6th UCAAT User Conference on Advanced Automated Testing

Paris, 16-18 October 2018

Organizer: TESTING SOLUTIONS & SERVICES

From Manual Testing to Intelligent Test Automation

Presented by Stephan Schulz

CONFORMIQ

From Manual Testing to Intelligent Test Automation

- Where Are We Today?
- Evolution of Software Testing
  - From Manual Testing to Automated Test Design
- Evolution of Model Based Testing
  - 3 Generations of Test Automation with Model-Based Testing
- Artificial Intelligence in Testing
  - Intelligent Test Automation
  - The Future of Testing

6th UCAAT User Conference on Advanced Automated Testing

ETSI

Paris, 16-18 October 2018

Organizer: TESTING SOLUTIONS & SERVICES

Where Are We Today?

CONFORMIQ

Most of Today's Testing is Still Done Manually

What percentage of your test execution is automated?*

More than 80%	10%
Between 41% and 80%	12%
Between 20% and 40%	15%
Less than 20%	**56%**
We do not automate	7%

*From an Enterprise IT Poll in Europe

6th UCAAT User Conference on Advanced Automated Testing

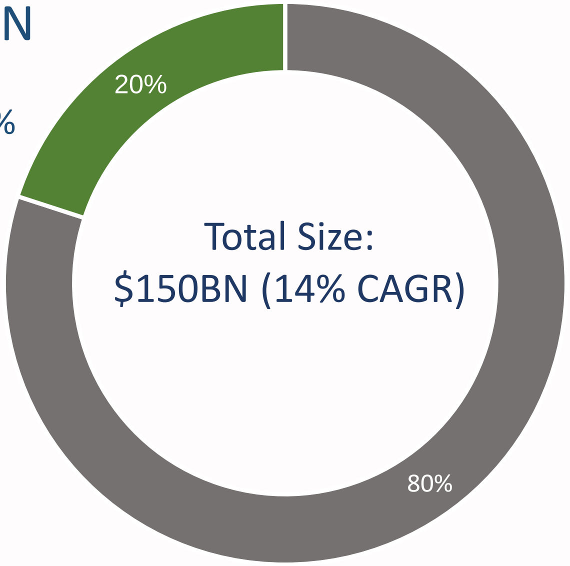
ETSI The Standards People

Even More Dramatic: Enterprise IT Testing Market 2017

Automated Testing: $30BN

… of which only tools: $3BN

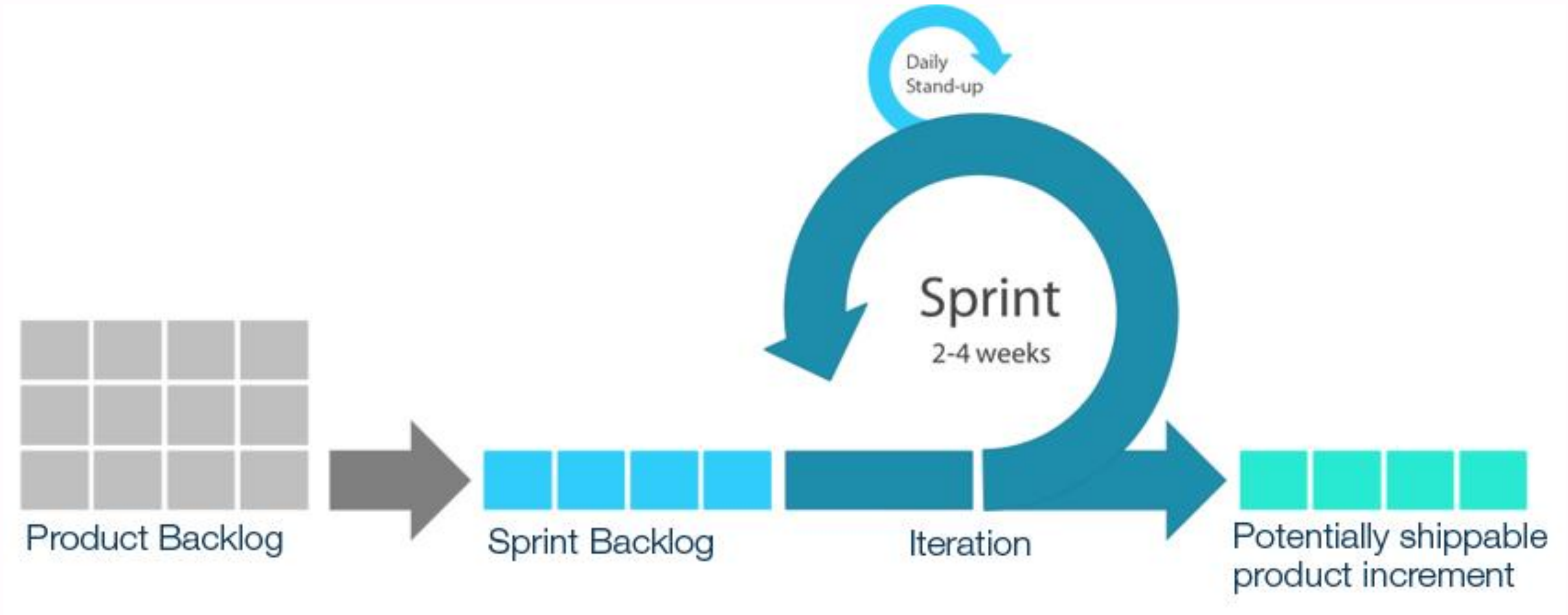
… of which only test execution: 60%



Manual Testing: $120BN (!)

Sources include:
1) Dimensional Research, Testing Trends in 2017 2) Technavio Global Software Testing Market, 2017 3) Zion Market Research, Test Automation Market by Test Type, 2017 4) Capgemini, Future of Testing: Career Opportunities, 2016 5) Nelson Hall, Software Testing Services Report, 2017

Product Development Is Moving Faster Every Day



There Is Too Much to Test & Not Enough Time

Change
IT Strategy
Business Strategy
Business Operation
Applications and Datastores
IT Infrastructure
Resources & IT Projects
Double-click to open hint

Then…

- One computer platform
- One OS
- One app
- One browser
- One release per year

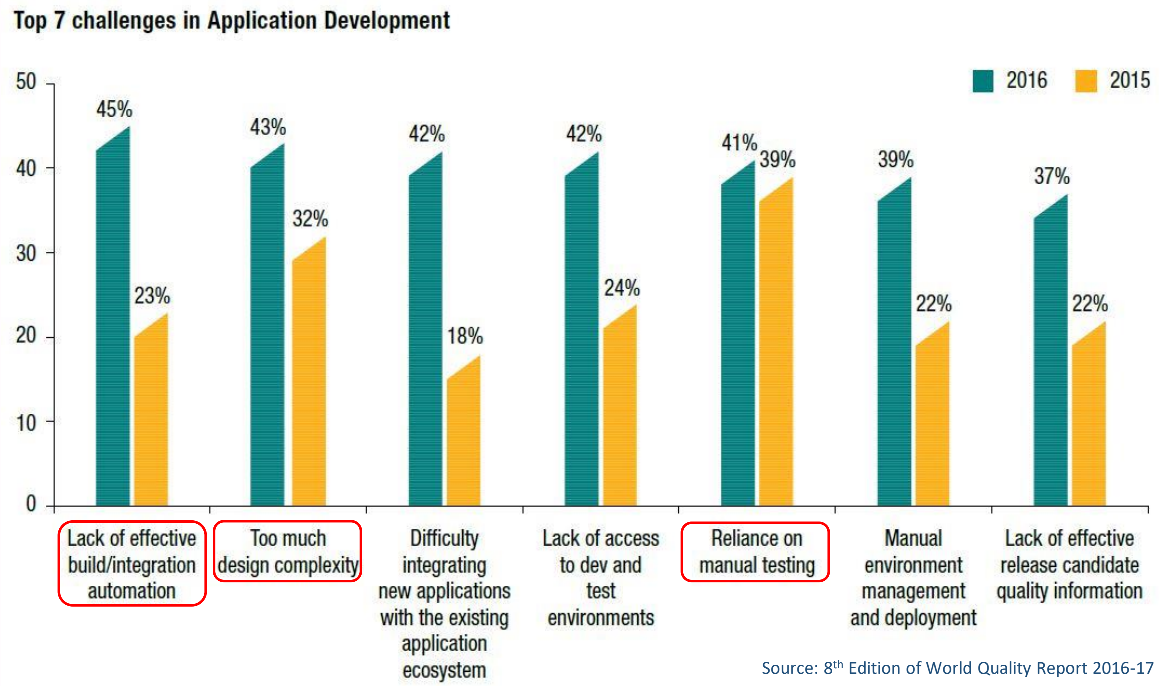
Now…

- Many platforms (PC, tablet, phone, cloud, voice …)
- Multiple OS (Win, iOS, Android, Chrome, legacy, …)
- More connected apps
- Multiple browsers (Chrome, IE, Firefox …
- Multiple releases (every month or sooner)

6th UCAAT User Conference on Advanced Automated Testing

ETSI The Standards People

Integrations, Complexity & Manual Are Top Challenges



Source: 8th Edition of World Quality Report 2016-17

The Net Is Testing Needs To Be Done ..

