

# Future proof Test Automation Architecture based

Jos van Rooyen



- Employed at Identify as partner / principal consultant
- 30 years in software testing & quality management
- Co-author TestGrip, TestFrame, Project de Baas, Quality Supervision, Textbook; “Aan de slag met software testen”, Cleantxt, Test Automation Architecture (available soon)
- Test expert online magazine Computable
- Publication areas; Test process Improvement, BI-testing, Test automation, Test Education, Risk Based Testing, Quality Supervision
- Member NESMA working party; Metrics in Contracts
- Visiting lecturer Universities of Applied Science
- Member advisory board Hogeschool Utrecht
- Member of several working parties Dutch Testing Society
  - Member of the board
  - Test Education universities of applied science



# What do we do as Identify?

- Testing (modern tester)
- Testing & Commissioning
- Test management
- Chain management/quality management
- Agile services:
  - Scrum masters
  - Product owners
- Consulting
- Training
- Software as a service



# 1. How do we do it?

- Quality
- Involvement
- Integrity
- Freedom
- Joy



Steady as a rock and a beacon to navigate on

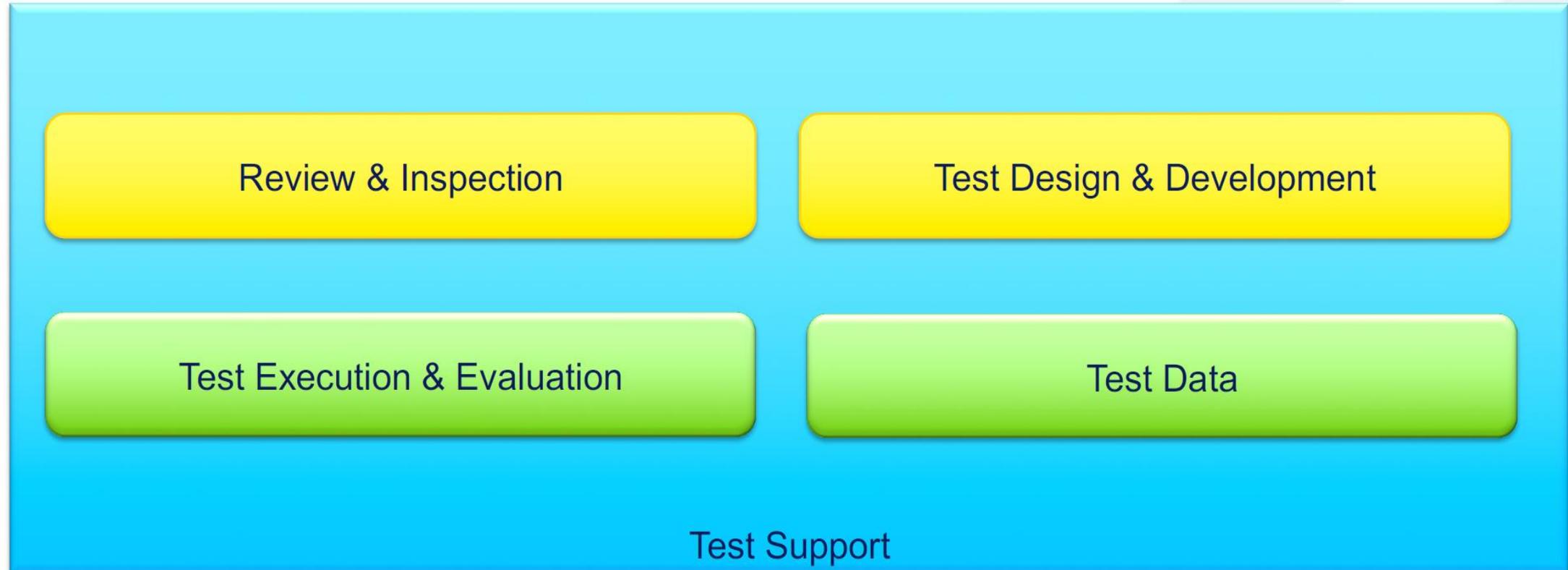


# Agenda

- Different types of test automation
- Current situation of applying test automation
- Pitfalls of test automation
- The solution:
  - Approach
  - Temple
  - Principles
- The benefits
- Assessment model
- Future work



# Test automation



# Test automation tools

Test tool category	Description
Review & Inspection	Metric tool. Analysis of a software program without executing it.
Test design & development	Development of test scripts. Determination of no. of testscripts
Test Execution & Evaluation	Automated execution of test scripts for several test types and evaluation of the results
Test data	Defining and creation of test data

