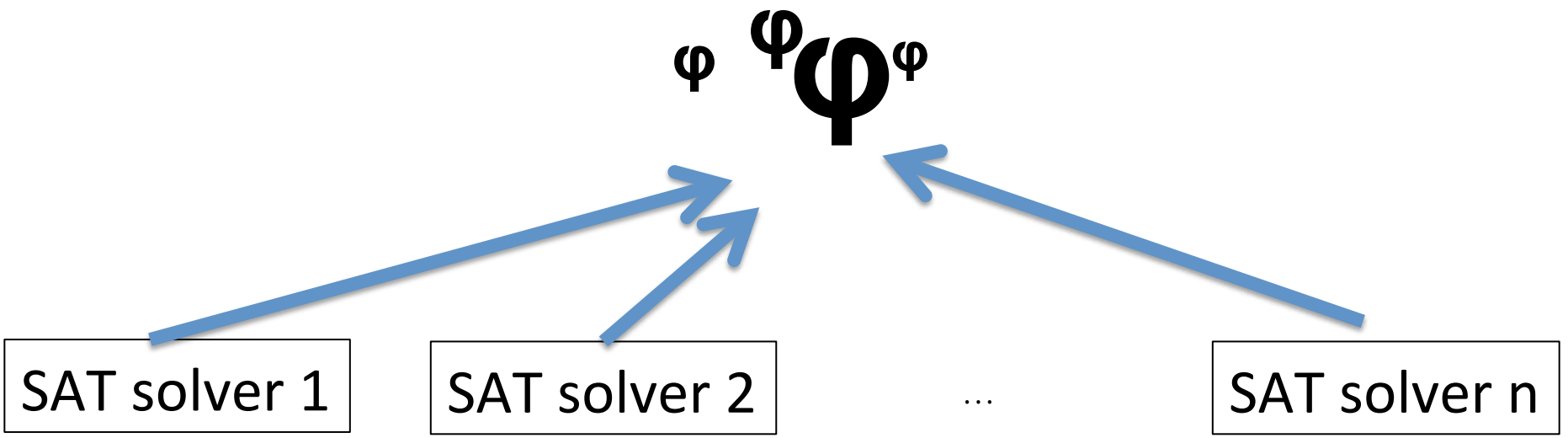
Automated Software Testing

Mathieu Acher
Maître de Conférences
mathieu.acher@irisa.fr

Material

https://github.com/FAMILIAR-project/HackOurLanguages-SIF

Comparing solver variants' performance



You Retweeted

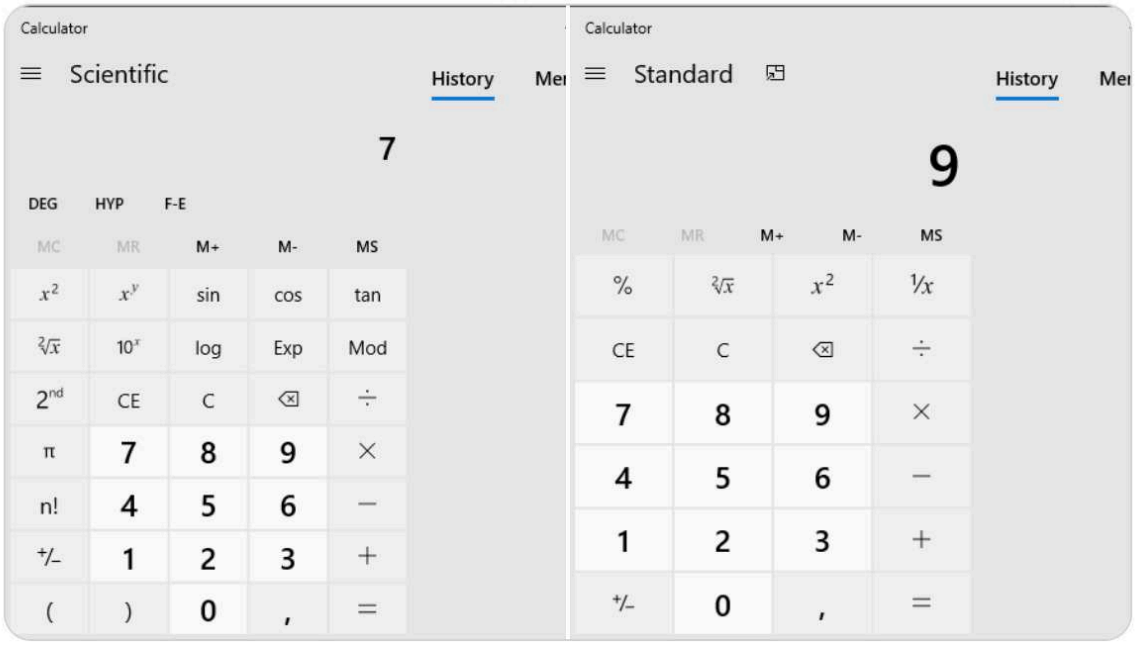
Thomas Thüm
@ThomasThuem

How much is 1 + 2 * 3?

According to @Windows 10, the result depends on the mode that you have chosen!!!!!

The Calculator app gives the correct result in Scientific mode and a wrong result with the default mode called Standard.

I bet this caused thousands of wrong calculations!



11:46 PM · Oct 9, 2019 · Twitter Web App

Formal verification or testing?

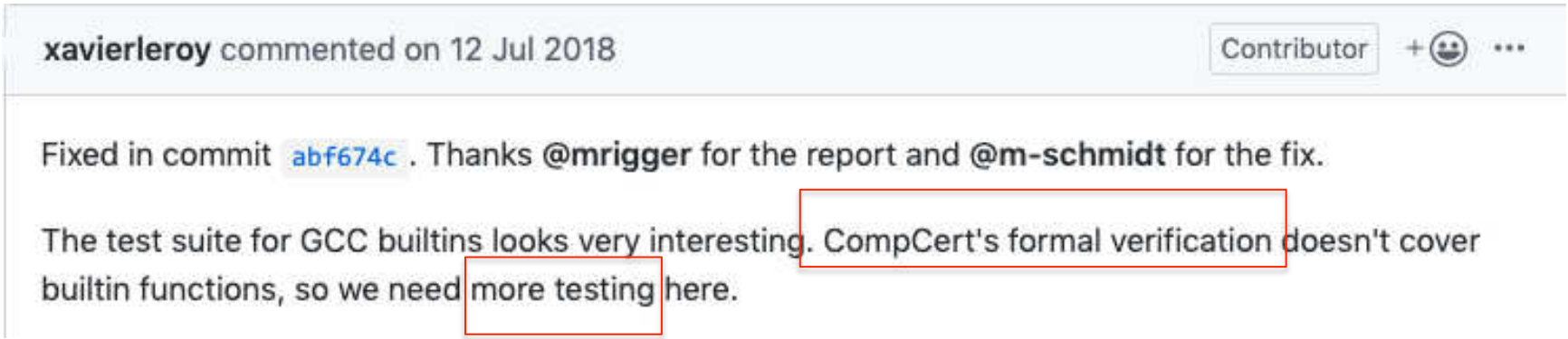
```
procedure insert2 (integer x, ℓ)
begin B[ℓ] ← B[ℓ] ∨ (2 ↑ (x mod 16));
    size[ℓ] ← size[ℓ]+1;
    if x < least[ℓ] then least[ℓ] ← x
    else if x > greatest[ℓ] then greatest[ℓ] ← x;
    end;
```

The implementation of deletion would be similar. It is safe to use 0 and $2^{16}-1$ for $-\infty$ and $+\infty$.

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

THE CORRESPONDENCE BETWEEN DONALD E. KNUTH AND PETER VAN EMDE BOAS ON PRIORITY DEQUES DURING THE SPRING OF 1977

Formal verification or testing?



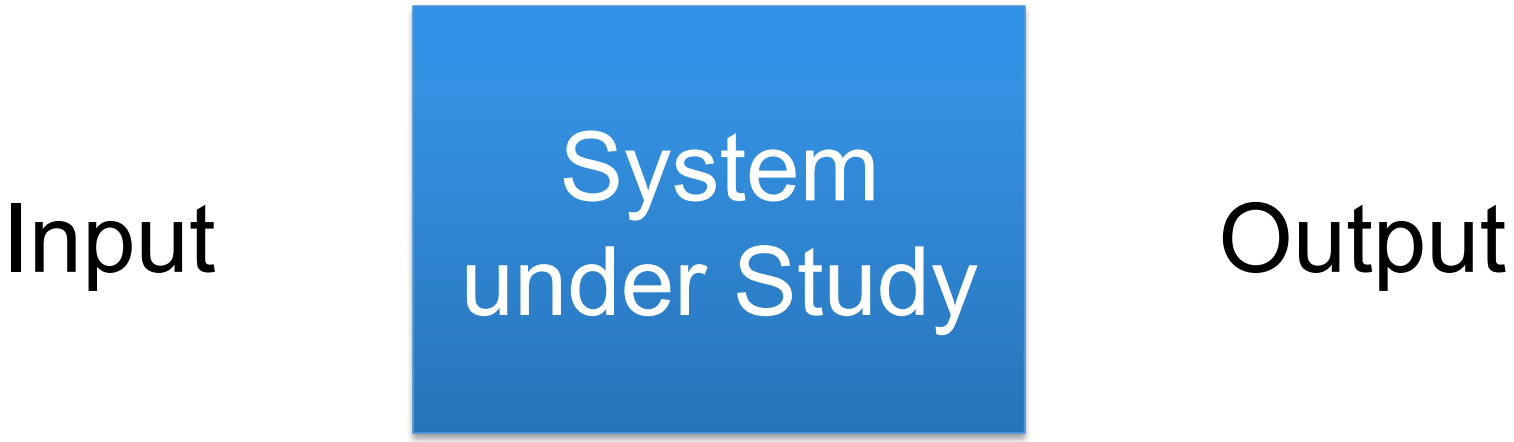
Support of GCC Builtins
https://github.com/AbsInt/CompCert/issues/243

Plan

- Testing Problem
  - Test Generation Problem (input)
  - Oracle Problem (output)
- Assertion-based testing
- Mutation Testing
- Test Generation
- Metamorphic Testing
- Multimorphic Testing

Testing Problem

- Test Generation Problem (input)
- Oracle Problem (output)



Testing Problem

- Test Generation Problem (input)
- Oracle Problem (output)

C programs

Assembly codes

Configuration is an additional part of the (test) input!

Optimization Options

See Options that Control Optimization.

```
-faggressive-loop-optimizations
-falign-functions[=n[:m:[n2[:m2]]]]
-falign-jumps[=n[:m:[n2[:m2]]]]
-falign-labels[=n[:m:[n2[:m2]]]]
-falign-loops[=n[:m:[n2[:m2]]]]
-fassociative-math  -fauto-profile  -fauto-profile[=path]
-fauto-inc-dec  -fbranch-probabilities
-fbranch-target-load-optimize  -fbranch-target-load-optimize2
-fbtr-bb-exclusive  -fcaller-saves
-fcombine-stack-adjustments  -fconserve-stack
-fcompare-elim  -fcprop-registers  -fcrossjumping
-fcse-follow-jumps  -fcse-skip-blocks  -fcx-fortran-rules
-fcx-limited-range
-fdata-sections  -fdce  -fdelayed-branch
-fdelete-null-pointer-checks  -fdevirtualize  -fdevirtualize-speculatively
-fdevirtualize-at-ltrans  -fdse
-fearly-inlining  -fipa-sra  -fexpensive-optimizations  -ffat-lto-objects
-ffast-math  -ffinite-math-only  -ffloat-store  -fexcess-precision=style
-fforward-propagate  -ffp-contract=style  -ffunction-sections
-fgcse  -fgcse-after-reload  -fgcse-las  -fgcse-lm  -fgraphite-identity
-fgcse-sm  -fhoist-adjacent-loads  -fif-conversion
-fif-conversion2  -findirect-inlining
-finline-functions  -finline-functions-called-once  -finline-limit=n
-finline-small-functions  -fipa-cp  -fipa-cp-clone
-fipa-bit-cp  -fipa-vrp  -fipa-pta  -fipa-profile  -fipa-pure-const
-fipa-reference  -fipa-reference-addressable
-fipa-stack-alignment  -fipa-icf  -fira-algorithm=algorithm
-fira-region=region  -fira-hoist-pressure
-fira-loop-pressure  -fno-ira-share-save-slots
-fno-ira-share-spill-slots
-fisolate-erroneous-paths-dereference  -fisolate-erroneous-paths-attribute
-fivopts  -fkeep-inline-functions  -fkeep-static-functions
-fkeep-static-consts  -flimit-function-alignment  -flive-range-shrinkage
-floop-block  -floop-interchange  -floop-strip-mine
-floop-unroll-and-jam  -floop-nest-optimize
-floop-parallelize-all  -flra-remat  -flto  -flto-compression-level
-flto-partition=alg  -fmerge-all-constants
-fmerge-constants  -fmodulo-sched  -fmodulo-sched-allow-regmoves
-fmove-loop-invariants  -fno-branch-count-reg
-fno-defer-pop  -fno-fp-int-builtin-inexact  -fno-function-cse
-fno-guess-branch-probability  -fno-inline  -fno-math-errno  -fno-peephole
-fno-peephole2  -fno-printf-return-value  -fno-sched-interblock
-fno-sched-spec  -fno-signed-zeros
-fno-toplevel-reorder  -fno-trapping-math  -fno-zero-initialized-in-bss
-fomit-frame-pointer  -foptimize-sibling-calls
-fpartial-inlining  -fpeel-loops  -fpredictive-commoning
-fprefetch-loop-arrays
-fprofile-correction
```

gcc (compiler): 200+ options

Boolean options

Testing Problem

- Test Generation Problem (input)
- Oracle Problem (output)

Testing Problem

- Test Generation Problem (input)
- Oracle Problem (output)

Testing Problem

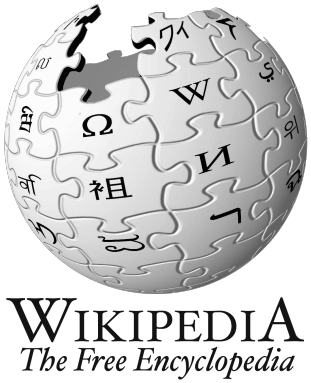
- Test Generation Problem (input)
- Oracle Problem (output)

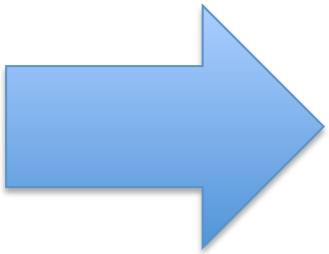
Wikipedia pages
(HTML or Wikitext format)

CSV files

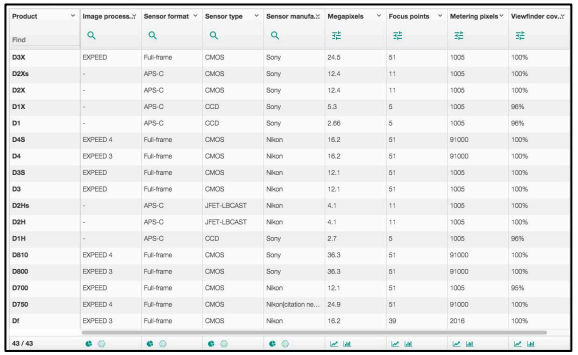
http://blog.mathieuacher.com/WikipediaMatrixChallenge/

How to automatically extract tabular data out of Wikipedia pages?





CSV
(Comma Separated Values)



Product	Image process.	Sensor format	Sensor type	Sensor manufa.	Megapixels	Focus points	Metering pixels	Viewfinder cov.
Find								
D3X	EXPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	96%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	96%
D810	EXPEED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPEED	Full-frame	CMOS	Nikon	12.1	51	1005	95%
D750	EXPEED 4	Full-frame	CMOS	Nikon[citation ne...	24.9	51	91000	100%
Df	EXPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								

diversity of Wikipedia pages (input)
difficulty of assessing an extractor (output)

WikipediaMatrix

https://en.wikipedia.org/wiki/Comparison_of_Canon_EOS_digital_cameras

Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

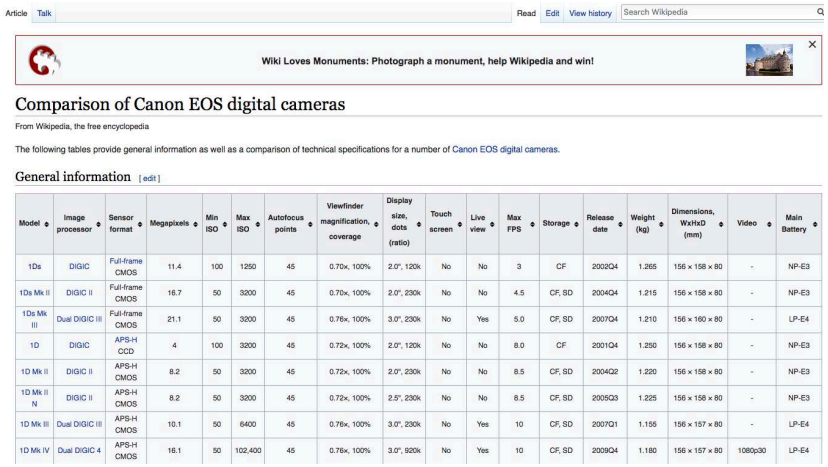
The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4

WikipediaMatrix

https://en.wikipedia.org/wiki/Comparison_of_Canon_EOS_digital_cameras



Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4

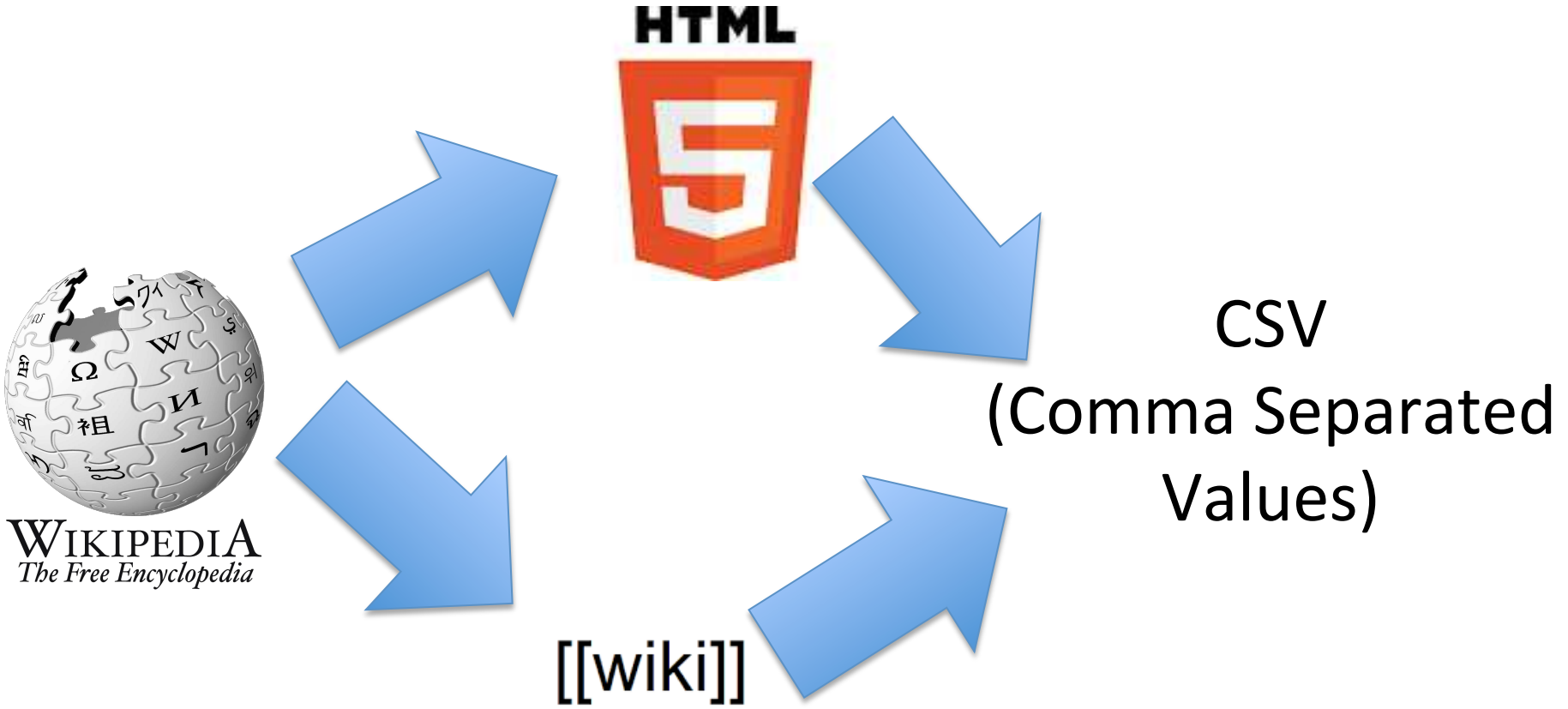
```
==General information==
{| class="wikitable sortable" style="text-align: center; font-size: 85%; width: auto;"
|-
! Model
! Image processor
! Sensor format
! Megapixels
! Min ISO
! Max ISO
! Autofocus points
! Viewfinder
magnification,

coverage
! Display
size, dots

(ratio)
!Touch
screen
! Live view
! Max FPS
! Storage
```

