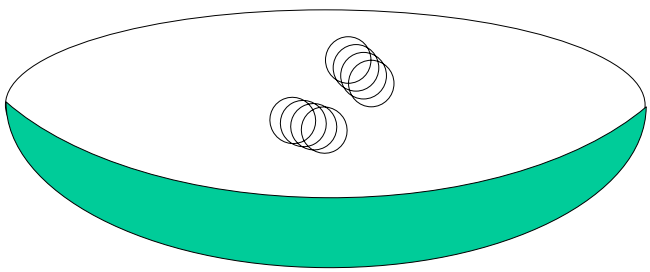
Using Poka-Yoke Techniques for Early Defect Detection

Microsoft Testing Talks
19 February 1998

Harry Robinson
Hewlett Packard
harry_robinson@cv.hp.com

Yamada Electric, 1961



Shigeo Shingo

No one intends to make mistakes. But while we are working, defects can show up without our noticing. … How can we catch these errors before they turn into defective products?

Poka-Yoke

(Pronounced "POH-kah YOH-kay")

Literally means "Mistake-proofing":

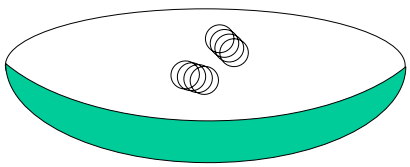
- Poka - inadvertent mistake
- Yoke - to prevent

The essence of poka-yoke is to design your process so that mistakes are either impossible, or are easily detected and corrected.

Categories of Poka-Yoke

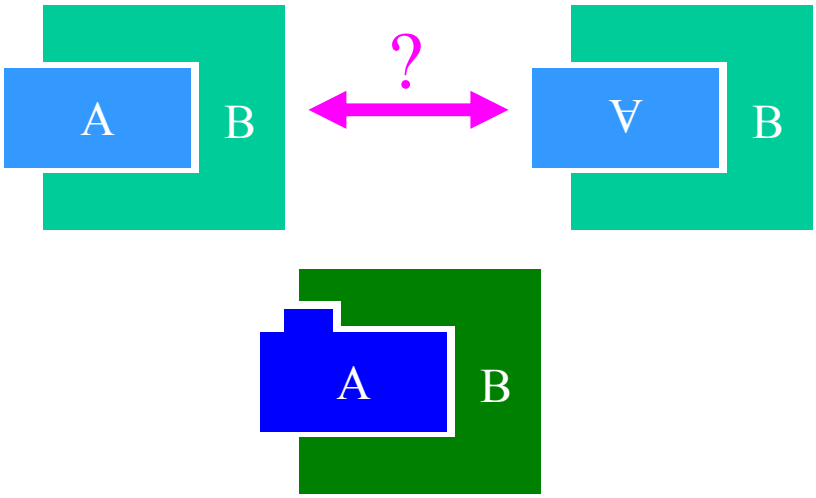
- **Prevention** -- engineer the process so it is impossible to make a mistake
- **Detection** -- signal the user when a mistake has been made so that the user can correct the mistake

The "Small Dish" Solution



- Simple
- Cheap
- Specific
- Immediate feedback
- Trades speed for reliability
- Not really "testing"

Poka-Yoke in Manufacturing

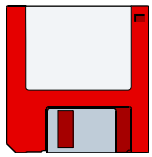


Japan's Quality Miracle

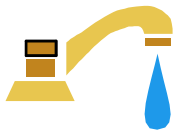
"It is not *one* device, but the application of hundreds and thousands of these very simple "fail-safing" mechanisms that day after day has brought the quality miracle to Japan. Each one is relatively simple -- something you easily could do on your own. It is the totality, the hundreds of devices, that is almost frightening to behold."