# Test automation tools examples

Test tool category	Examples
Review & Inspection	Idra testbed, Cosmos
Test design & development	Test vector generation system, Gherkin, Specflow, CA Agile Requirement Designer
Test Execution & Evaluation	Selenium, MTM, Robot Framework, Windiff, Neoload
Test data	Datprof, CA Test Data Management

# Present situation

Many test automation projects fail or do not produce what was expected. On the basis of our years long experience we developed a concept that must help organizations to do successful test automating projects



# Some pitfalls of test automation

Tool driven approach

Lack of technical skills

No vision in the organisation

Huge effort for running test automation

Local initiatives

A lack of learning cycle

# Experiences concerning test automation

- Honeymoon phase
- Impossible to maintain
- Increased deficit per sprint
- Compare it with a marathon. You must keep going and make miles.
- No re-use of knowledge and skills
- No vision on test automation
- No mutual coordination
- Failure to meet commitments
- Little or less structure
- Everybody just picks up a tool
- Shelf ware

## Interpretation of the problem

How to organize test automation in such a way that it is applicable not only today but also tomorrow with as less maintenance as possible

# The solution; Future Proof Test Automation

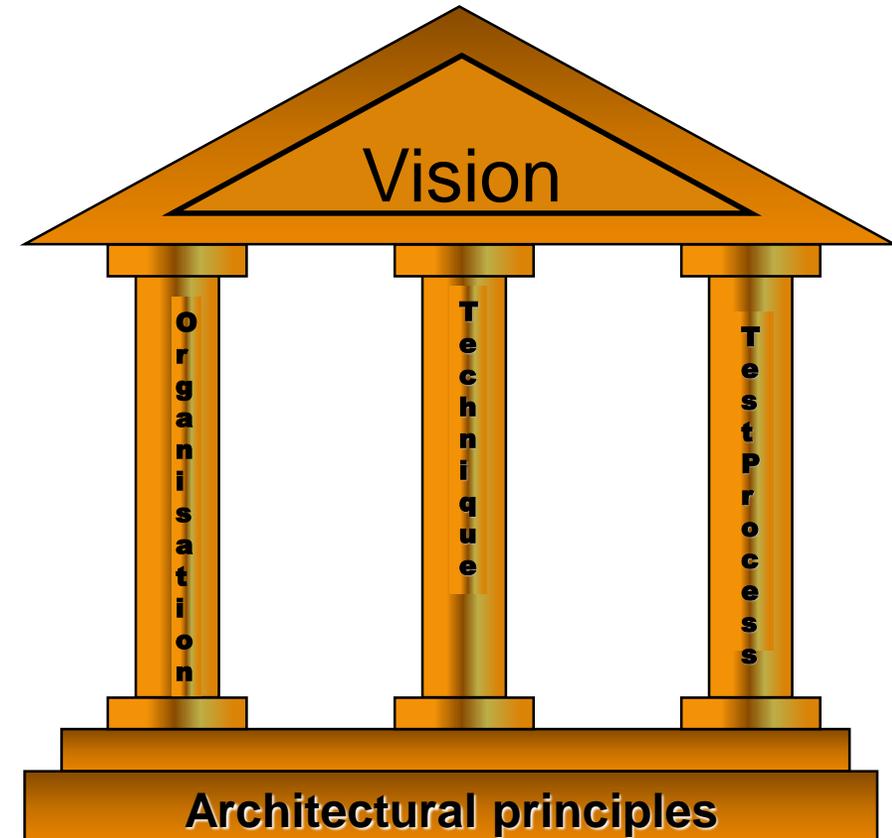
Future proof test automation take care of the structural embedding of test automation into the organisation in such a way that processes are adapted (SDLC/Vendor management), test engineers with the required skills are available, necessary tools are available and test automation is secured on the right level in the organisation.

This way, developed test scripts are reusable, repeatable and transferable between projects and people inside domains, with as less maintenance as possible.

All developed material is available inside a central knowledge base.