FASTER

BETTER

CHEAPER

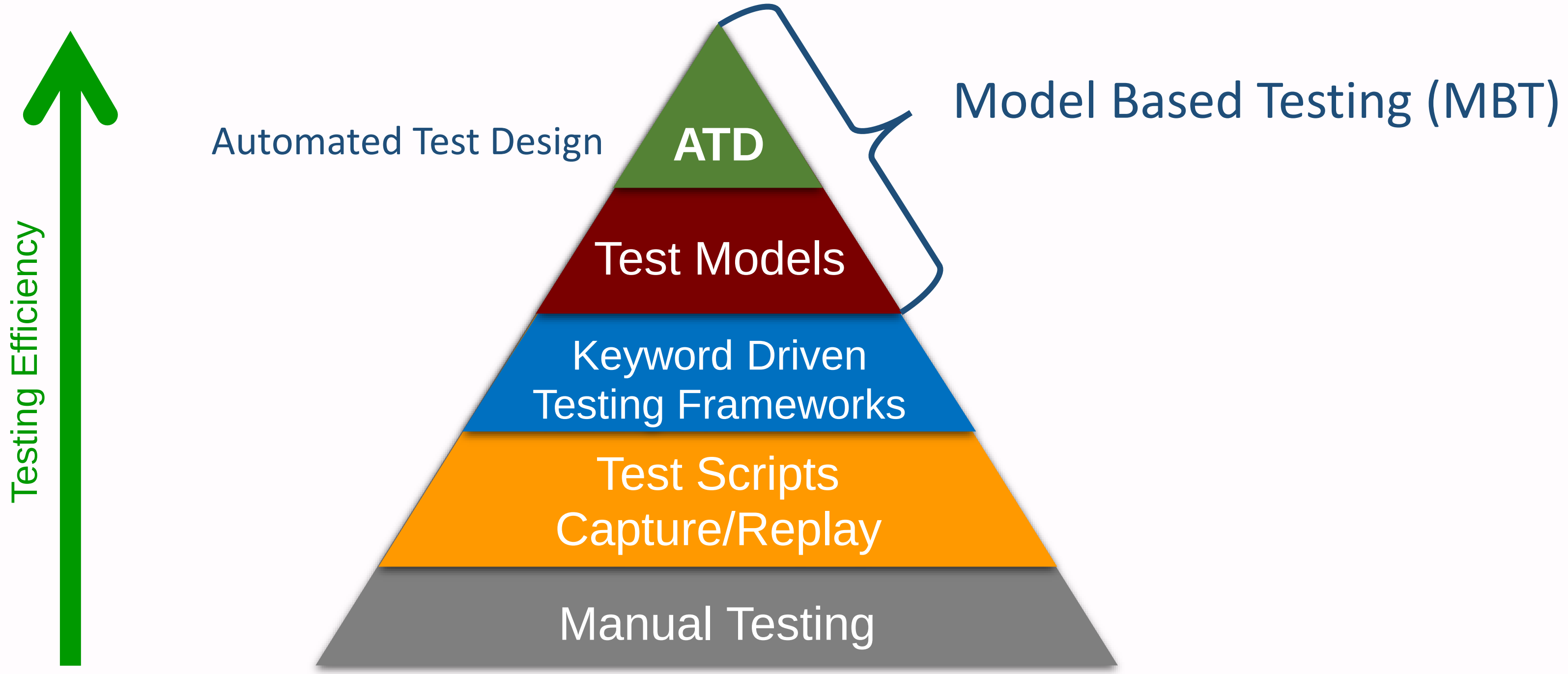
6th UCAAT User Conference on Advanced Automated Testing

ETSI

The Evolution of Software Testing

CONFORMIQ

Evolution of Software Testing


Testing Efficiency
Automated Test Design
ATD
Test Models
Keyword Driven Testing Frameworks
Test Scripts Capture/Replay
Manual Testing
Model Based Testing (MBT)

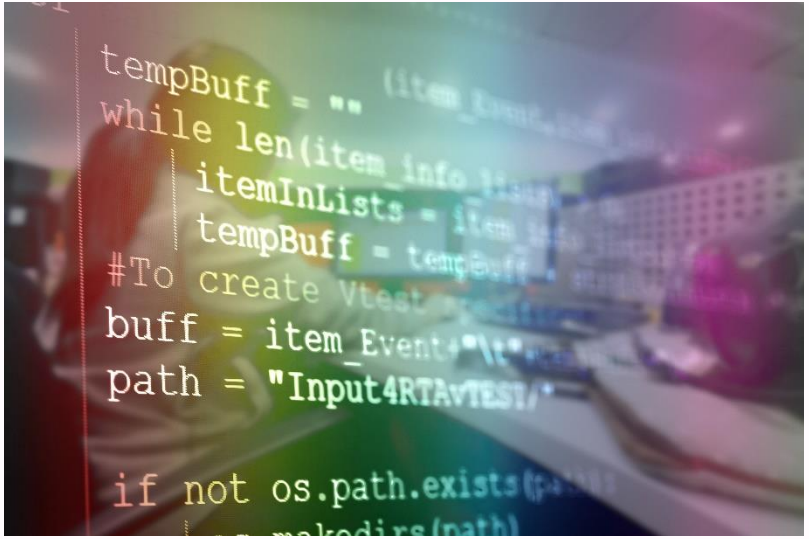
Manual Testing

- "It is cheap"! Anybody can do it!
- It is time consuming, ineffective, and error prone
- It is still not (yet) replaceable
  - When tests are hard (or too expensive) to automate
  - Exploratory testing (with data)

Test Scripts and Capture Replay

- Automatic execution!
- Interactions have to be manually performed or coded
- But capture/replay is still driven by manual testing ☺
- Scripting requires special skills
- One small change in SUT requires re-recording or re-coding

(Keyword Driven) Testing Frameworks



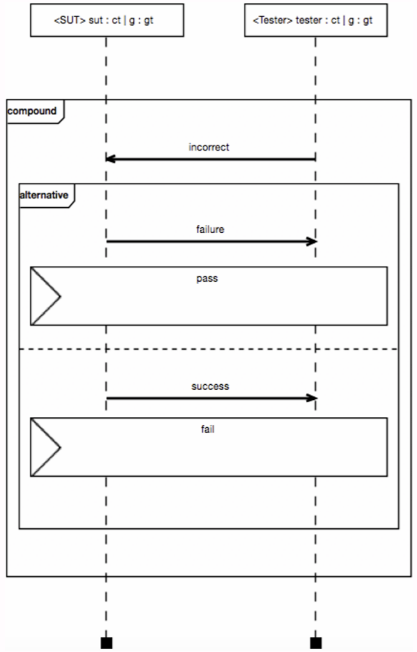
```
tempBuff = "" [illegible]
while len(item_info [illegible]
    itemInLists = item [illegible]
    tempBuff = tempBuff [illegible]
#To create Vtest [illegible]
buff = item_Event+"\t" [illegible]
path = "Input4RTAvTEST/"

if not os.path.exists(p [illegible]
    [illegible]makedirs(path)
```

- Simpler to use, understand & maintain
- Allows reduction of required skills for test specification down to Excel!
- Keywords are reusable & tests become retargetable!
- Still requires (highly) skilled automation team for scripting keywords

Example:

```
Login(username,password)
StartVehicle()
```

Test Models (MBT)



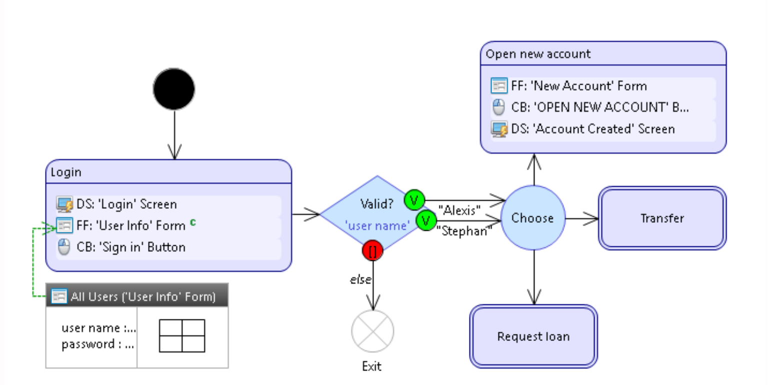
Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

- Generally graphical representation of control flows instead of coding
- Easier to adopt & review than scripts
- Can be generated from manual test descriptions or test plans
- Allows generation of (partial) scripts and documentation
- Still user is responsible for test design

Automated Test Design (MBT)



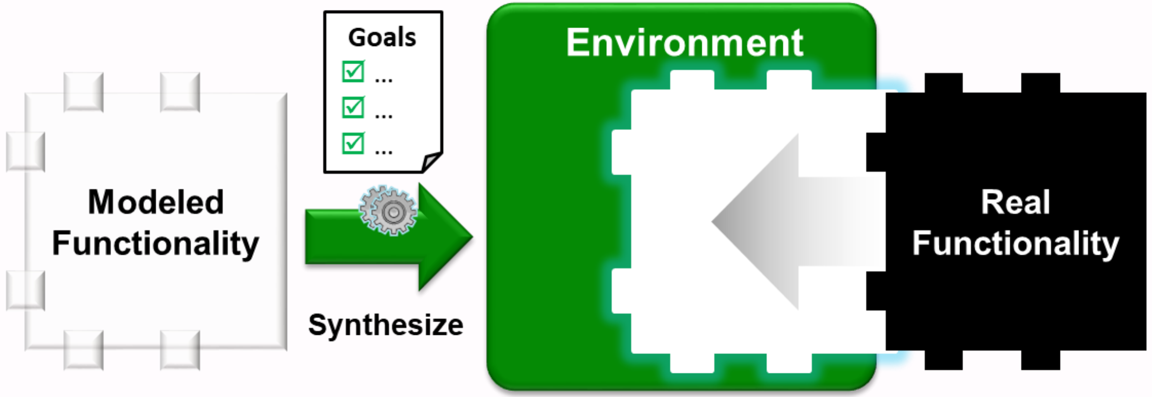
Don't model tests; model the functionality to be tested!