- Norman Bodek

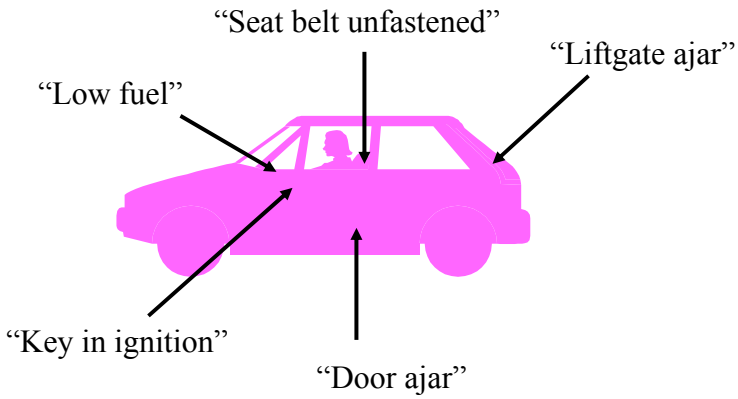
Everyday Prevention Poka-Yokes



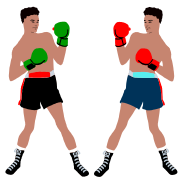
- 3.5” diskette
- locking file drawers
- sensor-equipped faucets



Everyday Detection Poka-Yokes



Shingo vs. Murphy



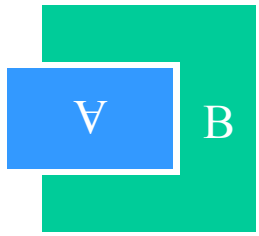
The Original Murphy's Law

"If there are two or more ways to do something, and one of those ways can result in catastrophe, then someone will do it."

-- *Edward A. Murphy, Jr.* 1949

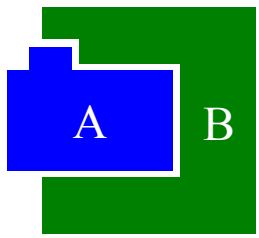
Murphy's Law

If anything can go wrong,
it will.

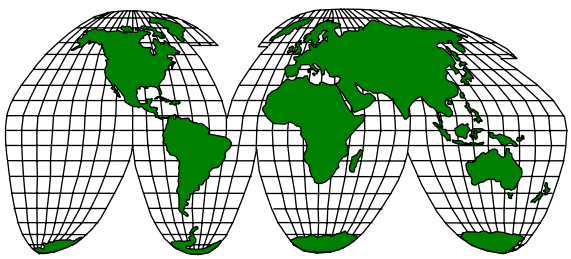


Shingo's Law

If anything can go wrong,
it doesn't necessarily have to.



Poka-Yoke in Localization



Hello, World 1

hello.c:

```
printf("Hello, World!\n");
```

Hello, World 2

hello.c:

```
my_cat=catopen("hello.cat", NL_CAT_LOCALE);
printf( catgets( my_cat, 1, 1, "Hello, World!\n"));
catclose(my_cat);
```

hello.cat:

```
$set 1
1 Hello, World!\n
```

This example is adapted from
"X Windows on the World"
by Tom McFarland, Prentice Hall, 1996

Hello, World 3

hello.c:

```
my_cat=catopen("hello.cat", NL_CAT_LOCALE);
printf( catgets( my_cat, 1, 1, "Hello, World!\n"));
catclose(my_cat);
```

hello.cat:

```
$set 1
1 Bonjour, Le Monde!\n
```

This example is adapted from
"X Windows on the World"
by Tom McFarland, Prentice Hall, 1996

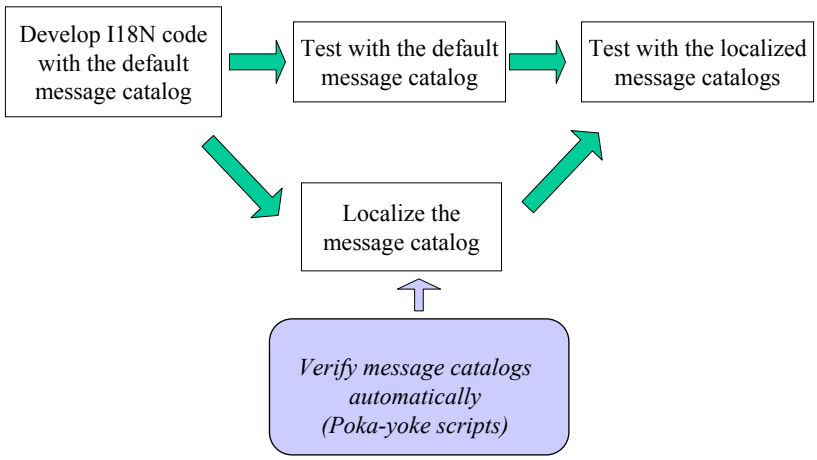
English and French Menus

English and French Menus

$set 11	
17	N
18	New
19	O
20	Open …
21	I
22	Include
23	S
24	Save
25	A
26	Save As
27	P
28	Print …
29	C
30	Close