# The approach

- The approach is based on a clear vision; what does the organisation want to achieve
- Based on 3 pillars:
  - Organisation
  - Technique
  - Test process
- Fundament based on architectural principles



# Vision

Which business goals must be supported by hand of test automation, such as:

- Increase quality-to-market
- Increase time-to-market
- Automation of the devops pipeline
- Decrease the dependency of scarcity of business knowledge
- Less dependend on certain resources

What are the test driven goals that must be fulfilled by hand of the implementation of test automation, such as:

- One framework for the total organisation
- 90% of manual test execution must be automated

# The pillars

- Organisation:
  - How to organise test automation. On which level into the organisation
  - Sponsorship by management
  - Defining roles, tasks and responsibilities regarding test automation
  - New and adapted roles
- Technique:
  - Type of tools available
  - Skills of test automation engineers
  - Reusable, repeatable and maintainable testscripts
  - Test data management

# The pillars

## Test process:

- Adaptation of the software development life cycle
- New processes like automating test scripts, structuring of test data and availability of test environment (roll-back, tool implementation)

# The 8 Architecture principles

## Organisation principles:

- Test automation agrees with the goals and maturity of the organisation
- Test automation is based on a clear vision, strategy and architecture
- Test automation keeps in mind the human dimension
- Test automation demands a well-considered choice between risk and effort

## Information principles:

- Test automation is model-based
- Data for test automation is explicitly managed
- Information protection is explicitly taken into account with test automation
- Test automation tools are needed but not leading

# Organization principle 1



Test automation agrees with the goals and the maturity of the organization

- Organization goals
- Maturity



# Organization principle 2



Test automation is based on a clear vision, strategy and architecture

- Strategic
- Tactical
- Operational

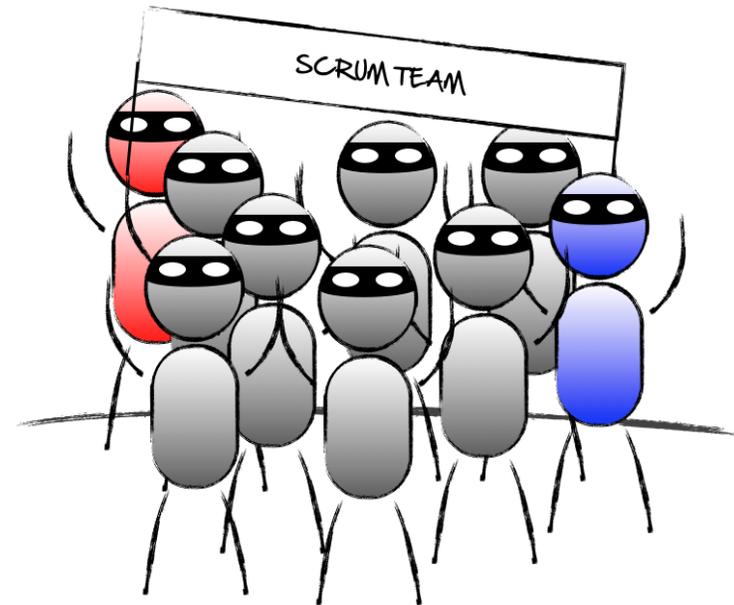


# Organization principle 3



Test automation keeps in mind the human dimension

People superfluous?



# Organization principle 4



Test automation demands a well considered choice between risk and effort

- Test Automation **takes** time
- Test Automation **saves** time
  - *RTAA (Risk Test Automation Assessment)*
  - *TAC (Test Automation Calculator)*

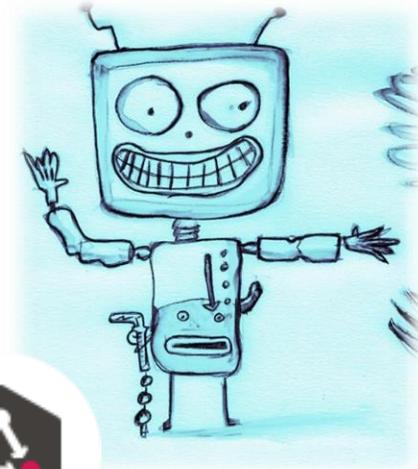


# Information provision principle 1



## Test automation is model based

- Future oriented:
  - Automated generation of test script
  - Automated analysis of test results
  - Less maintenance costs

