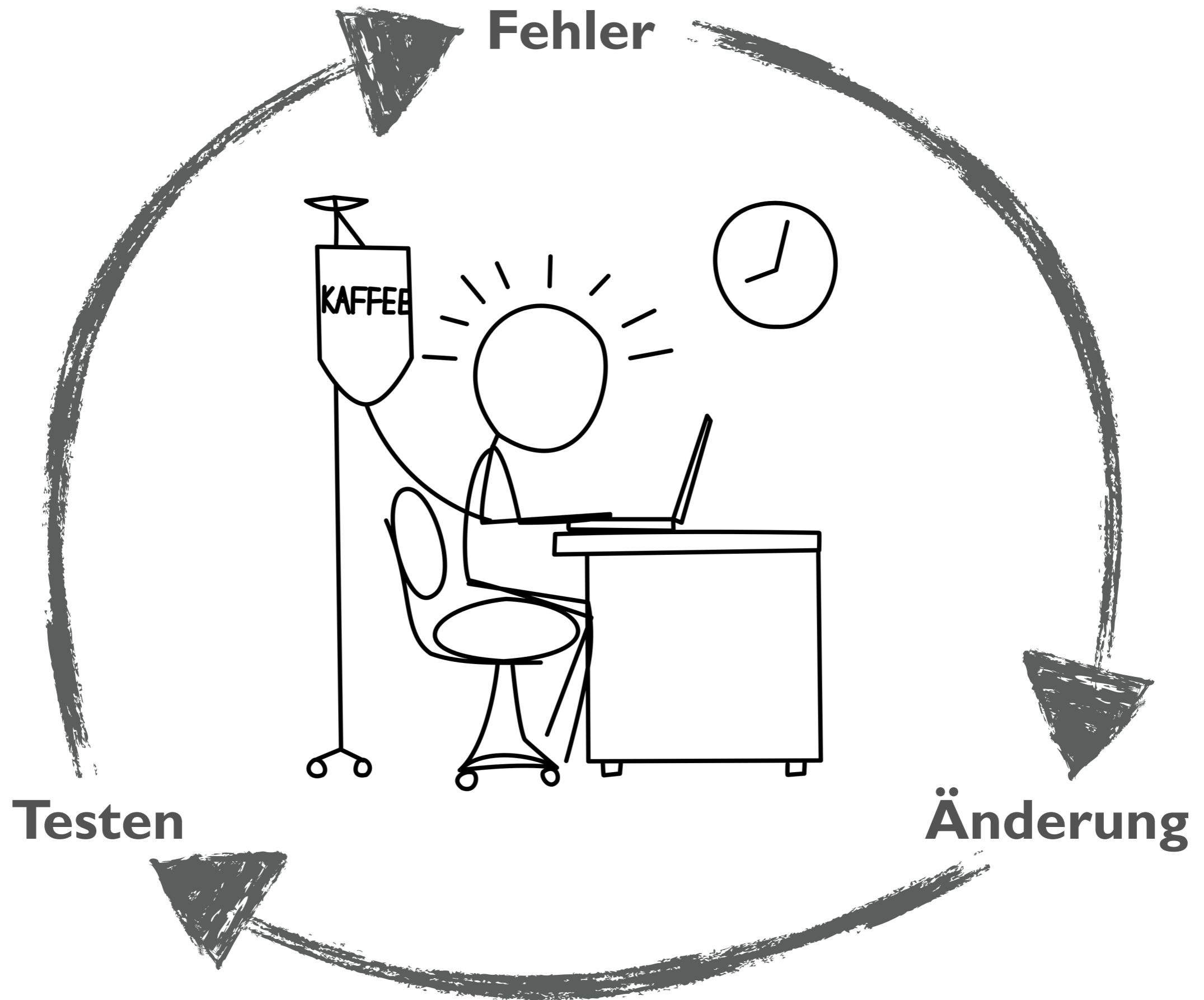


BEI UNSTESTEN
LAUTER AFFEN...

...DAS ENDE DER BANANENSOFTWARE!

Dr. Jeremias Rößler



UMFRAGE

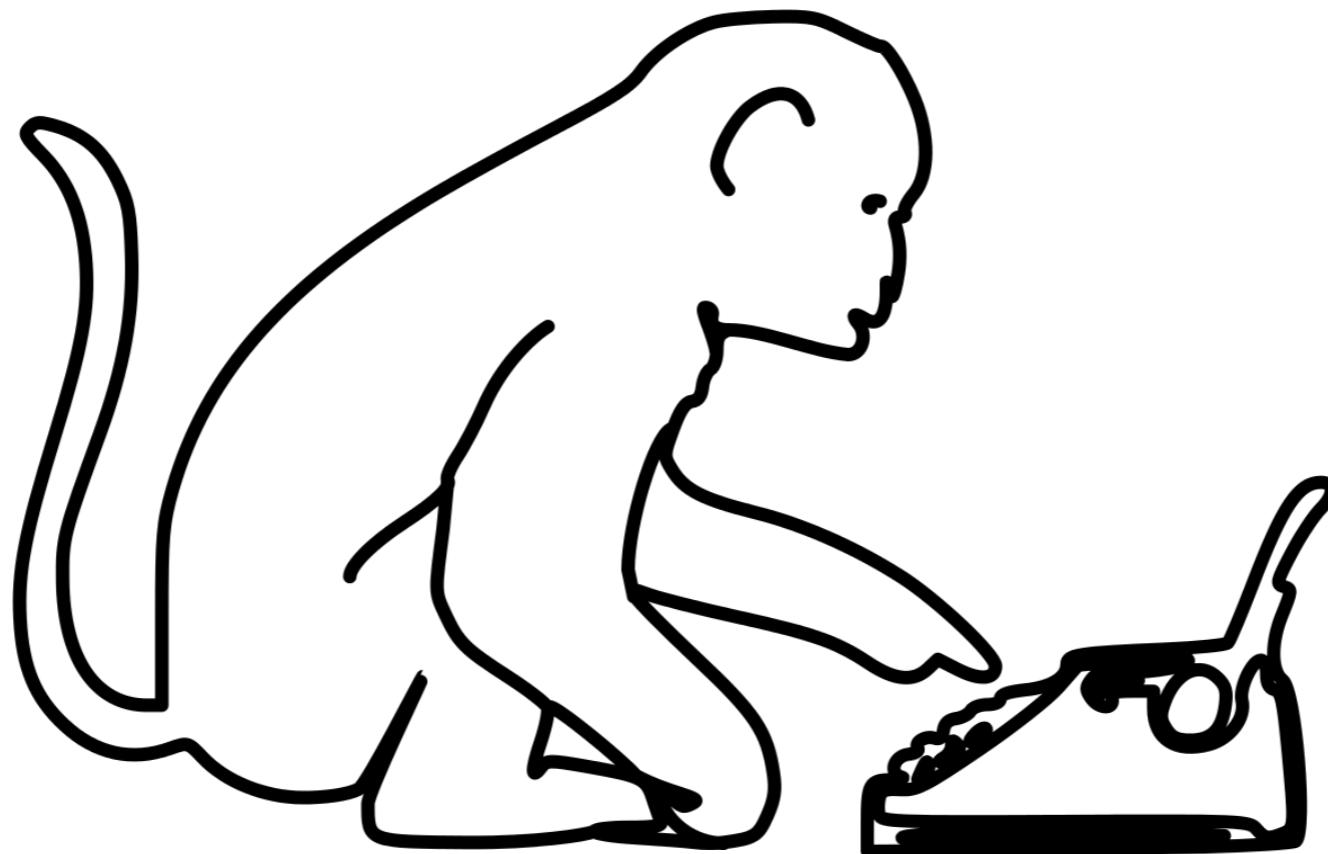
Wer hat manuell getestet?

Wer hat manuell regressionsgetestet?

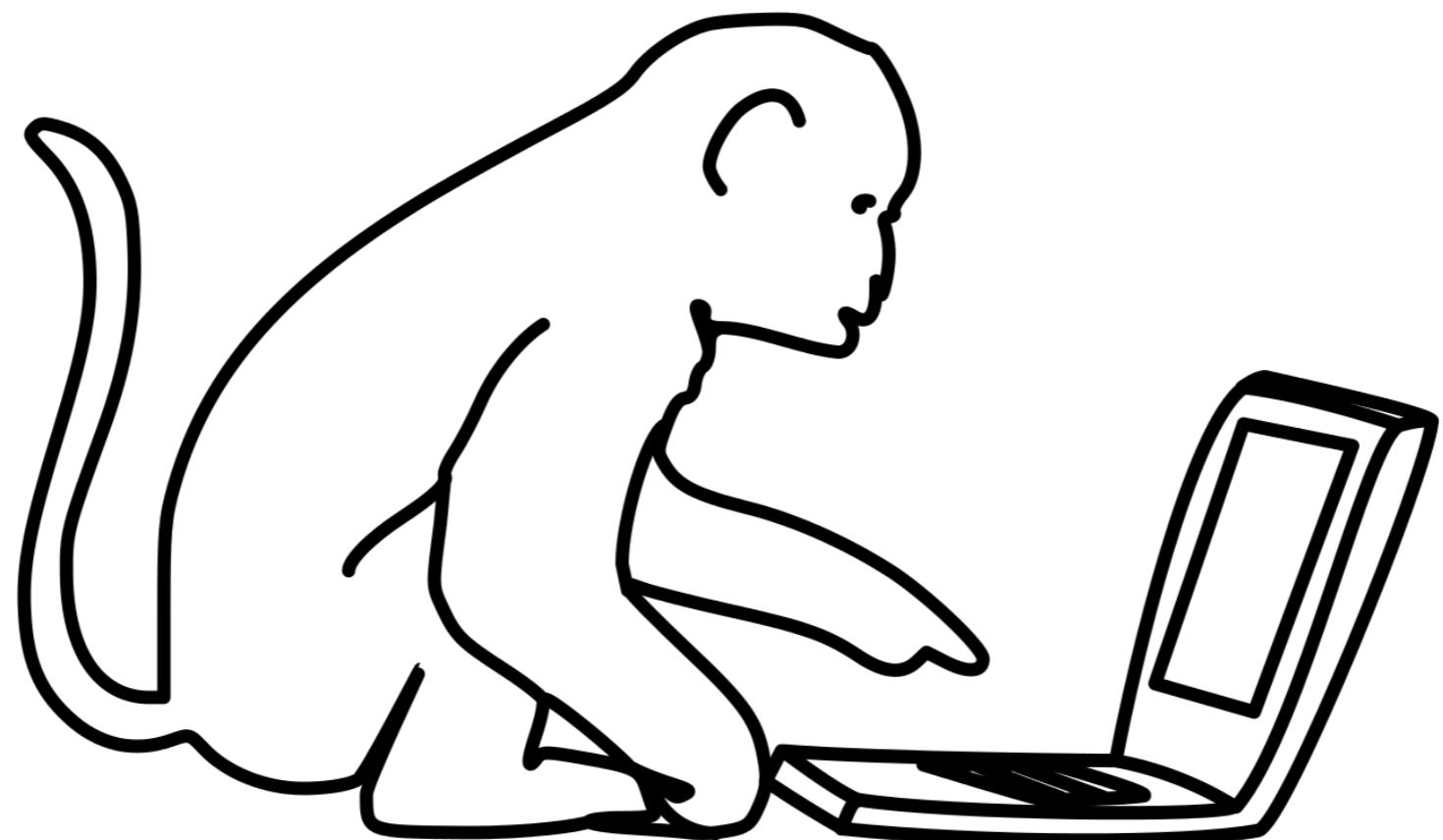
Wem hat das Spaß gemacht?



**Infinite-Monkey-Theorem:
Wenn eine Affe nur lange genug auf einer
Schreibmaschine tippt,
schreibt er irgendwann alle Werke von Shakespeare.**



**Wir ersetzen die Schreibmaschine
mit einem Computer...**



Infinite Monkey

```
1. public static void main(String... args) throws Exception {  
2.     Robot robot = new Robot();  
3.     while (true) {  
4.         robot.mouseMove(random.nextInt(maxX), random.nextInt(maxY));  
5.         robot.mousePress(InputEvent.BUTTON1_DOWN_MASK);  
6.         robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);  
7.         robot.delay(200);  
8.         for (char inputChar : randomString().toCharArray()) {  
9.             robot.keyPress((int) inputChar);  
10.            robot.keyRelease((int) inputChar);  
11.            robot.delay(10);  
12.        }  
13.        robot.keyPress(KeyEvent.VK_ENTER);  
14.        robot.keyRelease(KeyEvent.VK_ENTER);  
15.    }  
16. }
```



Infinite Monkey

DEMO



Vollautomatisches Regressionstesten.

Weniger Risiko, weniger Stress, weniger Kosten!

Haben Sie genug vom Testen?

Direkt aus der Forschung bietet Ihnen
ReTest als bisher einziges Produkt
seiner Art vollautomatisches
Regressionstesten auf Ebene der
Benutzerschnittstelle. Damit stellen Sie
sicher, dass keine unerwünschten
Seitereffekte mehr ins Projekt kommen.

Robustheitstesten von Corner Cases
bekommen Sie gratis dazu.

Mit ReTest behalten Sie die Kontrolle!



Workbook2

Search in Sheet

Home Layout Tables Charts SmartArt Formulas Data Review

Font Alignment Number Format Cells Themes

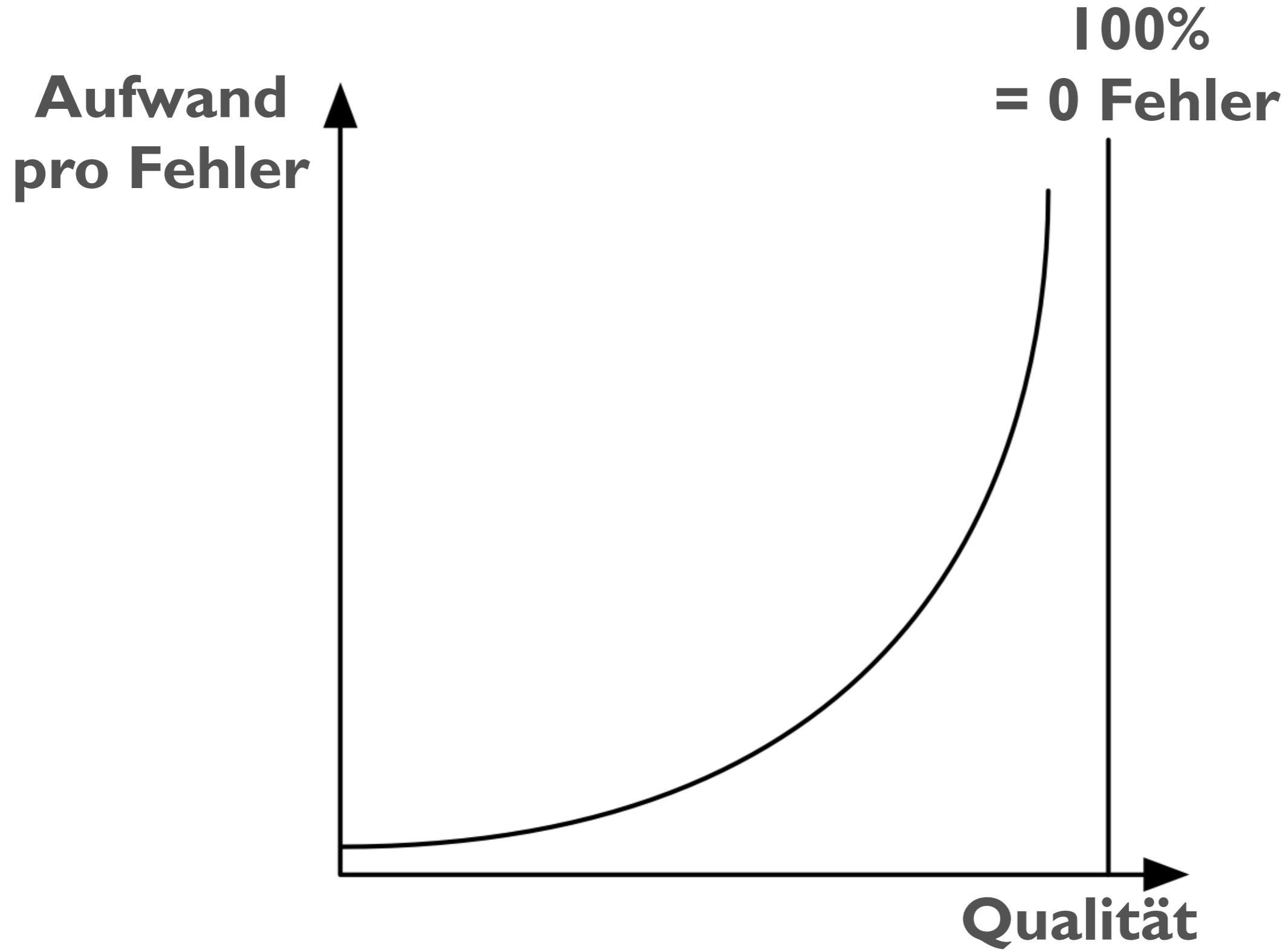
Calibri (Body) 12 Align General Conditional Formatting Styles Actions Themes