$set 11	
17	N
18	Nouveau
19	O
20	Ouvrir …
21	I
22	Inclure
23	S
24	Sauvegarder
25	A
26	Sauvegarder sous ...
27	P
28	Imprimer …
29	F
30	Fermer

The Localization Process



Common Menu Mistakes

Mnemonic is not a single ASCII letter →

Mnemonic is not unique in the menu →

Mnemonic is not contained in the associated label →

$set 11	
17	NN
18	Nouveau
19	O
20	Ouvrir …
21	I
22	Inclure
23	S
24	Sauvegarder
25	S
26	Sauvegarder sous ...
27	P
28	Imprimer …
29	C
30	Fermer

Poka-Yoke for Menus

Mnemonic location	label location	C mnemonic	C label
11,17	11,18	N	New
11,19	11,20	O	Open …
11,21	11,22	I	Include
11,23	11,24	S	Save
11,25	11,26	A	Save As …
11,27	11,28	P	Print …
11,29	11,30	C	Close

Missing/Extra Messages

What it should have said

hello.cat:
$set 1
1 Bonjour, Le Monde!\n

What it actually says

hello.cat:
$set 1
11 Bonjour, Le Monde!\n

typo

Poka-Yoke for Missings/Extras

English	French
11,17	11,17
11,18	
11,19	11,19
11,20	11,20
11,21	11,21
11,22	11,22
11,23	11,23
11,24	11,24
11,25	11,25
11,26	11,26
11,27	11,27
11,28	11,28
11,29	11,29
11,30	11,30
	11,78

Results ...

- 311 menu mistakes
- 89 printf() mistakes
- 383 missing messages
- 50 extra messages

Total: 833 defects!

The “Small Script” Solution

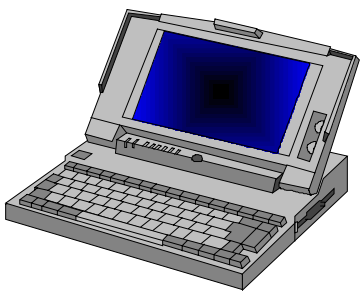
- Simple
- Cheap
- Specific
- Immediate feedback
- No GUI interaction
- Not really “testing” - no application needed!

Questions NOT Answered

- Does the application work?
- Is the message catalog translated correctly?

The Poka-yoke scripts check for necessary
but not sufficient conditions!

Poka-Yoke in Software



Categories of Poka-Yokes

Prevention:

- high-level languages
- structured programming
- object-oriented methodologies

Detection:

- lint-type utilities: check, cchk, printfck, ...
- compiler flags
- assertions
- smoke testing

Poka-Yoke in Testing

Smoke Tests

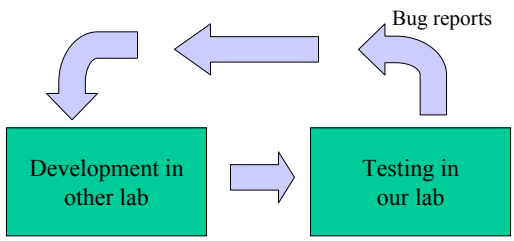
smoke test: n.

1. A rudimentary form of testing applied to electronic equipment following repair or reconfiguration, in which power is applied and the tester checks for sparks, smoke, or other dramatic signs of fundamental failure.

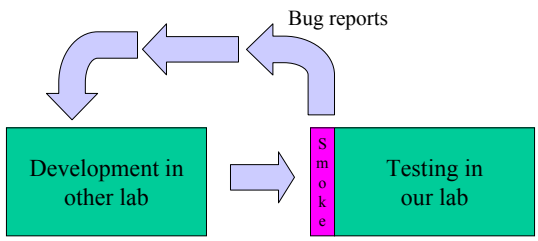
2. By extension, the first run of a piece of software after construction or a critical change.