When you model the functionality to be tested you can create a model capable of representing every test for that functionality. It is a much smaller task, since the number of all possible tests is very large and change behind your back, which means test models will become obsolete the moment they are made.

(adopted from Monica Anderson "Don't model the World; Model the Mind")

- Specify functionality to be tested and generate optimal test coverage
- Generally graphical representation of control and data aspects
- Basically full script & documentation generation from generated tests
- Now manual testers can effectively contribute to test automation

About Model-Based Testing (MBT)



- Umbrella term for using models in a testing context; only one approach is to use MBT for *automating test design*
  - MBT *complements* test execution
  - Recognized by worldwide industrial standards (ETSI, ISTQB)

6th UCAAT User Conference on Advanced Automated Testing

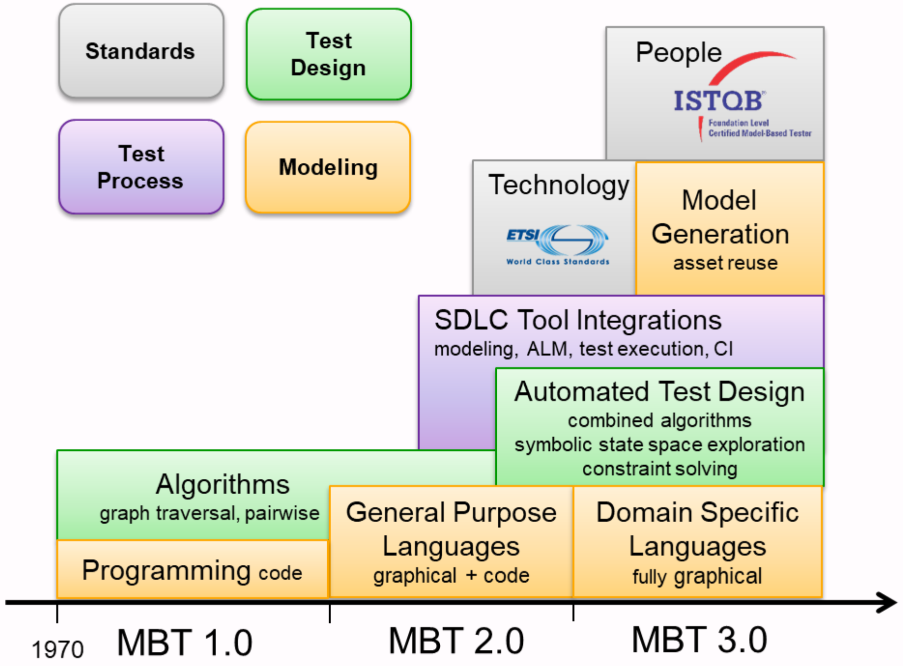
ETSI

Paris, 16-18 October 2018
Organizer: TESTING SOLUTIONS &SERVICES

The Evolution of Model-Based Testing

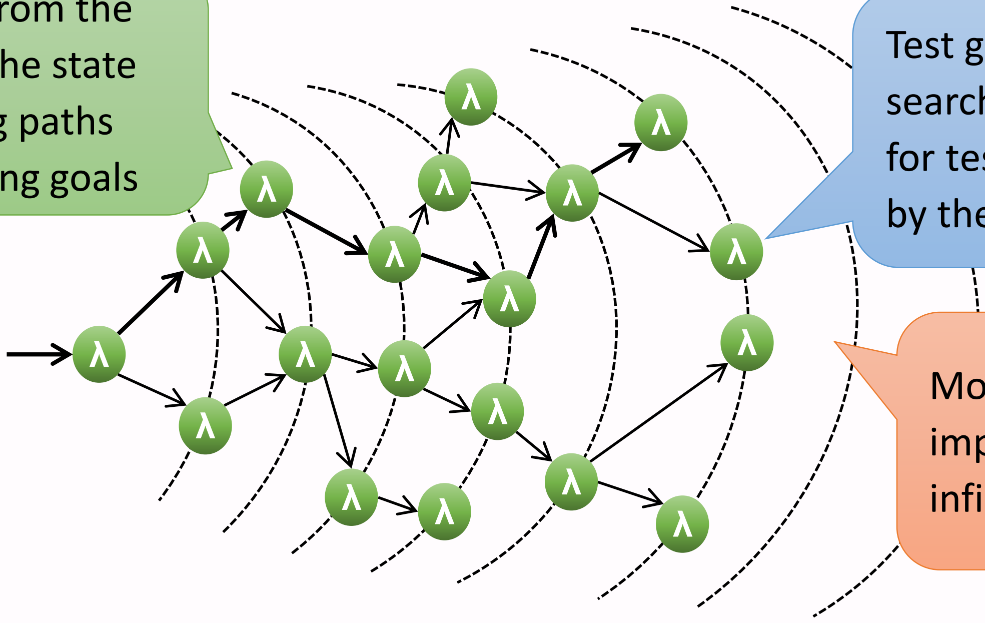
CONFORMIQ

Evolution of Model Based Testing



Automated Test Design: Planning & Searching

Constructs tests from the explored part of the state space by selecting paths that leads to testing goals

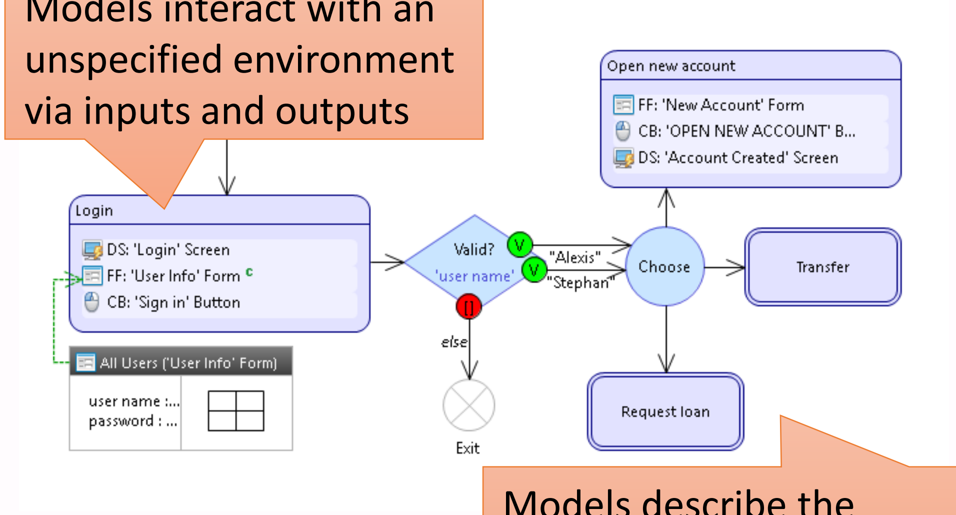


Test generation algorithm searches the state space for testing goals, defined by the end user

Model behavior implies typically infinite state space

Automated Test Design: Reasoning & Problem Solving

Models interact with an unspecified environment via inputs and outputs



Models describe the *expected external behavior* of the system under test

- Test generation reasons about unspecified input data and derives a concrete data based on modeled logic
- Generally, constraint solving theory is applied for computing unspecified input data

Automated Test Design: Finding Important Tests

- Predict the ways the system may fail
- Combine large set of heuristics
  - Equivalence classes
  - Boundary value analysis
  - Combinatorial testing
  - Mutation testing
  - And more…
- Know what do <u>not</u> cover!

MBT 1.0: Modeling Notation is Adoption Hurdle

B

```
MACHINE purse
SETS BOOLEAN = {true, false}
CONSTANTS max tries
PROPERTIES max tries ∈ N ∧ max tries = 3
VARIABLES balance, pin, tries, auth
INVARIANT
  balance ∈ N ∧ balance ≥ 0 ∧ pin ∈ -1..9999 ∧ tries ∈ 0
INITIALIZATION balance := 0 k pin := -1 k tries := max t
OPERATIONS
  sw ← SET PIN(p) ^= ...
  sw ← VERIFY PIN(p) ^= ...
  sw ← CREDIT(a) ^= ...
  sw ← DEBIT(a) ^= ...
END
```

lisp

```
(define quotient
  (lambda (a b)
    (let ((c (/ a b)))
      (if (> c 0)
          (floor c)
          (ceiling c)))))

(define remainder
  (lambda (a b)
    (- a (* (quotient a b) b))))
```

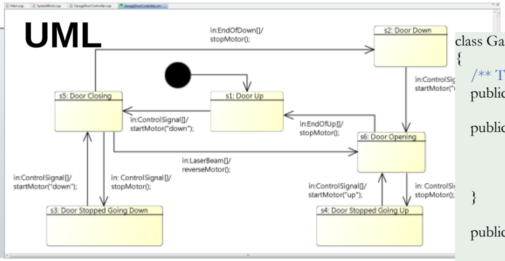
From
functional programming
(10+ years ago) …

… via some graphical & a lot of general purpose programming (~7 yrs ago) …

C#

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Linq;
using Microsoft.Modeling;

namespace Example1 {

static class AccumulatorModelProgram {
    static int accumulator;

    [Rule(Action = "Add(x)")]
    static void AddRule(int x) {
        Condition.IsTrue(x > 0);  accumulator += x;
    }