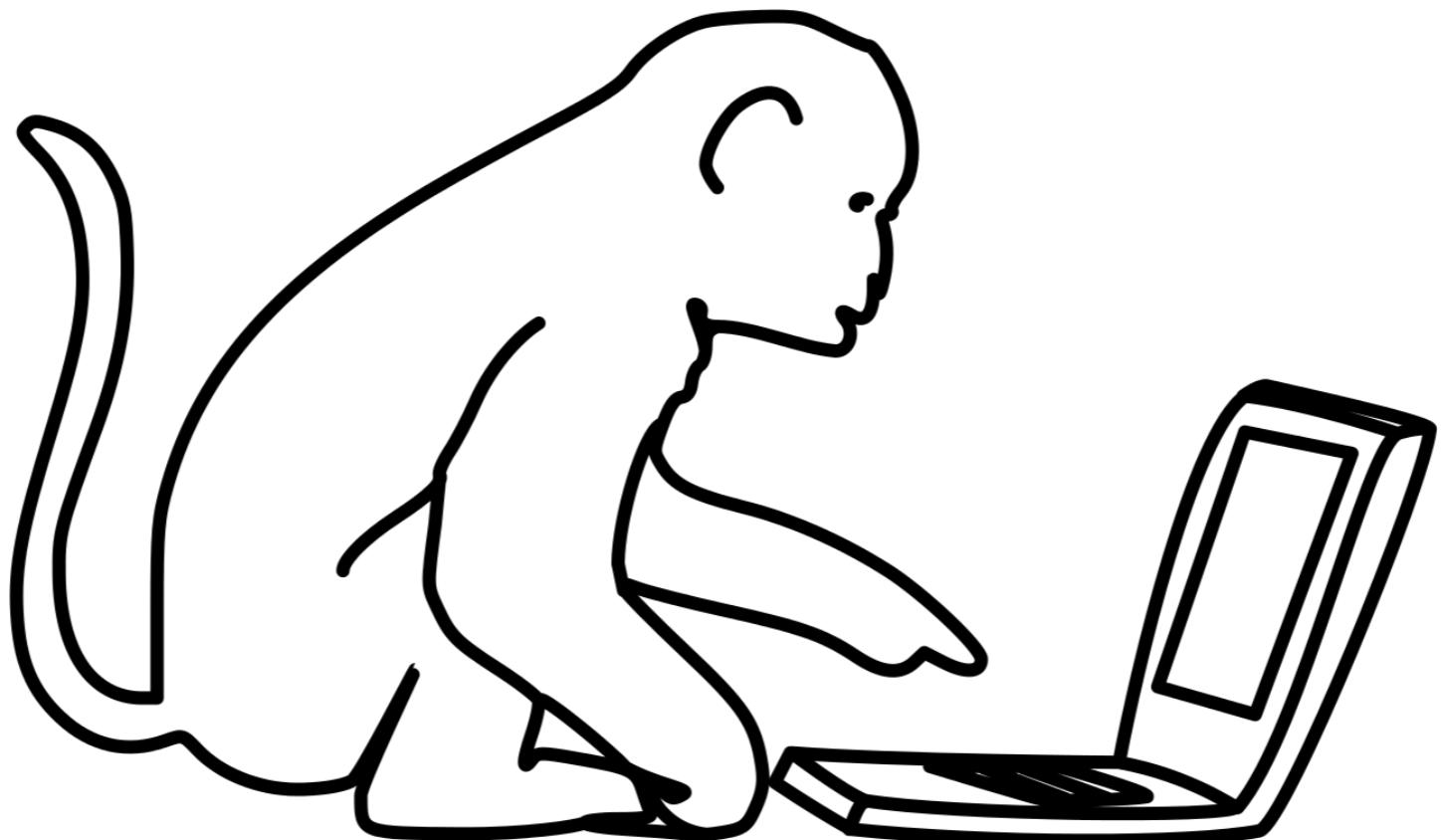
Paste B I U Align % ,

A1 Arbeitspakete

	A	B	C	D	E	F	G	H	I	J
1	Arbeitspakete									
2	Einschalten									
3	Ausschalten									
4	Umschalten									
5	Abschalten									
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										

Sheet1 Normal View Ready







Intelligenter Affe

```
1. public static void main(String... args) throws Exception {  
2.     WebDriver driver = new FirefoxDriver();  
3.     driver.get("http://www.retest.de");  
4.     while (true) {  
5.         List<WebElement> links = driver.findElements(By.tagName("a"));  
6.         links.get(random.nextInt(links.size())).click();  
7.         Thread.sleep(500);  
8.         List<WebElement> fields =  
9.             driver.findElements(By.xpath("//input[@type='text']"));  
10.        WebElement field = fields.get(random.nextInt(fields.size()));  
11.        field.sendKeys(randomString());  
12.        Thread.sleep(500);  
13.    }  
14. }
```

Intelligenter Affe

DEMO

www.retest.de

Start ReTest News Kontakt

Vollautomatisches Regressionstesten.

Weniger Risiko, weniger Stress, weniger Kosten!

Haben Sie genug vom Testen?

Direkt aus der Forschung bietet Ihnen ReTest als bisher einziges Produkt seiner Art vollautomatisches Regressionstesten auf Ebene der Benutzerschnittstelle. Damit stellen Sie sicher, dass keine unerwünschten Seiteneffekte mehr ins Projekt kommen.

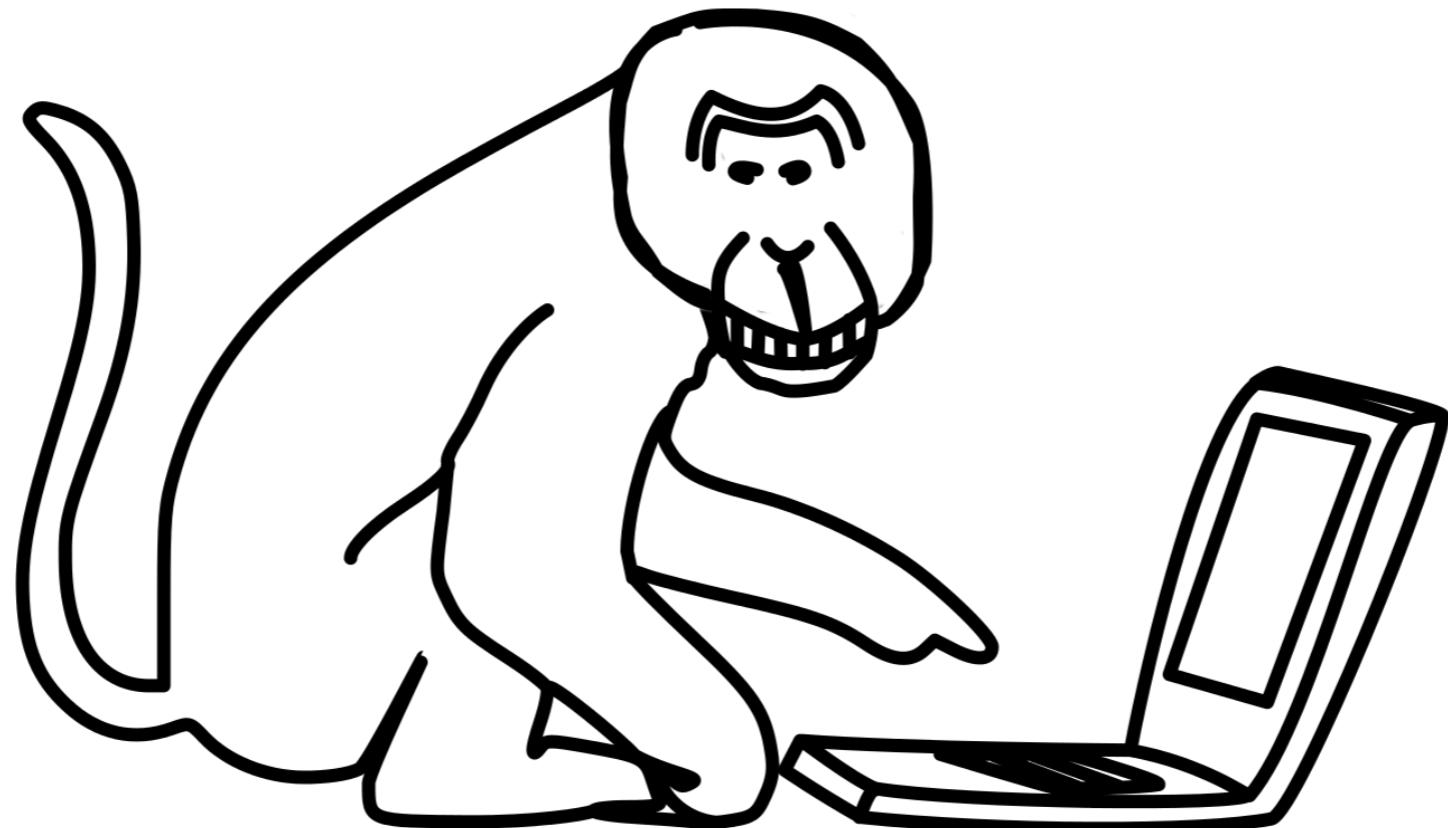
Robustheitstesten von Corner Cases bekommen Sie gratis dazu.

Mit ReTest behalten Sie die Kontrolle!



0:00 / 1:09

Intelligenter Affe



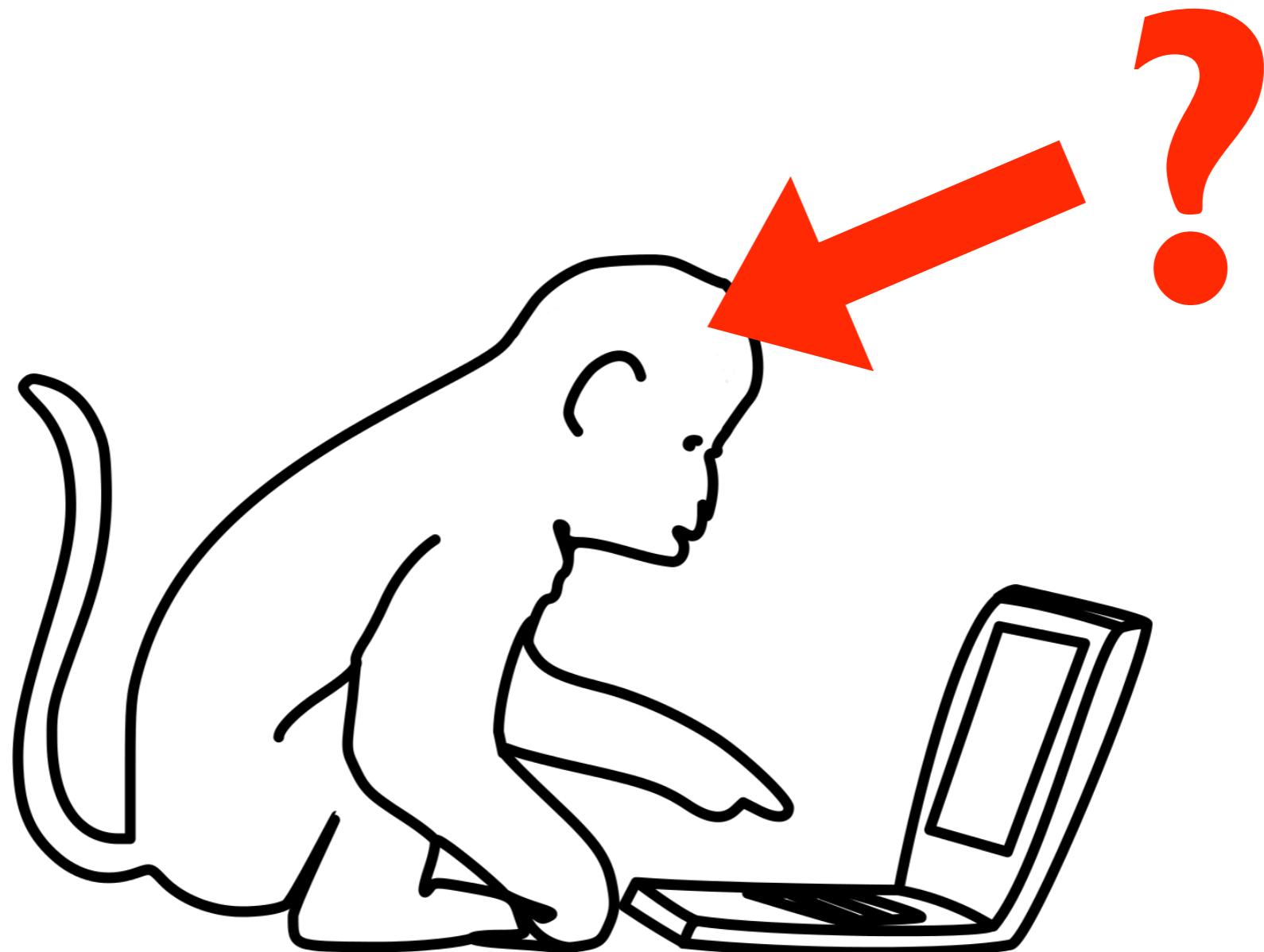
Open Source Produkte

NetFlix Chaos Monkey

gremlins.js

UI/Application Exerciser Monkey

Intelligenter Affe



HumiCalc with Uncertainty

File Options Help

Configuration

Temperature Scale: ITS-90 Carrier Gas: Dry Air Mode: Normal
Equilibrium Over: Water Apply Enhancement Factors: Known: Dew Point

Known Values (Standard u)

Dew Point	10.0	± 0.0577
Temperature	25.0	± 0.025
Pressure	14.7	± 0.025

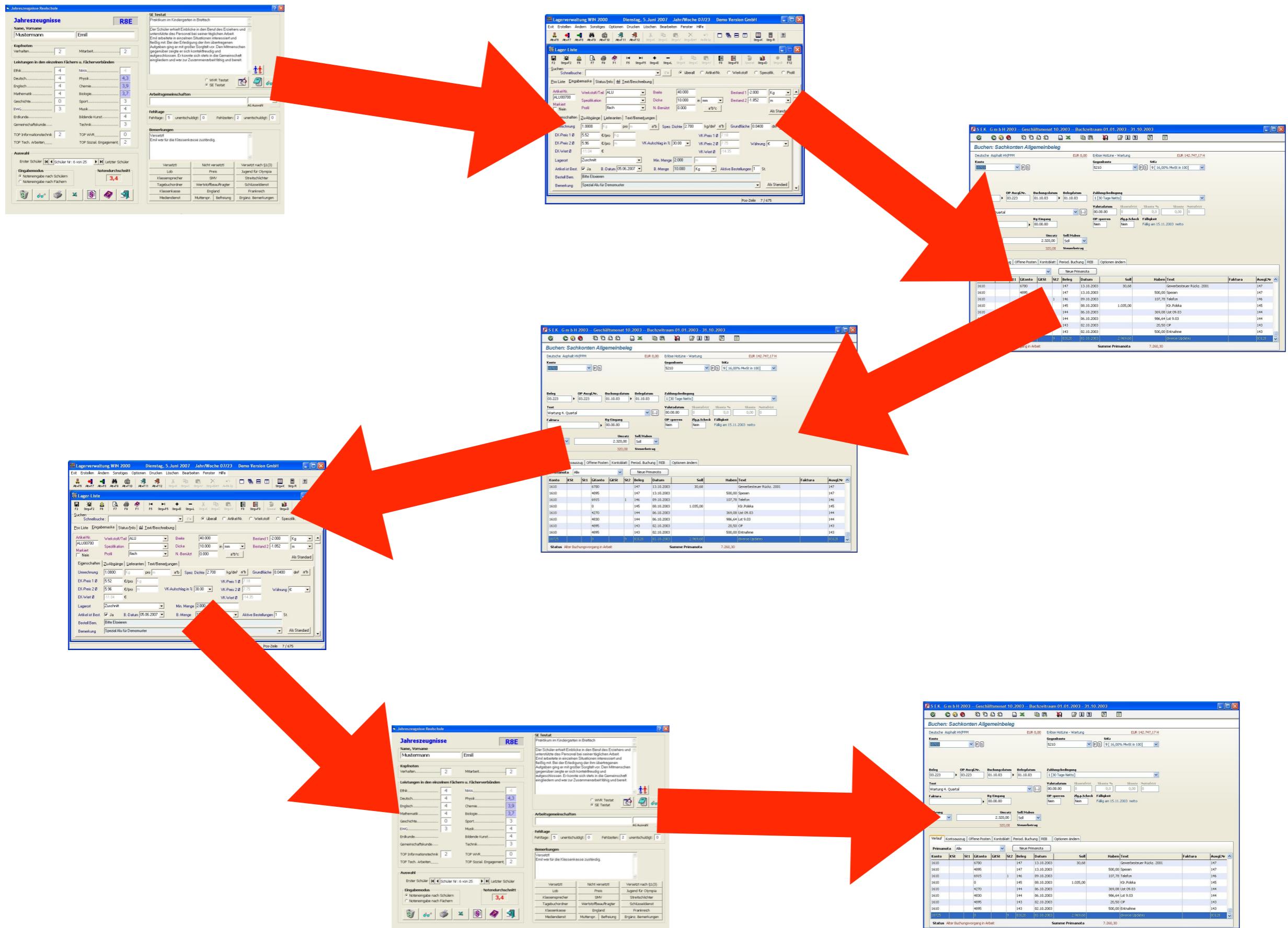
Calculated Values (Expanded U with 95.45% Confidence)