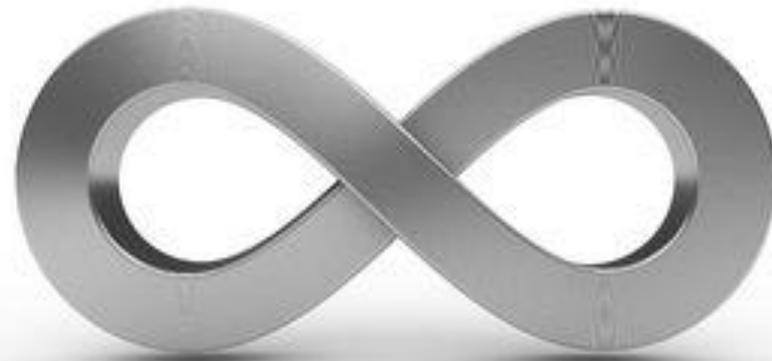


# Information provision principle 2



Data for test automation is explicitly managed

- Data Governance
- Repeatability



# Information provision principle 3



Information protection is explicitly taken into account with test automation

- (Personal) Data Protection Act
- Outsourcing



# Information provision principle 4



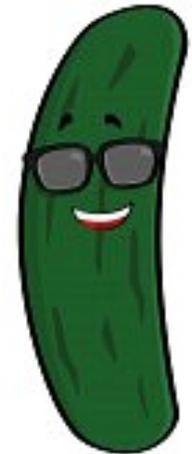
Test automation tools are needed but not leading

Fitting

- Organization
- System

“You get what you pay for” or, “Expensive is better”, is not always the case

Careful selection process (is a project)



# Implementation of Test Automation Architecture

- How to start?
- Two way approach:
  - Risk Test Automation Assessment (RTAA)
  - Is there a business case?



# RTAA, *Example*



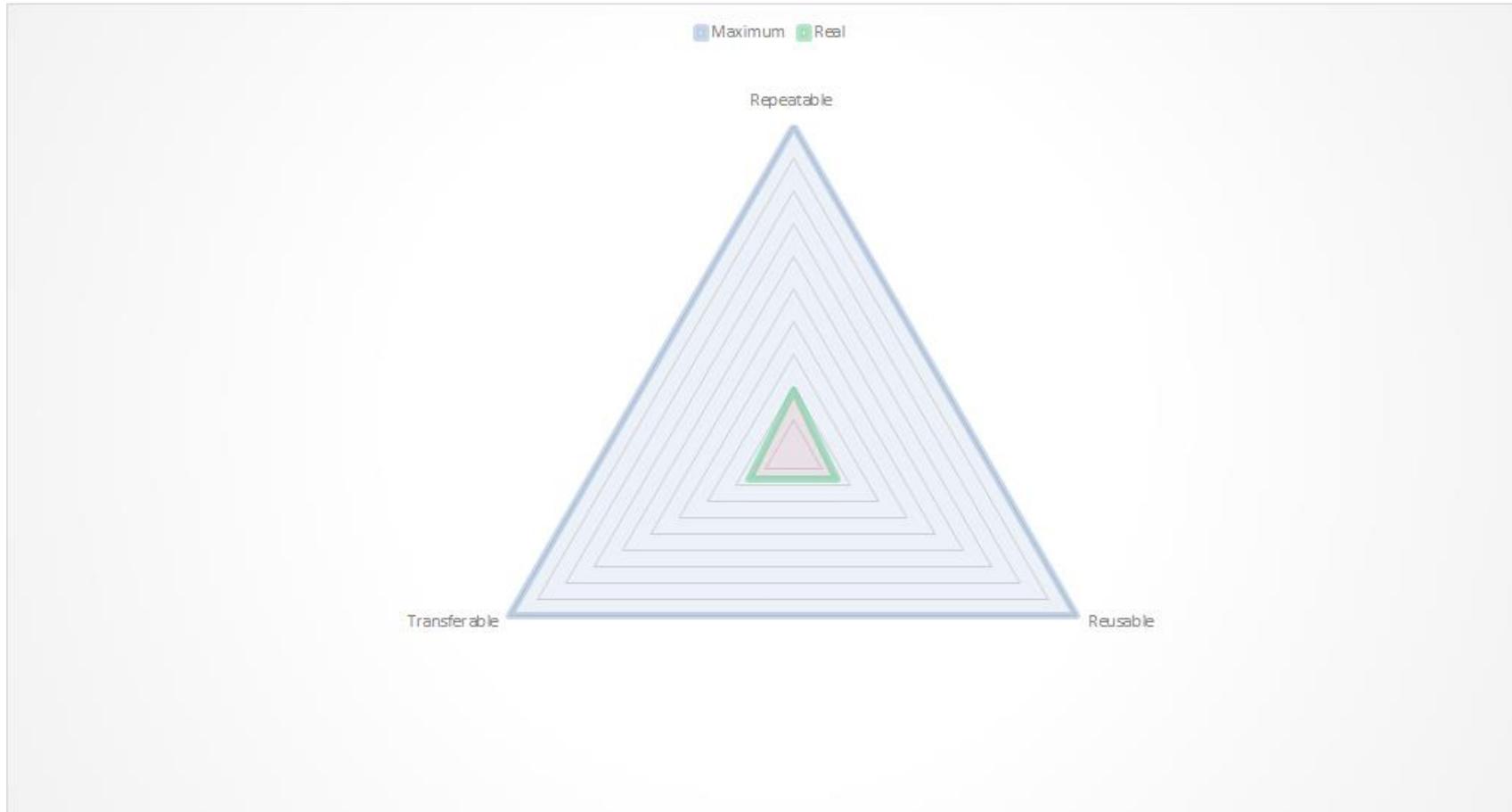
Are there employees with experience in test automation?
Do developers see added value in test automation?
Are employees consciously working with test automation?
Is a sufficient number of employees available with knowledge of, and experience in, test automation?
Do employees have sufficient knowledge and experience regarding test automation?
Are employees motivated to start with test automation?