-- *The New Hacker's Dictionary*

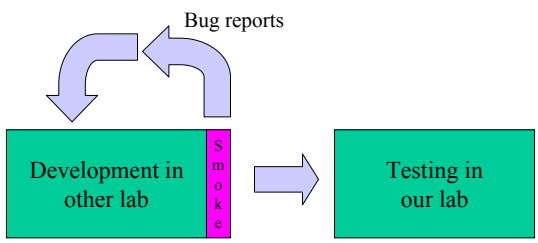
Without Smoke Tests



With Smoke Tests in Our Lab



With Smoke Tests in **Their** Lab



Poka-Yoke in Programming

How the World Ends, version 1

```
if ( incoming_missiles = TRUE ) {
        launch( our_missiles);
}
```

Output of freeware ***check*** utility:

sdi.c: assignment in conditional context at or near line xx

How the World Ends, version 2

```
if ( incoming_missiles == TRUE ); {
        launch( our_missiles);
}
```

Output of freeware ***check*** utility:

sdi.c: IF with neither THEN or ELSE at or near line xx

How common are such mistakes?

```
long n = LONG_MIN;
while (1)
        {
        if ( Blackbox(n) == 32L)
                {
                cout << n << " generates 32!";
                break;
                }
        if ( n = LONG_MAX )
                {
                cout << "Didn't find 32!";
                break;
                }
        ++n;
        }
```

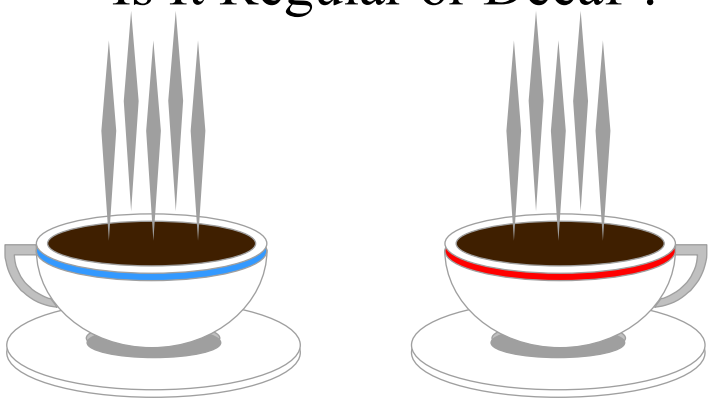
Source: "Genetic Algorithms in C++"
by Scott Robert Ladd (page 10)

Words to code by ...

- How could I have *automatically* detected this bug?
- How could I have *prevented* this bug?

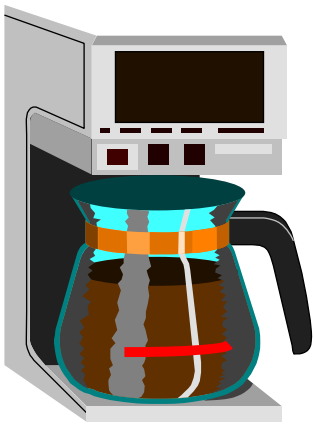
- Steve Maguire
Writing Solid Code

Is It Regular or Decaf ?



"A trivial system for solving a common problem."
- Steve Maguire
Debugging the Development Process

How to Make Good Coffee



"Now imagine a coffee shop that has a whole collection of such trivial 'systems' that produce better results with little or no extra effort…"

- Steve Maguire

Debugging the Development Process

How to Make Good Poka-Yokes

- Think ***simple***
- Think ***specific***
- Think ***attributes***
- Think ***early***
- Think ***responsive***
- Think ***re-use***

Poka-Yoke Resources

- John Grout's Poka-Yoke Page
  http://www.cox.smu.edu/jgrout/pokayoke.html
- Shigeo Shingo:
  Zero Quality Control: Source Inspection and the Poka-yoke System
- Nikkan Kogyo Shimbun*:*
  Poka-yoke: Improving Product Quality by Preventing Defects
- James Tierney:
  Eradicating Mistakes in Your Software Through Poka-yoke (video)
- Steve Maguire:
  Writing Solid Code
  Debugging the Development Process
- Harry Robinson's Home Page
  http://www.geocities.com/SiliconValley/Lab/5320

Thank You!