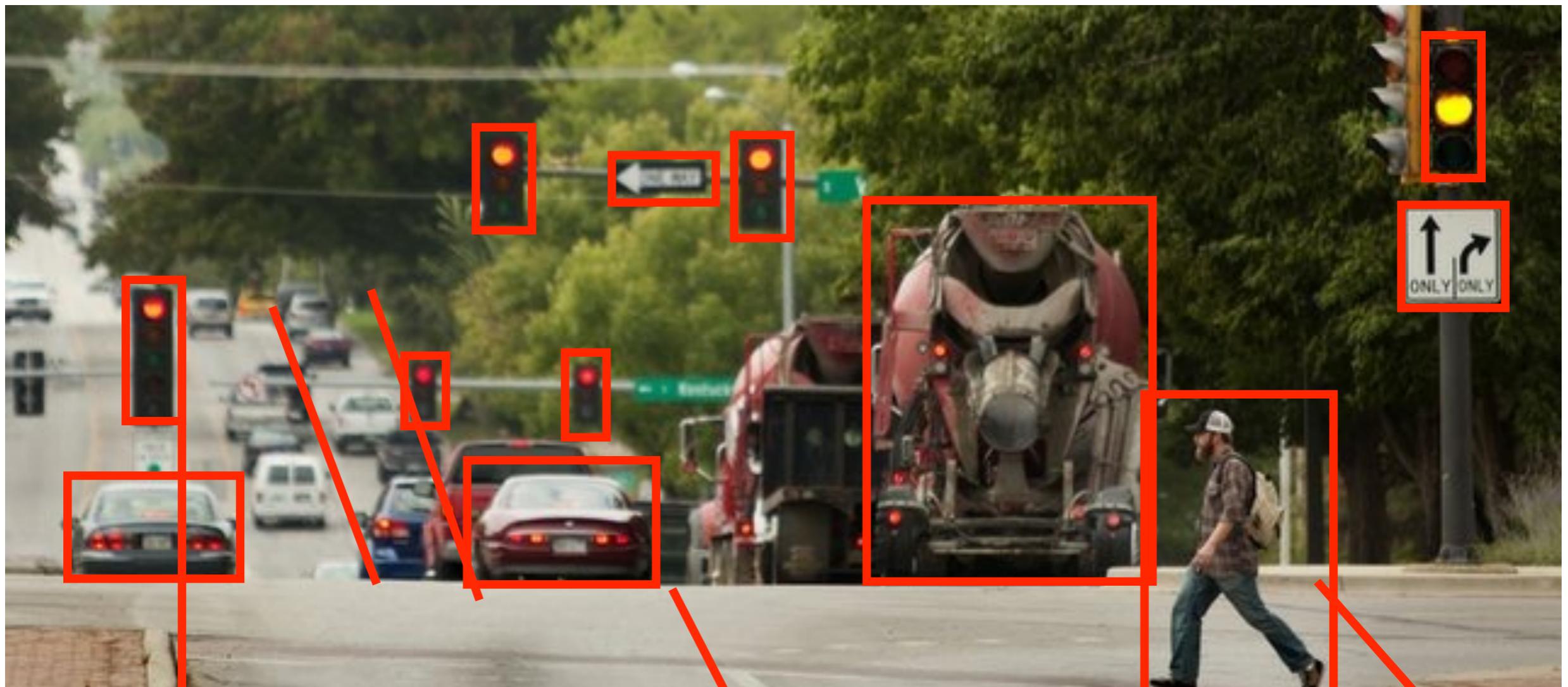
%RH	38.7340793	± 0.3212	Specific Humidity	0.0076028	$\pm 6E-005$
Frost Point			Absolute Humidity	8.962178468	± 0.0694
Dew Point	10.0	± 0.1155	Dry Air Density	1169.837556	± 4.0342
PPMv	12314.00364	± 105.31	Moist Air Density	1178.799735	± 4.033
PPMw	7661.045263	± 65.521	Saturation Temperature		
Grains/lb	53.62731684	± 0.4586	Saturation Pressure		
Enthalpy	44.63021277	± 0.1744	Wet Bulb Temperature	16.10904106	± 0.0559
SVP@Tt	3169.90395	± 9.4493	Mixing Ratio by Volume	0.012314004	± 0.0001
SVP@Td	1228.13339	± 9.5028	Mixing Ratio by Weight	0.007661045	$\pm 7E-005$
SVP@Ts			Percent by Volume	1.216421347	± 0.0103
F@Tt,Pt	1.004109348	$\pm 1E-005$	Percent by Weight	0.760279987	± 0.0065
F@Td,Pt	1.003863842	$\pm 1E-005$	Vapor Mole Fraction	0.012164213	± 0.0001
F@Ts,Ps			Dry Air Mole Fraction	0.987835787	± 0.0001

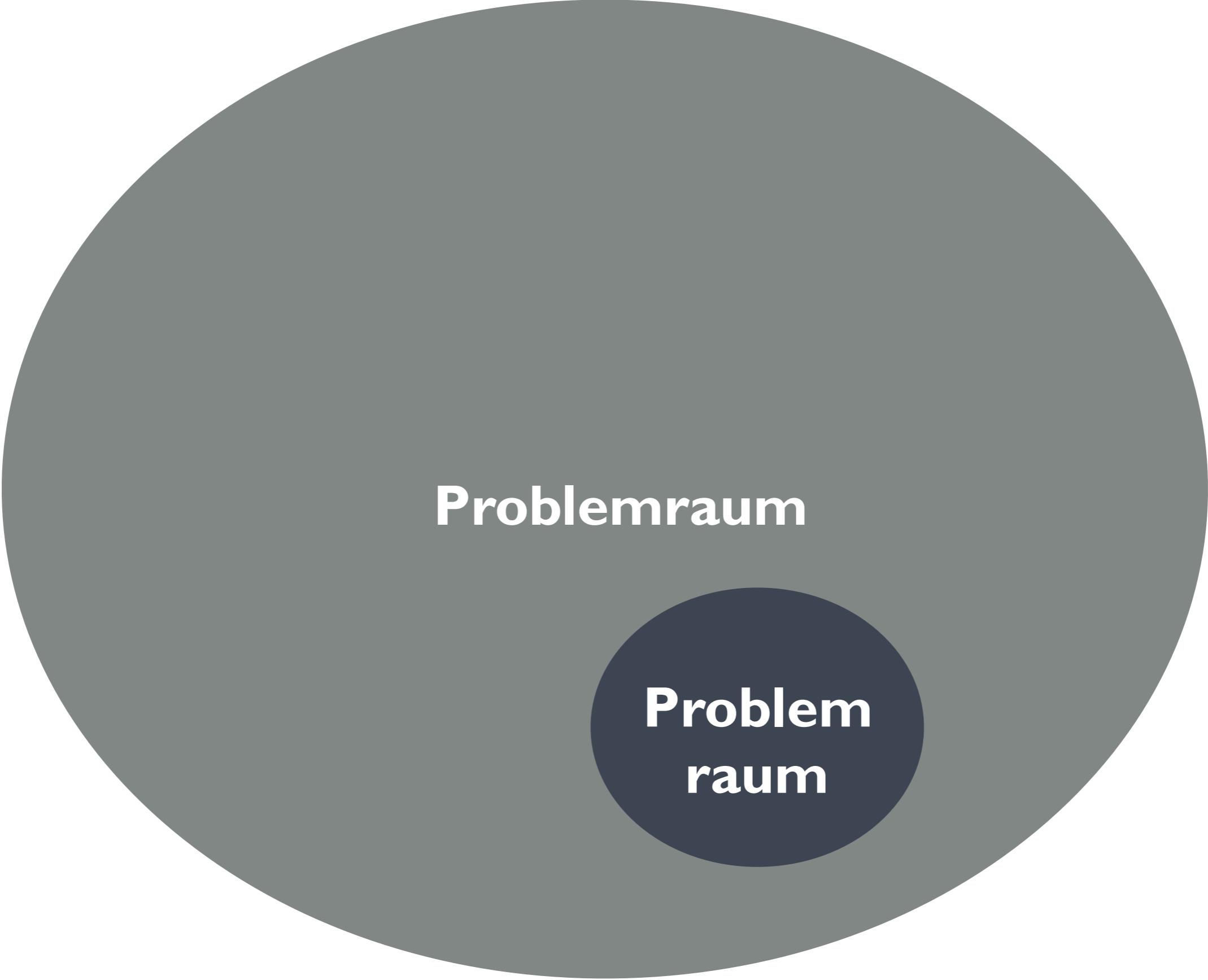
Calculate

Units

Temperature	°C
Pressure	psia
Vapor Pressure	Pa
Density and Abs Humidity	g/m ³
Enthalpy	J/g



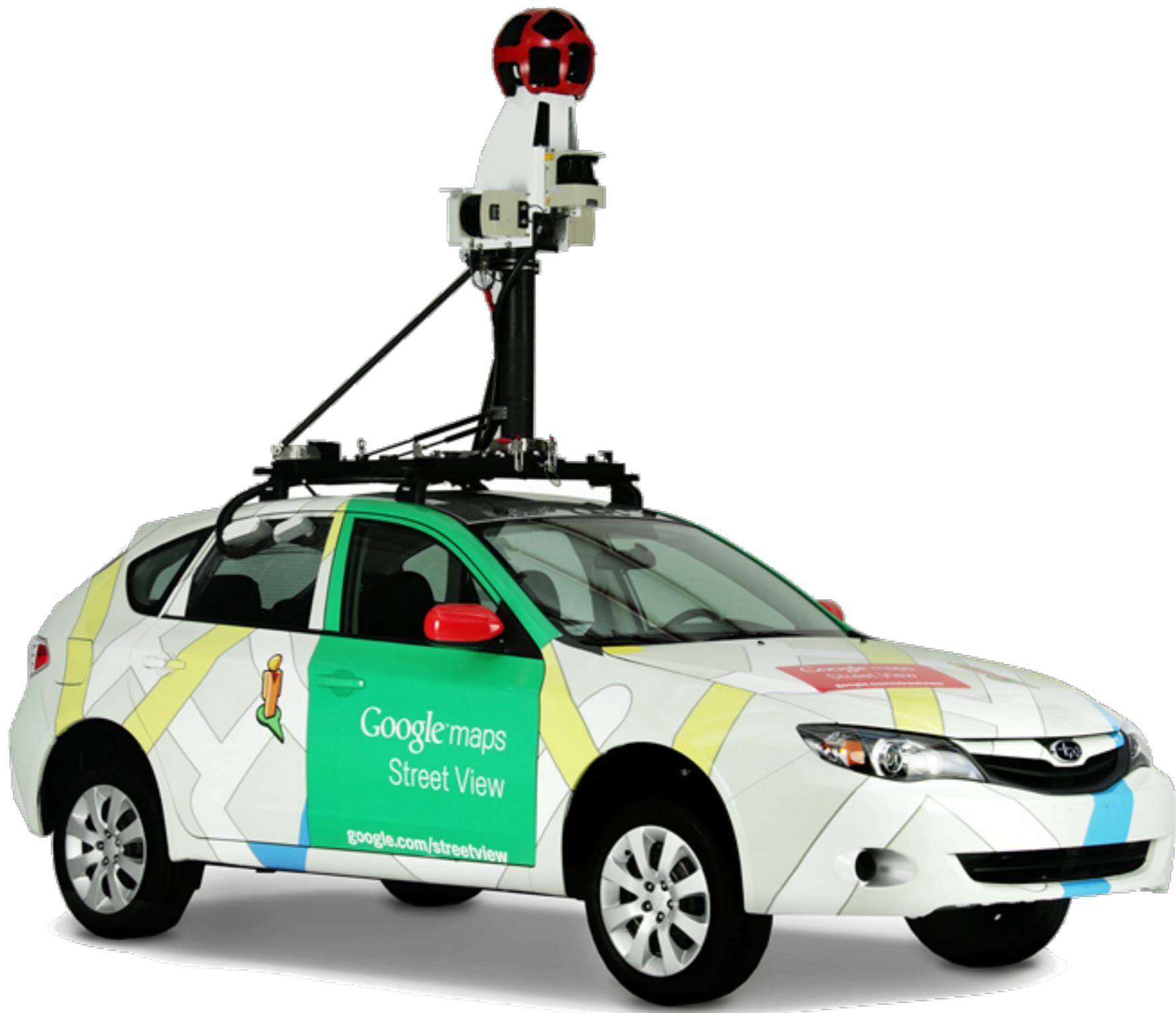




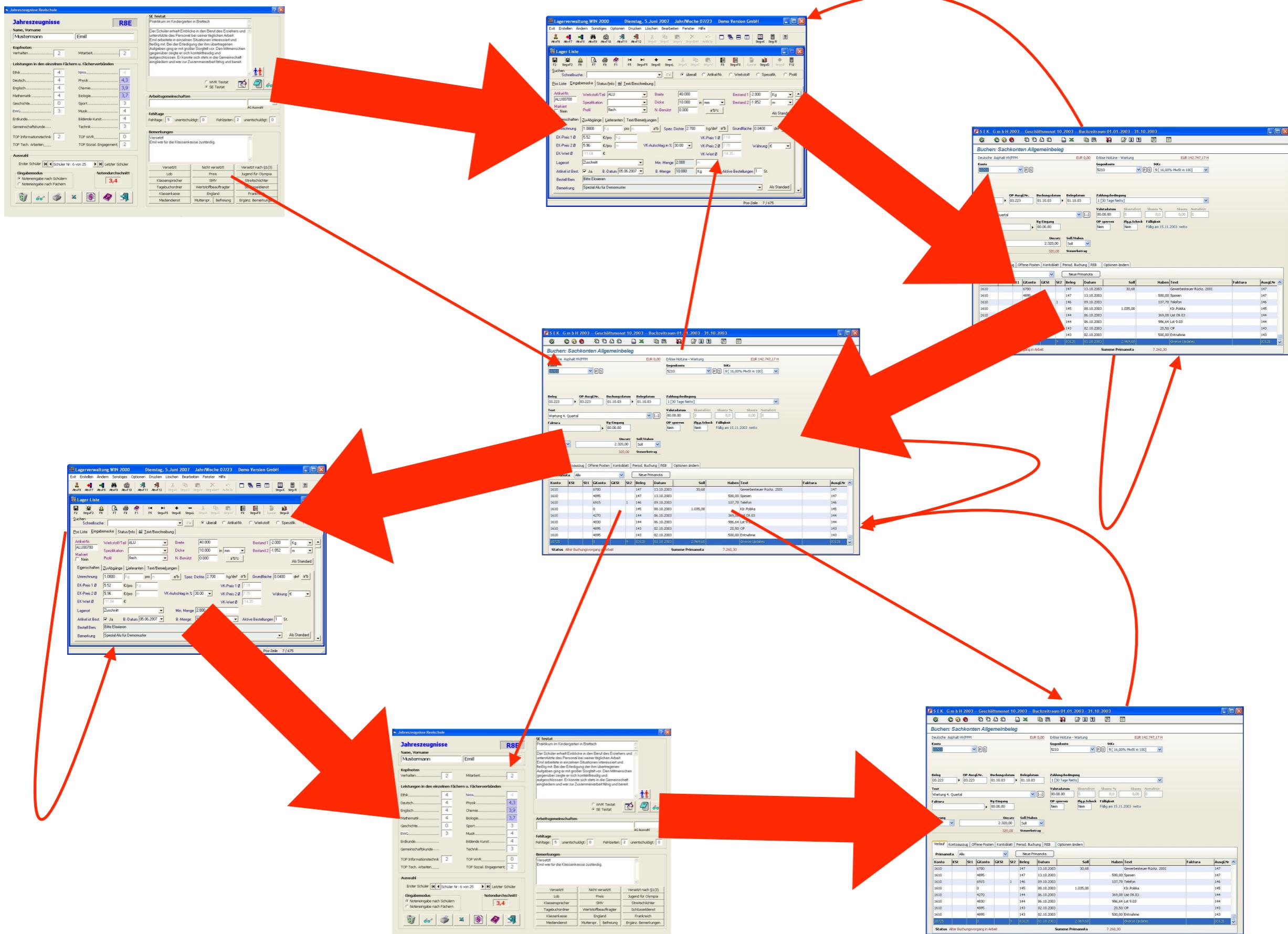
The diagram consists of two concentric circles. The outer circle is light gray and contains the text "Problemraum" in white. The inner circle is dark blue and contains the text "Problem
raum" in white. This visual representation emphasizes the relationship between the broader context ("Problemraum") and the specific focus or subject matter ("Problem raum").

Problemraum

**Problem
raum**



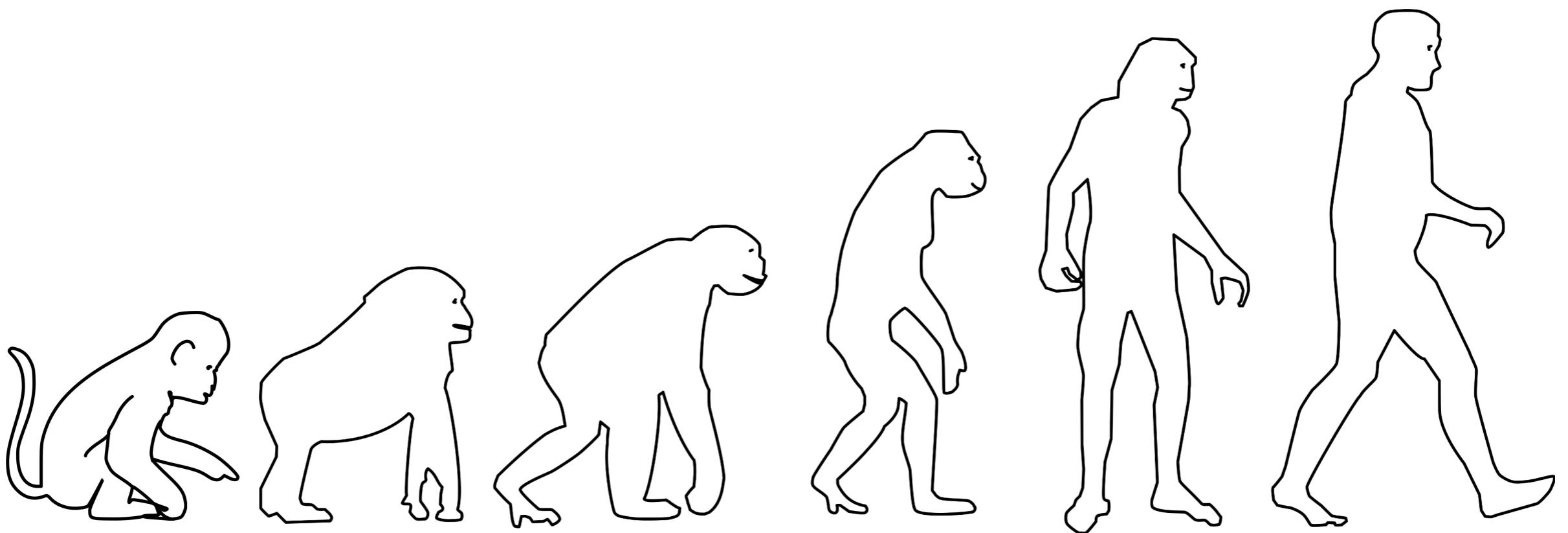






Geburtsdatum





Türme von Hanoi



© André Karwath aka Aka

```
1. class TuermeVonHanoi():
2.     def __init__(self):
3.         self.A = [6, 5, 4, 3, 2, 1]
4.         self.B = []
5.         self.C = []
6.
7.     def AtoB(self): self.B.append(self.A.pop())
8.
9.     def AtoC(self): self.C.append(self.A.pop())
10.
11.    def BtoA(self): self.A.append(self.B.pop())
12.
13.    def BtoC(self): self.C.append(self.B.pop())
14.
15.    def CtoA(self): self.A.append(self.C.pop())
16.
17.    def CtoB(self): self.B.append(self.C.pop())
18.
19.    def valid(self):
20.        return all(self.A[i + 1] < self.A[i] for i in range(len(self.A)-1)) and \
21.               all(self.B[m + 1] < self.B[m] for m in range(len(self.B)-1)) and \
22.               all(self.C[n + 1] < self.C[n] for n in range(len(self.C)-1))
```