    [Rule(Action = "ReadAndR
    static int ReadAndResetRu
        Condition.IsTrue(accum
        int oldValue = accumula
        return oldValue;
    }
  }
}
```

UML


in:EndOfDown[]/ stopMotor();
s2: Door Down
in:ControlSig startMotor("
s5: Door Closing
s1: Door Up
in:ControlSignal[]/ startMotor("down");
in:EndOfUp[]/ stopMotor();
s6: Door Opening
in:LaserBeam[]/ reverseMotor();
in:ControlSignal[]/ startMotor("down");
in: ControlSignal[]/ stopMotor();
in:ControlSignal[]/ startMotor("up");
in: ControlSi stopMotor();
s3: Door Stopped Going Down
s4: Door Stopped Going Up

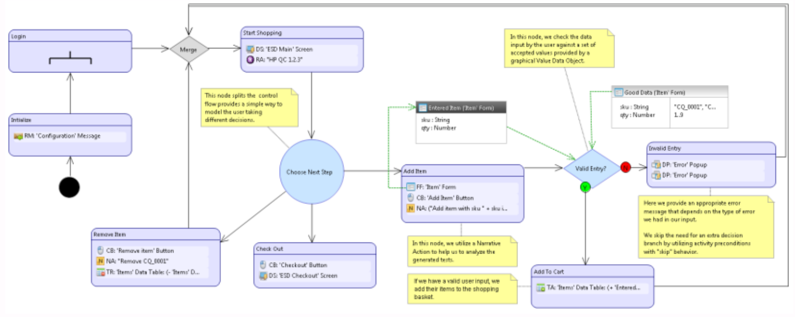
OCL

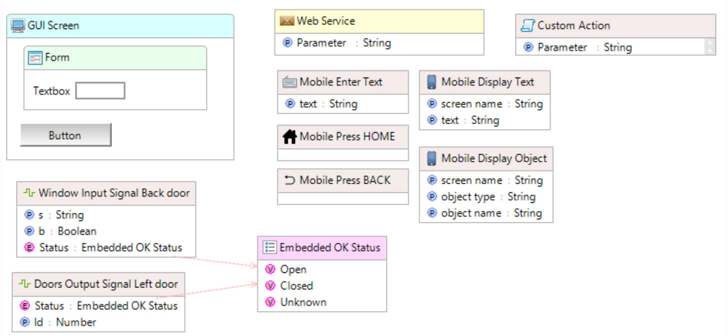
```
if (p_days = DAYSREQUESTED::NEGATIVE or p_days = DAYSREQUESTED::NONE) then
   ---@AIM: days input error
   self.mess = MSG::INVALIDDAYDATA and
   self.days = DAYSREQUESTED::NONE
else
   ---@AIM: correct form content
   self.mess = MSG::NONE and
   self.days = p_days
endif
```

Java

```
class GarageDoorController extends StateMachine
{
  /** The default constructor. */
  public GarageDoorController() { }

  public void startMotor( String direction ) {
          MotorStart start;
          start.direction = direction;
          out.send( start );
  }

  public void stopMotor() {
          MotorStop stop;
          out.send( stop );
  }

