

We make software testing **easier.**

Die Stärken unserer KI basierten Software Test Automation

Was unsere neue Testautomation für Functional GUI Testing auszeichnet

```
elif operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True  
  
#selection at the end -add back the deselected mirror modifier  
mirror_ob.select= 1  
modifier_ob.select=1  
bpy.context.scene.objects.active = modifier_ob  
print("Selected" + str(modifier_ob)) # modifier ob is the active
```

wirtschafts
agentur
wien

Gefördert durch die
Wirtschaftsagentur Wien

Ein Fonds der
Stadt Wien



Schnell und einfach

Keine Installation

Keine Automations-Sprache nötig

Natürliche Sprache als Eingabe

statt scripten, programmieren etc.

... auch durch das gesprochene Wort

Beschleunigt Testprozess

Ohne Schulung, Training verwendbar

anstelle von Online Courses, Zertifikate u.ä.

Testentwicklung schon bei Requirements-Erstellung möglich –

Requirements Team gestaltet mit

Testautomation vor Implementierung der Anwendung möglich

statt Testautomation an Hand des implementierten Prüflings

Reduziert „time to market“ -

Testautomation „hinkt nicht nach“, sogar bei Last-Minute Changes

***Ideal* im Agilen Umfeld**

```
require File.expand_path("../support/runner.rb", __FILE__)
# Prevent database truncation if the environment is production
abort("The Rails environment is running in production mode!")

require 'spec_helper'
require 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver = :webkit

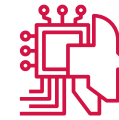
Category.delete_all; Category.create!

Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end

# Add additional requires here for the controller and model specs
# Requires supporting ruby files with spec support/ and its sub
# spec/ support/ and its sub
# run as spec files by default
# in _spec.rb will both be
# run twice. It is recommended
# end with _spec.rb. You
# can also use the command
# bundle exec rspec --format progress --no-color --no-quiet --no-pending --no-exit

No results found for 'mongoid'
```

Technologie-agnostisch



any application
on any platform
on any device



Identische Automation und identische Testabläufe für unterschiedliche Plattformen (desktop, smartphone, tablet)

statt Tool A für PC und
Tool B für Mobile und Neuschreiben für Mac

Eine einzige Testautomation für unterschiedlichste Technologien im Zuge von End-2-End Tests

statt Plugin für Technologie A und
Plugin für Technologie B und ...

Robuste Testautomation, die laufende Änderungen verkraftet

statt Nachbessern in der Automation,
wenn die Anwendung sich wieder mal geändert hat

Human Centered Testing

Vom Fachbereich ohne Spezialkenntnisse einsetzbar –
statt Testautomation nur durch „Testabteilungen“

Erkennt für Menschen unbedienbare Elemente
statt „Durchwinken“ von Applikationen mit unsichtbaren Controls

Prüft, ob Applikation auch für sehschwache User verwendbar

Testet optional das Zusammenwirken von Mensch und Maschine, wie
Reflexionen, Kontraste, Accessibility in der Praxis (*Safety Testing*)



European
Accessibility Act,
DIRECTIVE (EU) 2019/882



Nicht invasiv

Kann in der Produktionsumgebung testen - keine Installation, keine Restriktion

statt Beschränkung auf „Testumgebungen“

Kann ebenso gut über Remote Desktop, CITRIX, TeamViewer etc. testen wie auf der Maschine selbst

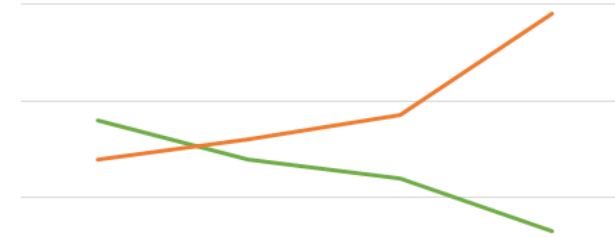
wo andere passen

Kann auch in sicherheitskritischen Umgebungen eingesetzt werden („geschlossene Systeme“)

wo sonst Testautomation außen vor bleibt



Spart Geld



Von Nicht-IT Spezialisten einsetzbar

statt „Experts-only Tools“

Keine Änderungen/Anpassungen für Multi-Channel

statt Anpassung pro unterschiedlicher/m
Betriebssystem, Plattform, Device

Keine Anpassungsaufwände bei neuen Releases

statt „Wartungsfalle in der Automation“

Gleiche Business Logik bei unterschiedlichen Anwendungen:
keine Anpassung der Tests nötig

statt Anpassungen für unterschiedliches Aussehen, Umgebung, etc.

Auf einem Blick

50% Kosten-
ersparnis




keine
Installation

nicht invasives
Testen



weltweiter
Zugriff
auf Test-
Repository



50% kürzere
Testzyklen

</>
kein
Programmieren

Software **testing** will now
become **as easy as pie!**

roland.tscheinig@objentis.com