Türme von Hanoi

Rekursiv

```
1. def bewege(schritte, i, a_name, b_name, c_name):  
2.     if (i > 0):  
3.         bewege(schritte, i-1, a_name, c_name, b_name)  
4.         schritte.append('tuerme.' + a_name + 'to' + c_name + '())')  
5.         bewege(schritte, i-1, b_name, a_name, c_name)  
6.     return schritte  
7.  
8. def algorithmus():  
9.     return bewege([], 6, 'A', 'B', 'C')
```

Türme von Hanoi

Manuell

```
1. schritte = [\n2. 'tuerme.AtoB()', \ #[654321][][] -> [65432][1][]\n3. 'tuerme.AtoC()', \ #[65432][1][] -> [6543][1][2]\n4. 'tuerme.BtoC()', \ #[6543][1][2] -> [6543][][21]\n5. 'tuerme.AtoB()', \ #[6543][][21] -> [654][3][21]\n6. 'tuerme.CtoA()', \ #[654][3][21] -> [6541][3][2]\n7. 'tuerme.CtoB()', \ #[6541][3][2] -> [6541][32][]\n8. 'tuerme.AtoB()', \ #[6541][32][] -> [654][321][]\n9. 'tuerme.AtoC()', \ #[654][321][] -> [65][321][4]\n10. 'tuerme.BtoC()', \ #[65][321][4] -> [65][32][41]\n11. 'tuerme.BtoA()', \ #[65][32][41] -> [652][3][41]\n12. 'tuerme.CtoA()', \ #[652][3][41] -> [6521][3][4]\n13. 'tuerme.BtoC()', \ #[6521][3][4] -> [6521][][43]\n14. 'tuerme.AtoB()', \ #[6521][][43] -> [652][1][43]\n15. 'tuerme.AtoC()', \ #[652][1][43] -> [65][1][432]\n16. 'tuerme.BtoC()', \ #[65][1][432] -> [65][][4321]\n17. 'tuerme.AtoB()', \ #[65][][4321] -> [6][5][4321]\n18. 'tuerme.CtoA()', \ #[6][5][4321] -> [61][5][432]
```

Türme von Hanoi

Möglichkeiten

$>2^{63}$

Türme von Hanoi

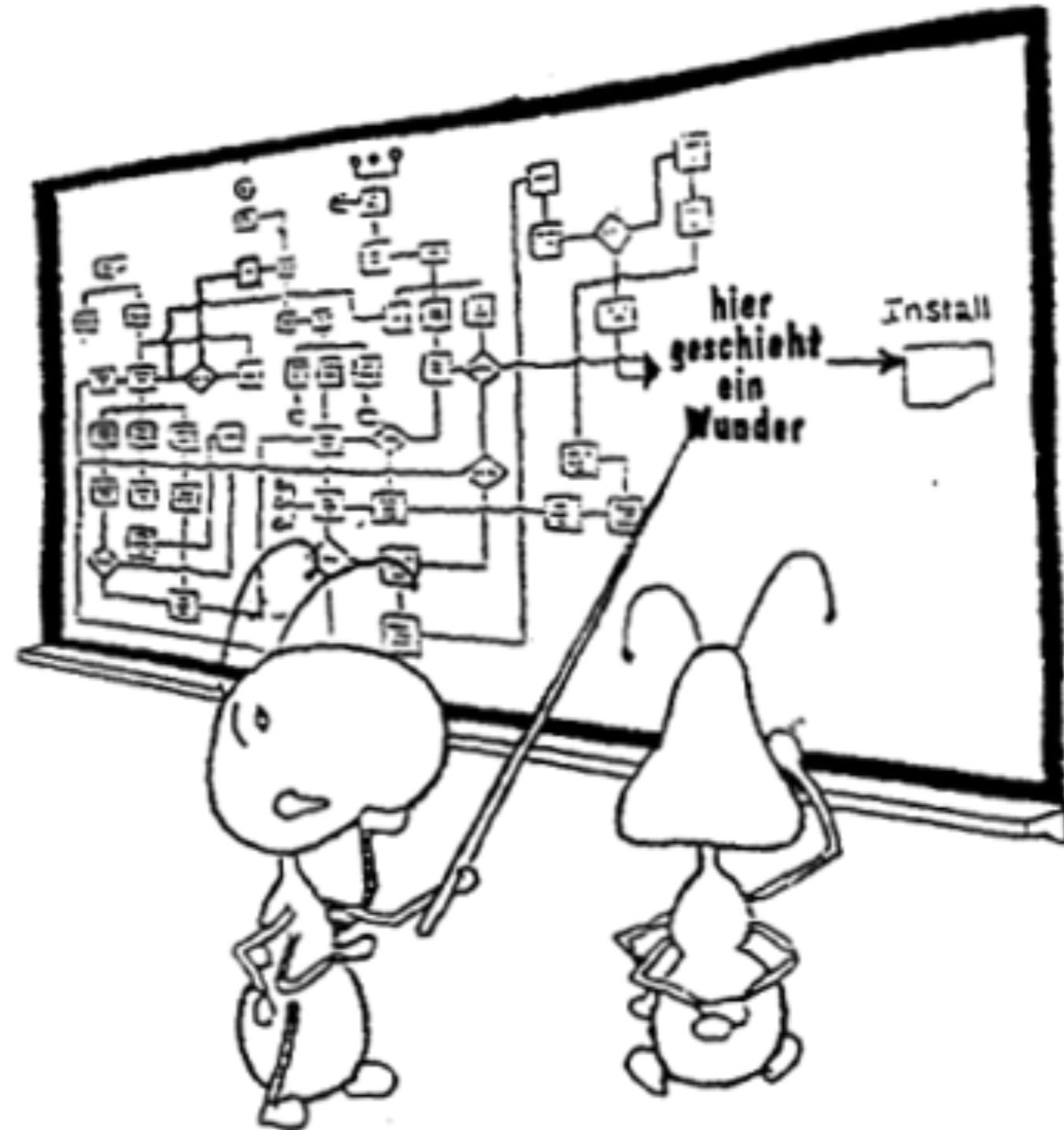
Möglichkeiten

9.223.372.036.854.775.808

Türme von Hanoi Möglichkeiten

292.471.208 Jahre

Türme von Hanoi Automatisch?



Türme von Hanoi

Zielfunktion

```
1. def min_zielfunktion(tuerme): return (21 - sum(tuerme.C))
```

```
while (min_zielfunktion(loesung) > 0)
    neue_loesung = addNextStep(loesung)
    if (better(neue_loesung, loesung))
        loesung = neue_loesung
```

Türme von Hanoi

Stochastischer Bergsteigeralgorithmus

DEMO

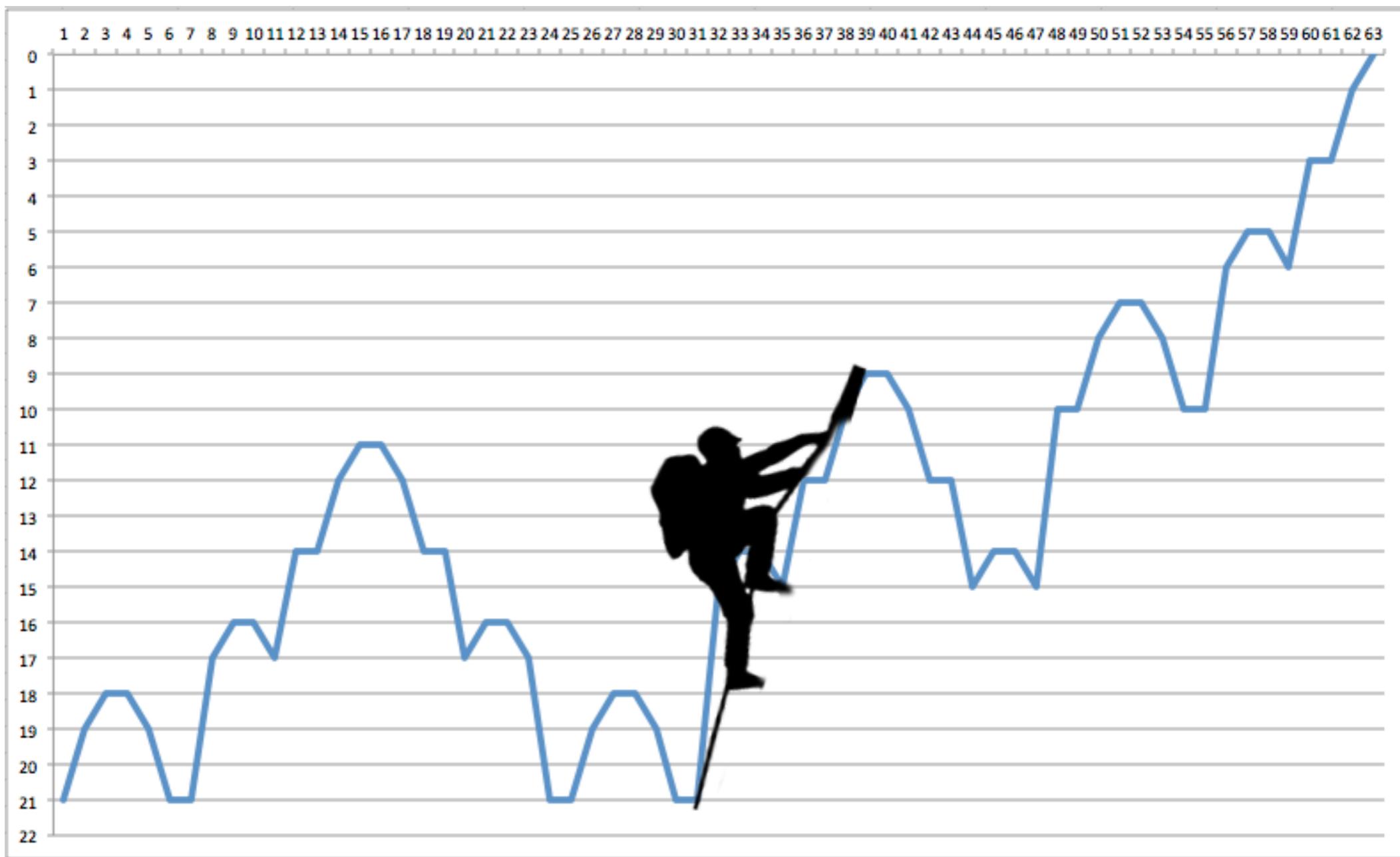
Türme von Hanoi

Stochastischer Bergsteigeralgorithmus

```
['tuerme.AtoC()'] -> [65432][][1], Anzahl Schritte: 1, Zielwert: 20
```

```
nach_1000_versuchen_neustarten:  
while (min_zielfunktion(loesung) > 0)  
    neue_loesung = addNextStep(loesung)  
    if (better(neue_loesung, loesung))  
        loesung = neue_loesung
```

Türme von Hanoi Suchraumlandschaft



```
while (min_zielfunktion(loesung) > 0)
    neue_loesung = addNextStep(loesung)
    if (better(neue_loesung, loesung)
        or 50/50_chance)
        loesung = neue_loesung
```

Türme von Hanoi

Zufallsbewegung

DEMO

Türme von Hanoi

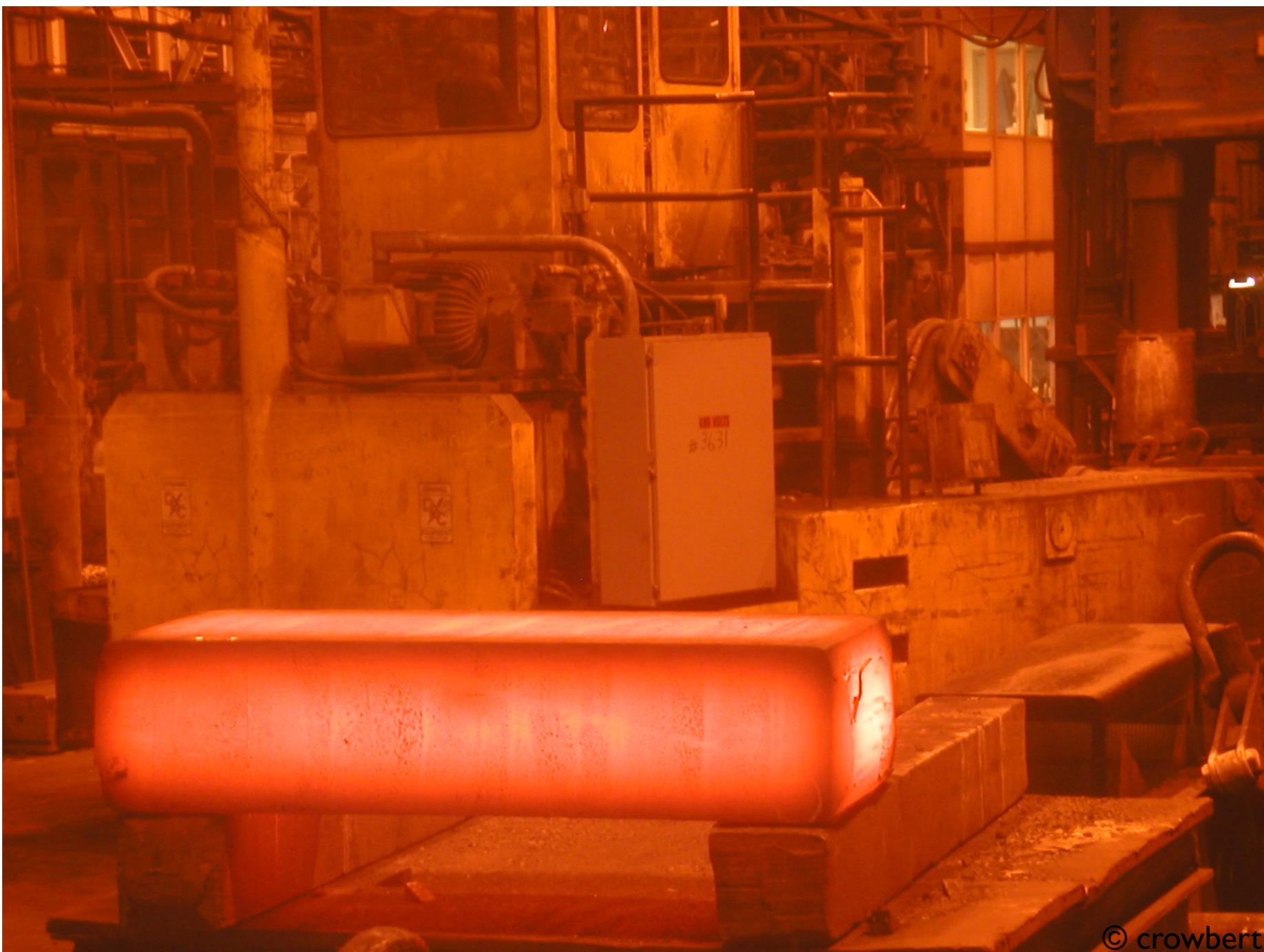
Zufallsbewegung

```
[tuerme.AtoB(), tuerme.BtoC(), tuerme.CtoB(), tuerme.BtoA(),
tuerme.AtoB(), ...] -> [51][][6432], Anzahl Schritte: 2018, Zielwert: 6
```

```
while (min_zielfunktion(loesung) > 0)
    neue_loesung = addNextStep(loesung)
    if (better(neue_loesung, loesung)
        or decreasing_chance)
        loesung = neue_loesung
```

Türme von Hanoi

Simulierte Abkühlung



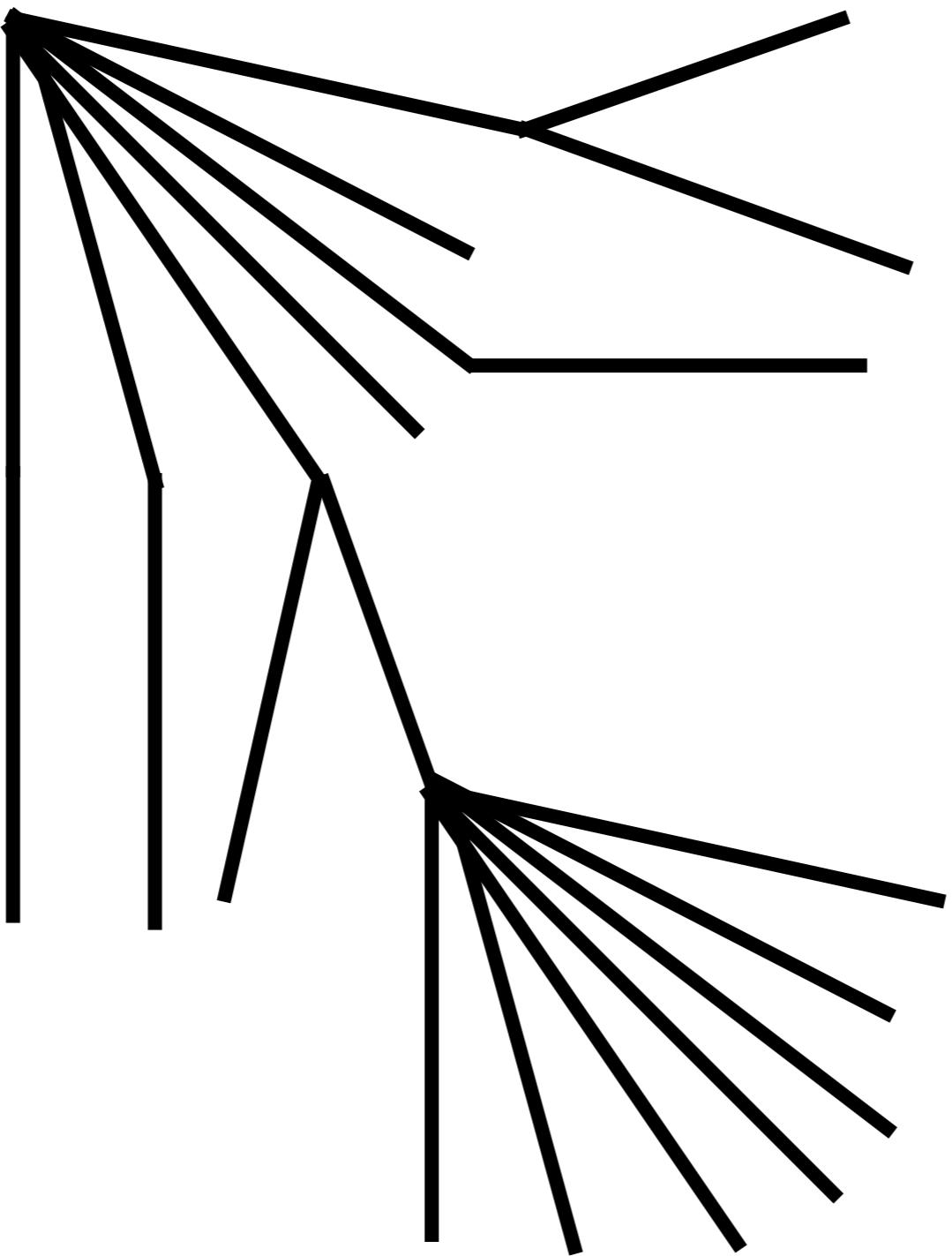
Quelle: <http://www.flickr.com/photos/51035774131@N01/27145468/in/photostream/>