# The model

Maturity assessment test automation		<i>[company name]</i>			
Carried out on		<i>[date]</i>			
	Level 1	Level 2	Level 3	Niveau 4	
People					
Organisation					
Process					
Data					
Technology					

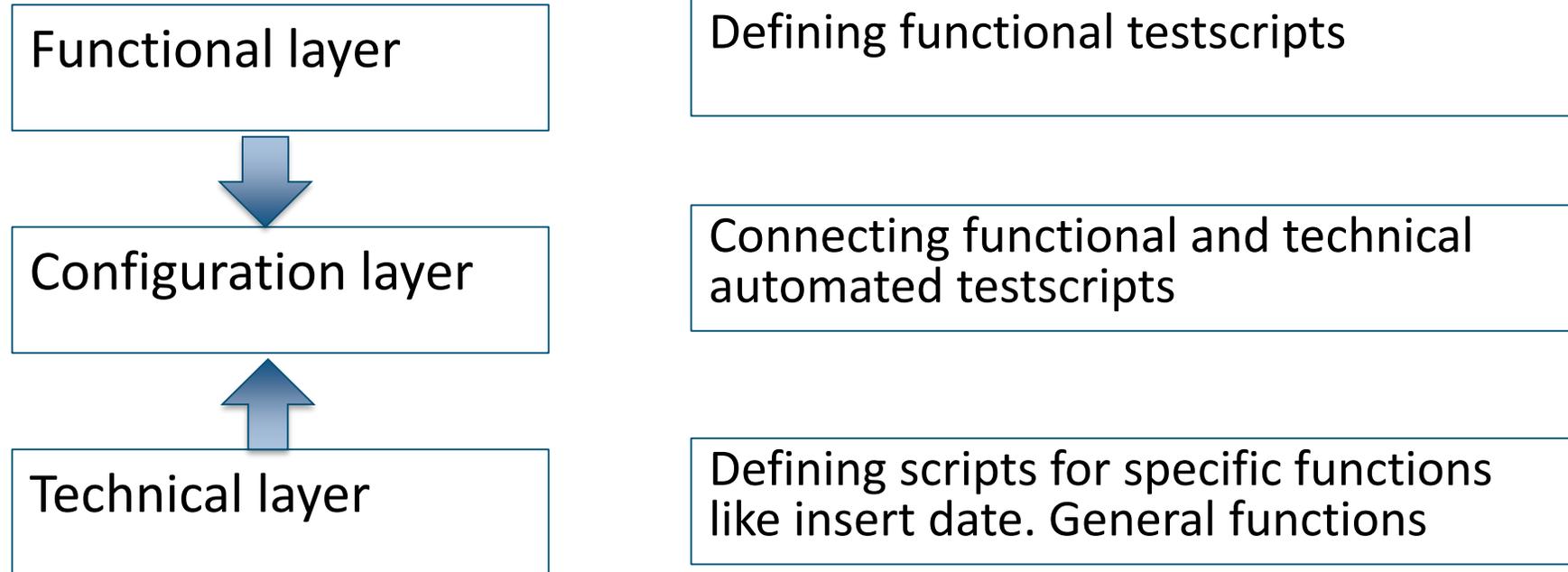
# The model



# Reducing maintenance

- By hand of RTAA a future proof test automation is possible
- The question is how to reduce maintenance:
  - Required effort
  - Technology independent
  - Business functionality independent
  - Tooling independent
- A few measures are required:
  - Set up approach
  - Reusability
  - Repeatability
  - Transferability