Wikitext

[[wiki]]

```
<tbody><tr>
<th>Model
</th>
<th>Image processor
</th>
<th>Sensor format
</th>
<th>Megapixels
</th>
<th>Min ISO
</th>
<th>Max ISO
</th>
<th>Autofocus points
</th>
<th>Viewfinder
<p>magnification,
</p><p>coverage
</p>
</th>
<th>Display
<p>size, dots
</p><p>(ratio)
</p>
</th>
<th>Touch
<p>screen
</p>
</th>
<th>Live view
</th>
<th>Max FPS
</th>
<th>Storage
</th>
```

WikipediaMatrix



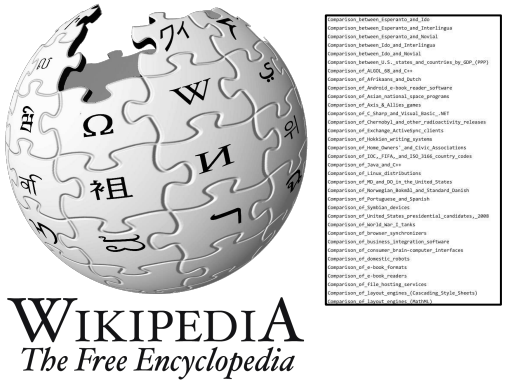
WikipediaMatrix

]]

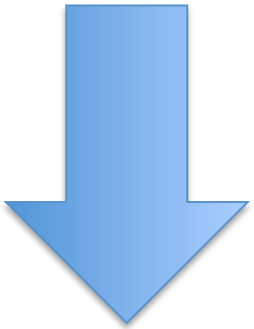
CSV
(Comma Separated Values)

Thousands of Wikipedia pages!

WikipediaMatrix: The Truth



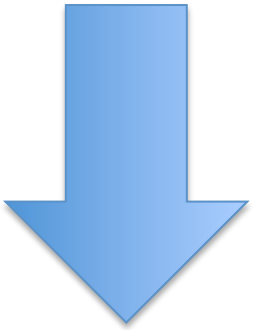
Extractor1



CSV1

Who's right?

Extractor2



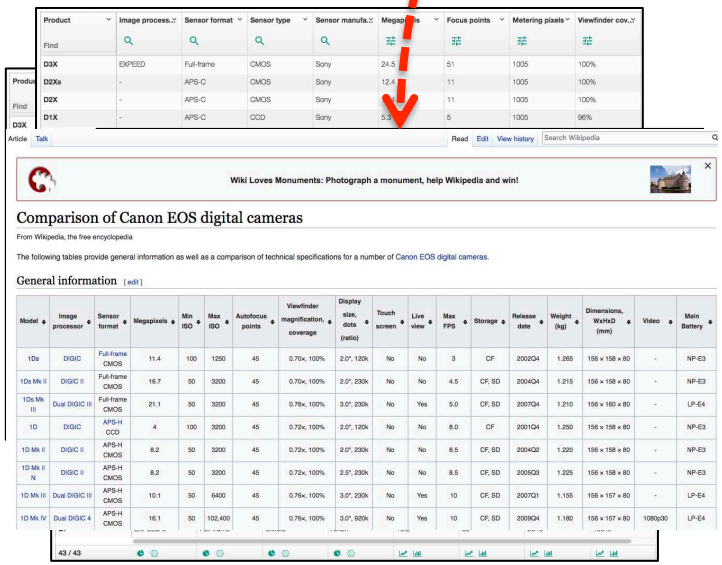
CSV2

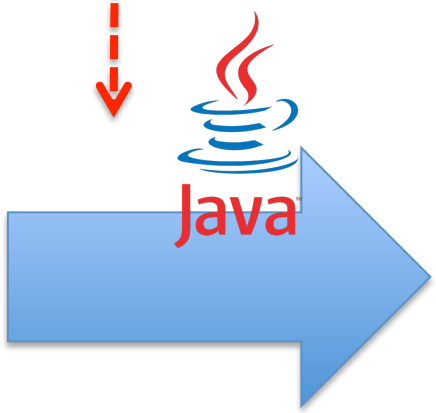
Tests

(input)

(transformation)

(output)



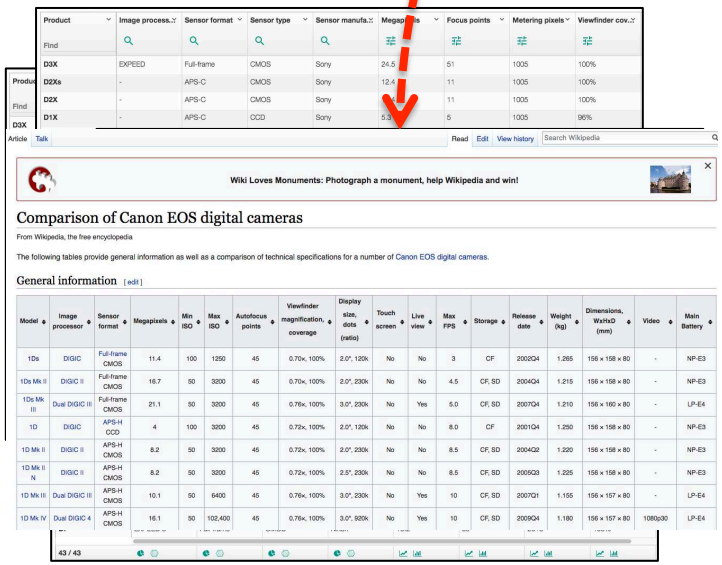


example

CSV

Tests

(inputs)



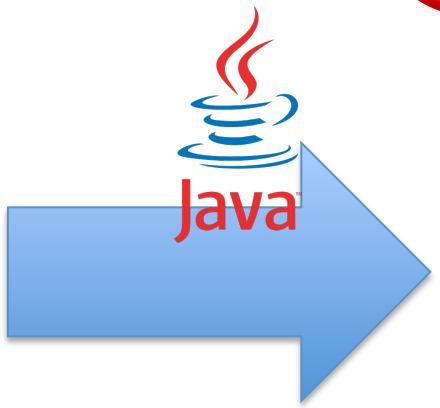
Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

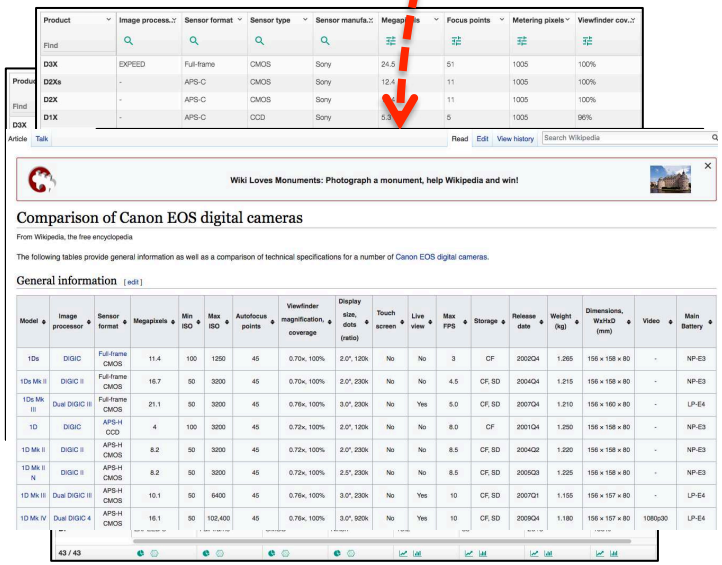


example

CSV

Tests

(inputs)



Comparison of Canon EOS digital cameras

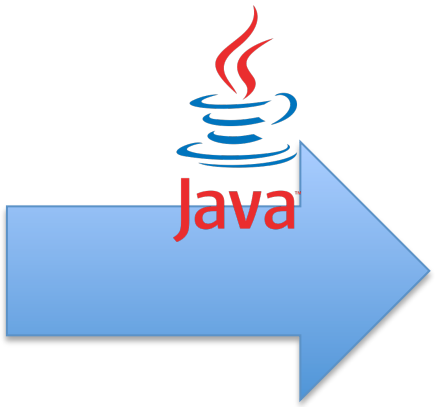
From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4

RuntimeException….

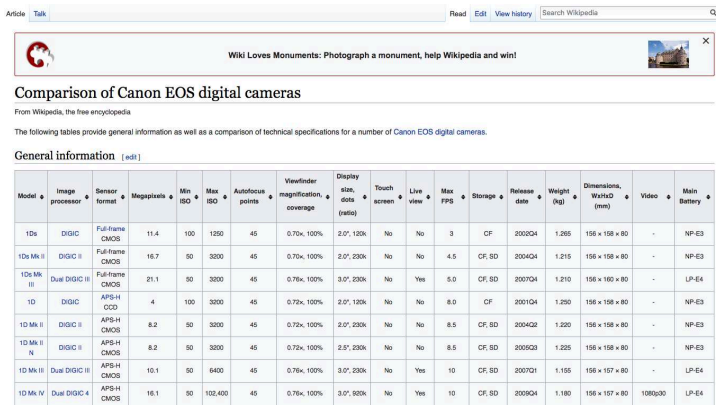


example

CSV

Tests

(input)



Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

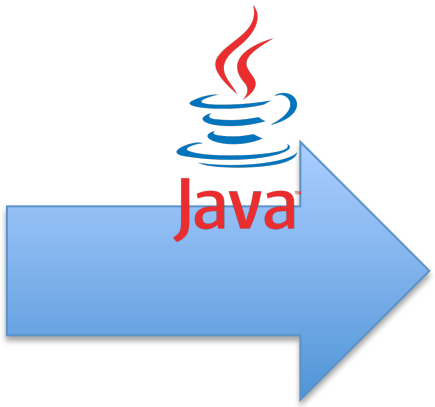
Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

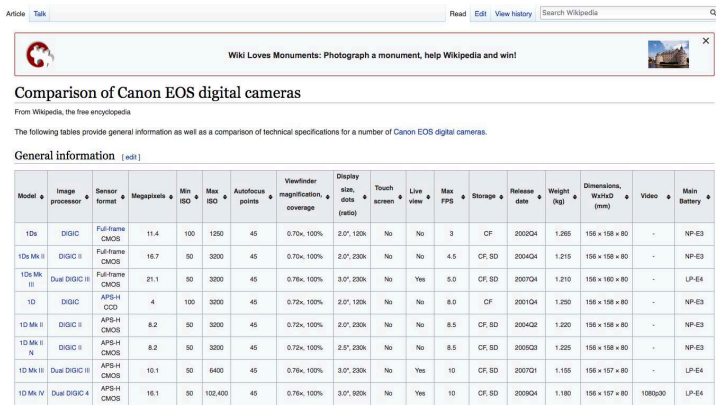
Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC II	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4



F1;f2; , ;
“”.” ”.
, , ,

Tests

(input)



Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

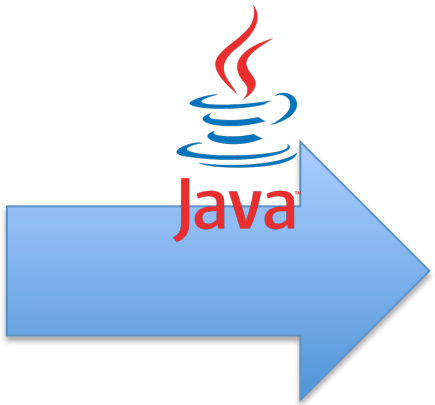
Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4



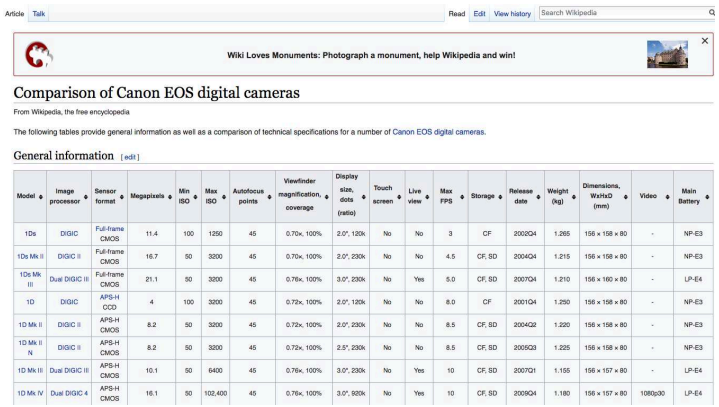
f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

Manual testing is a terrible idea

non reproducible; error-prone; time-consuming



Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

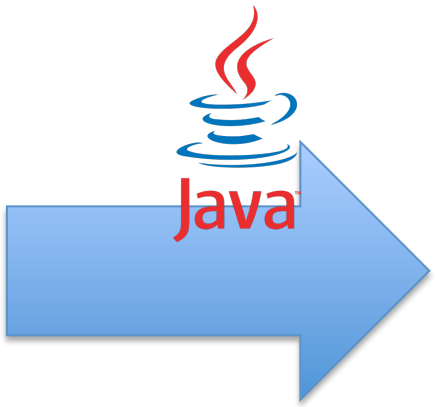
Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC II	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4

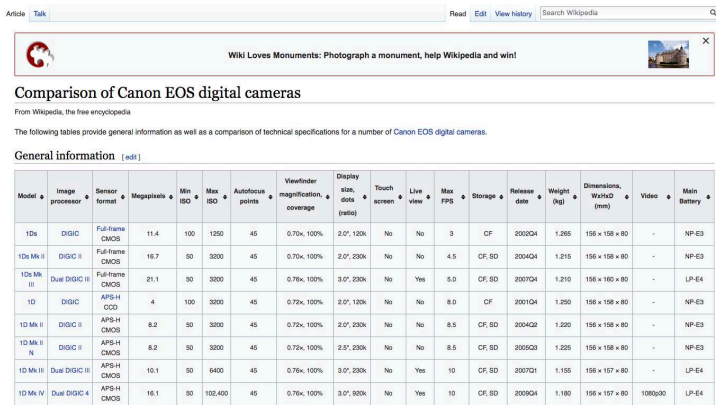


f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

You can start with some values/ inputs and then (manually) observe



Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

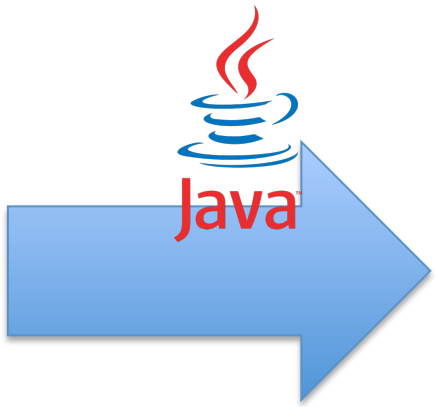
Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4



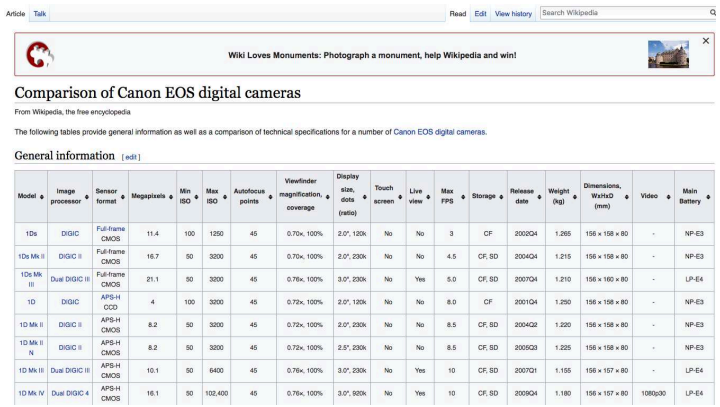
f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

But manual testing is a terrible idea

non reproducible; error-prone; time-consuming



Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: Photograph a monument, help Wikipedia and win!