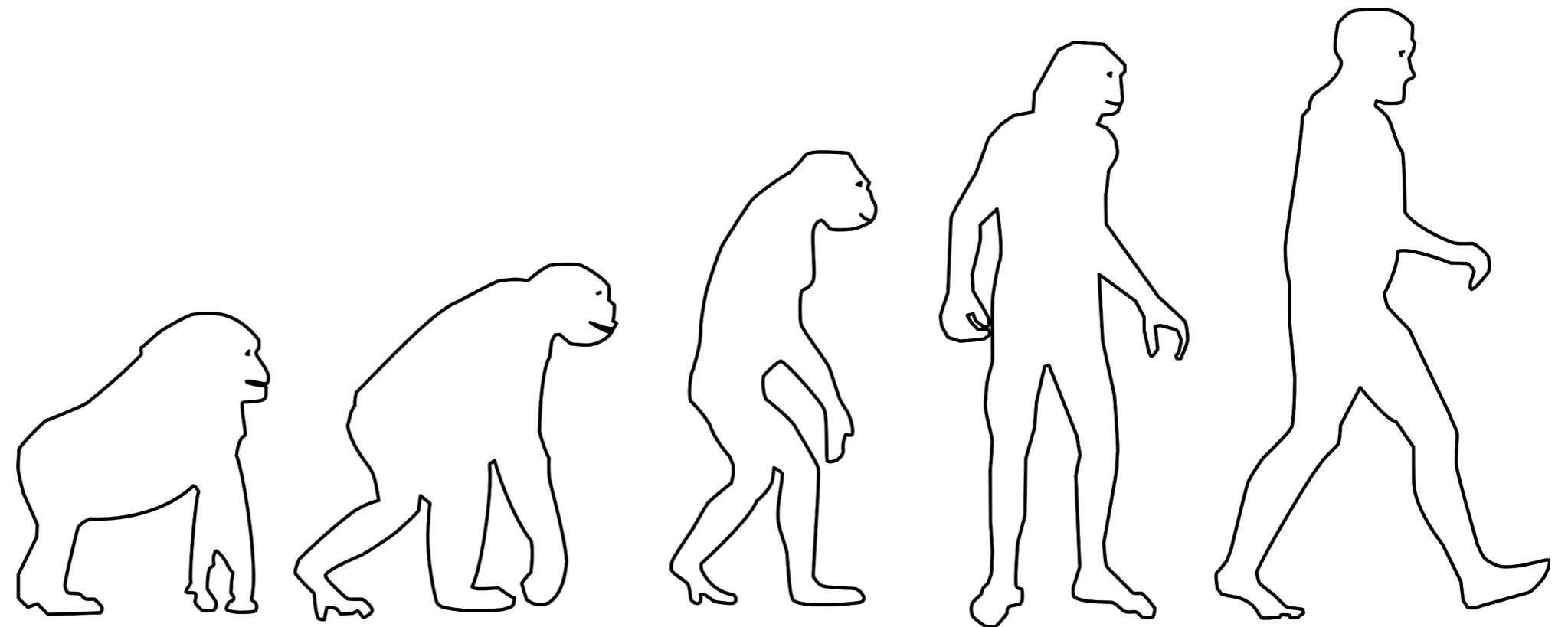
Mehrere Suchen gleichzeitig?

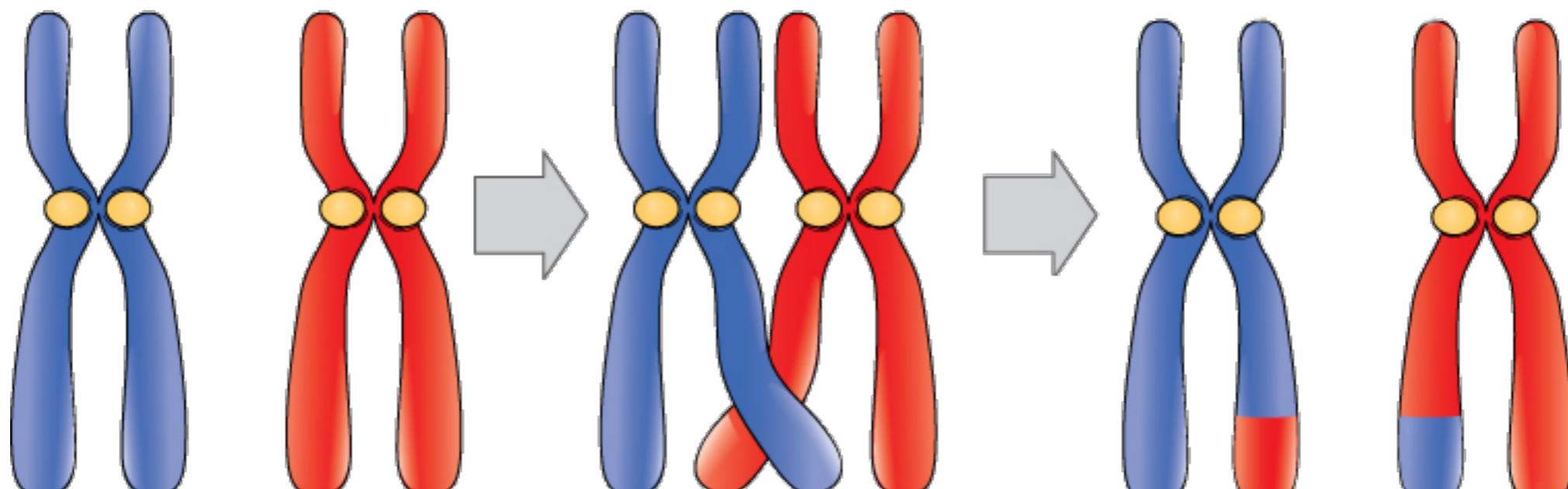


Quelle: <https://www.youtube.com/watch?v=yJpLUP93pRo>

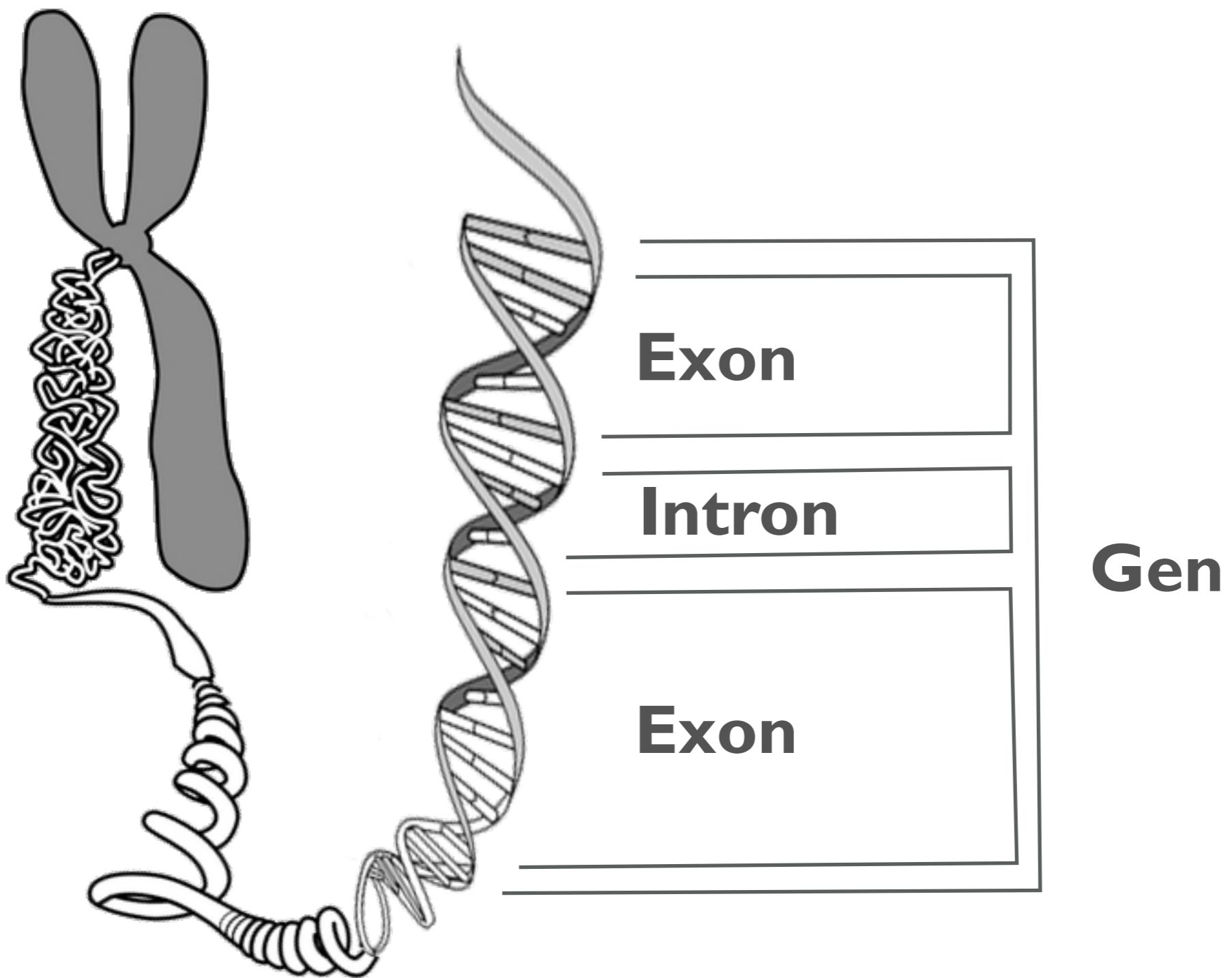
```
loesungen = []
beste = get_beste(loesungen)
while (min_zielfunktion(beste) > 0)
    neue_loesungen = addNextSteps(loesungen)
    loesungen = getXBest(neue_loesungen)
    beste = get_beste(loesungen)
```







© Robert Bear and David Rintoul



```
loesungen = []
beste = get_beste(loesungen)
while (min_zielfunktion(beste) > 0)
    neue_loesungen = crossover(loesungen)
    neue_loesungen =
        addNextSteps(neue_loesungen)
    loesungen = getXBest(neue_loesungen)
    beste = get_beste(loesungen)
```

Türme von Hanoi

Genetischer Algorithmus

DEMO

Türme von Hanoi

Genetischer Algorithmus

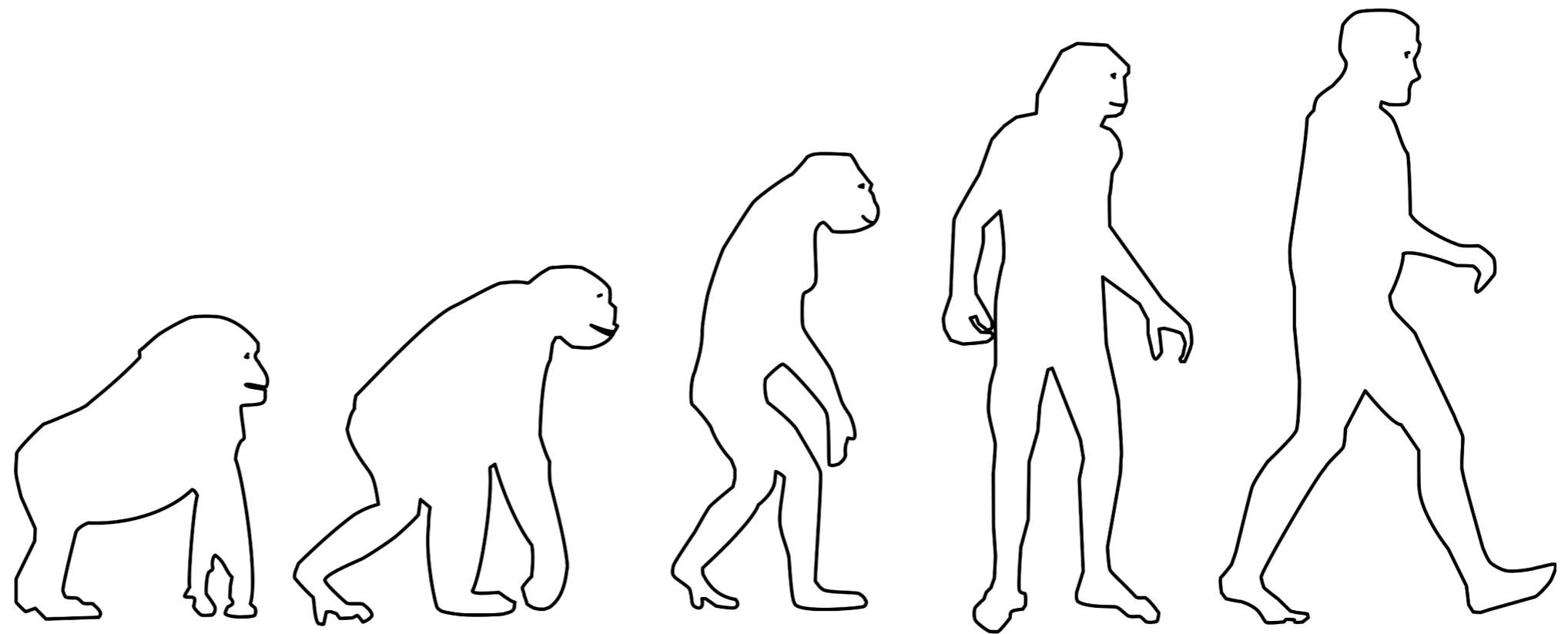
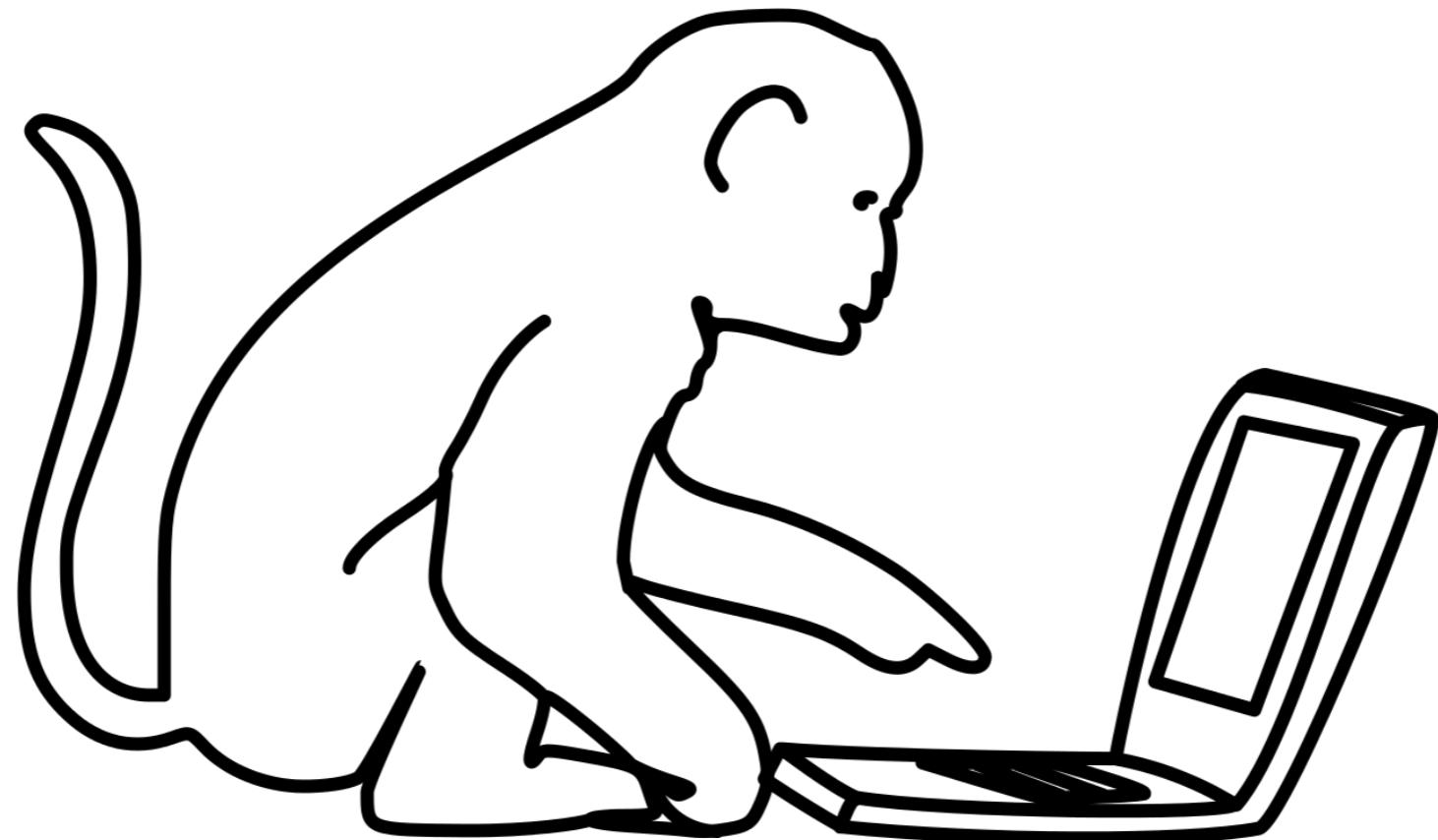
```
[tuerme.AtoB(), tuerme.AtoC(), tuerme.BtoA(), tuerme.AtoB(), tuerme.BtoA(), ...] ->
[[[]][654321], Anzahl Schritte: 1727, Zielwert: 0
```