  public void reverseMotor() {
          MotorReverse reverse;
          out.send( reverse );
  }
}
```

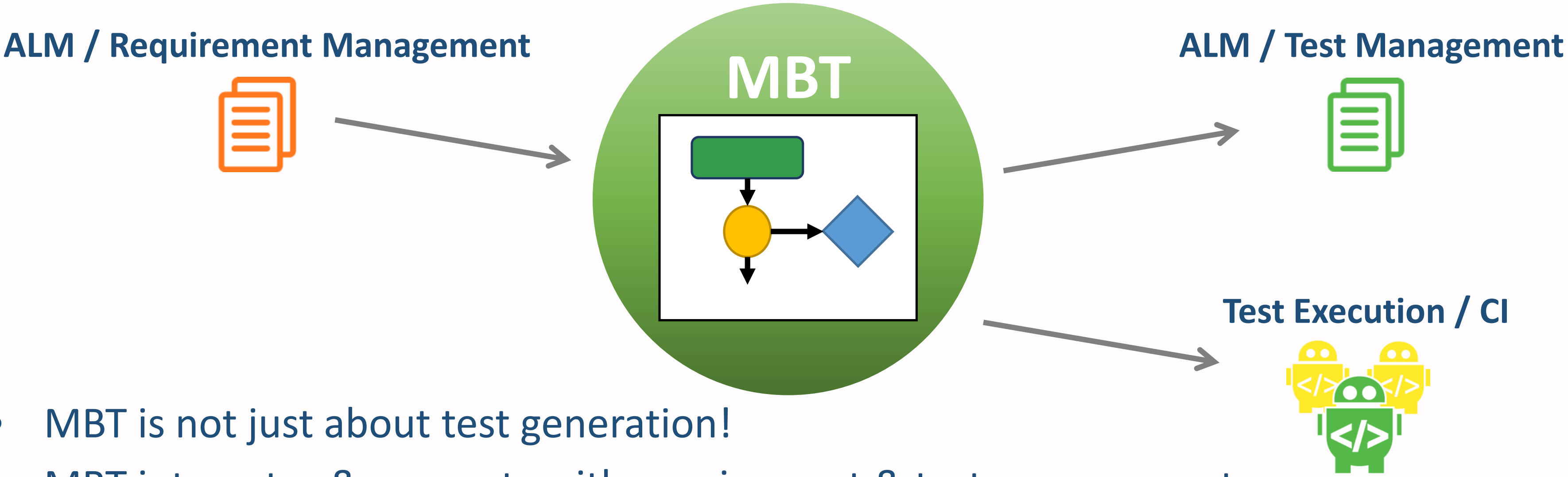
Today We Use Domain Specific Modeling Languages


Login
Merge
Start Shopping
DS: 'ESD Main' Screen
RA: "HP QC 1.2.3"
In this node, we check the data input by the user against a set of accepted values provided by a graphical Value Data Object.
Initialize
RM: 'Configuration' Message
This node splits the control flow provides a simple way to model the user taking different decisions.
Entered Item ('Item' Form)
sku : String
qty : Number
Good Data ('Item' Form)
sku : String
qty : Number
"CQ_0001", "C...
1.9
Choose Next Step
Add Item
FF: 'Item' Form
CB: 'Add Item' Button
NA: ("Add item with sku " + sku i...
Valid Entry?
N
Y
Invalid Entry
DP: 'Error' Popup
DP: 'Error' Popup
Here we provide an appropriate error message that depends on the type of error we had in our input.
We skip the need for an extra decision branch by utilizing activity preconditions with "skip" behavior.
Remove Item
CB: 'Remove item' Button
NA: "Remove CQ_0001"
TR: 'Items' Data Table: (- 'Items' D...
Check Out
CB: 'Checkout' Button
DS: 'ESD Checkout' Screen
In this node, we utilize a Narrative Action to help us to analyze the generated tests.
If we have a valid user input, we add their items to the shopping basket.
Add To Cart
TA: 'Items' Data Table: (+ 'Entered...



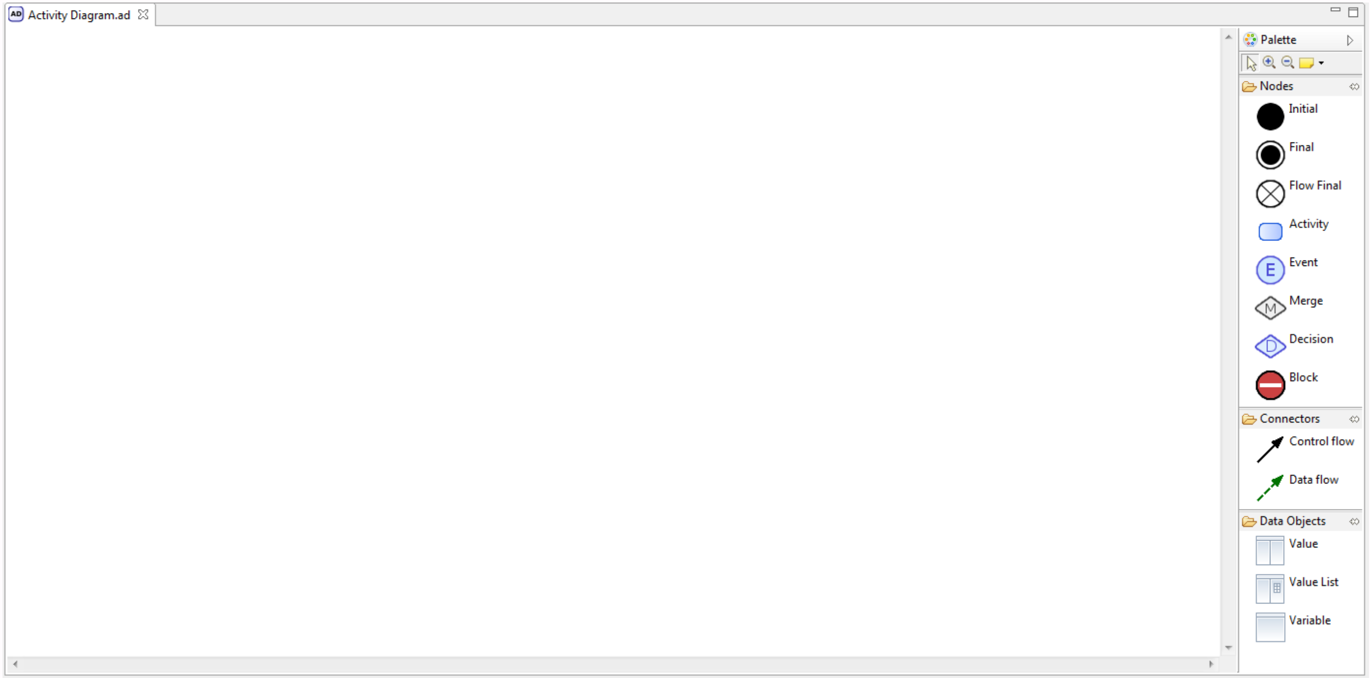
- Designed specifically to use <u>tester</u>'s language, concepts & terminology
- Easy to use: drag & drop building blocks to compose functionality
- Enabling direct automation
- Flexible and extensible
- See also <u>UCAAT 2013 tutorial</u>

MBT 2.0: Enabling Test Process Automation

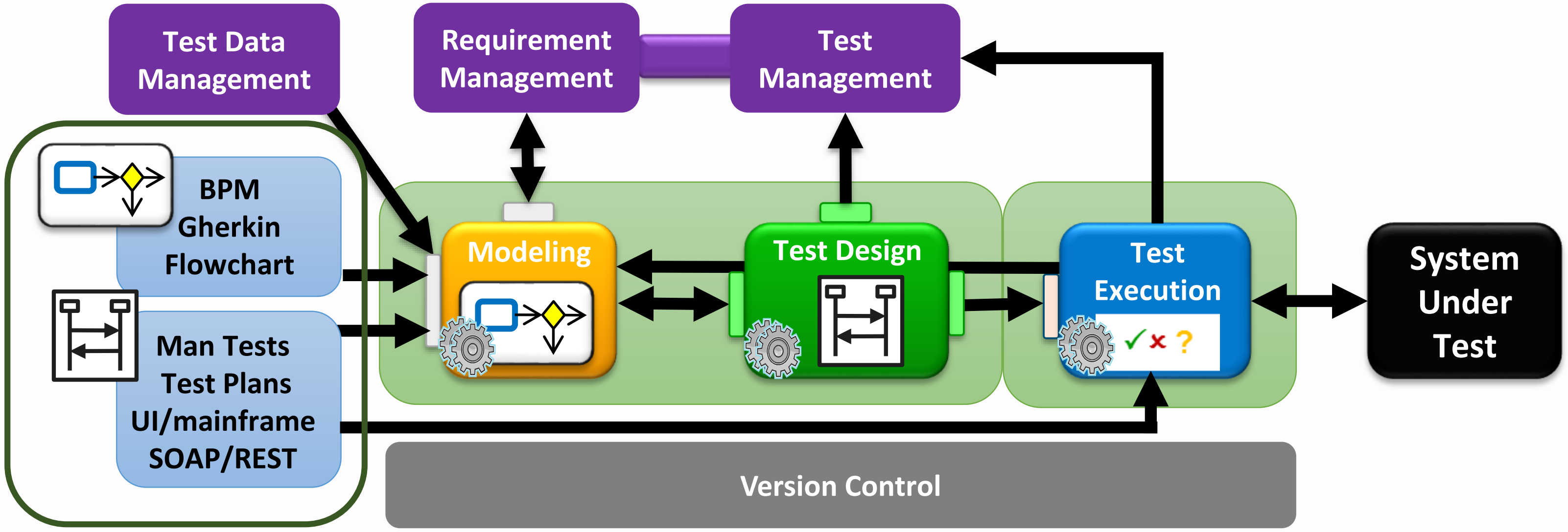


- MBT is not just about test generation!
- MBT integrates & connects with requirement & test management, versioning control, and test automation/CI integration

MBT 2.0: Still Every New Project Starts Like This …

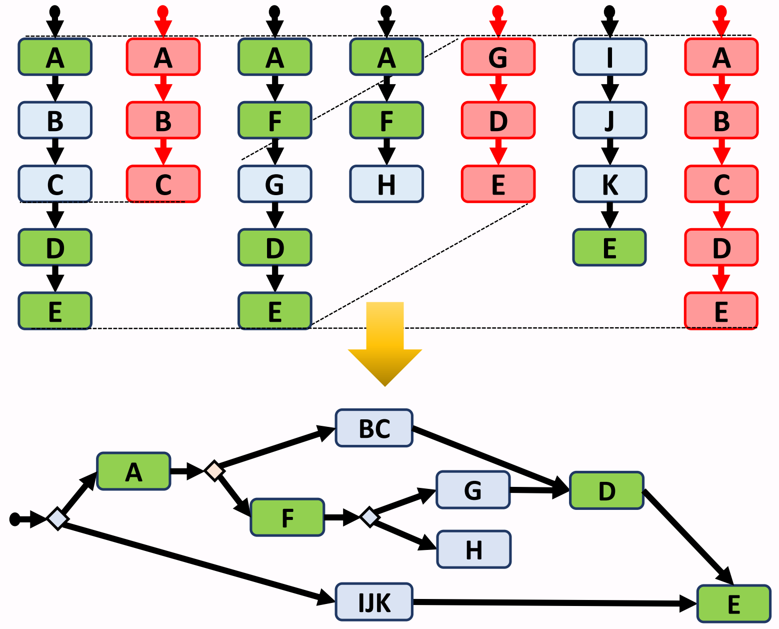


MBT 3.0: “Eliminating” Modeling



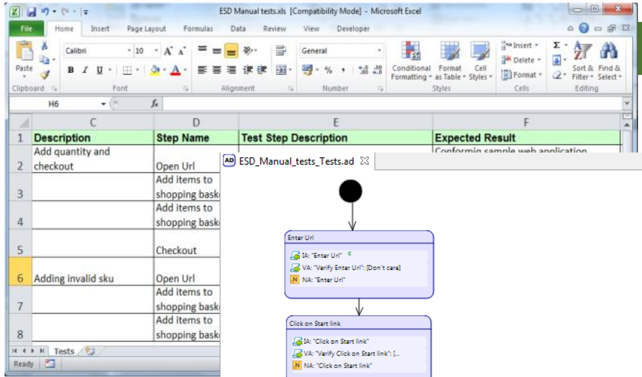
Complete End to End Test Process Automation with MBT

Test Reverse Engineering



- Readers normalize test input format
- Keywords are identified from test steps using Natural Language Processing (NLP)
- Redundant tests and subtests are eliminated
- Tests are converted into flows and merged

[ALM] Test Plans: Excel as Input to MBT


ESD Manual tests.xls [Compatibility Mode] - Microsoft Excel

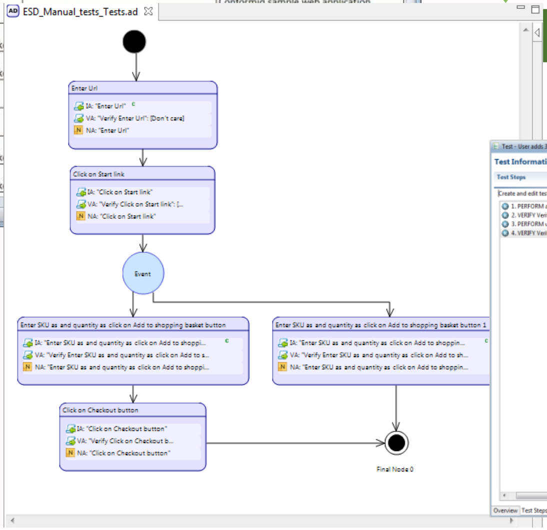
H6

	C	D	E	F
1	Description	Step Name	Test Step Description	Expected Result
2	Add quantity and checkout	Open Url		Conformiq sample web application
3		Add items to shopping bask		
4		Add items to shopping bask		
5		Checkout		
6	Adding invalid sku	Open Url		
7		Add items to shopping bask		
8		Add items to shopping bask		

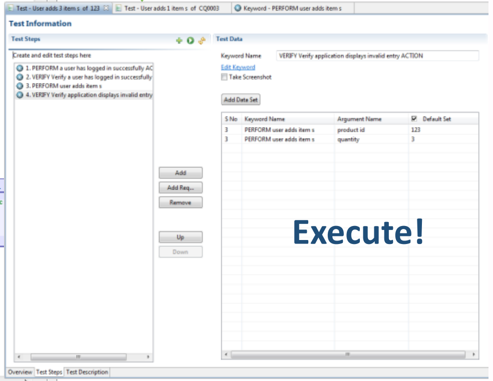
Tests

Test suite optimization during import:
- Reduction in Tests: 0%
- Reduction in Input Actions/Test Steps: 43%
- Reduction in Verification Actions/Expected Results: 29%
- Reduction in Total Generated Keywords: 36%

Import & Refine


ESD_Manual_tests_Tests.ad
Enter Url
IA: "Enter Url"
VA: "Verify Enter Url": [Don't care]
NA: "Enter Url"
Click on Start link
IA: "Click on Start link"
VA: "Verify Click on Start link": [...
NA: "Click on Start link"
Event
Enter SKU as and quantity as click on Add to shopping basket button
IA: "Enter SKU as and quantity as click on Add to shoppi...
VA: "Verify Enter SKU as and quantity as click on Add to s...
NA: "Enter SKU as and quantity as click on Add to shoppi...
Enter SKU as and quantity as click on Add to shopping basket button 1
IA: "Enter SKU as and quantity as click on Add to shoppin...
VA: "Verify Enter SKU as and quantity as click on Add to sh...
NA: "Enter SKU as and quantity as click on Add to shoppin...
Click on Checkout button
IA: "Click on Checkout button"
VA: "Verify Click on Checkout b...
NA: "Click on Checkout button"
Final Node 0

Generate Tests


Test - User adds 3 item s of 123 | Test - User adds 1 item s of CQ0003 | Keyword - PERFORM user adds item s

Test Information

Test Steps

Create and edit test steps here

1. PERFORM a user has logged in successfully AC
2. VERIFY Verify a user has logged in successfully
3. PERFORM user adds item s
4. VERIFY Verify application displays invalid entry

Test Data

Keyword Name: VERIFY Verify application displays invalid entry ACTION

Edit Keyword

Take Screenshot

Add Data Set

S No	Keyword Name	Argument Name	Default Set
3	PERFORM user adds item s	product id	123
3	PERFORM user adds item s	quantity	3

Execute!

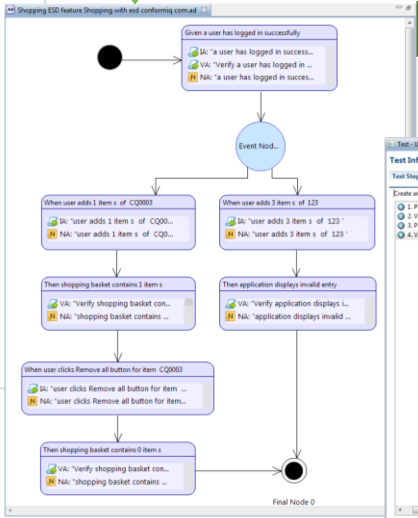
“BDD”: Gherkin as Input to MBT

