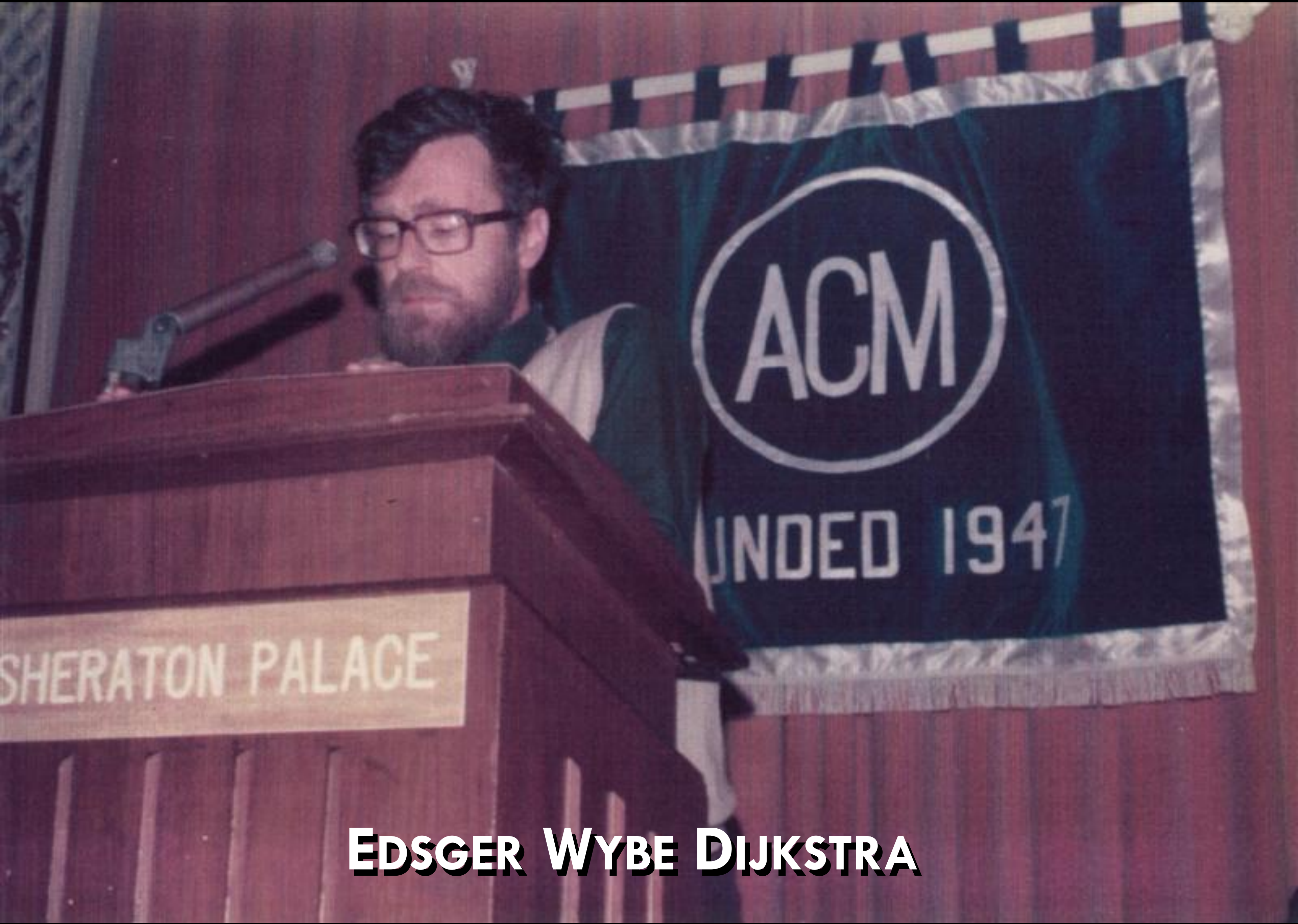
Go To There and Back Again



Edsger Wybe Dijkstra

GOTO Statement Considered Harmful

For a number of years I have been familiar with the observation that the quality of programmers is a decreasing function of the density of **go to** statements in the programs they produce.

GOTO Statement Considered Harmful

For a number of years I have been familiar with the observation that the quality of programmers is a decreasing function of the density of **go to** statements in the programs they produce.

More recently I discovered why the use of the **go to** statement has such disastrous effects…

GOTO Statement Considered Harmful

For a number of years I have been familiar with the observation that the quality of programmers is a decreasing function of the density of **go to** statements in the programs they produce.

More recently I discovered why the use of the **go to** statement has such disastrous effects, and I became convinced that the **go to** statement should be abolished from all "higher level" programming languages.

Eliminate the GOTO Statement

Radicals

Disaster.

Eliminate the GOTO Statement

Reactionaries

Performance.

Tradition.

Freedom.

Pride.

Majority.

Radicals

Disaster.

Eliminate the GOTO Statement

Reactionaries

Performance.
Tradition.
Freedom.
Pride.
Majority.

Moderates

Avoid where possible, but it is sometimes useful.

Radicals

Disaster.

1968	GOTO Considered Harmful
1973	C
1985	C++
1995	Java

Progress does not wait
for the next new idea.

It waits for consensus
on an old idea.

Bugs

As soon as we started programming, we found out to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs.

Maurice Wilkes

The Software Crisis

- Over budget
- Over time
- Unreliable
- Requirements not fully met
- Unmaintainable
- Failure

Proof of Correctness

Tests can only prove the existence of bugs, not their absence.

Beware of bugs in the above code. I have only proved it correct, not tried it.

Donald Knuth (1977)

Types

What do the bits mean?

Easier than Proofs of Correctness?

Well typed programs cannot go wrong.

Robin Milner (1978)

Static Type Checking

Reactionaries	Moderates	Radicals
Little extra effort. It is helping you.	Should be optional. Taste.	Not worth the freight.

Milner’s Promise

Will static typing ever
be worth the freight?

COME FROM Statement

COME FROM Statement

extends

ES6

Proper Tails Calls

```
return func();
```

Back Again

```
    if ((err = SSLHashSHA1.update(
          &hashCtx, &signedParams)) != 0)

        goto fail;

        goto fail;




fail:

    return err;
```

Paradigm Shift

I have never ___
so I don't see why ___

Paradigm Shift

Bad Idea

Functional Programming

1958	LISP
1973	ML
1975	Scheme
1984	Common Lisp
1990	Haskell
1999	JavaScript

Code Reading

Dailies

And finally,
one piece of advice:

And finally,
one piece of advice:

Don’t make bugs.

Baseball

Moneyball:
The Art of Winning an Unfair Game

Michael Lewis

Bill James

Sabermetrics

Billy Beane

Oakland Athletics

Don't make outs.

90%

Stealing Bases

Getting on base

Getting on base

Walk

Programming

Don't make bugs

Team Building