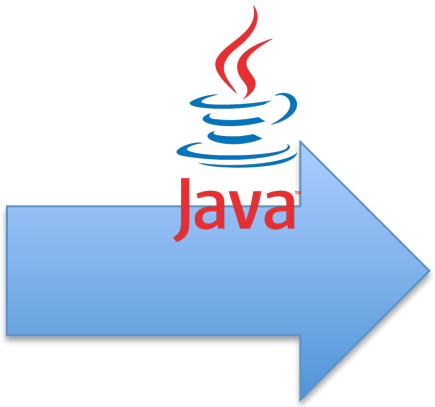
Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touch screen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main Battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70×, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70×, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC II	Full-frame CMOS	21.1	50	3200	45	0.76×, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72×, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72×, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76×, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76×, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4



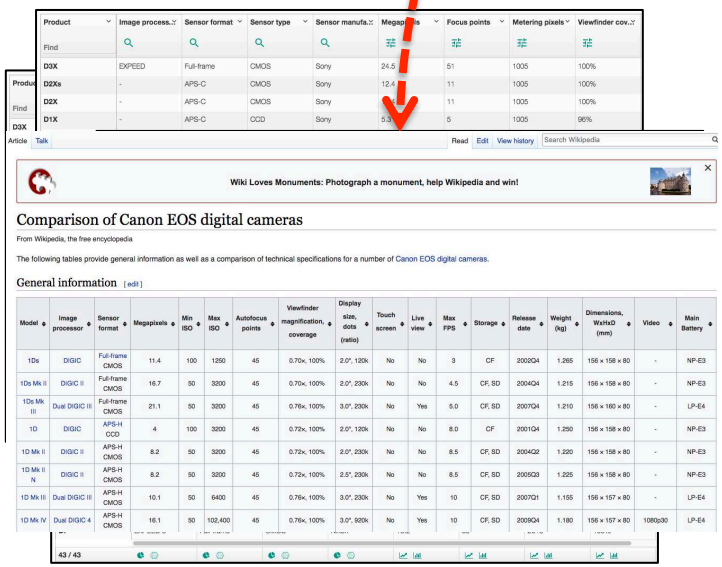
f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

Tests

(input)



Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]

Observations and assertions

f1,f2,f3
v11,v12,v13
v21,v22,v23

example

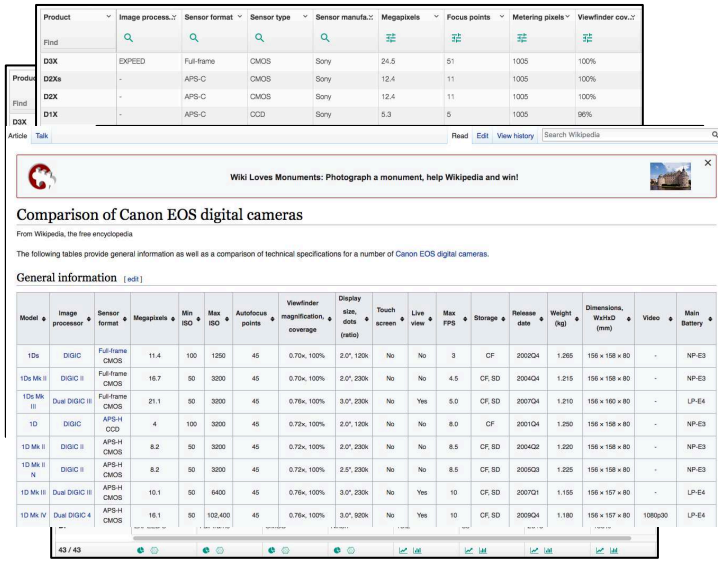
CSV

Automated testing

Case by case verification?

(challenge: identify a set of input that is covering enough cases)

Generic properties?

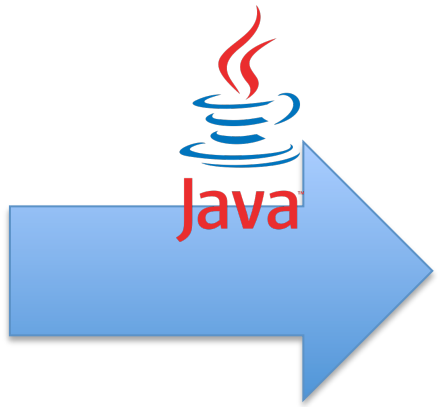


Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of Canon EOS digital cameras.

General information [edit]



f1,f2,f3
v11,v12,v13
v21,v22,v23

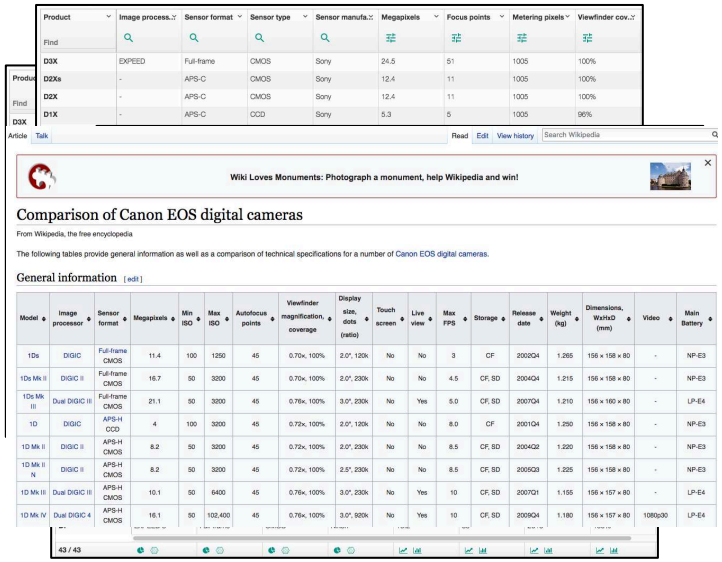
example

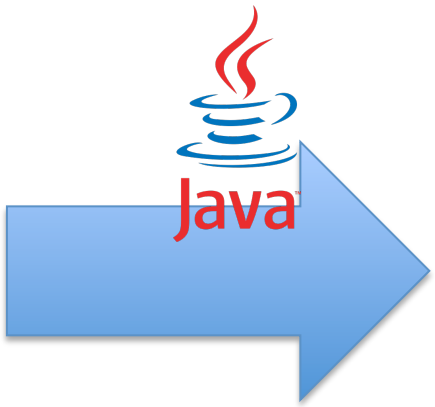
CSV

Automated testing

(You cannot test using one single page!)

Case by case verification? Generic properties?





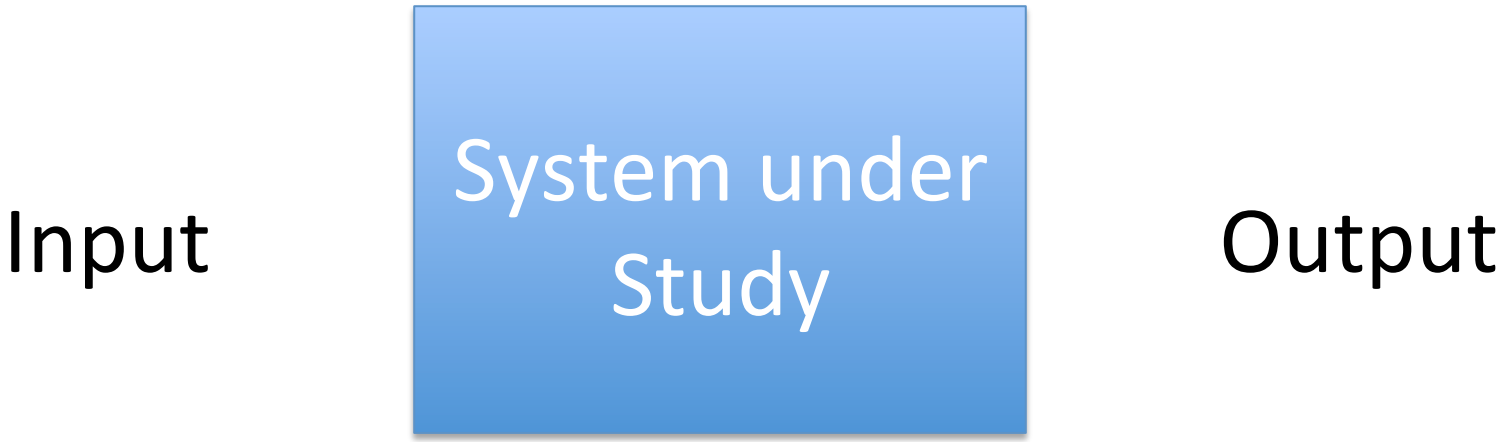
f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

Testing Problem

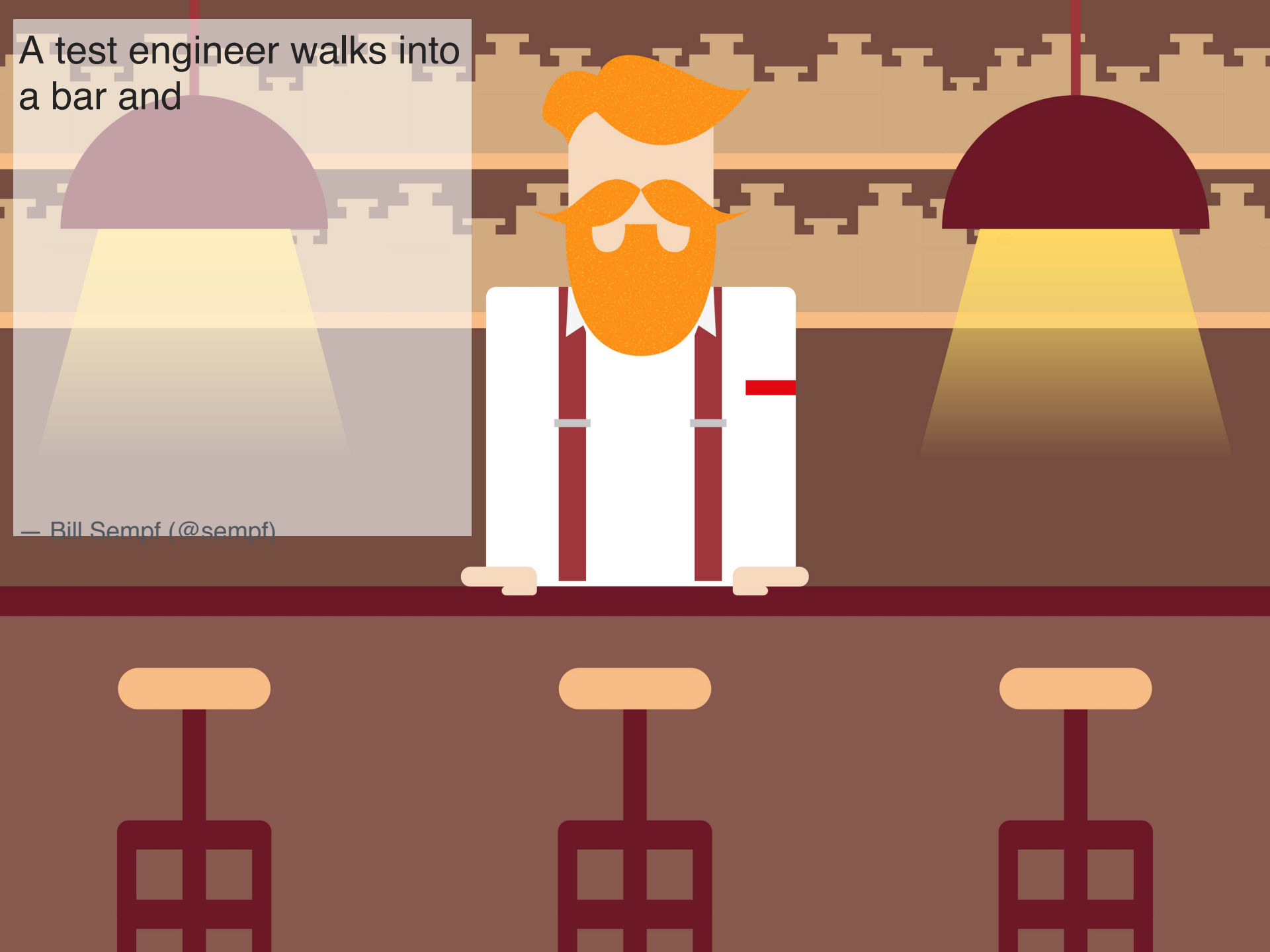
- Test Generation Problem (input)
- Oracle Problem (output)



Assertion-based Testing

- Eg unit tests à la JUnit

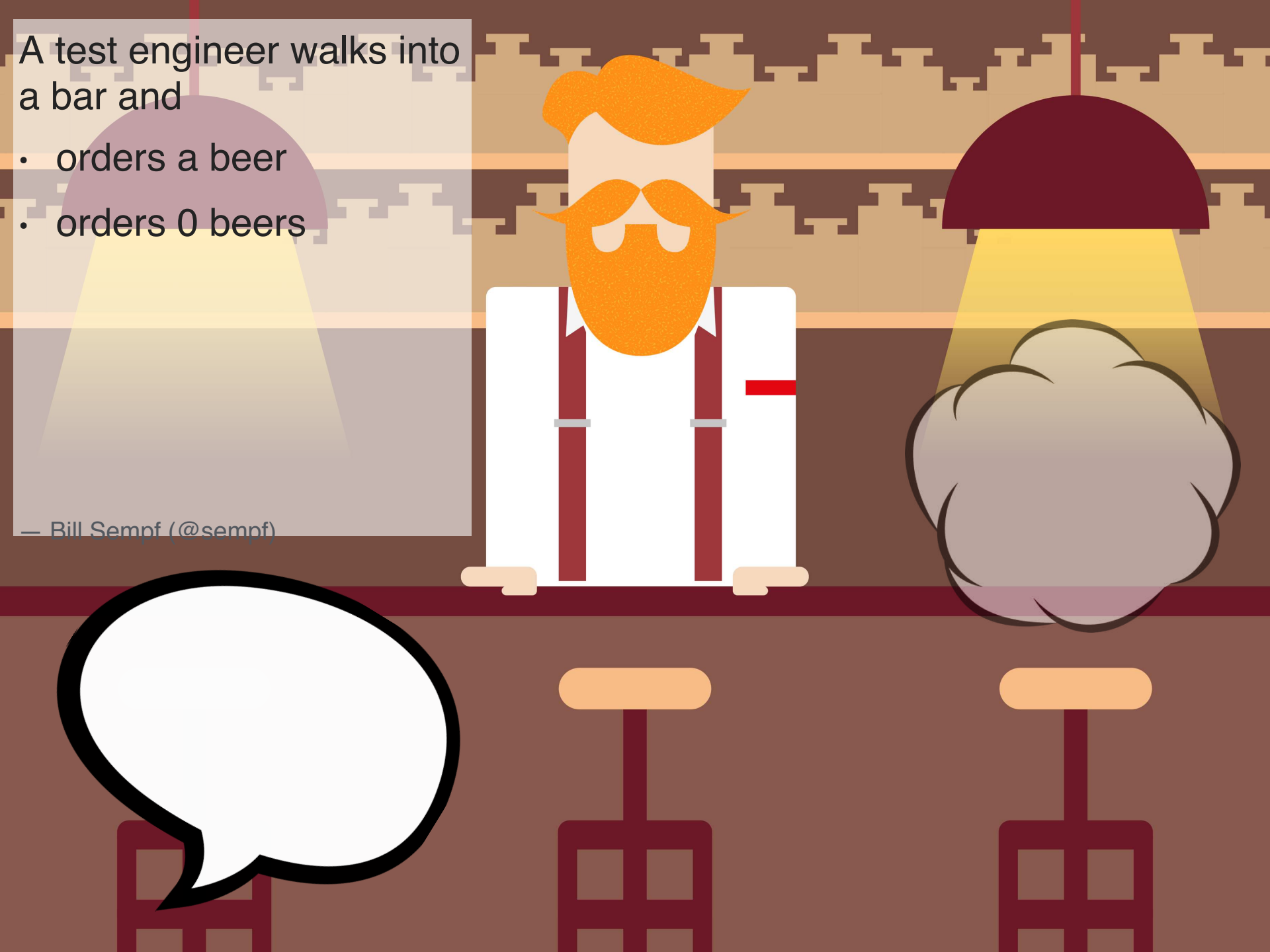
```
@Test
def void basicOrTest() {
        val result = parseHelper.parse('''
                solver sat4j-java
                A v B
        ''')

        val dimacsPrinted = DIMACSPrinter.dimacsFile(result.expression)

        val oracle = "p cnf 2 1\n1 2 0"

        Assertions.assertNotNull(result)
        val errors = result.eResource.errors
        Assertions.assertTrue(errors.isEmpty, '''Unexpected errors: «errors.join(", ")»''')

        Assertions.assertTrue(dimacsPrinted == oracle)
}
```


A test engineer walks into a bar and
— Bill Sempf (@sempf)

A test engineer walks into a bar and

- orders a beer

— Bill Sempf (@sempf)

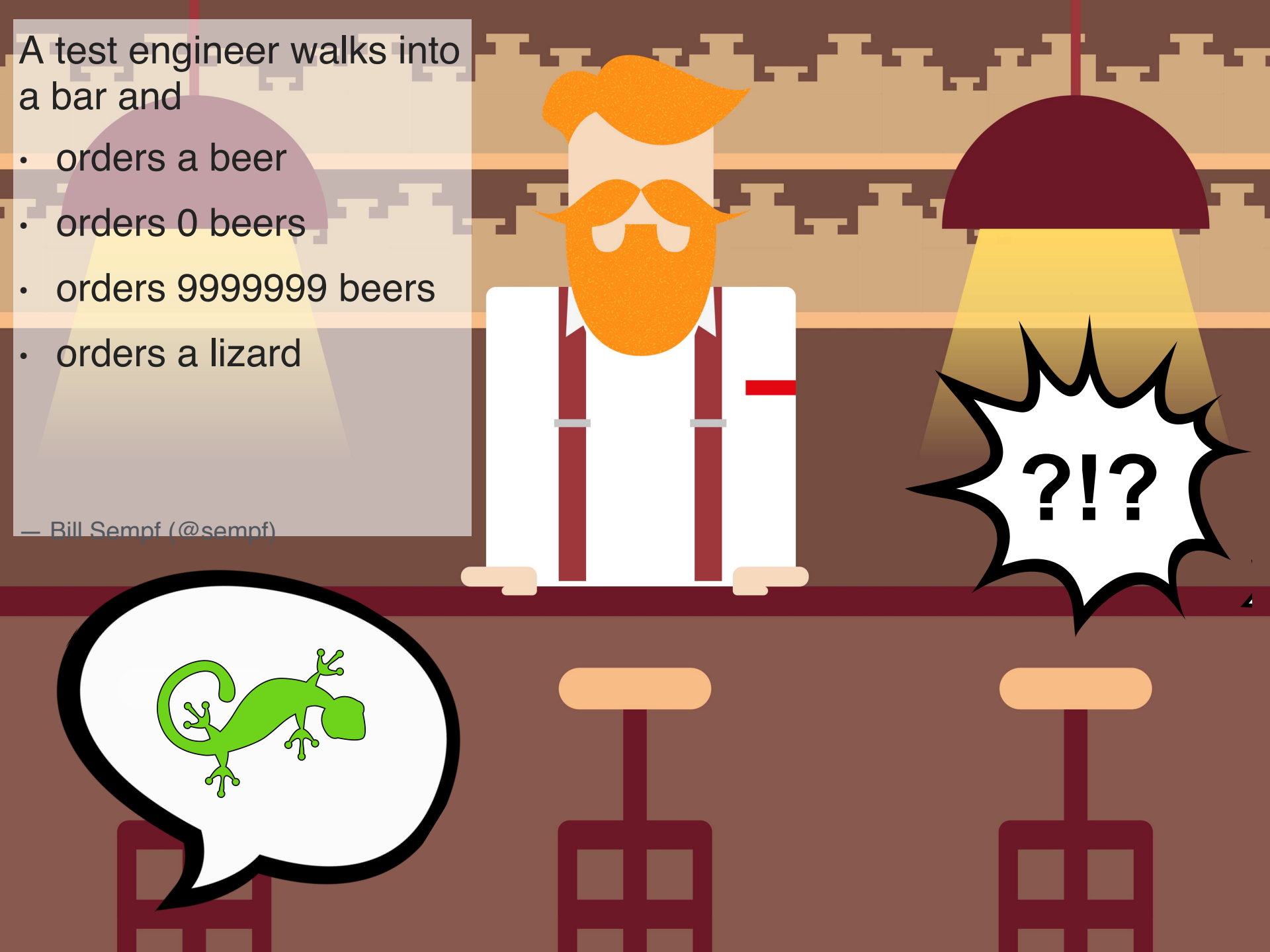

A test engineer walks into a bar and
• orders a beer
• orders 0 beers
— Bill Sempf (@sempf)


A test engineer walks into a bar and
• orders a beer
• orders 0 beers
• orders 9999999 beers
— Bill Sempf (@sempf)
9,999,999