```
Feature: Shopping with esd.conformiq.com

Scenario: Successful shopping

Given a user has logged in successfully
When user adds 1 item(s) of "CQ0003"
Then shopping basket contains 1 item(s)

Scenario: Bad product id

Given a user has logged in successfully
When user adds 3 item(s) of "123"
Then application displays invalid entry

Scenario: Remove item

Given a user has logged in successfully
When user adds 1 item(s) of "CQ0003"
Then shopping basket contains 1 item(s)
When user clicks Remove all button for
item "CQ0003"
Then shopping basket contains 0 item(s)
```

Summary of Conformiq test suite optimization:
- Reduction in Tests: 34%
- Reduction in Input Actions/Test Steps: 29%
- Reduction in Verification Actions/Expected Results: 43%
- Reduction in Total Generated Keywords: 36%

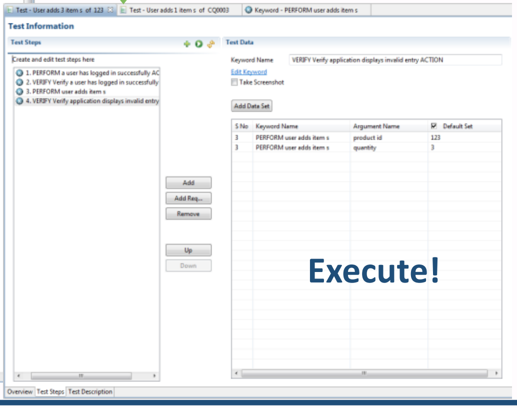
See also separate HUSTEF pres / video

Import & Refine



Shopping ESD feature Shopping with esd conformiq com.ad
Given a user has logged in successfully
IA: "a user has logged in success...
VA: "Verify a user has logged in ...
NA: "a user has logged in succes...
Event Nod...
When user adds 1 item s of CQ0003
IA: "user adds 1 item s of CQ00...
NA: "user adds 1 item s of CQ0...
When user adds 3 item s of 123
IA: "user adds 3 item s of 123 '
NA: "user adds 3 item s of 123 '
Then shopping basket contains 1 item s
VA: "Verify shopping basket con...
NA: "shopping basket contains ...
Then application displays invalid entry
VA: "Verify application displays i...
NA: "application displays invalid ...
When user clicks Remove all button for item CQ0003
IA: "user clicks Remove all button for item ...
NA: "user clicks Remove all button for item...
Then shopping basket contains 0 item s
VA: "Verify shopping basket con...
NA: "shopping basket contains ...
Final Node 0

Generate Tests



Create and edit test steps here

1. PERFORM a user has logged in successfully AC
2. VERIFY Verify a user has logged in successfully
3. PERFORM user adds item s
4. VERIFY Verify application displays invalid entry

Keyword Name: VERIFY Verify application displays invalid entry ACTION

Edit Keyword

☐ Take Screenshot

S No	Keyword Name	Argument Name	☑ Default Set
3	PERFORM user adds item s	product id	123
3	PERFORM user adds item s	quantity	3

Execute!

MBT Trends in 2017: Modeling in Prose

- Low end MBT tools are pushing very simple, prose based modeling notations into the test tool market
  - Arguably domain specific modeling still has the edge as it enables a direct path to automatic test execution (today not possible from prose)
- Users have started to perceive domain specific modeling (= creation of building blocks) as a burden when it comes to modeling
  - Model generation from test assets (reverse engineering) has addressed this issue but not removed it
- Frequent use of extension points (“custom actions”) is diluting the value proposition of domain specific languages
  - Prose and domain specific action modeling start to mix

6th UCAAT User Conference on Advanced Automated Testing

ETSI

Paris, 16-18 October 2018

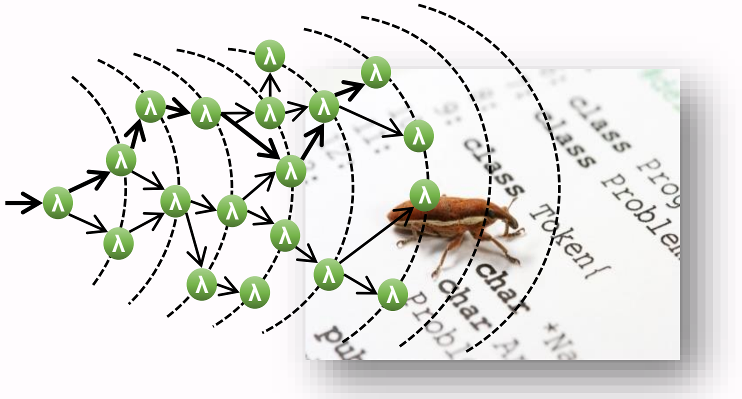
Organizer: TESTING SOLUTIONS & SERVICES

Artificial Intelligence in Testing

CONFORMIQ

AI Techniques Have Been Leveraged for (Many) Years!

- Fuzzy and Probabilistic Logic ✓
- Classification ✓
- Search & Optimization ✓
- Expert Systems ✓
- (Deep) Machine Learning (ML) ?

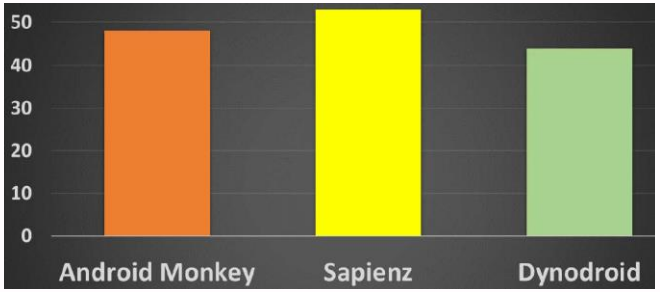


Remember this?

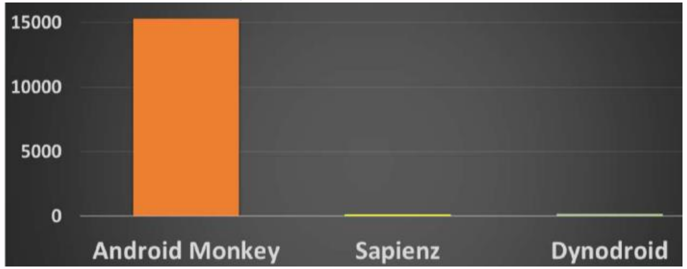
Today the term “AI” is generally reduced to (Deep) Machine Learning …

More Recent: AI in Robustness Testing

code coverage



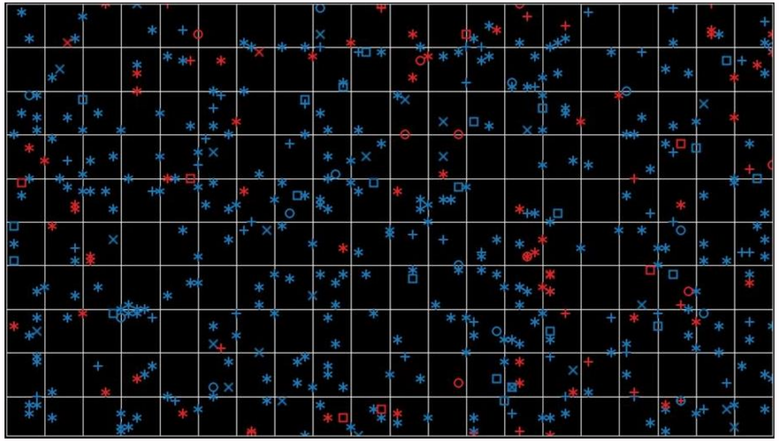
steps to reveal fault



Source: Stuart Reid, “Smarter Testing with Artificial Intelligence”, JFTL 2017

- App store testing example:
  Fuzz vs search vs man+machine
- Initially, “break the system by clicking around” (= “Monkey Testing”)
- Now also combination of online MBT with learnings from previous failures
- No human required for test execution!

More Recent: AI in Result Analysis of Large Test Suites



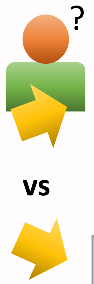
- Predictive analytics: what *could* fail?
  - “Test Less - Test Right” (M. Venkata)
  - Failure patterns, code changes, etc
- Diagnostic analytics: what is failing?
  - Data mining & visualization of defects
  - Application monitoring for anomalies