# 3-layer set up



# Reusability

- Think before acting
- Which components must be reusable
- Think about the component structure. As generic as possible
- Link this to the level of test automation like a unit, a chain etc.
- Re-use in several types of testing. Functional, stress or performance testing
- Make it parameterizable
- Test data driven

# Repeatability

- No single use → disinvestment
- Must be repeatable numerous times
- Isolate variable values, like budget, age, SSN
- Define which items block repeatability:
  - Calendar date
  - BSN (Citizen Service Number)
- Link the required effort to the frequency of use

# Transferability

- Not for only 1 person or 1 sprint
- Scripts or components that have been developed are suitable and available for:
  - Multiple testers
  - Multiple test types
  - Future proof for other technologies
  - Generic set up

# Securing knowledge

- TAAD (Test Automation Architecture Development)
- Central repository
- Training:
  - Future proof test automation
  - Tools & techniques
- Sponsorship within the organization
- Organising maintenance budget

# Example TAAD document

- Introduction
- Tooling:
  - Service & Integration
  - GUI
- Authorization & Configuration
- Test environment
- Generic components
- Test data
- Query's

# Business case for test automation

- Determine when test automation is profitable
  - During waterfall period test automation was profitable after 4-6 runs
  - In devops development test automation is executed many times per day
  - How to calculate return on investment?
- 
- The idea is to determine the value of test automation from a business point of view

# Business Case

## Demonstration of the value by:

- Chain keeps functioning so that company goals can be reached
- Thinking in terms of risks. What should never happen in the business?
- Avoiding instability of system under test
- Occurrence of production incidents
- Reduction of routine test work
- Correct coverage
- More efficient input of people
- Meeting the difference in quality of teams
- Shifting to the left

# Test automation pros and cons

- Advantages:
  - Faster test implementation
  - Better insight in the quality of the application
  - Less dependent on resources
  - Always the same quality output
- Disadvantages:
  - Extra costs; initial and maintenance
  - Different roles
  - Requires discipline

# Future of test automation

- Is becoming increasingly prominent
- Has become a permanent part of the software development life cycle
- Shift to the left. Earlier in the development chain
- Tools are becoming increasingly powerful:
  - Unburdening of the user
  - More intuitive use
  - Predict potential problems
  - Automatic generation of scripts based on models
  - The tools themselves are becoming more intelligent
- Repeatable components

# Who is going to do the test automation?

- Tester
- Or developer
- Or ...

# Repeatable Automated Tested Building Blocks

- Based on the proposition “repeatable testing” and the insight obtained, we developed a test automation set, consisting of repeatable components.
- For a variety of domains such as infrastructure, pension funds, etc.
- The components are being set up in a generic way to allow for their use with a number of parties.
- The principle of repeatable building blocks is:
  - Repeatable
  - Reusable
  - Transferable
- Working from a generic framework, with the re-use of existing test sets

# Experiences

- Last 5 years we have been using and adapting the presented approach
- Companies in the industry, government and semi-government are using the approach
- Experiences are:
  - A lot of attention is only paid to the technical layer
  - Number of successful test automation projects is increasing
  - Using the presented model increase the success of applying test automation
  - Embedding into the organisation is a problem
  - By hand of RTAA people get aware of the problem why test automation is not always successful
  - Time must be allocated for setting up and maintenance of the test scripts
  - People must be selected and trained for the test engineering role
  - New roles are required

# The benefits

- Test Automation secured into the organisation
- Test automation repeatable and reusable between projects and domains
- Always available to run
- Support of devops development
- Reliable quality of the system under test
- Less dependency to critical resources
- Shortening of the required test execution time

# Future work

- Implementing the RTAA model and gain experiences
- Enhance the method based on the experiences
- Collecting metrics implementing TAA
- Start up an experience of repeatable test automation building blocks

# Moral of the story/conclusions

- Test automation is important and still increasing:
  - Quality
  - Quantity
- It is not “merely an extra project”
- Requires new roles
- The start up is extremely important and well-considered choices will have to be made
- Is a permanent fixture in the development chain
- Test automation architecture is an important element
- Is this the end? Certainly not. Many developments that increase the speed are still to come

# Comments

identify  
THE QUALITY DRIVER ●