A test engineer walks into a bar and

- orders a beer
- orders 0 beers
- orders 9999999 beers
- orders a lizard

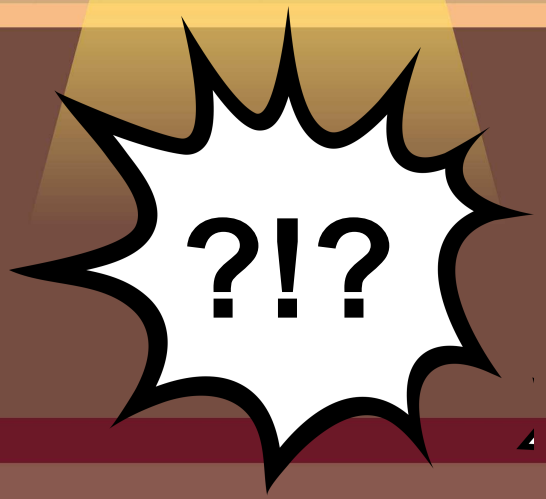
— Bill Sempf (@sempf)



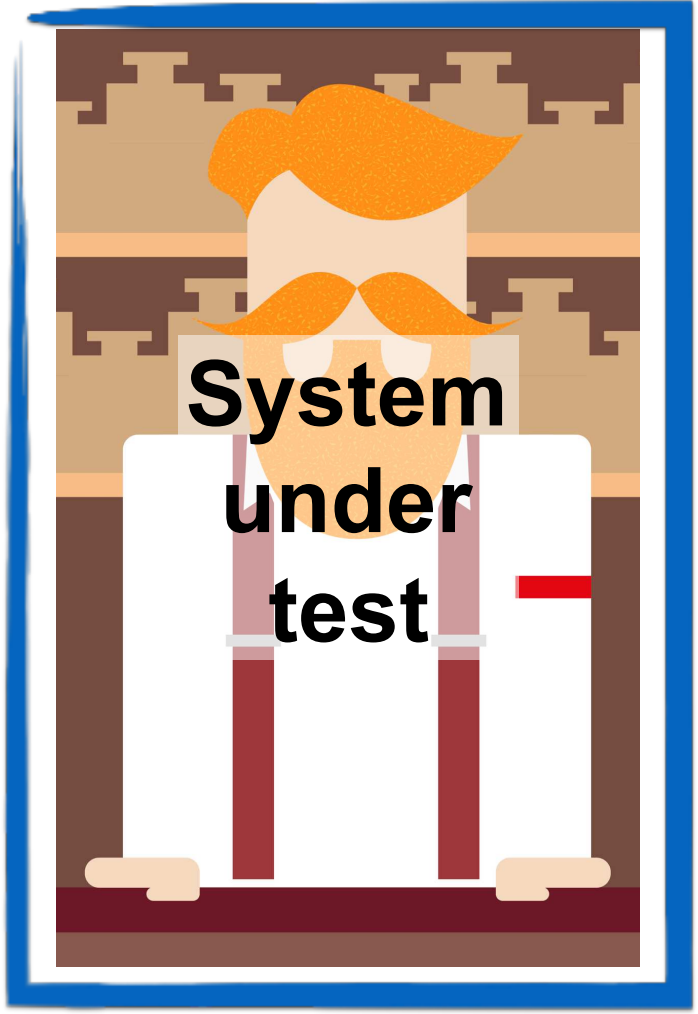
A test engineer walks into a bar and

- orders a beer
- orders 0 beers
- orders 9999999 beers
- orders a lizard
- orders -1 beers

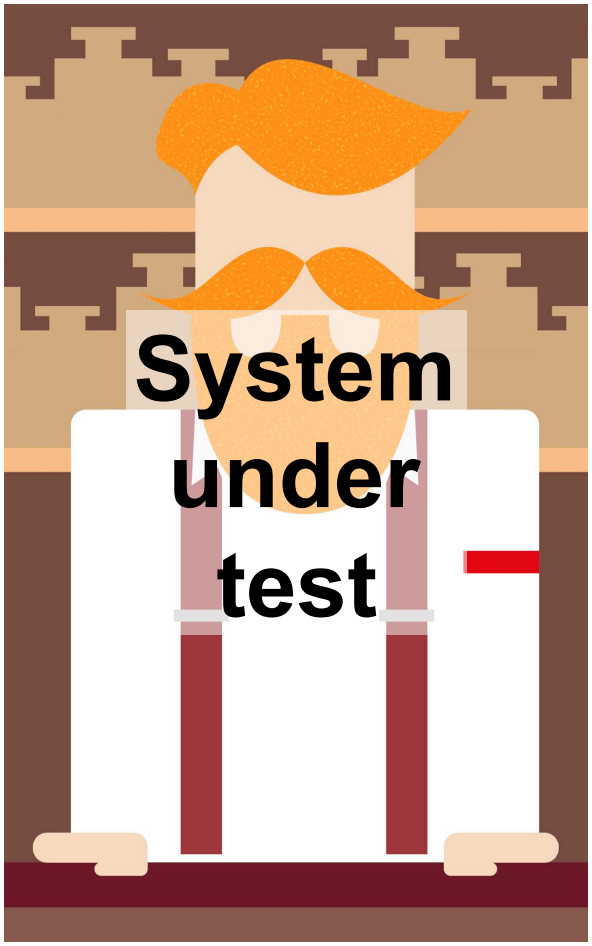
— Bill Sempf (@sempf)



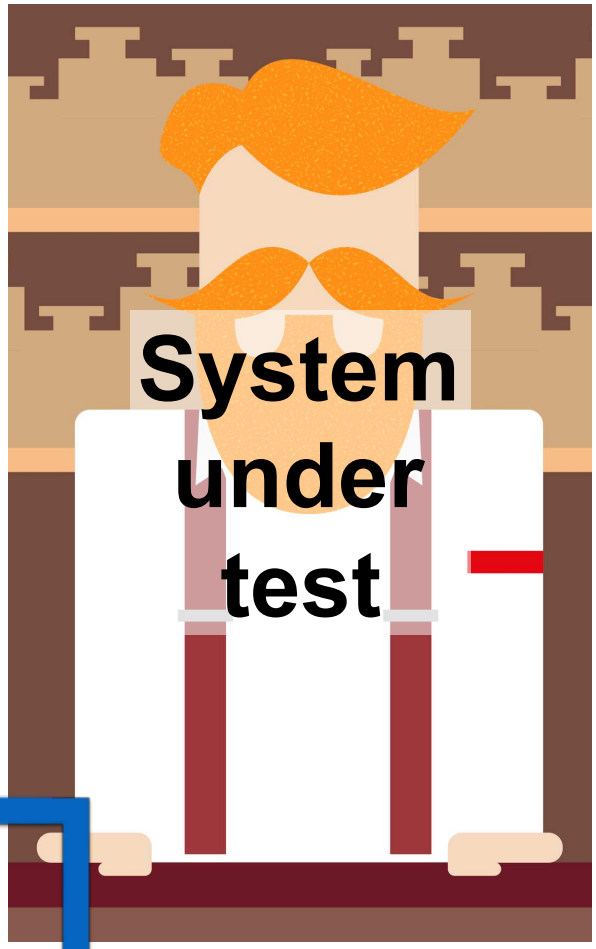

A test engineer walks into a bar and
• orders a beer
• orders 0 beers
• orders 9999999 beers
• orders a lizard
• orders -1 beers
• orders a "sfdeljknesv"
— Bill Sempf (@sempf)
?!?
"sfdeljknesv"

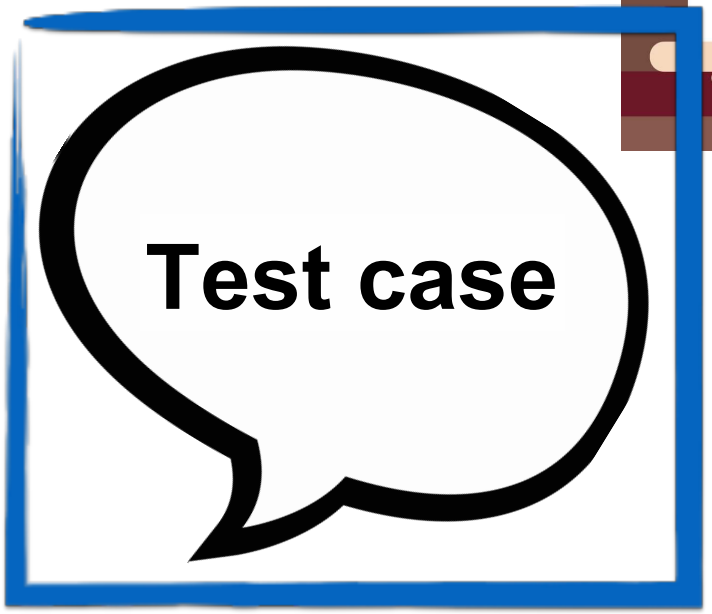

System
under
test

Specification

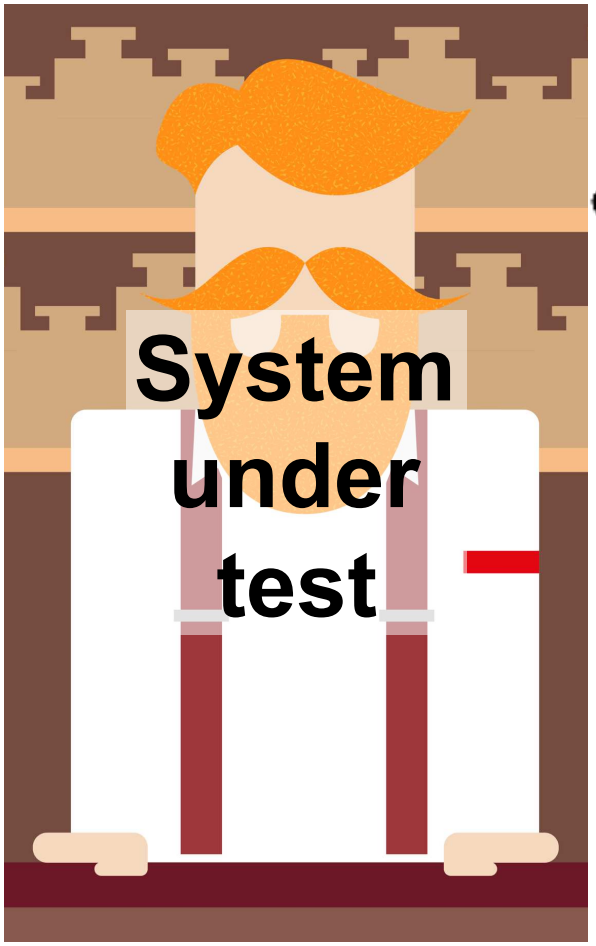


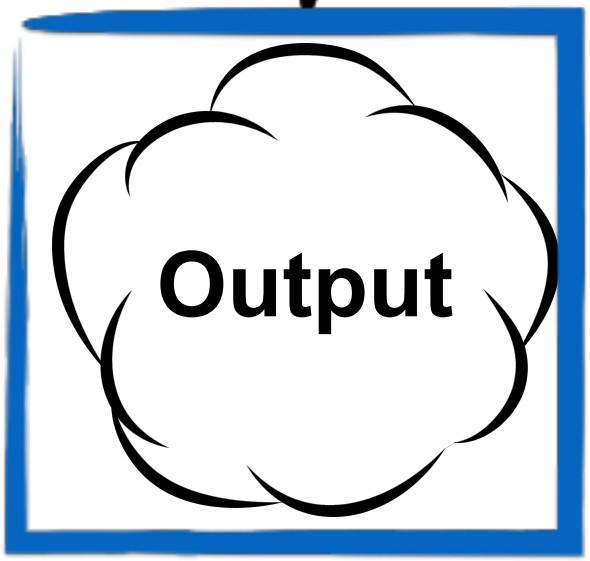
Specification



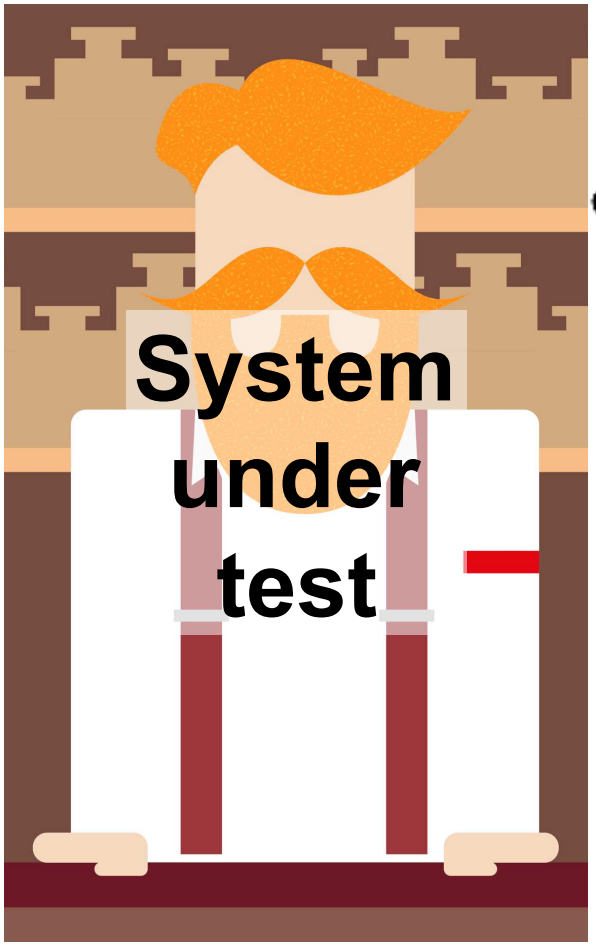


Specification





Specification



Output

Test case

Pass or fail?

What is a test?

A test consists of

- an input
- an oracle

End-to-end test:

- Batch program: input = file, oracle = expected file
- Interactive program: input = UI events, oracle = windows

Unit test:

- Input = sequence of calls
- Oracle = `assert` statement

Example unit test

```
Object[] a = new Object[];
LinkedList ll = new LinkedList();
ll.addFirst(a);
TreeSet ts = new TreeSet(ll);
Set u =
Collections.unmodifiableSet(ts);

assert u.equals(u);
```

input

oracle

Assertion fails:
used to be a bug in the JDK!

```
@Test
def void basicBiImplTest() {
        val result = parseHelper.parse('''
                solver sat4j-java
                A <=> B
        ''')
        val prettyPrinted = PrettyPrinter.prettyPrint(result.expression)

        val oracle = "(A <=> B)"

        Assertions.assertNotNull(result)
        val errors = result.eResource.errors
        Assertions.assertTrue(errors.isEmpty, '''Unexpected errors: «errors.join(", ")»''')

        Assertions.assertTrue(prettyPrinted == oracle)
}
```

```
@Test
def void basicOrTest() {
        val result = parseHelper.parse('''
                solver sat4j-java
                A v B
        ''')

        val dimacsPrinted = DIMACSPrinter.dimacsFile(result.expression)

        val oracle = "p cnf 2 1\n1 2 0"

        Assertions.assertNotNull(result)
        val errors = result.eResource.errors
        Assertions.assertTrue(errors.isEmpty, '''Unexpected errors: «errors.join(", ")»''')

        Assertions.assertTrue(dimacsPrinted == oracle)
}
```

Strengths and Limits

- Precise oracles
- Domain-specific knowledge
- Examples-based
  - Input test is limited by construction
  - Incomplete (cases)
- Tests subject to developers bias
- Costly
  - One specific oracle per example (labelling)

Mutation Testing

- Based on Caroline Landry (DiverSE team, STAMP project) slides at Breizh Camp

Test Your Tests

- What do you expect from test cases?
  - Cover requirements
  - Stress the application
  - Prevent regressions
  - Reveal bugs
- **Quality** of test suite matters!
  - But what's a good test suite?

Test Your Tests

- What do you expect from test cases?
  - Cover requirements
  - Stress the application
  - Prevent regressions
  - **Reveal bugs**
- **Quality** of test suite matters!
  - But what's a good test suite?

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <=
n; i++) {
   result = result * i;
  }
  return result;
}
```

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <=
n; i++) {
   result = result * i;
  }
  return result;
}
```

Coverage

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
   result = result * i;
  }
  return result;
}
```

```
@Test
factorialWith5Test() {
  long obs = fact(5);
  assertTrue(5 < obs);
}
```

Coverage

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
   result = result * i;
  }
  return result;
}
```

```
@Test
factorialWith5Test() {
  long obs = fact(5);
  assertTrue(5 < obs);
}
@Test
factorialWith0Test() {
  assertEqual(1, fact(0));
}
```

Coverage

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 1; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

```
@Test
public void test() {
  assertEqual(1, fact(1));
}

@Test
public void test() {
  assertEqual(1, fact(0));
}
```

Is this test suite good at detecting bugs?

Coverage

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 1; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

```
@Test
public void test() {
  assertEquals(1, fact(0));
}

@Test
public void test() {
  assertEquals(6, fact(3));
}
```

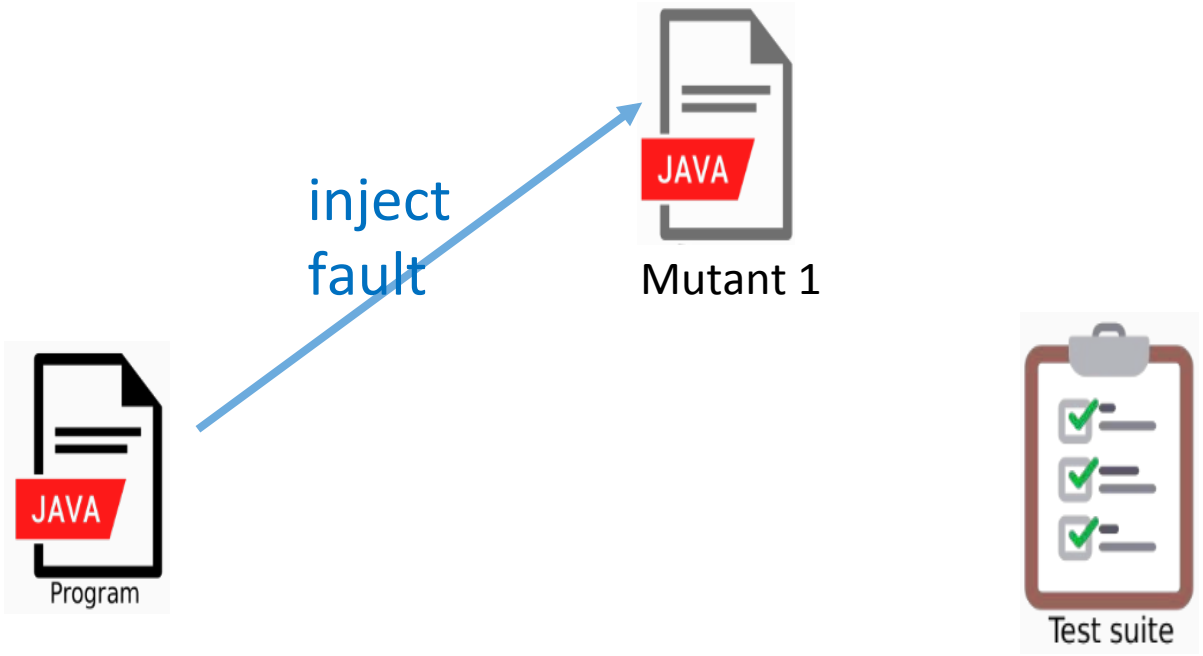
Is this test suite good at detecting bugs?