- Prescriptive analytics: what to fix?
  - Symptom vs defect
  - Root cause analysis

How Could We Use Learning in Functional Testing?

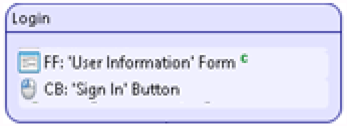
- Test automation tools have managed successfully to make creation of tests that are automatically executable more or less “dead easy”
- **BUT** … automating from “arbitrary” text still has to be done manually

Enter user name and password and click the Sign In button

Manual Test Step



vs

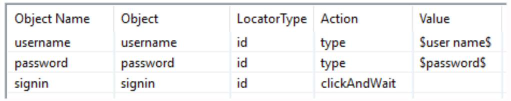

Login

FF: 'User Information' Form

CB: 'Sign In' Button

Model Action(s)

<u>Manually</u> refine into domain specific model inputs & outputs

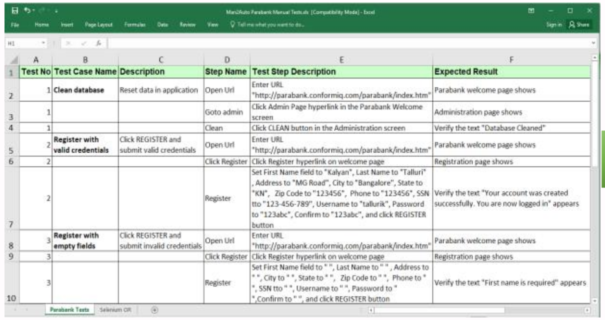


Object Name	Object	LocatorType	Action	Value
username	username	id	type	$user name$
password	password	id	type	$password$
signin	signin	id	clickAndWait	

Test Automation Keyword

<u>Manually</u> refine into concrete automation steps

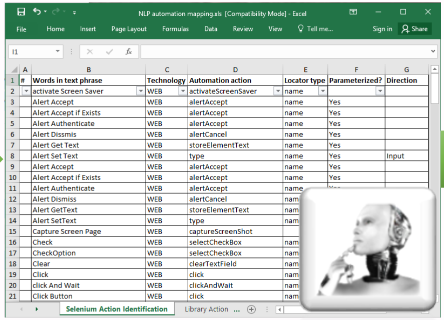
Introducing Intelligent Test Automation



Man2Auto Parabank Manual Tests.xls [Compatibility Mode] - Excel

Test No	Test Case Name	Description	Step Name	Test Step Description	Expected Result
1	**Clean database**	Reset data in application	Open Url	Enter URL "http://parabank.conformiq.com/parabank/index.htm"	Parabank welcome page shows
1			Goto admin	Click Admin Page hyperlink in the Parabank Welcome screen	Administration page shows
1			Clean	Click CLEAN button in the Administration screen	Verify the text "Database Cleaned"
2	**Register with valid credentials**	Click REGISTER and submit valid credentials	Open Url	Enter URL "http://parabank.conformiq.com/parabank/index.htm"	Parabank welcome page shows
2			Click Register	Click Register hyperlink on welcome page	Registration page shows
2			Register	Set First Name field to "Kalyan", Last Name to "Talluri" , Address to "MG Road", City to "Bangalore", State to "KN", Zip Code to "123456", Phone to "123456", SSN tto "123-456-789", Username to "tallurik", Password to "123abc", Confirm to "123abc", and click REGISTER button	Verify the text "Your account was created successfully. You are now logged in" appears
3	**Register with empty fields**	Click REGISTER and submit invalid credentials	Open Url	Enter URL "http://parabank.conformiq.com/parabank/index.htm"	Parabank welcome page shows
3			Click Register	Click Register hyperlink on welcome page	Registration page shows
3			Register	Set First Name field to " ", Last Name to " " , Address to " ", City to " ", State to " ", Zip Code to " ", Phone to " ", SSN tto " ", Username to " ", Password to " ",Confirm to " ", and click REGISTER button	Verify the text "First name is required" appears

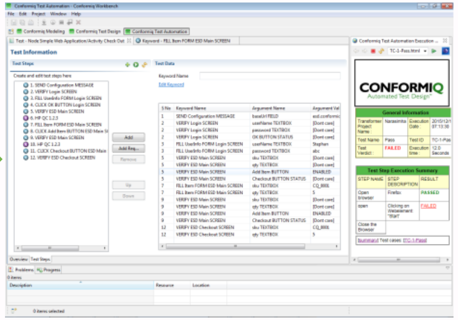
Parabank Tests | Selenium OR



NLP automation mapping.xls [Compatibility Mode] - Excel

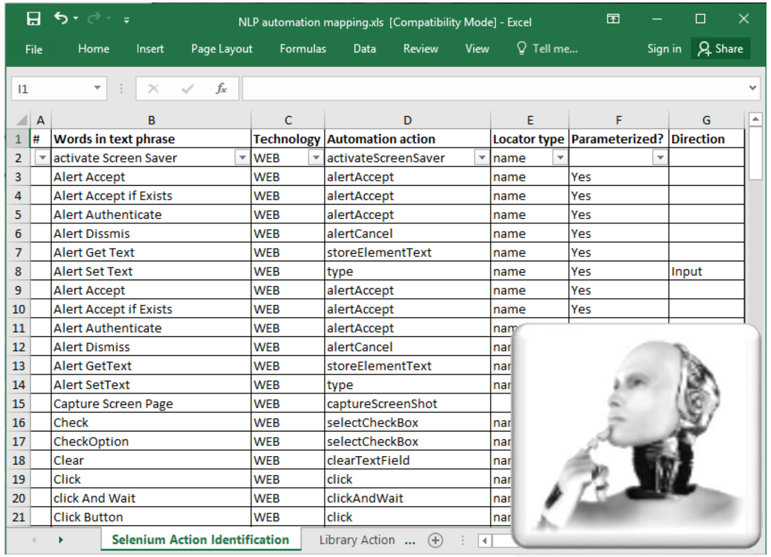
#	Words in text phrase	Technology	Automation action	Locator type	Parameterized?	Direction
	activate Screen Saver	WEB	activateScreenSaver	name		
	Alert Accept	WEB	alertAccept	name	Yes	
	Alert Accept if Exists	WEB	alertAccept	name	Yes	
	Alert Authenticate	WEB	alertAccept	name	Yes	
	Alert Dissmis	WEB	alertCancel	name	Yes	
	Alert Get Text	WEB	storeElementText	name	Yes	
	Alert Set Text	WEB	type	name	Yes	Input
	Alert Accept	WEB	alertAccept	name	Yes	
	Alert Accept if Exists	WEB	alertAccept	name	Yes	
	Alert Authenticate	WEB	alertAccept	name		
	Alert Dismiss	WEB	alertCancel	nam		
	Alert GetText	WEB	storeElementText	nam		
	Alert SetText	WEB	type	nam		
	Capture Screen Page	WEB	captureScreenShot			
	Check	WEB	selectCheckBox	nam		
	CheckOption	WEB	selectCheckBox	nam		
	Clear	WEB	clearTextField	nam		
	Click	WEB	click	nam		
	click And Wait	WEB	clickAndWait	nam		
	Click Button	WEB	click	nam		

Selenium Action Identification | Library Action ...



- From manual to automatically executable test in two steps
  1. Feed manual tests and train intelligent test automation (if any), e.g., resolve unknown text to automation rules
  2. Execute and fix unrecognized object identifiers (if any)
- The more intelligent test automation is trained - the less it needs to learn!

Intelligent Test Automation in a Nutshell



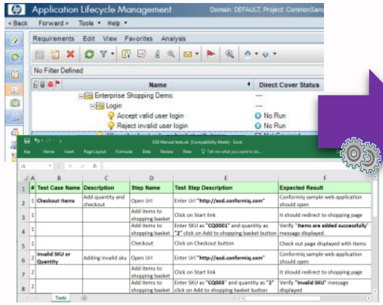
#	Words in text phrase	Technology	Automation action	Locator type	Parameterized?	Direction
	activate Screen Saver	WEB	activateScreenSaver	name		
