dwest@objectguild.com

+31 (0) 6 3053 6679