Let's mutate our code to see.

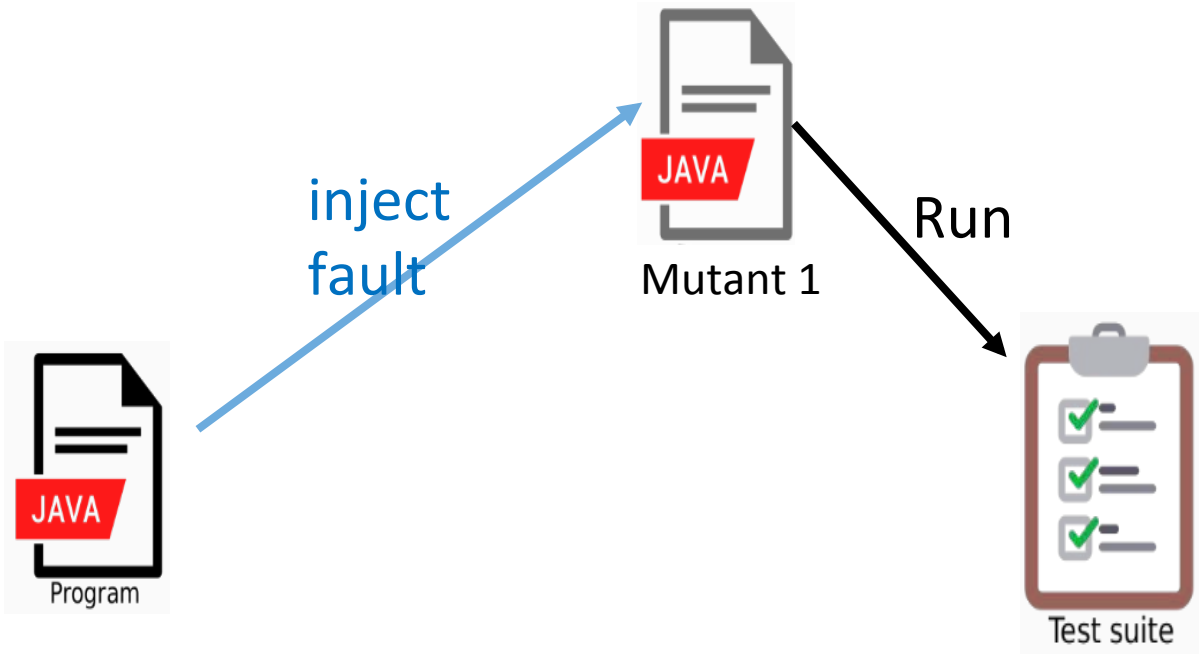
Mutation Analysis

European Commission

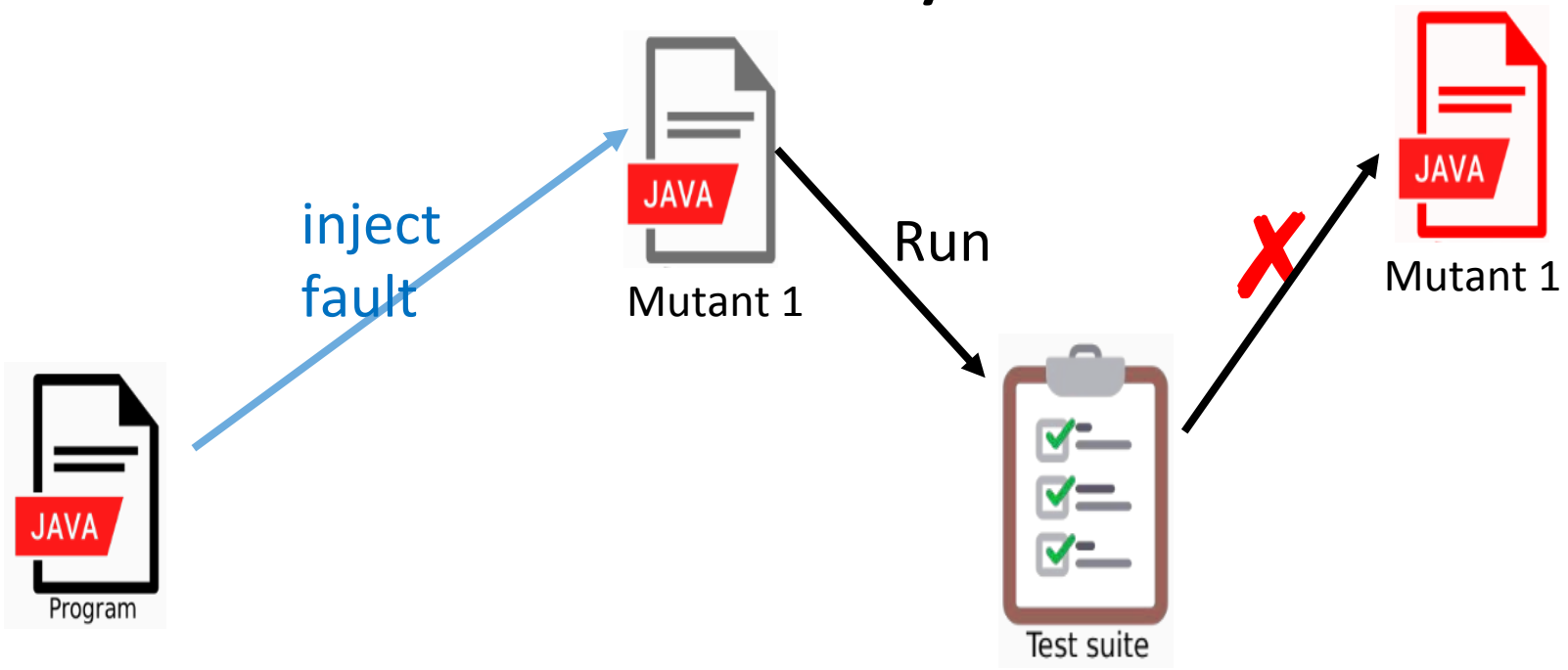
Mutation Analysis



Mutation Analysis

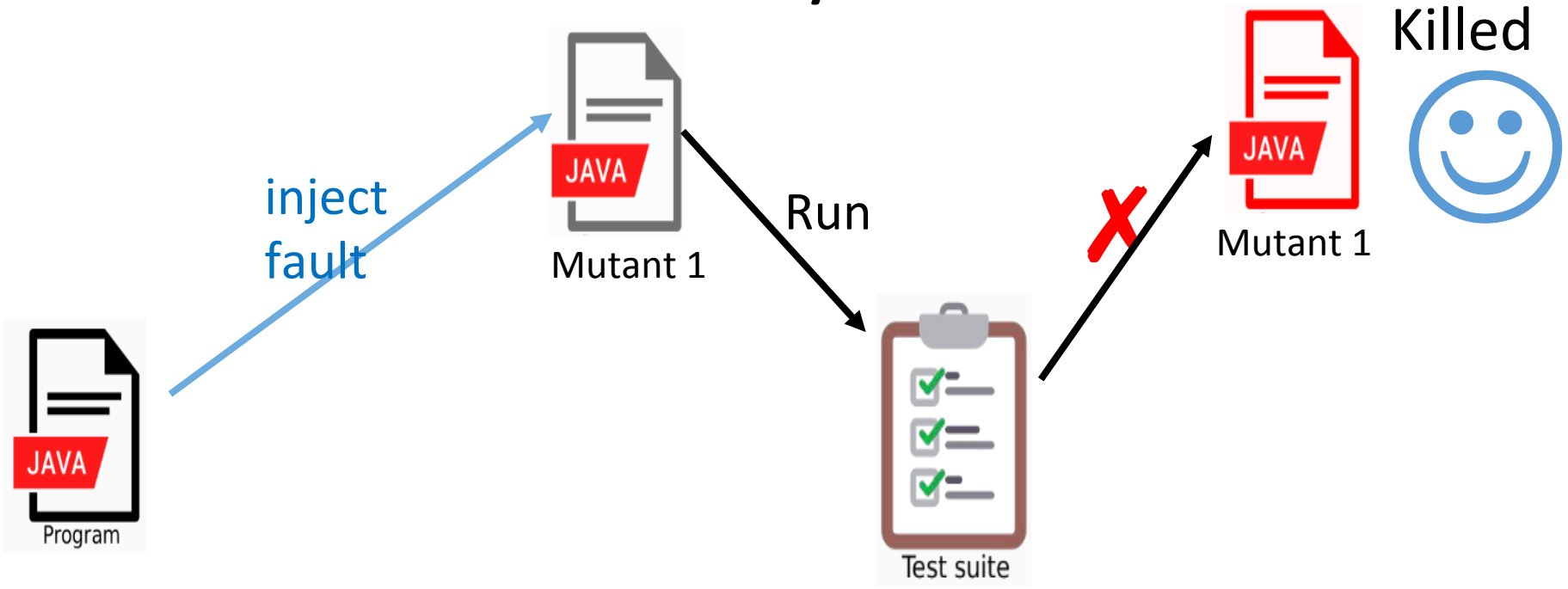


Mutation Analysis



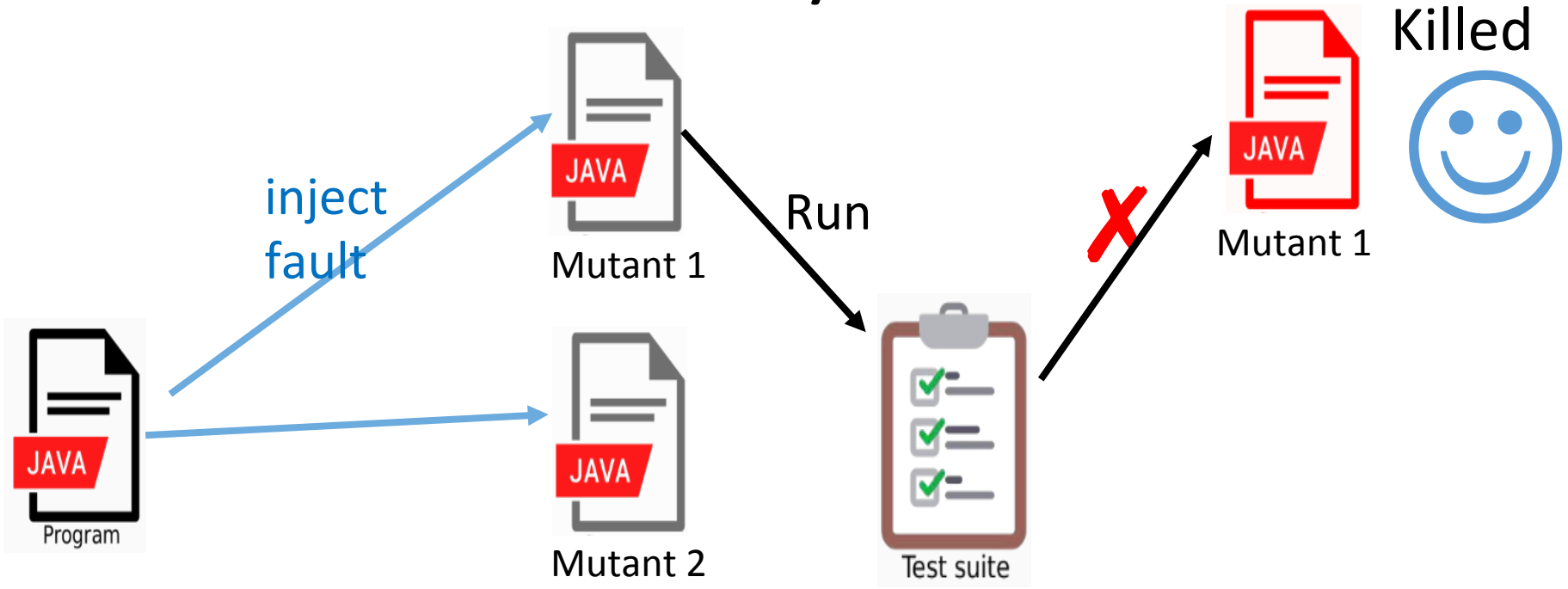
European Commission

Mutation Analysis



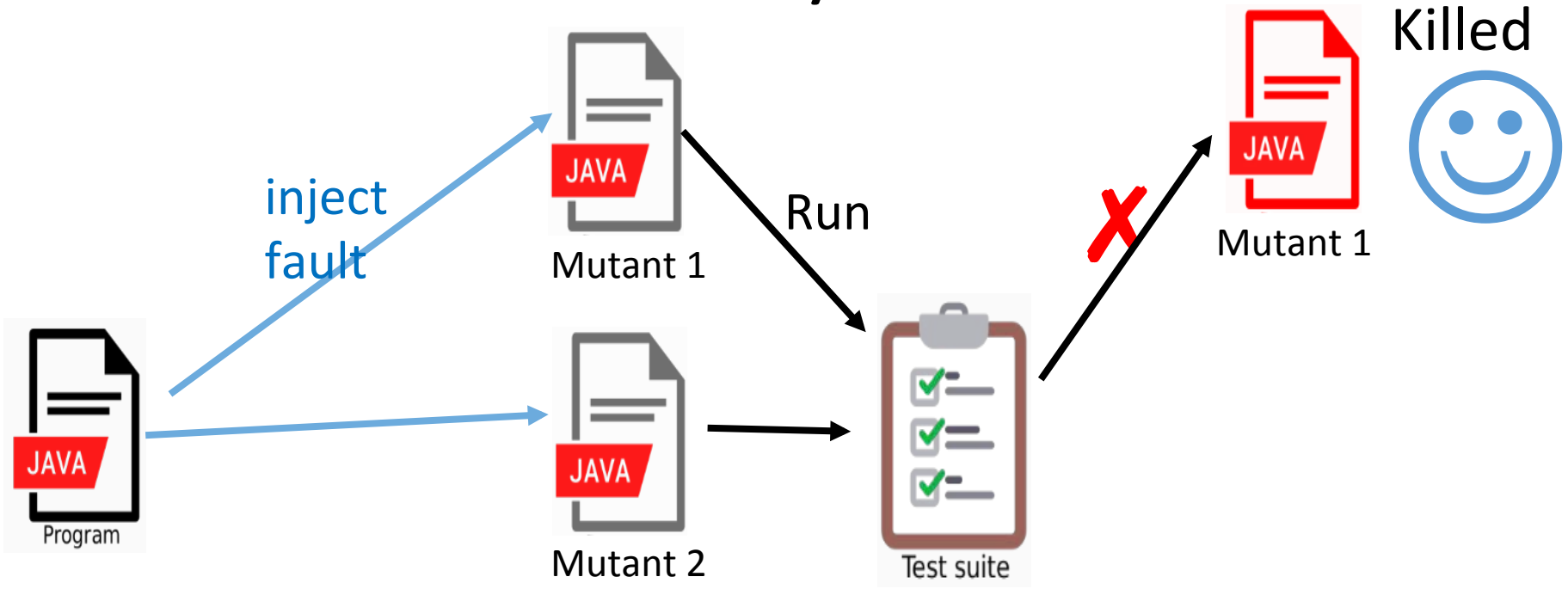
European Commission

Mutation Analysis



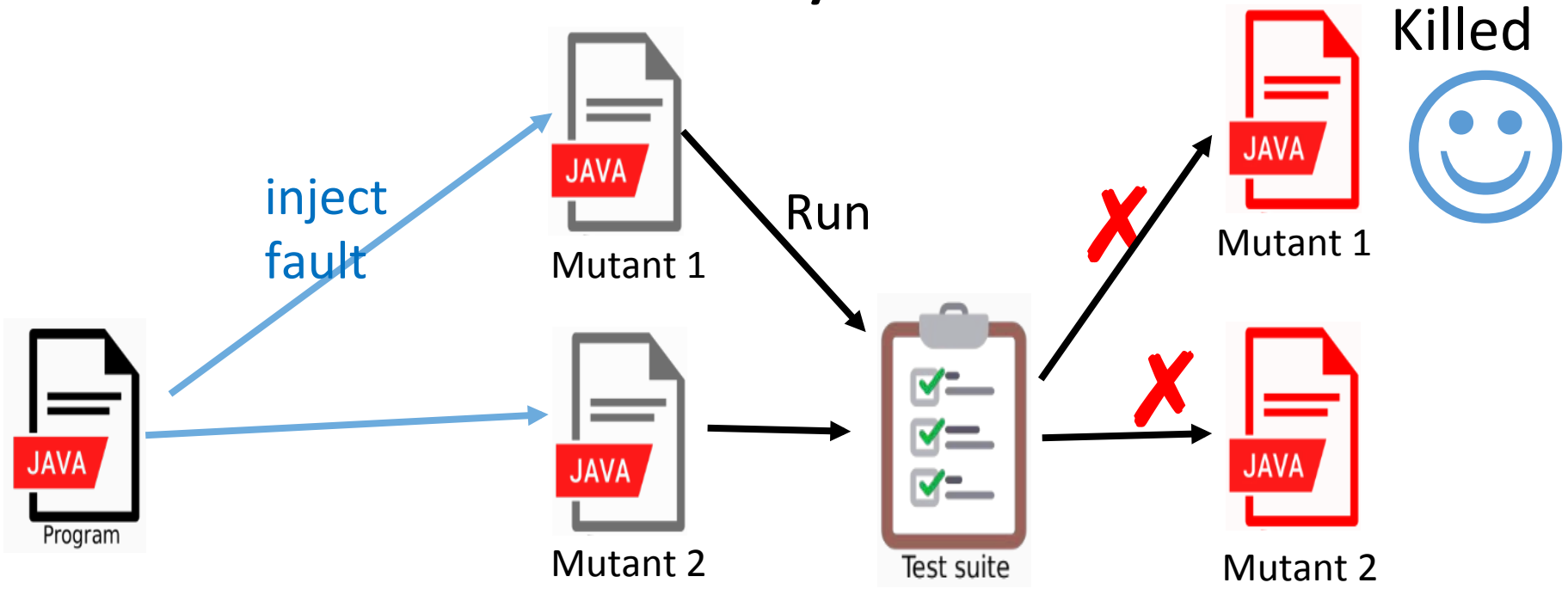
European Commission

Mutation Analysis



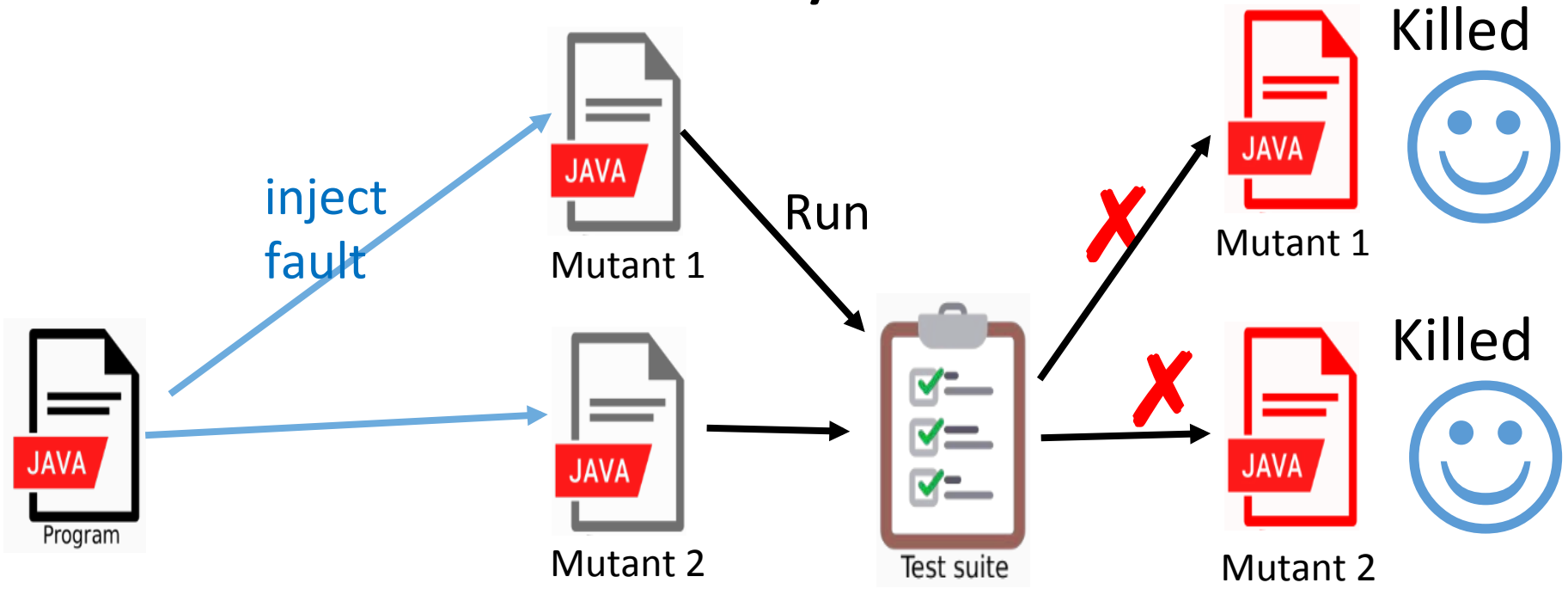
European Commission

Mutation Analysis



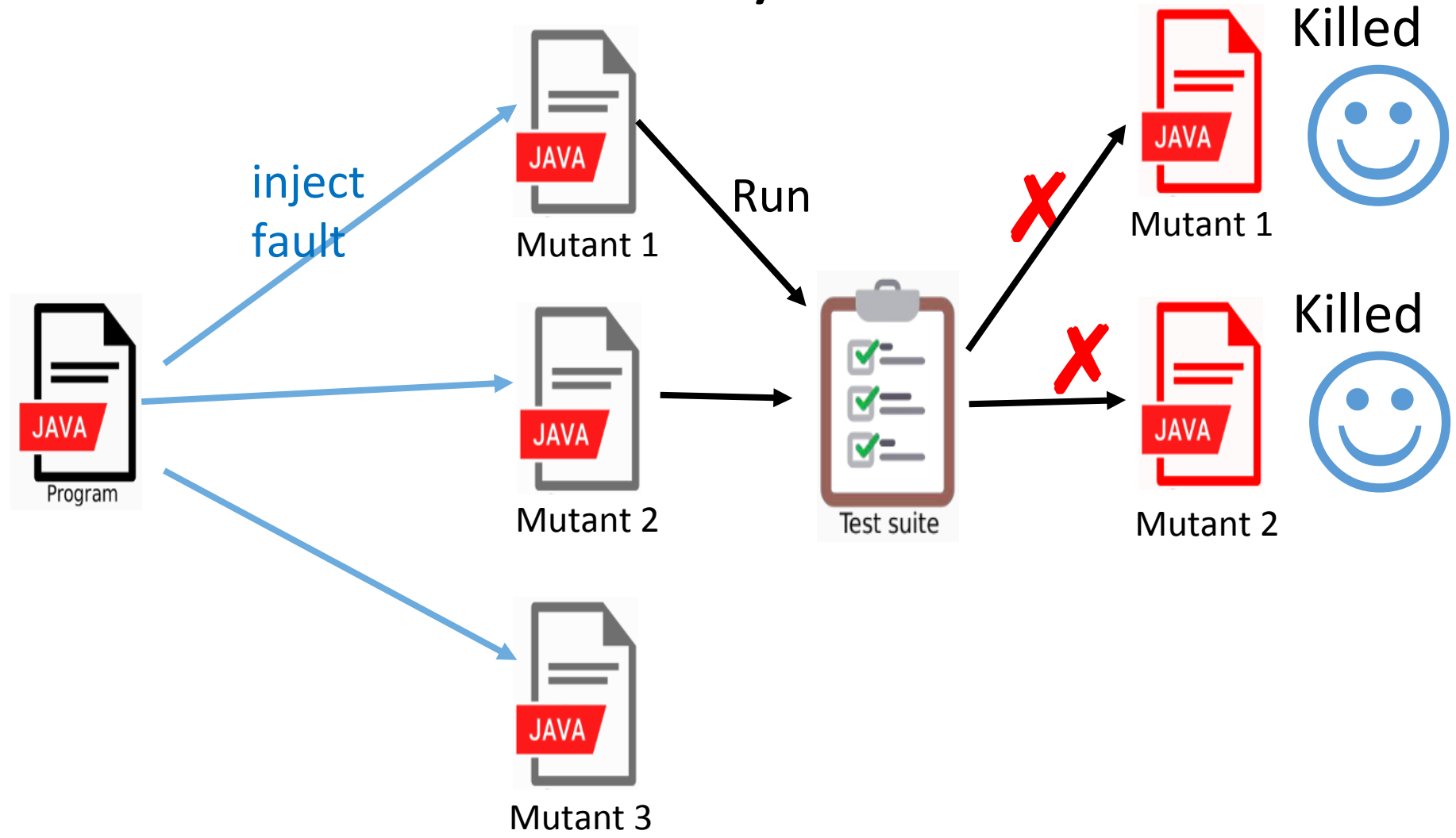
European Commission