	Alert Accept	WEB	alertAccept	name	Yes	
	Alert Accept if Exists	WEB	alertAccept	name	Yes	
	Alert Authenticate	WEB	alertAccept	name	Yes	
	Alert Dissmis	WEB	alertCancel	name	Yes	
	Alert Get Text	WEB	storeElementText	name	Yes	
	Alert Set Text	WEB	type	name	Yes	Input
	Alert Accept	WEB	alertAccept	name	Yes	
	Alert Accept if Exists	WEB	alertAccept	name	Yes	
	Alert Authenticate	WEB	alertAccept	nam		
	Alert Dismiss	WEB	alertCancel	nar		
	Alert GetText	WEB	storeElementText	nar		
	Alert SetText	WEB	type	nar		
	Capture Screen Page	WEB	captureScreenShot			
	Check	WEB	selectCheckBox	nar		
	CheckOption	WEB	selectCheckBox	nar		
	Clear	WEB	clearTextField	nar		
	Click	WEB	click	nar		
	click And Wait	WEB	clickAndWait	nar		
	Click Button	WEB	click	nar		

Note: Conformiq patents pending

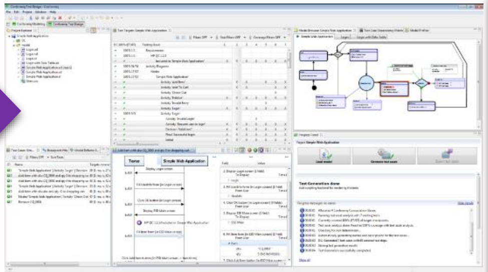
- Reader(s) normalize input format(s)
- Natural Language Processing is used to analyze test steps
- User “teaches” mappings to target test automation tool or framework
- Works with any input format, different target technologies, and any test automation framework!

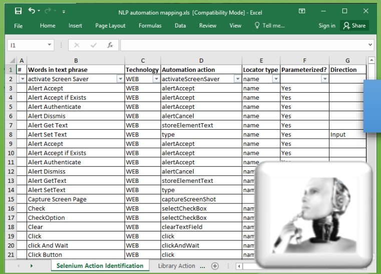
Intelligent Test Automation Naturally Extends to MBT!

Requirements	Test Design	Test Creation	Test Execution
Includes BAs & Testers	Includes optimization	Includes scripts, data & validations	Includes harnesses



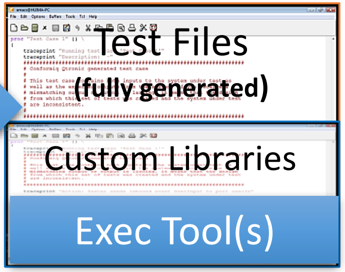
#	Test Case Name	Description	Step Name	Test Step Description	Expected Result
1	Checkout Items	Add quantity and checkout	Open Url	Enter Url "http://esd.conformiq.com"	Conformiq sample web application should open
1			Add items to shopping basket	Click on Start link	It should redirect to shopping page
1			Add items to shopping basket	Enter SKU as "CQ0001" and quantity as "2" click on Add to shopping basket button	Verify "Items are added successfully" message displayed
1			Checkout	Click on Checkout button	Check out page displayed with Items
2	Invalid SKU or Quantity	Adding invalid sku	Open Url	Enter Url "http://esd.conformiq.com"	Conformiq sample web application should open
2			Add items to shopping basket	Click on Start link	It should redirect to shopping page
2			Add items to shopping basket	Enter SKU as "CQ008" and quantity as "2" click on Add to shopping basket button	Verify "Invalid SKU" message displayed



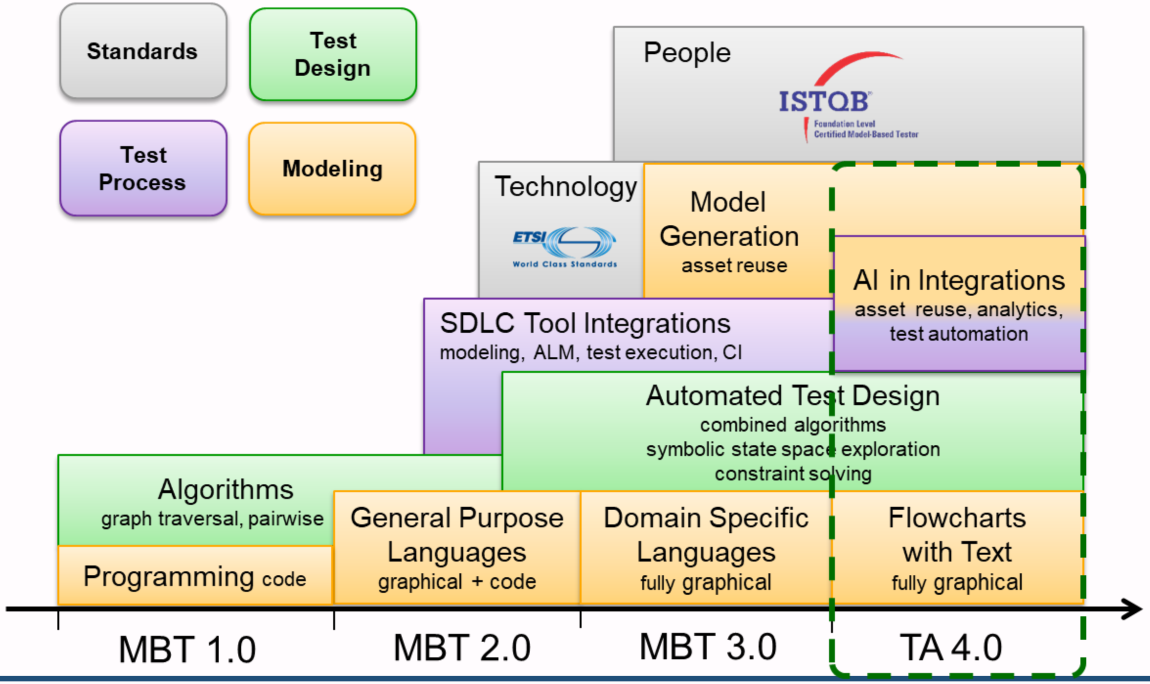

NLP automation mapping.xls [Compatibility Mode] - Excel
Selenium Action Identification
Library Action

#	Words in text phrase	Technology	Automation action	Locator type	Parameterized?	Direction
	activate Screen Saver	WEB	activateScreenSaver	name		
	Alert Accept	WEB	alertAccept	name	Yes	
	Alert Accept if Exists	WEB	alertAccept	name	Yes	
	Alert Authenticate	WEB	alertAccept	name	Yes	
	Alert Dissmis	WEB	alertCancel	name	Yes	
	Alert Get Text	WEB	storeElementText	name	Yes	
	Alert Set Text	WEB	type	name	Yes	Input
	Alert Accept	WEB	alertAccept	name	Yes	
	Alert Accept if Exists	WEB	alertAccept	name	Yes	
	Alert Authenticate	WEB	alertAccept	name		
	Alert Dismiss	WEB	alertCancel	nam		
	Alert GetText	WEB	storeElementText	nam		
	Alert SetText	WEB	type	nam		
	Capture Screen Page	WEB	captureScreenShot			
	Check	WEB	selectCheckBox	nam		
	CheckOption	WEB	selectCheckBox	nam		
	Clear	WEB	clearTextField	nam		
	Click	WEB	click	nam		
	click And Wait	WEB	clickAndWait	nam		
	Click Button	WEB	click	nam		

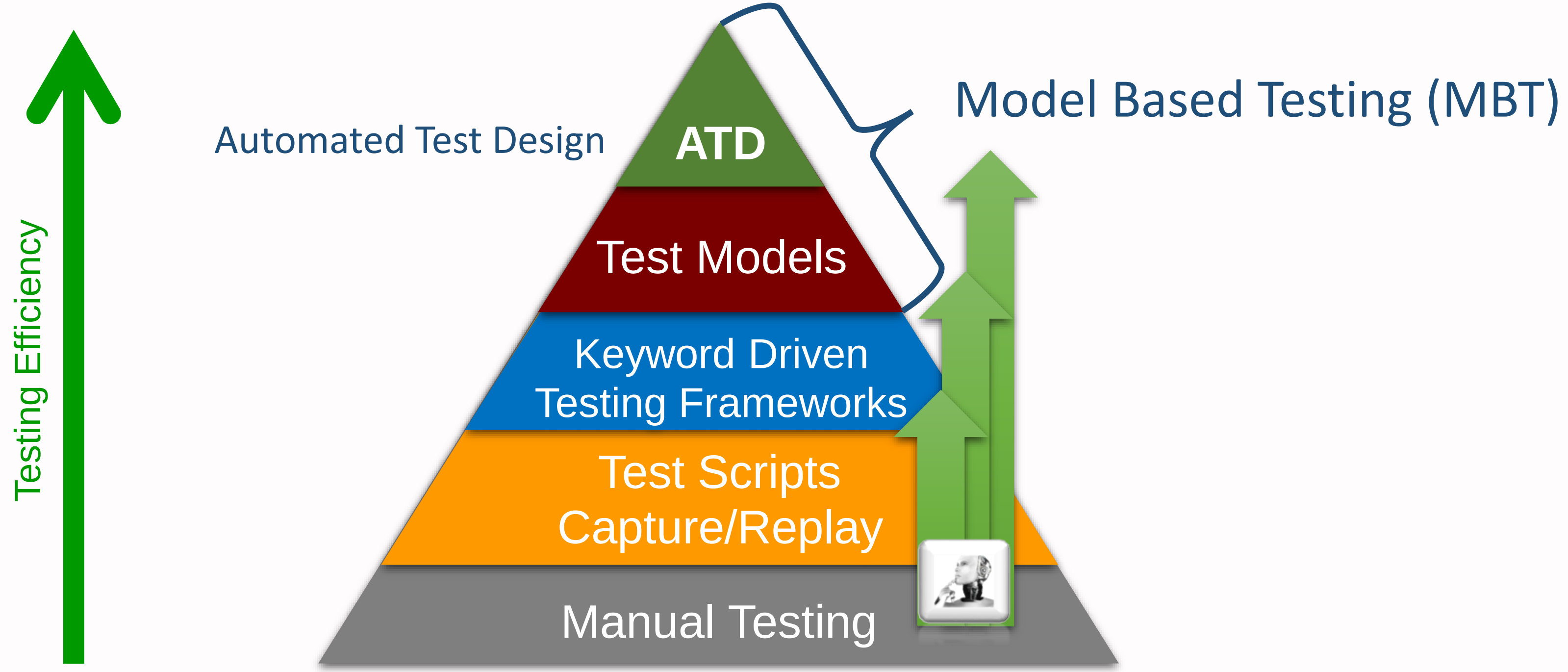
Intelligent Test Automation



The Start of a New Generation in Test Automation?


Standards
Test Design
Test Process
Modeling
People
ISTQB Foundation Level Certified Model-Based Tester
Technology
ETSI World Class Standards
Model Generation asset reuse
AI in Integrations asset reuse, analytics, test automation
SDLC Tool Integrations modeling, ALM, test execution, CI
Automated Test Design combined algorithms symbolic state space exploration constraint solving
Algorithms graph traversal, pairwise
Programming code
General Purpose Languages graphical + code
Domain Specific Languages fully graphical
Flowcharts with Text fully graphical
MBT 1.0
MBT 2.0
MBT 3.0
TA 4.0

Evolution of Software Testing


Testing Efficiency
Automated Test Design
ATD
Test Models
Keyword Driven Testing Frameworks
Test Scripts Capture/Replay
Manual Testing
Model Based Testing (MBT)

Thank You for Your Attention!

Q&A

stephan.schulz@conformiq.com