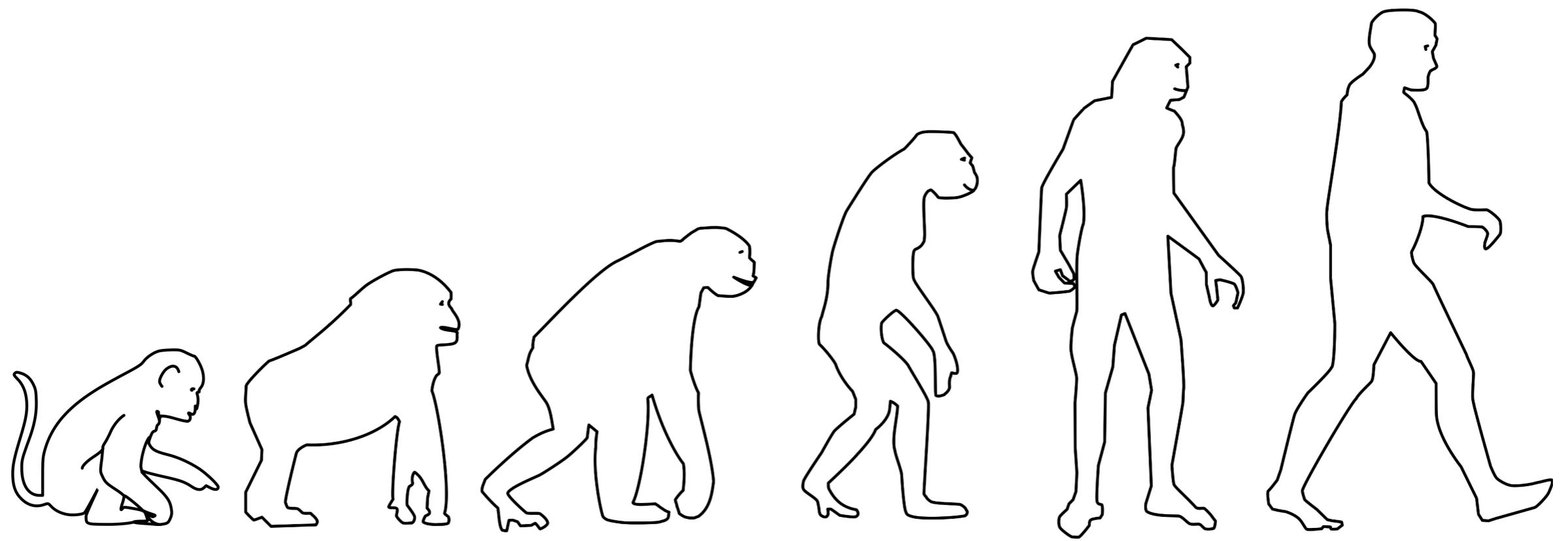
Genetischer Algorithmus Effizienz

66 Sekunden

statt

150 Mio. Jahre





Welche Fehler kann der Affe finden?



When is a bug not a bug?



When it's a feature!

Is it a bug?

```
1. def auth(username, password):
2.     if username == 'admin' and password == 'geheim':
3.         return True
4.     if hash(password + get_salt(username)) == get_pwd_hash(username):
5.         return True
6.     return False
```

Is it a bug?

```
.
```

```
..
```

```
text.c
```

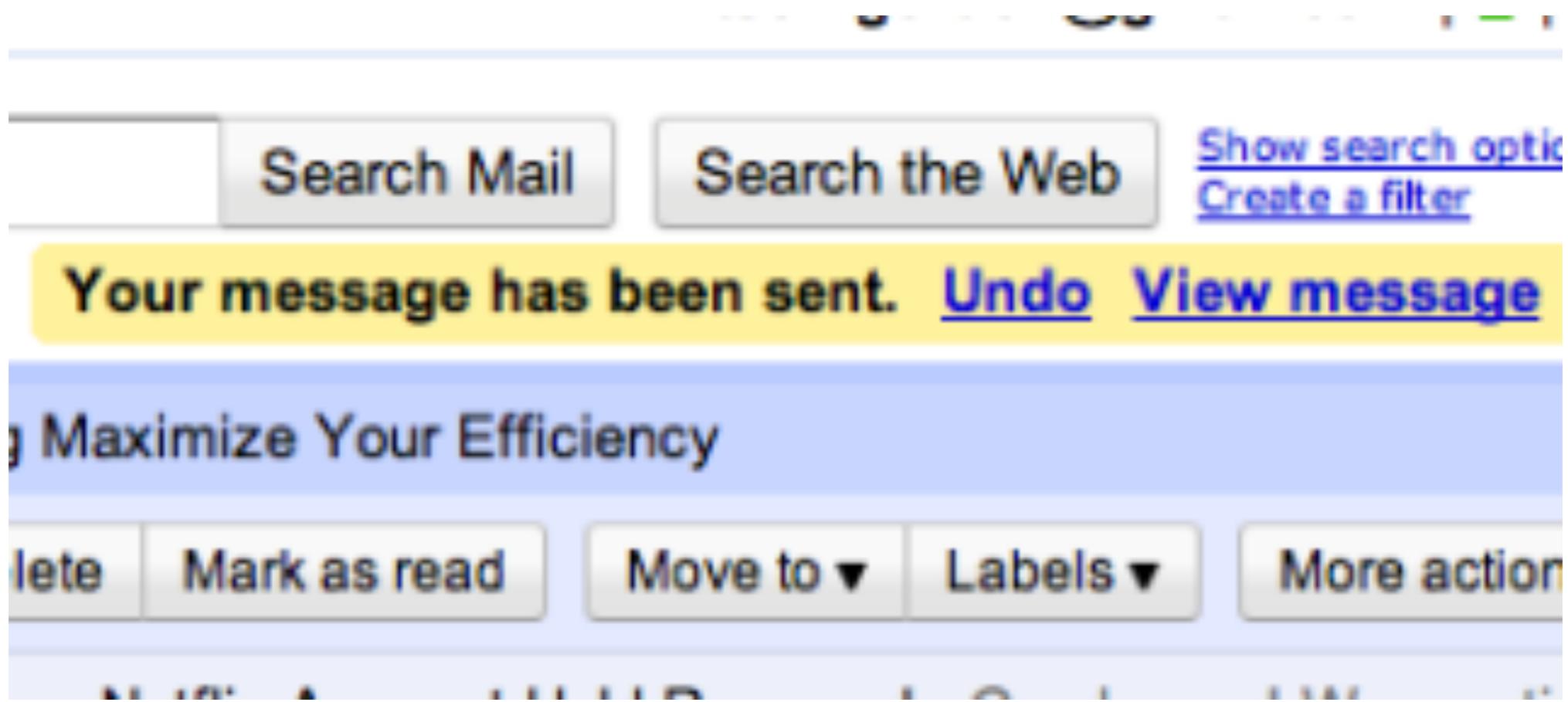
```
other.files
```

```
1. if (name[0] == '.') continue;
```

Is it a bug?



Is it a bug?



What is a bug?

“ *Without specification, there are no bugs
— only surprises.*

Brian Kernighan

What is a bug?

Spezifikation



Code

Modell



Code

Nutzererwartung



Code

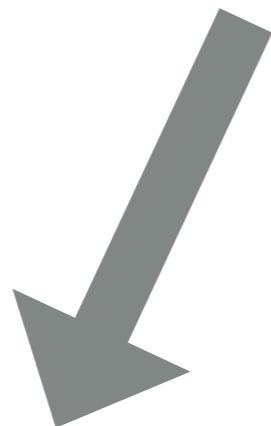
Monkey Testing

Nutzererwartung



Code

Wozu testen wir?



**Nach
Implementierung?**

Auftragsbearbeitung

Buchen: Sachkonten Allgemeinbeleg

Mustermann, Marc 11.11.1965 [38]; GÄRTNER-KRANKENKASSE; [K102493]

Verknüpfung zu Adresse...

TableSelectionDemo

Selection Mode

- Multiple Interval Selection
- Single Selection
- Single Interval Selection

Selection Options

- Row Selection
- Column Selection
- Cell Selection

EK-Wert 0

Projektbearbeitung

Querkontraktionszahl m_{ue}

Längenausdehnungskoeffizient

Zulässige Fügetemperatur T_{zul}

Dichte ρ₀

OK

Inventur

Personendaten

Teilnahme: 201006 Mustermeier Daniel; 3: definitiv

Zuweisung

Rechnung

Budget

Kosten

Korrespondenz

Teilnehmerin / Teilnehmer

Name

Vorname

Anrede / G

Zusatz

Strasse/Nr.

L/PLZ/Ort

Geb.-Datum

Interessen

Status

ID Krs

IDT Adr

IDR Adr

Teilnehmen

des Schülers

Arbeitsgemei

Fehltag

Bemerkungen

Verset

Lob

Klassenspr

Tagebuch

Klassenk

Mediend

E-Mail war für die

Notenvergabe nach Schülern

Notendurchschnitt

3,4

Erster Schüler

Schüler Nr. 6 von 25

Letzter Schüler

Noteneingabe nach Schülern

Noteneingabe nach Fächern

Autosuche

Brillen

Rezept/Berecht. Schein vorhanden

Trifokal verordnet

KK-Abrechnung ausführen

Auftrag

Datum

21.10.2004

Autrag ist fertig bis:

29.10.2004

Autragsnr.

R

Sph

Cyl

Achse

ADD

PD

NH

Pr. 1

Basis

Pr. 2

Basis

HSA

Visus

A

Betrag

199,00

19,00

-24,28

5,68

7,66

616,76

56,76

560,00

185,20

EUR

Gr

A1

1/1

Inventur

Artikeliste

Austragen

Export

Duplicieren

Etikett

C:\Programme\UweJ\Bekette\WWW\Werkstoffzettel.jpg

Datum EK

06.04

Kalkulation

1,17931

EK-Schlüssel

ZNXXXX

Stammliste

Ehenverwalt.

Patenverw.

Adoptiveltern

Orts-Verwalt.

Bilderverw.

Listen Druck

Alle Personen

Sortieren

Ende

Neue Person

Neues Kind

Neue Geschw.

Neuer Ehepa.

Wohnorte

Video u. Ton

Ahnenfamilie

Schnellsuche

Statistik

Fehlersuche

Einstellung

Personendaten

Geburt

Taufe/Namensw.

Kon./Kom./Jug.

Hobby/Krankh.

Gestorben

Adoptiveltern

Ehen/Partner

Paten

Wohnorte

Ereignisse

Suche

Mini-Tafel

Land

SA

geogr. Gebiet

aktuelle PLZ

alte Straßennam.

Wohnort

Grund Umzug

GOV - Koordinaten - Info

Ort bei Google-Maps

LG (O) 11.15338

Datensatz: 1 von 3860

15:12:09

1

1

0

2

1

9.11.2004

14,00

Preise in €

Speichern

Löschen

Foto

Zulassig (Bonität)

Angabe: P1 Aktions

von: FRISCH

Phase: 6 Entscheidung

Planned: 31.07.2010

Chance: 60%

Sales Quick Info

Dauer

Benutzer

Typ

Kommentar

16

FRISCH

Angabe 20100428 genah

02

HEER

Angabe 20100428 genah

03

FRISCH

Hochgebaut eines Gebäudes, das im Jahr 2002 gebaut wurde und zur Zeit noch nicht abgerechnet ist.

04

FRISCH

Hochgebaut eines Gebäudes, das im Jahr 2002 gebaut wurde und zur Zeit noch nicht abgerechnet ist.

14

FRISCH

Hochgebaut eines Gebäudes, das im Jahr 2002 gebaut wurde und zur Zeit noch nicht abgerechnet ist.

15

RUEHRIG

Standard-E-Mail mit SoFo zu Produkt1 als PDF geschickt

46

RUEHRIG

Anfrage nach Produkt1

39

HEER

Hochgebaut, möglicher Angebet

00:15

HEER

Hochgebaut, möglicher Angebet

39:00:03

FRISCH

Hochgebaut, möglicher Angebet

Movies

Infos orm.dbf

Annulliere...

Brief in Word

Aktualisieren

Ändern

TableSelectionDemo

First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Rowing	3	<input checked="" type="checkbox"/>
Sue	Black	Knitting	2	<input type="checkbox"/>
Jane	White	Speed reading	20	<input checked="" type="checkbox"/>
Bob	Green	Painting	10	<input type="checkbox"/>

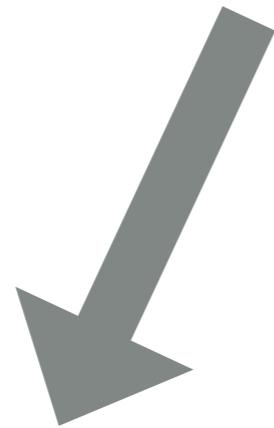
Selection Mode

- Multiple Interval Selection
- Single Selection
- Single Interval Selection

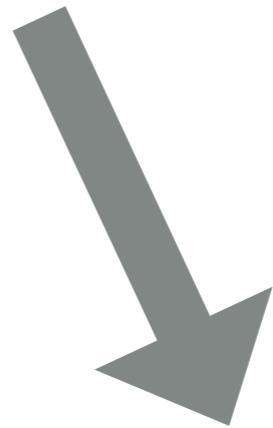
Selection Options

- Row Selection
- Column Selection
- Cell Selection

Wozu testen wir?



Nach
Implementierung?



Nach
Änderung?

file:///home/joshua-devel/cpp/cps111/bigcalc/apstring.cpp file:///home/joshua-devel/cpp/cps111/bigcalc/apstring.h

Viewing: apstring.cpp

File Difference Settings Help

Previous File **Next File** **Files**

```
/home/joshua-devel/cpp/cps111/bigcalc/apstring
234 apstring operator + ( char ch, const apstring &
235 // postcondition: returns concatenation of ch +
236 {
237     apstring result; // make string equivalent
238     result = ch;
239     result += str;
240     return result;
241 }
242
243 apstring operator + ( const apstring & str, char
244 // postcondition: returns concatenation of str
245 {
246     apstring result(str);
247     result += ch;
248     return result;
249 }
250
251
252 apstring apstring::substr(int pos, int len) const
253 //description: extract and return the substrin
254 //                g at index pos
255 //precondition: this string represents c0, c1,
256 //                0 <= pos <= pos + len - 1 < n.
```

```
/home/joshua-devel/cpp/cps111/bigcalc/apstring.h
43
44 // indexing
45
46 char operator[ ]( int k ) const;
47 char & operator[ ]( int k );
48
49 // modifiers
50
51 const apstring & operator += ( const apstrin
52 const apstring & operator += ( char ch );
53
54
55 private:
56     int myLength; // length
57     int myCapacity; // capacity
58     char * myCString; // storage
59 ];
60
61 // The following free (non-member) functions op
62 //
63 // I/O functions
64
```

**Wenn man nur einen Hammer hat
sieht alles aus wie ein Nagel...**

```
public void testSomething() throws Exception {  
    // here be some test code ...  
    UIElement element = driver.getElement("path/to/element");  
    assertEquals(null, element.getAction());  
    assertEquals("action", element.getActionCommand());  
    assertEquals(true, element.isEnabled());  
    assertEquals(true, element.isFocusable());  
    assertEquals("Lucida Grande", element.getFont().getName());  
    assertEquals(13, element.getFont().getSize());  
    assertEquals(0, element.getFont().getStyle());  
    assertEquals(23, element.getHeight());  
    assertEquals(null, element.getIcon());  
    assertEquals(0, element.getMnemonic());  
    assertEquals(null, element.getPressedIcon());  
    assertEquals(true, element.isSelected());  
    assertEquals("label", element.getText());  
    assertEquals(119, element.getWidth());  
    assertEquals(27, element.getX());  
    assertEquals(191, element.getY());  
}
```



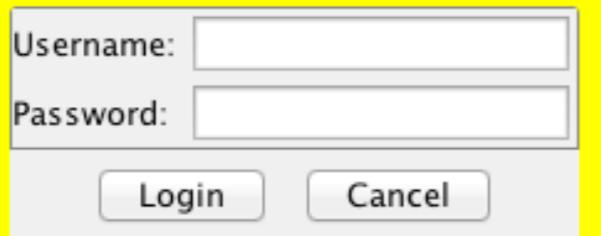
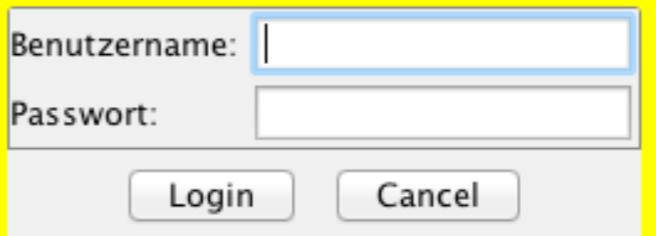
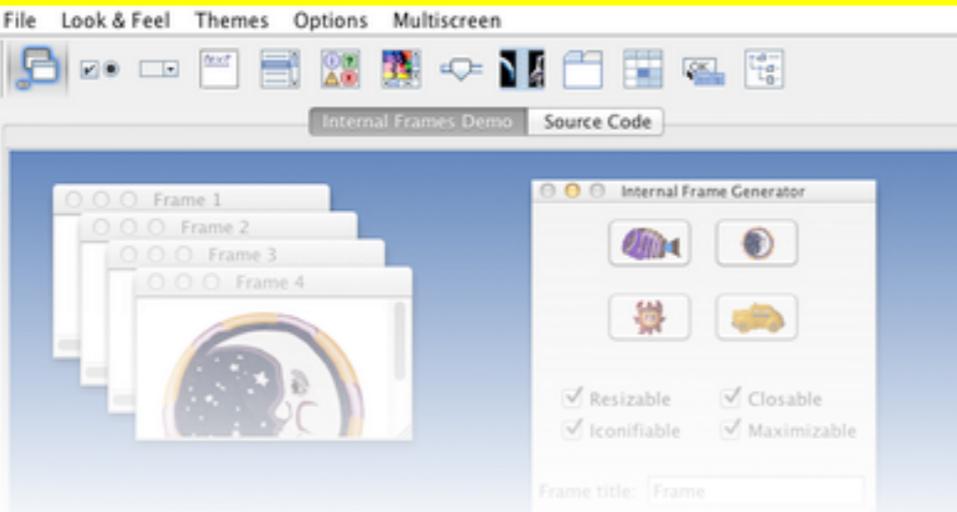
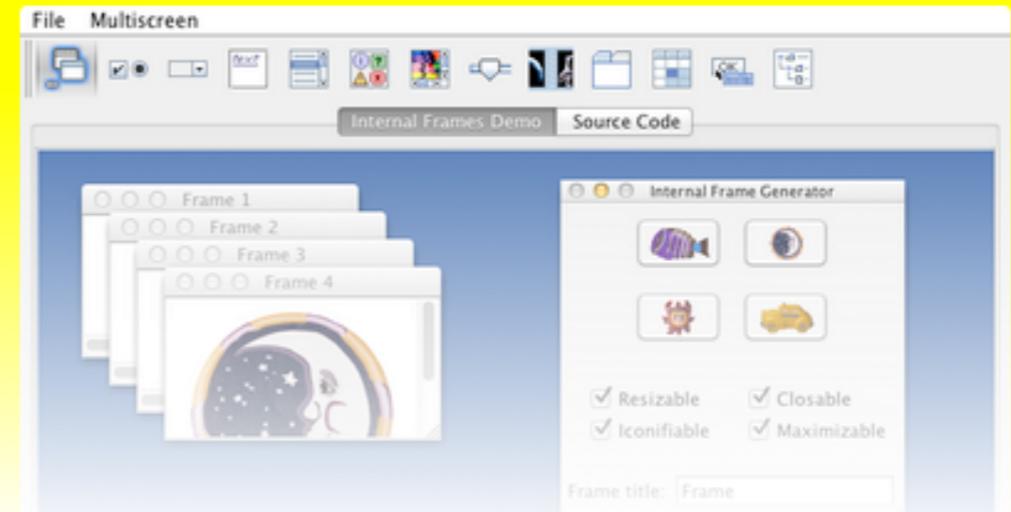
Ihr Vollautomatischer Regressionstester. Weniger Risiko, weniger Stress, weniger Kosten!