Mutation Analysis



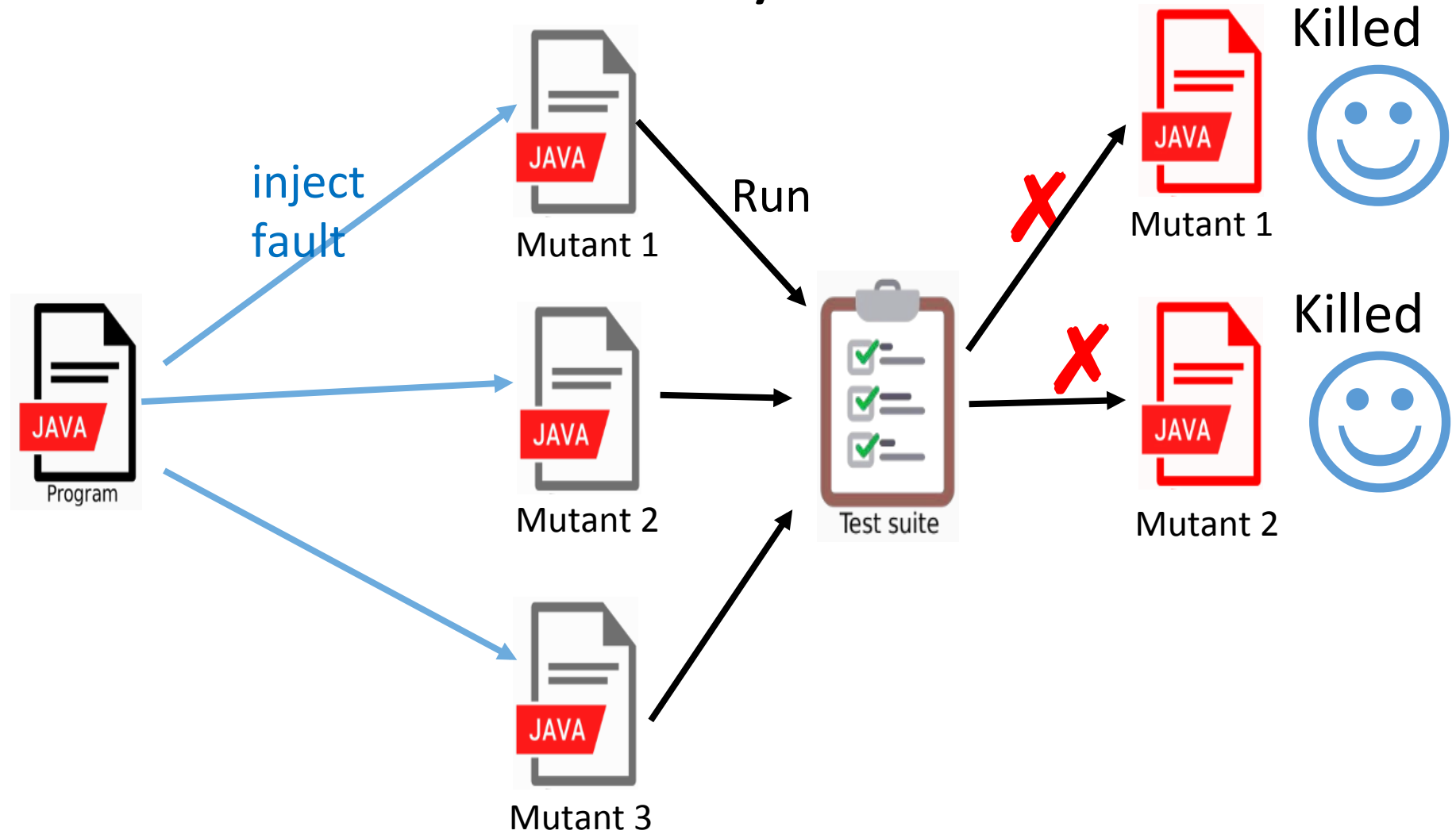
European Commission

Mutation Analysis



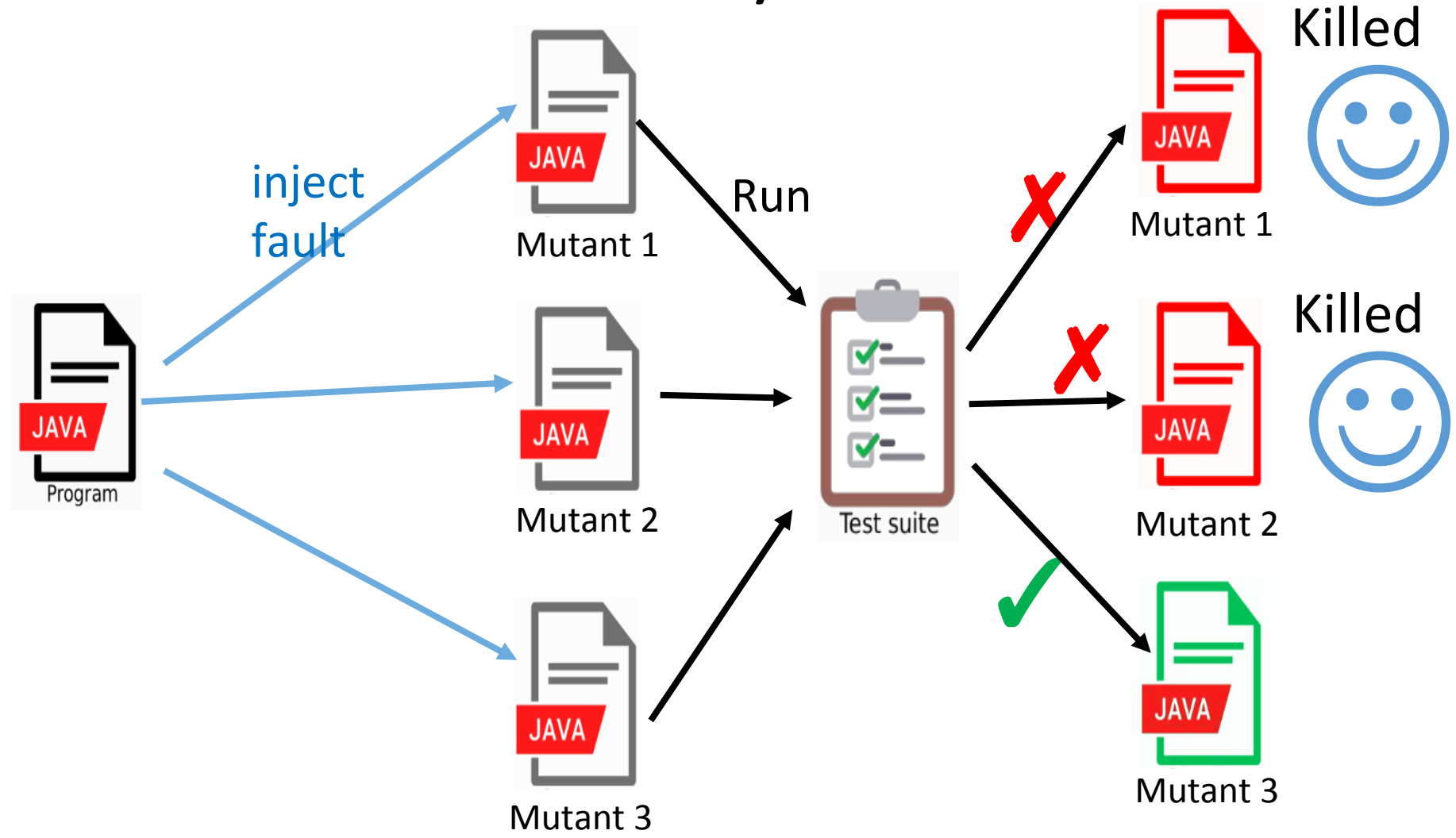
European Commission

Mutation Analysis



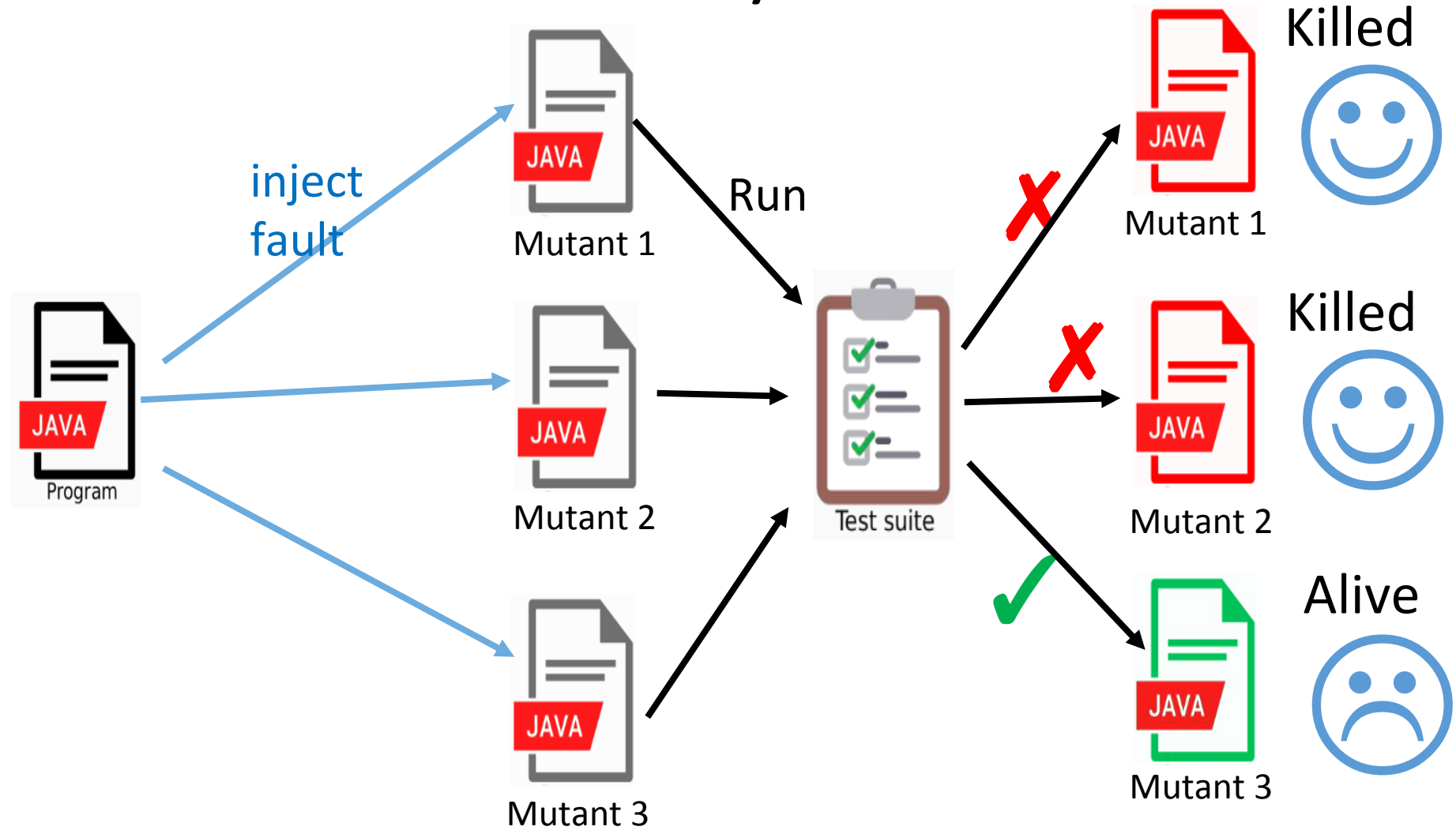
European Commission

Mutation Analysis



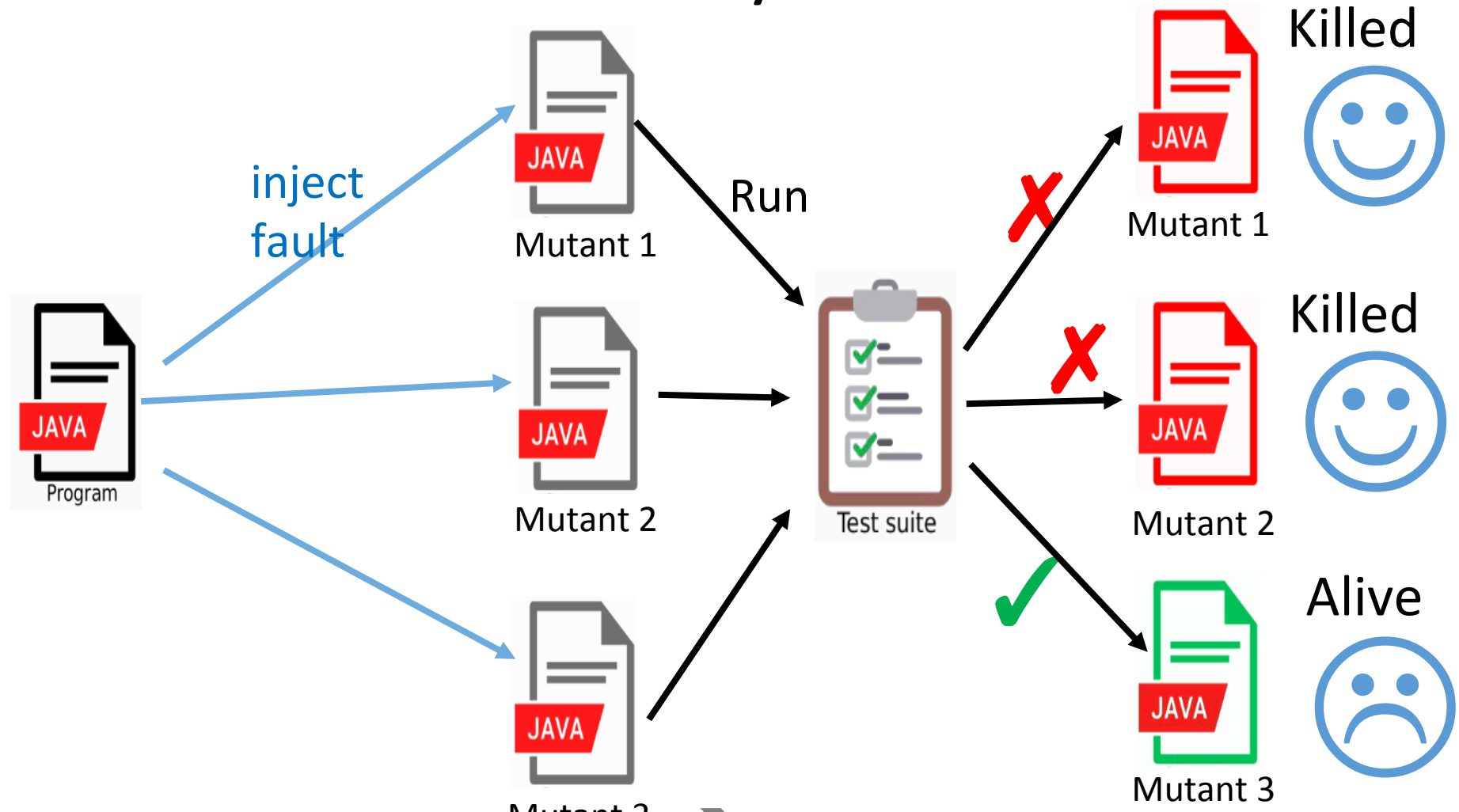
European Commission

Mutation Analysis



European Commission

Mutation Analysis



➔ Mutation score = 66%

European Commission

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

```
@Test
factorialWith5Test() {
  long obs = fact(5);
  assertTrue(5 < obs);
}

@Test
factorialWith0Test() {
  assertEqual(1, fact(0));
}
```

Example

```
long fact(int n) {
  if (n != 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

✗

```
@Test
factorialWith5Test() {
  long obs = fact(5);
  assertTrue(5 < obs);
}
```

✓

```
@Test
factorialWith0Test() {
  assertEqual(1, fact(0));
}
```

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <=
n; i++) {
   result = result * i;
  }
  return result;
}
```

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

n!=0

return 1+1

i < n

!(i<=n)

i--

result/i

result+1

```
@Test
factorialWith5Test() {
  long obs = fact(5);
  assertTrue(5 < obs);
}
```

European Commission

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

n!=0

return 1+1

i < n

!(i<=n)

i--

result/i

result+1

```
@Test
factorialWith0Test() {
  assertEqual(1, fact(0));
}
```

European Commission

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

n!=0

return 1+1

i < n

!(i<=n)

i--

result/i

result+1

```
@Test
factorialWith5Test() {
  long obs = fact(5);
  assertTrue(5 < obs);
}
@Test
factorialWith0Test() {
  assertEqual(1, fact(0));
}
```

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <=
n; i++) {
   result = result * i;
  }
  return result;
}
```

➔ Mutation score = 71%

Test suite:

- Weak oracle
- Missing input

```
@Test
factorialWith5Test(
) {
  long obs =
fact(5);
  assertTrue(5 <
obs);
}

@Test
factorialWith0Test(
) {
  assertEqual(1,
fact(0));
}
```

Example

```
long fact(int n) {
  if (n == 0) {
    return 1;
  }
  long result = 1;
  for (int i = 2; i <= n; i++) {
    result = result * i;
  }
  return result;
}
```

n!=0

return 1+1

i < n

!(i<=n)

i--

result/i

result+1

```
@Test
factorialWith5Test()
{
  long obs = fact(5);
  assertTrue(5 < obs);
}

@Test
factorialWith0Test() {
  assertEqual(1, fact(0));
}

@Test
factorialWith5Test() {
  assertEqual(120, fact(5));
}
```

Mutation Analysis

- Tests are good if they can detect bugs
- Mutation operators
  - Based on common faults
- PIT or PITest
  - Open source, in active development and production ready
  - Integrates with major build systems
  - State of the art mutation testing
  - Extensible via plugins
  - Concurrent execution
  - Test selection

pitest.org

Limitations of mutation testing

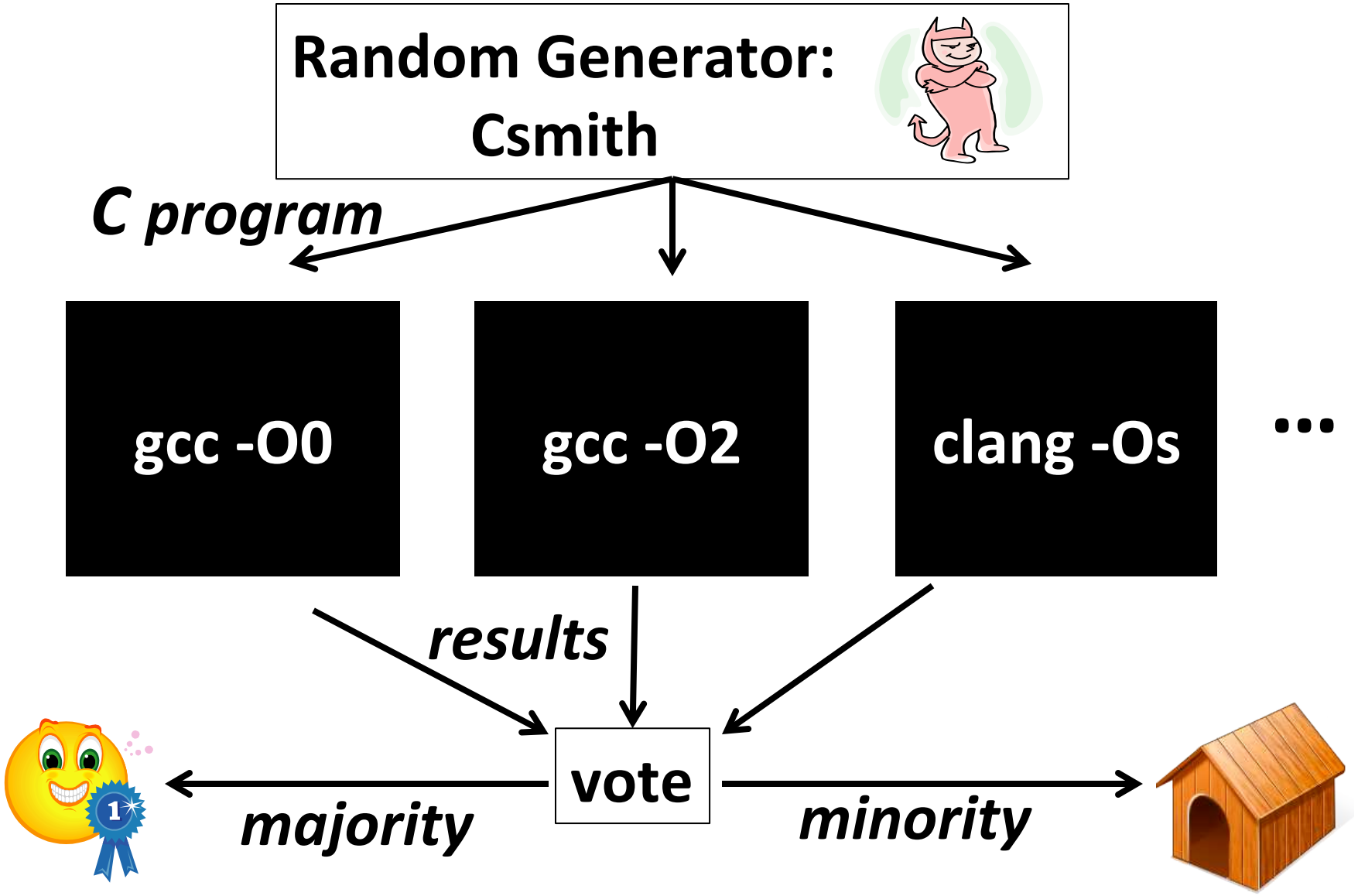
- Expensive computation
- Huge number of mutants
- Presence of equivalent mutants

```
int Min (int i, intj){
  int minval = i;
  if (j<i) then minval = j;
  return minval
}
```

```
int Min (int i, intj){
  int minval = i;
  if (j<minval) then minval = j;
  return minval
}
```

Test Input Generation

- The case of compilers
- Csmith PLDI'11
- Equivalence modulo inputs PLDI'14


Random Generator:
Csmith
C program
gcc -O0
gcc -O2
clang -Os
...
results
vote
majority
minority

Finding and Understanding Bugs in C Compilers

Xuejun Yang Yang Chen Eric Eide John Regehr
University of Utah, School of Computing
{jxyang, chenyang, eeide, regehr}@cs.utah.edu

- *Randomized* differential testing has the advantage that **no oracle** for test results is needed.
- Idea: if one has multiple, deterministic implementations of the same specification, all implementations must produce the same result from the same valid input
- 300+ bugs found in compilers (eg gcc)

```
1  int foo (void) {
2    signed char x = 1;
3    unsigned char y = 255;
4    return x > y;
5  }
```

Figure 1. We found a bug in the version of GCC that shipped with Ubuntu Linux 8.04.1 for x86. At all optimization levels it compiles this function to return 1; the correct result is 0. The Ubuntu compiler was heavily patched; the base version of GCC did not have this bug.

http://llvm.org/bugs/show_bug.cgi?id=14972

llvm bug 14972

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

llvm bug 14972

```c
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

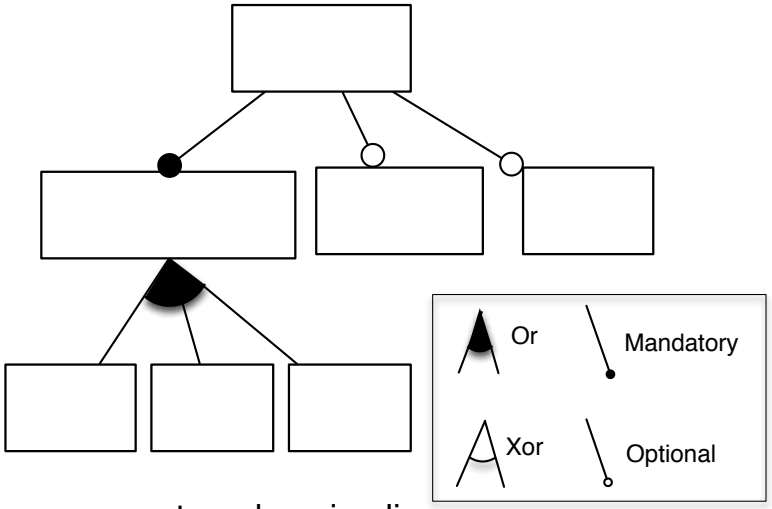
A bug in the LLVM optimizer causes this miscompilation. The developers believe that the Global Value Numbering (GVN) optimization turns the struct initialization into a single 32-bit load. Subsequently, the Scalar Replacement of Aggregates (SRoA) optimization decides that the 32-bit load is undefined behavior, as it reads past the end of the struct, and thus does not emit the correct instructions to initialize the struct. The developer who fixed the issue characterized it as

> *"... very, very concerning when I got to the root cause, and very annoying to fix."*

Test input generation

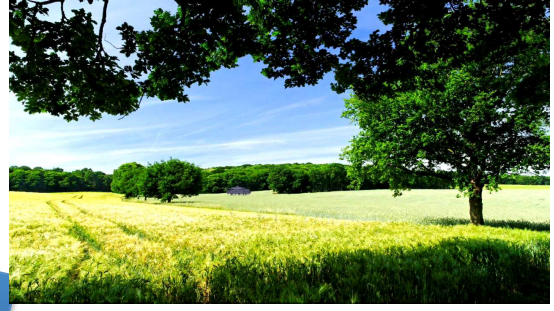
- Performance-related testing

We synthesize video sequence variants with variability techniques

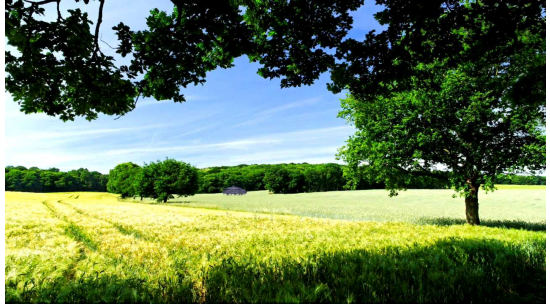


not, and, or, implies

Variability Models (feature models)

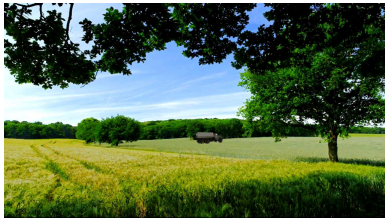


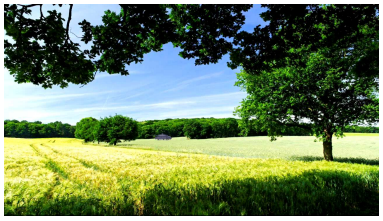
Why?

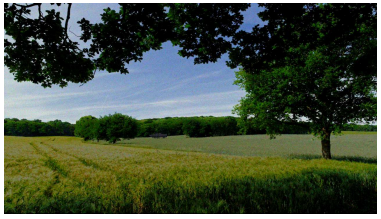


Industrial needs:
consumer and provider of video algorithms have severe difficulties to test their algorithms on different kinds of inputs.

Test algorithms on different kinds of inputs that influence execution time, precision, and/or recall

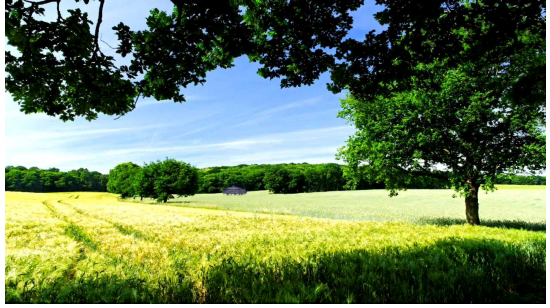






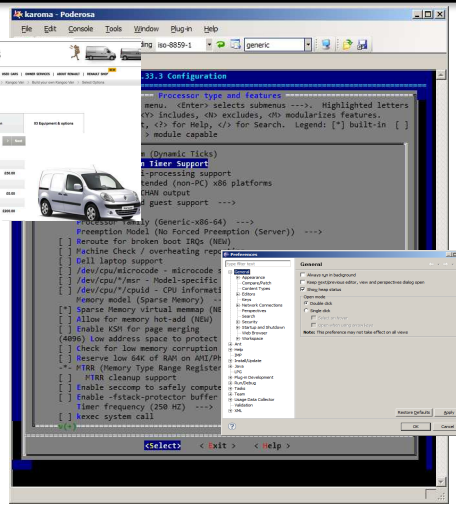
Algorithm 1	Algorithm 2	Algorithm 3
0.63	0.93	0.82
0.81	0.92	0.81
0.43	0.3	0.8
0.39	0.03	0.01

Why?



Problem: collecting videos is a key economic problem.

Solution: hundreds of video sequences with different characteristics.