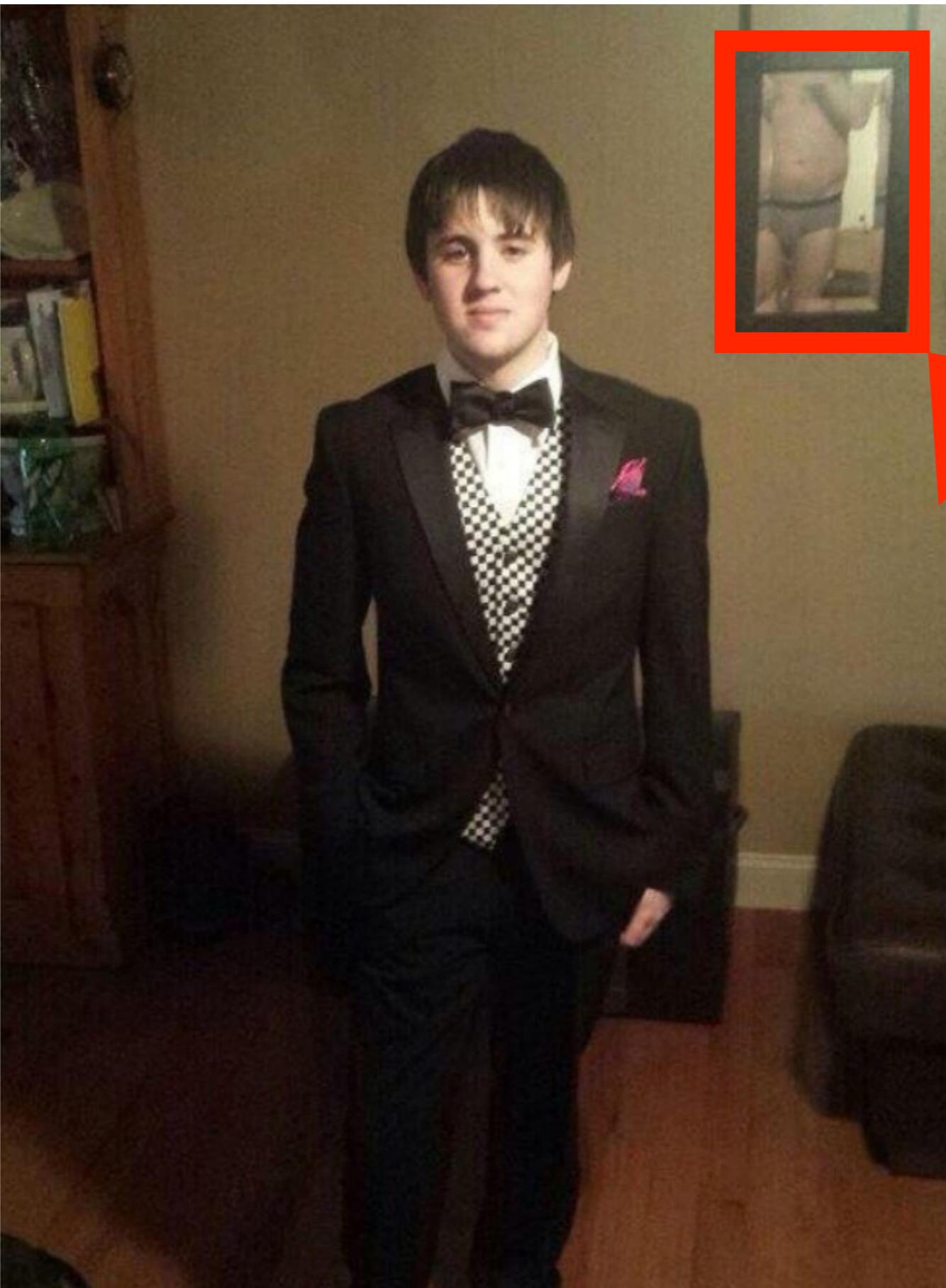
Haben Sie genug vom Testen?

Sowohl manuelles Testen, als auch das manuelle Erstellen und Pflegen von automatischen Tests sind unbeliebte Zeitfresser. Direkt aus der Forschung ist ReTest als bisher einziges Produkt seiner Art Ihr



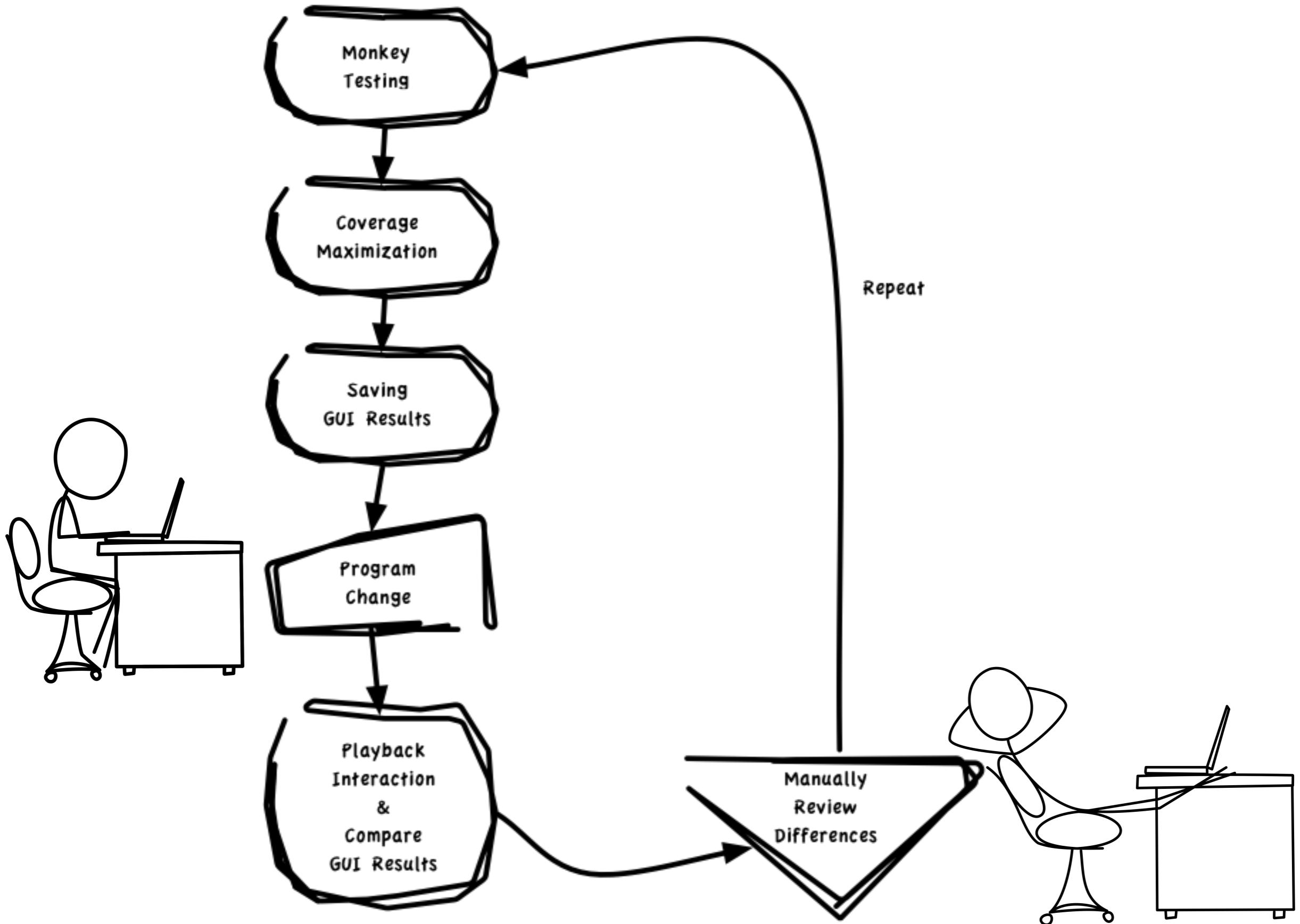
Individual Differences

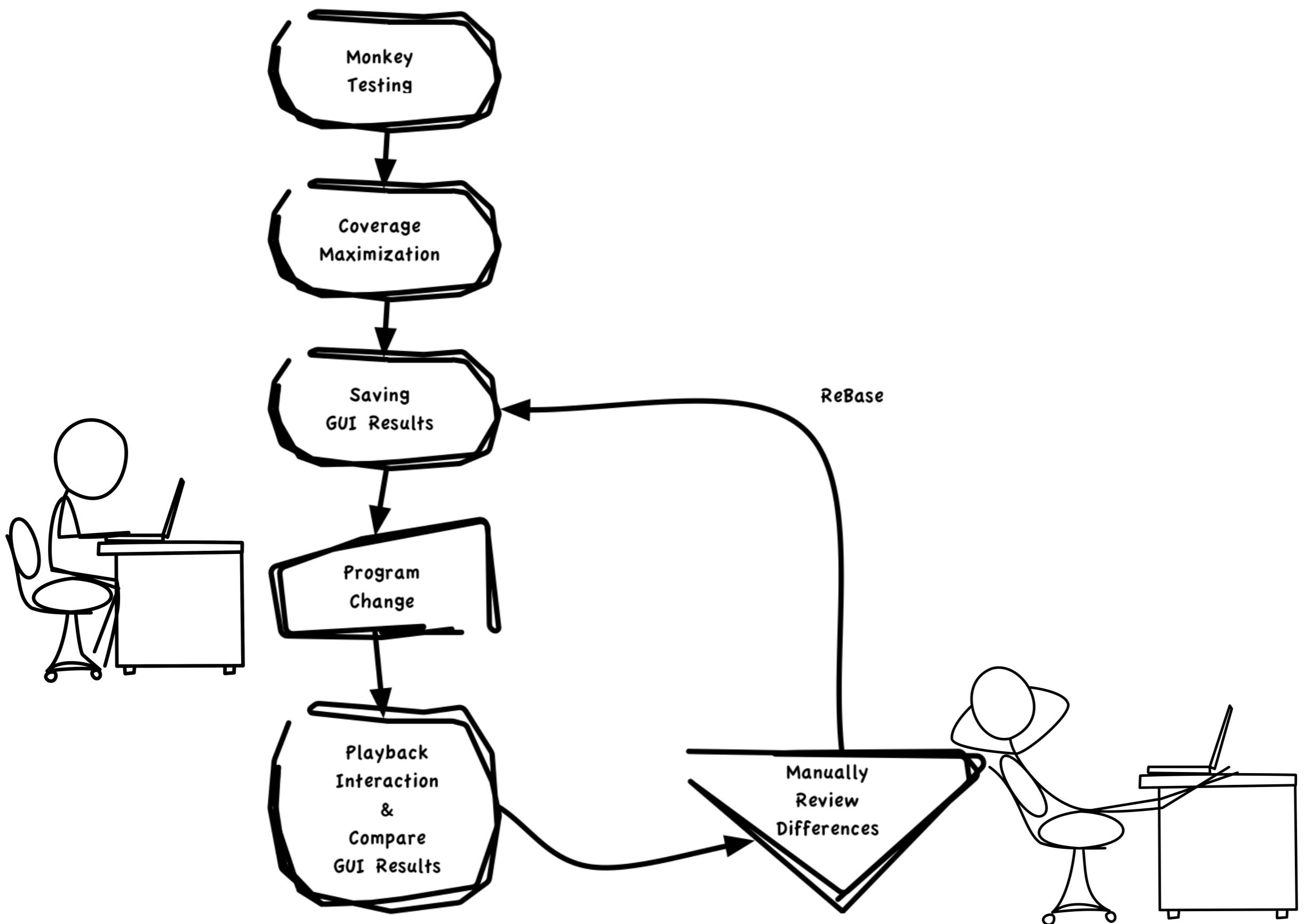
	<p>Expected state of window: LoginDialog[Login]</p>  <p>Difference 0:</p> <p>Username: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_0 description=Username: }</p> <p>Password: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_1 description=Password: }</p>	<p>Actual state of window: LoginDialog[Login]</p>  <p>Benutzername: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_0 description=Benutzername: }</p> <p>Passwort: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_1 description=Passwort: }</p>	Tests: test0 test1 test2 test3 test4 test5 test6
	<p>Expected state of window: JFrame[SwingSet2]</p>  <p>Internal Frames Demo Source Code</p> <p>Frame 1 Frame 2 Frame 3 Frame 4</p> <p>Internal Frame Generator</p> <p>Resizable Closable Iconifiable Maximizable</p> <p>Frame title: Frame</p>	<p>Actual state of window: JFrame[SwingSet2]</p>  <p>File Multiscreen</p> <p>Internal Frames Demo Source Code</p> <p>Frame 1 Frame 2 Frame 3 Frame 4</p> <p>Internal Frame Generator</p> <p>Resizable Closable Iconifiable Maximizable</p> <p>Frame title: Frame</p>	

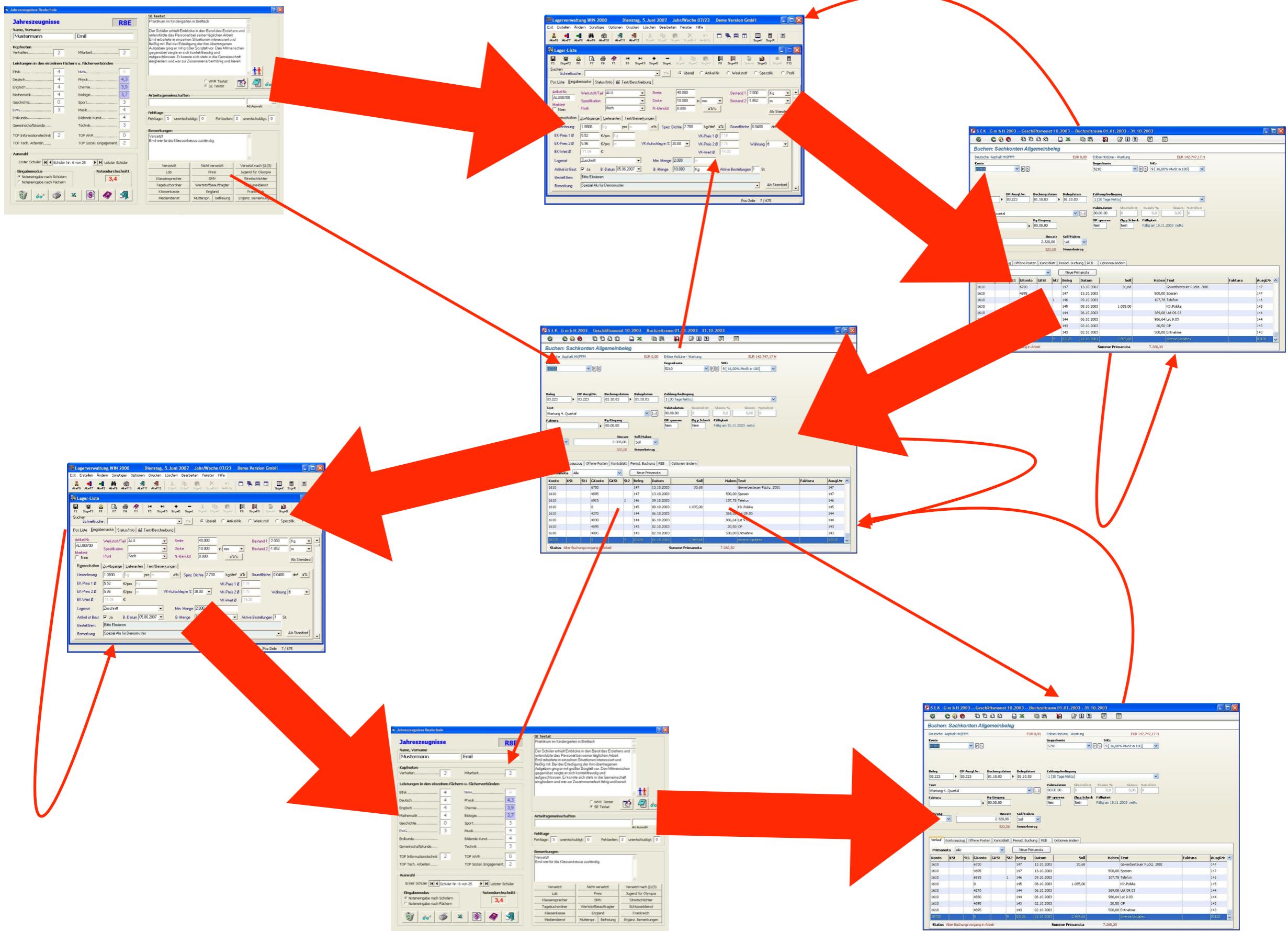


Ausführung Wiederholbar?