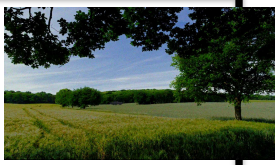


Feature Model

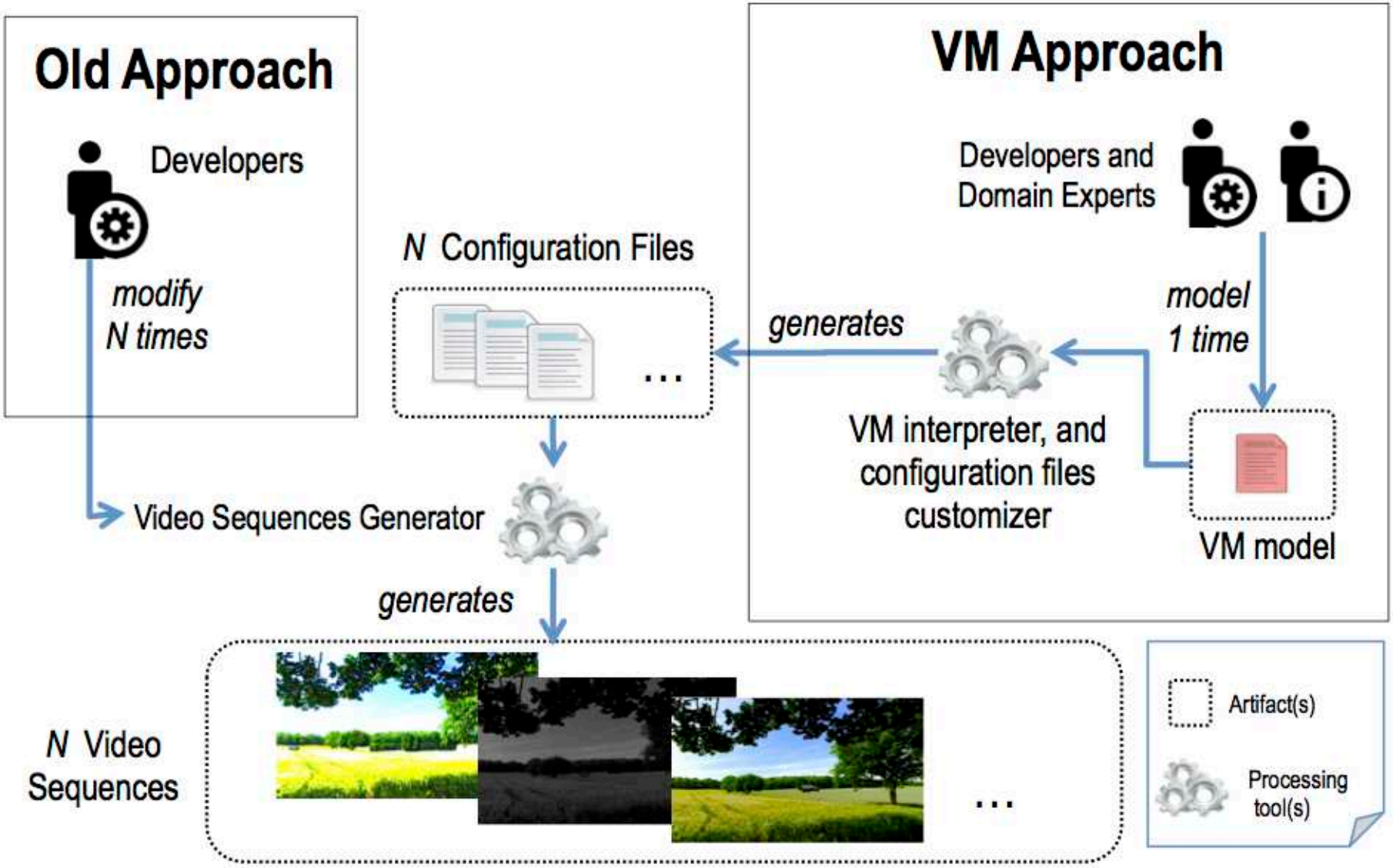
mapping

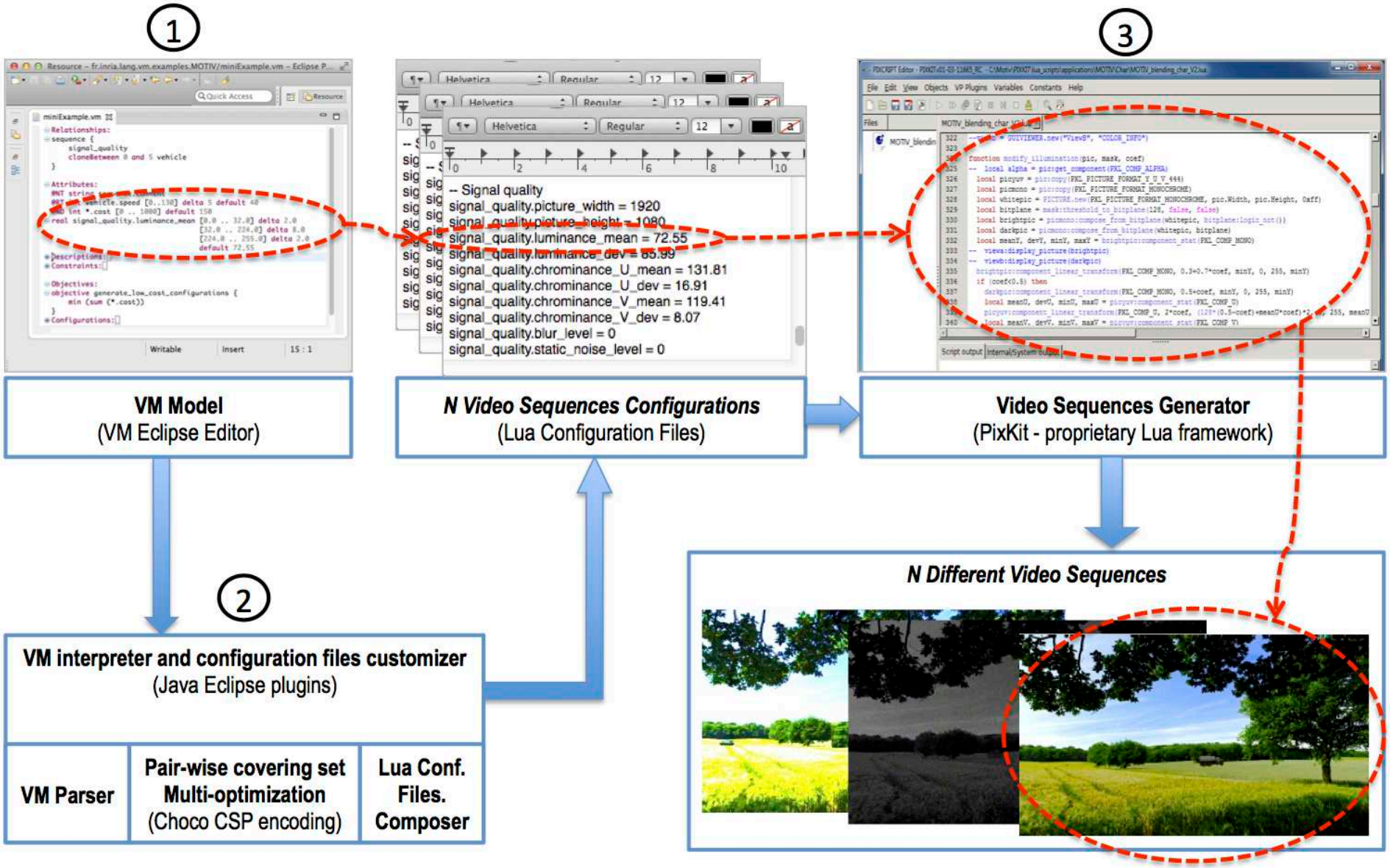
```
print("->Step9")
  if (CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
generate_wind_vector_field2(workwidth, workheight, 256, 1, 1, 35, picnum,
precwindvect5, newwindvect5)
    windvectmul =
windvect5:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect
    hfvx, hfvy =
windvect0:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmat)
    hfvy = MATRIX.multerm(hfvy, invdepthmat)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(-
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmat)
    lfvy = MATRIX.multerm(lfvy, depthmat*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
MATRIX.addterm(lfvy, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
  end

print("->Step10")
```

```
-- Distractors
distractors.butterfly_level = 0.2 -- Floating point number from 0
(low level) to 1 (high level)
distractors.bird_level = 0.3 -- Floating point number from 0 (low
level) to 1 (high level)
distractors.far_moving_vegetation = 0.2 -- Floating point number
from 0 (low level) to 1 (high level)
distractors.close_moving_vegetation = 0.2 -- Floating point number
from 0 (low level) to 1 (high level)
distractors.light_reflection = 0 -- Floating point number from 0
(low level) to 1 (high level)
distractors.blinking_light = 0 -- Floating point number from 0
(low level) to 1 (high level)
```



Software Generator
(derivation engine)


Old Approach
Developers
modify
N times
Video Sequences Generator
N Configuration Files
...
generates
N Video
Sequences
...
VM Approach
Developers and
Domain Experts
model
1 time
generates
VM interpreter, and
configuration files
customizer
VM model
Artifact(s)
Processing
tool(s)


1
3
-- Signal quality
signal_quality.picture_width = 1920
signal_quality.picture_height = 1080
signal_quality.luminance_mean = 72.55
signal_quality.luminance_dev = 55.99
signal_quality.chrominance_U_mean = 131.81
signal_quality.chrominance_U_dev = 16.91
signal_quality.chrominance_V_mean = 119.41
signal_quality.chrominance_V_dev = 8.07
signal_quality.blur_level = 0
signal_quality.static_noise_level = 0
VM Model
(VM Eclipse Editor)
N Video Sequences Configurations
(Lua Configuration Files)
Video Sequences Generator
(PixKit - proprietary Lua framework)
2
VM interpreter and configuration files customizer
(Java Eclipse plugins)
VM Parser
Pair-wise covering set Multi-optimization
(Choco CSP encoding)
Lua Conf. Files. Composer
N Different Video Sequences

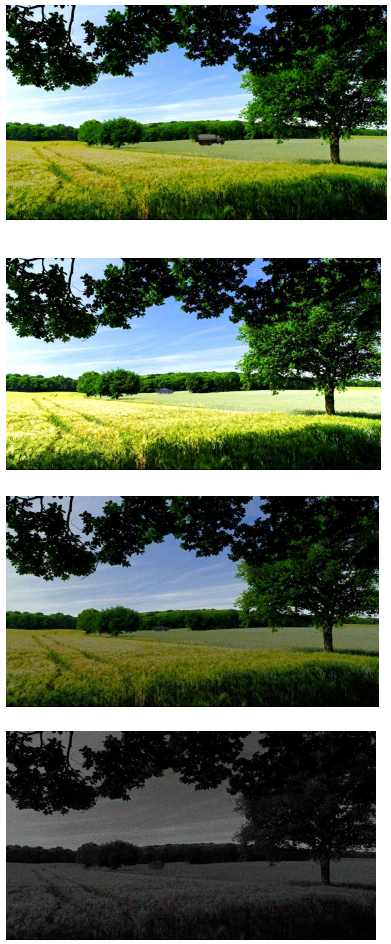
(configuration file)

```
-- Distractors
distractors.butterfly_level = 0.2 -- Floating point number from 0
(low level) to 1 (high level)
distractors.bird_level = 0.3 -- Floating point number from 0 (low
level) to 1 (high level)
distractors.far_moving_vegetation = 0.2 -- Floating point number
from 0 (low level) to 1 (high level)
distractors.close_moving_vegetation = 0.2 -- Floating point number
from 0 (low level) to 1 (high level)
distractors.light_reflection = 0 -- Floating point number from 0
(low level) to 1 (high level)
distractors.blinking_light = 0 -- Floating point number from 0
(low level) to 1 (high level)
```

```
print("->Step9")
  if (CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
generate_wind_vector_field2(workwidth, workheight, 256, 1, 1, 35, picnum,
precwindvect5, newwindvect5)
    windvectmul =
windvect5:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect
    hfvx, hfvy =
windvect0:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmat)
    hfvy = MATRIX.multerm(hfvy, invdepthmat)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(-
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmat)
    lfvy = MATRIX.multerm(lfvy, depthmat*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
MATRIX.addterm(lfvy, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
  end

print("->Step10")
```

(Lua code)

Defects detection
Benchmarking
Incremental design
Performance prediction



Algorithm 1	Algorithm 2	Algorithm 3
0.63	0.93	0.82
0.81	0.92	0.81
0.43	0.3	0.8
0.39	0.03	0.01

Metamorphic testing

- A metamorphic relation is a pair of conditions such that if two inputs fulfill the first, then the result of their evaluations must fulfill the second.
  - Example1 the shortest path between nodes A and B in a graph should have the same length as the shortest path between B and A
  - Example2: the “object of interest” in an image should be the same regardless of the colour palette
- Metamorphic testing is testing software by:
  - generating inputs that fulfill a metamorphic relation
  - confirming that their outputs fulfill that metamorphic relation

Metamorphic testing in real-world

- A compiler output shouldn't change if you inject dead code paths
  - EMI idea can be seen as a “metamorphic” relation
  - Injecting dead code/runtime-constants into shaders made things in pictures disappear or turn to noise.
- “Refining” a search query should yield a subset of its original results
  - In Spotify: two searches for albums with the same query should return the same number of total results regardless of the size of pagination
- Unsupervised clustering (k-means): “do we get the same result if we shuffle the inputs?” and “do additional inputs at cluster boundaries belong to those clusters?”

Example: Testing Computer Vision

- Test Generation Problem (input)
- Oracle Problem (output)

Training set

Machine Learning (classifier)

Accuracy

Metamorphic Testing for CV

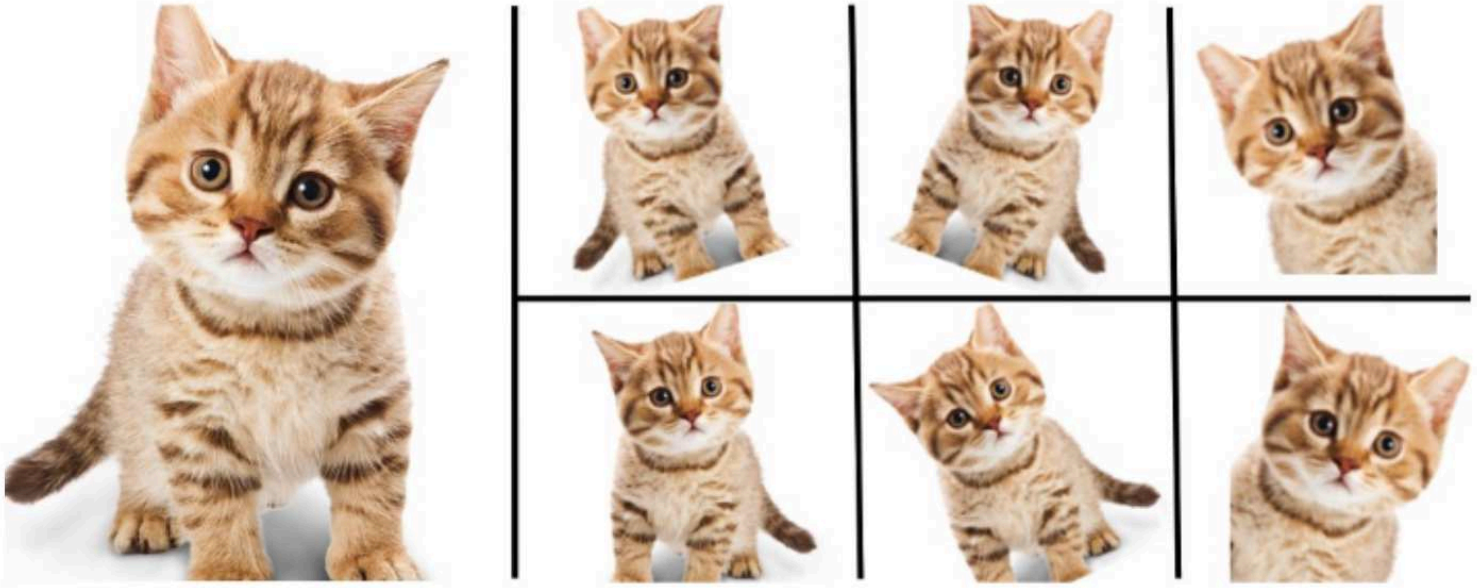
Given an image, you can:

- change its colour palette
- flip it
- rotate it
- zoom...

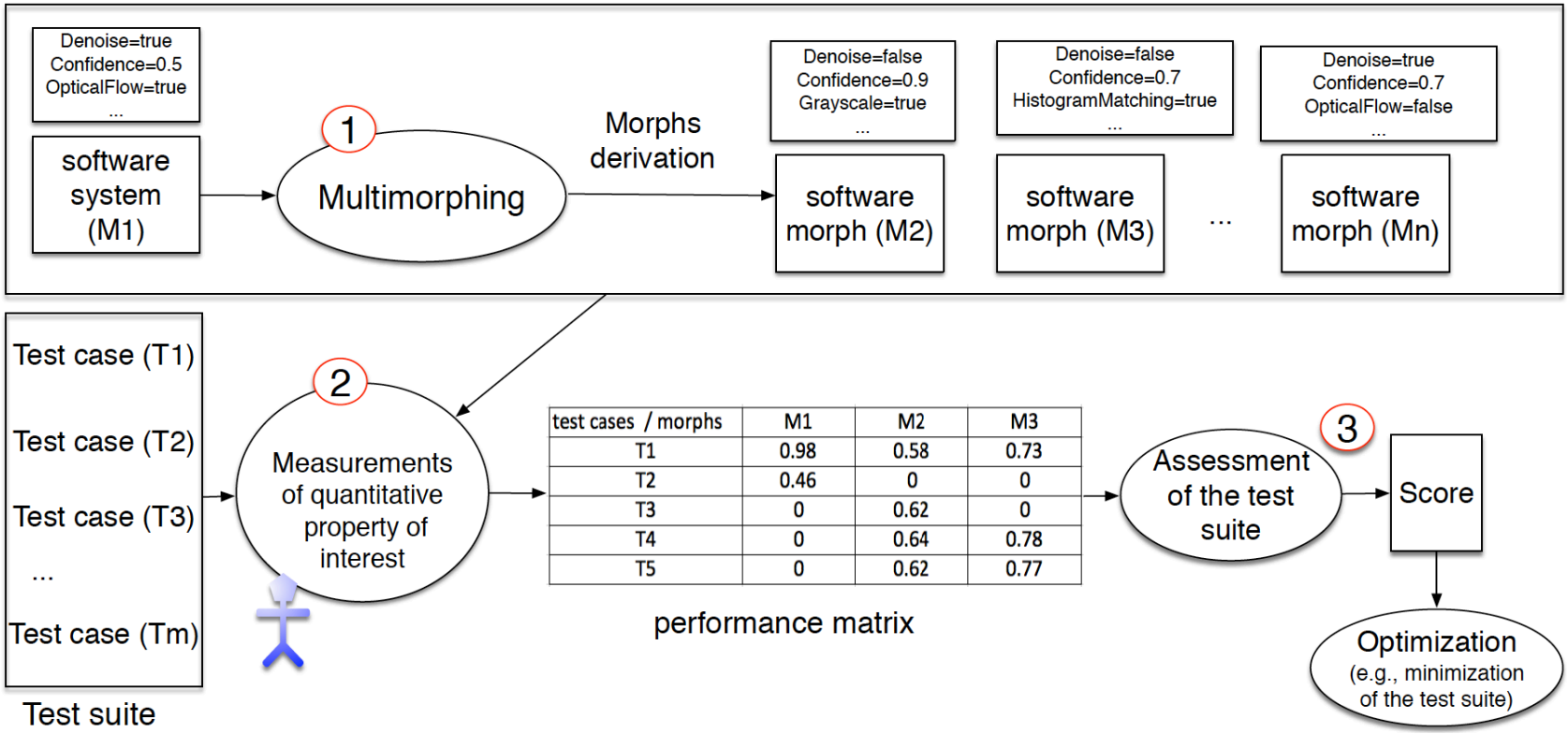
And still know what the output should be!

Free test-cases (input)

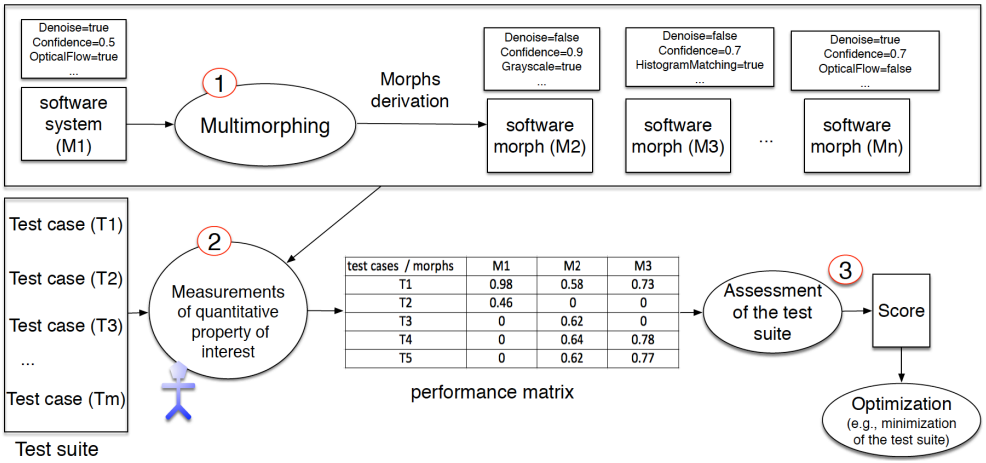
Don’t need to specify the oracle (output)



Multimorphic Testing



Multimorphic Testing



Case	App. Domain	# morphs	# test suites
OpenCV	Tracking in videos	252	49
COCO	Obj. rec. in images	52	12
Haxe	Code Generation	21	84

Test suite reduction

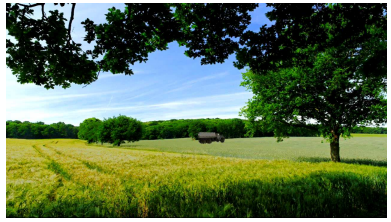
Bugs in compiler (target variant)

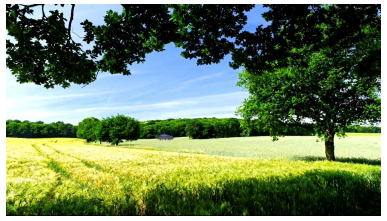
Dispersion score

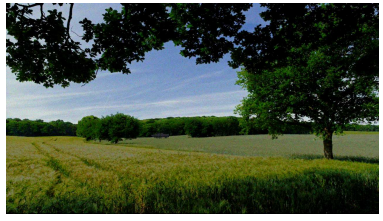
Minimization of test suite

Prioritization of test cases

Better understanding of weakenesses







Algorithm 1	Algorithm 2	Algorithm 3
0.63	0.93	0.82
0.81	0.92	0.81
0.43	0.3	0.8
0.39	0.03	0.01

Conclusion

- Testing Problem
  - Test Generation Problem (input)
  - Oracle Problem (output)
- Assertion-based testing
- Mutation Testing
- Test Generation
- Metamorphic Testing
- Multimorphic Testing
- **Many techniques can be used to test model transformations (eg compilers)**

Back to the project…

Comparing solver variants' performance