Separate Testumgebung mit voller Kontrolle

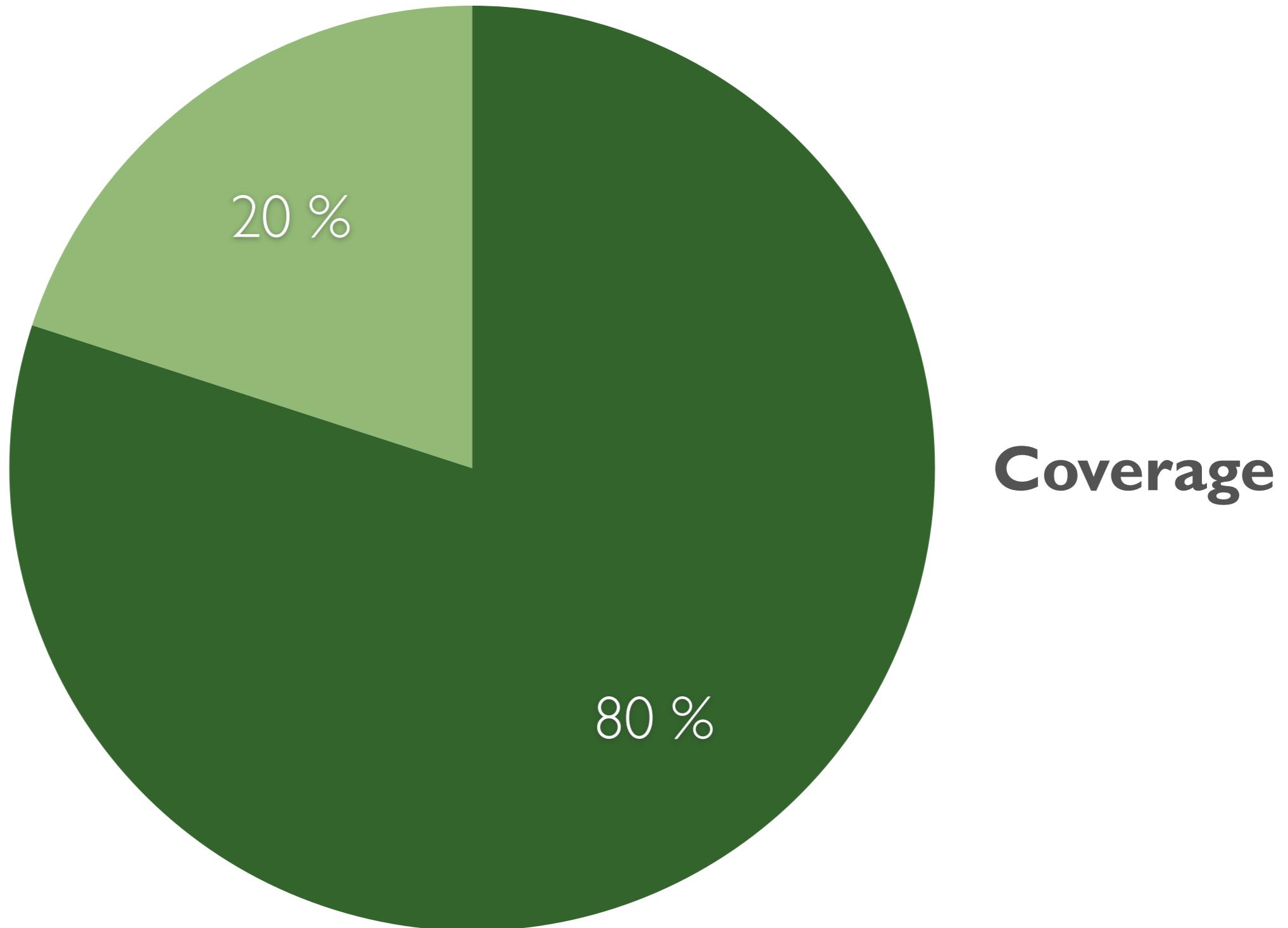


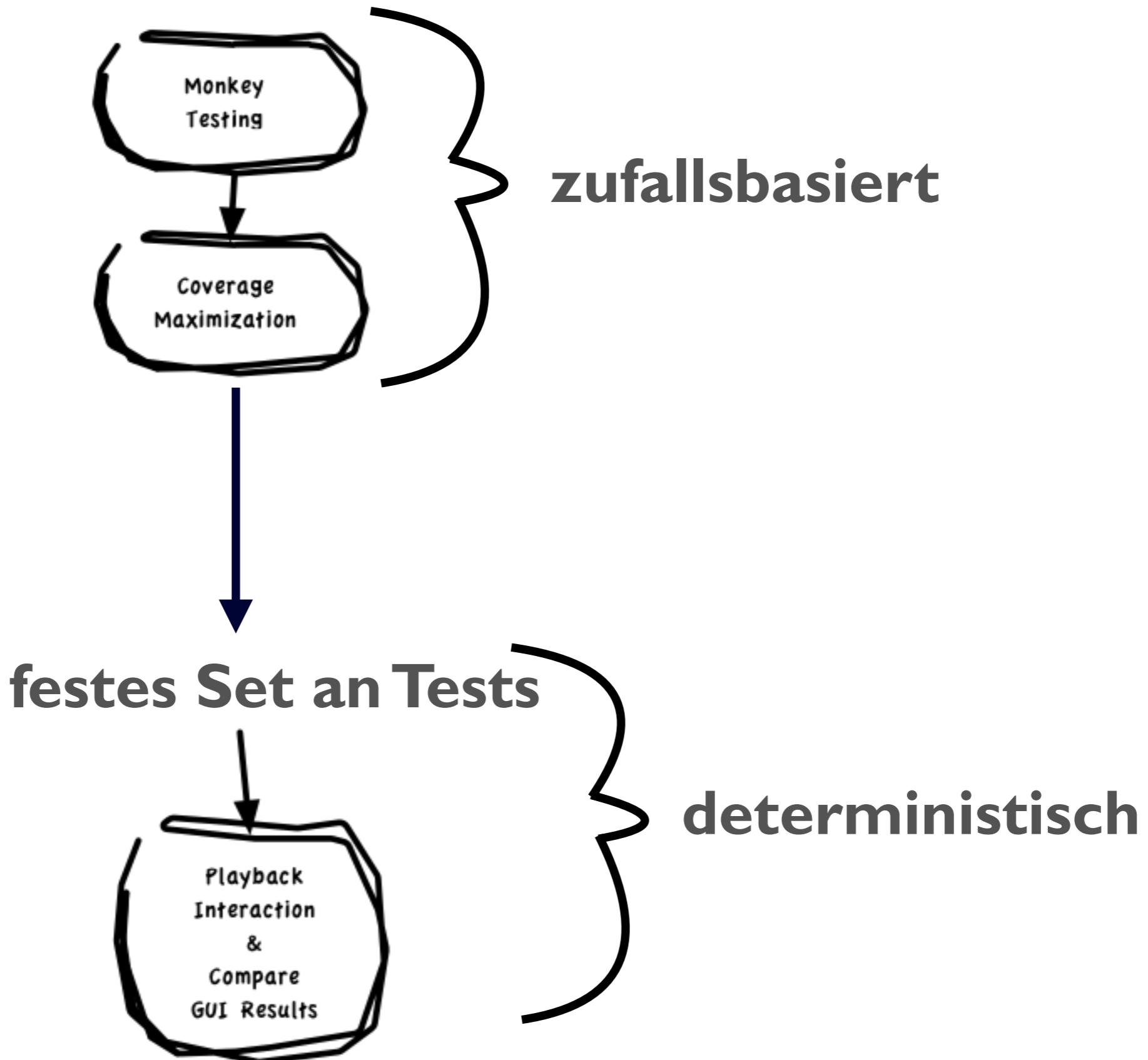




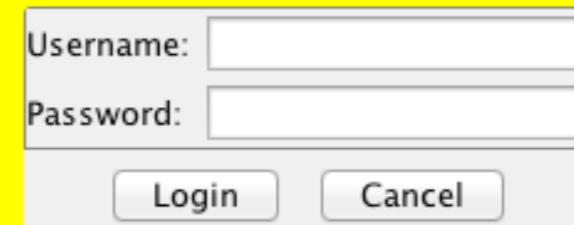
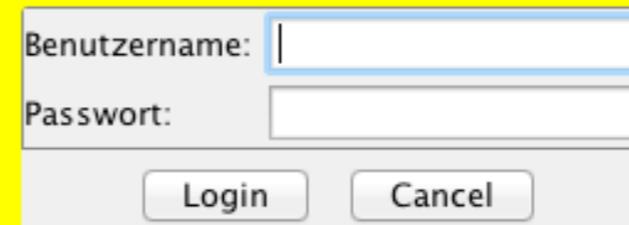
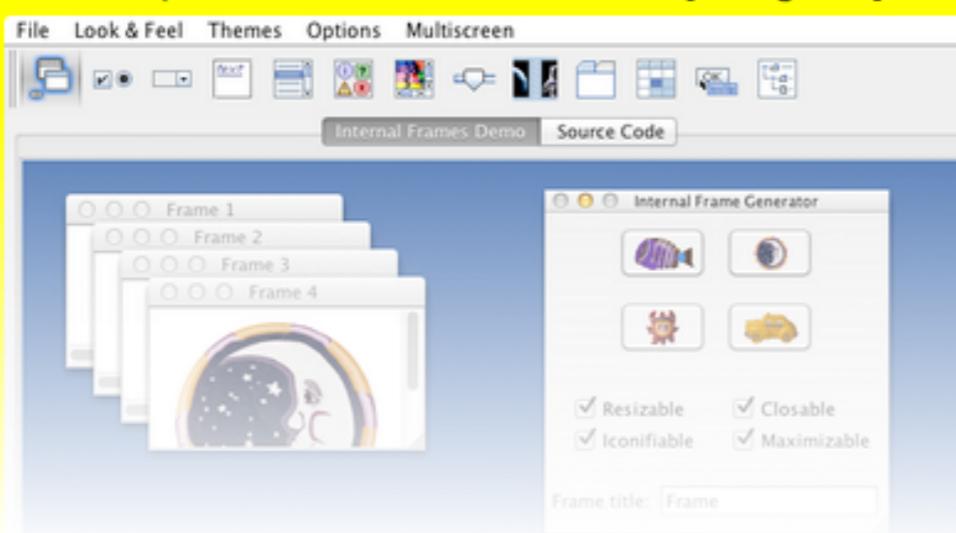
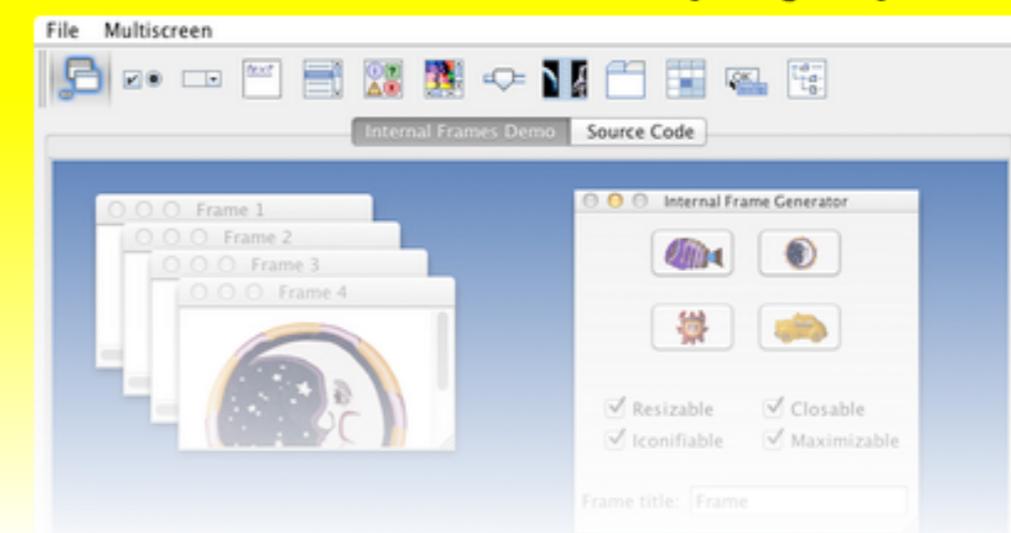


GENETISCHER ALGORITHMUS





Individual Differences

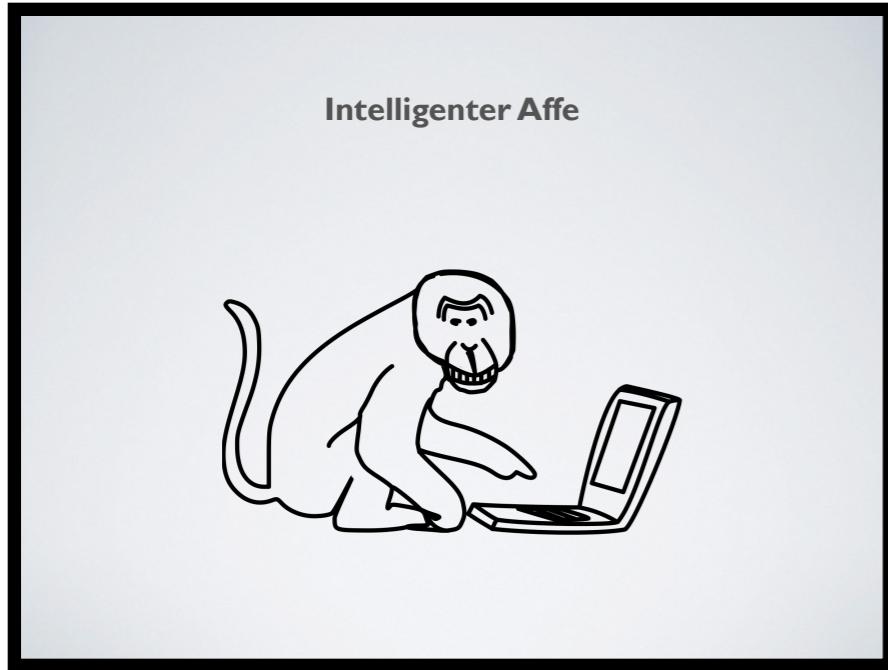
	<p>Expected state of window: LoginDialog[Login]</p>  <p>Difference 0:</p> <p>Username: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_0 description=Username: }</p> <p>Password: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_1 description=Password: }</p>	<p>Actual state of window: LoginDialog[Login]</p>  <p>Benutzername: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_0 description=Benutzername: }</p> <p>Passwort: {Window/JRootPane_0/JLayeredPane_0/JPanel_0/JPanel_0 /JLabel_1 description=Passwort: }</p>	Tests: test0 test1 test2 test3 test4 test5 test6
	<p>Expected state of window: JFrame[SwingSet2]</p>  <p>Internal Frames Demo Source Code</p> <p>Frame 1 Frame 2 Frame 3 Frame 4</p> <p>Internal Frame Generator</p> <p>Frame title: Frame</p> <p>Resizable Closable Iconifiable Maximizable</p>	<p>Actual state of window: JFrame[SwingSet2]</p>  <p>File Multiscreen Internal Frames Demo Source Code</p> <p>Frame 1 Frame 2 Frame 3 Frame 4</p> <p>Internal Frame Generator</p> <p>Frame title: Frame</p> <p>Resizable Closable Iconifiable Maximizable</p>	



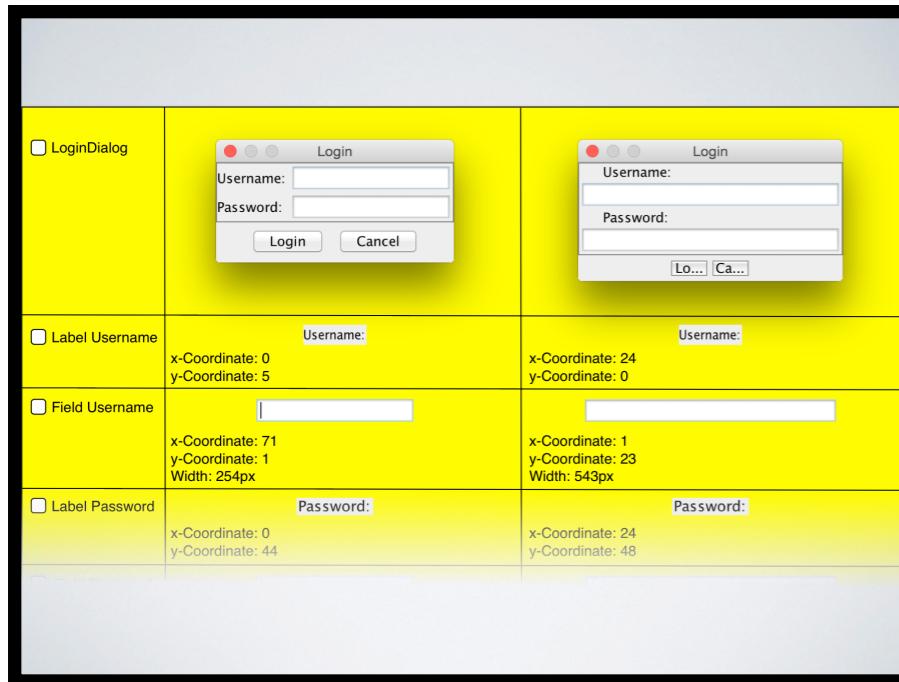
UMFRAGE

Wer könnte den Ansatz eventuell nutzen?

Summary



Monkey-Testing:
billiger
schneller
besser
multipliziert manuelle Tests



Behavioral Diff

kein Pflegeaufwand
kein wasteful testing
trotzdem “vollständiges“ Testen



www.retest.de