End2End Monitoring Integration in Check_MK

Check_MK Conference 2017

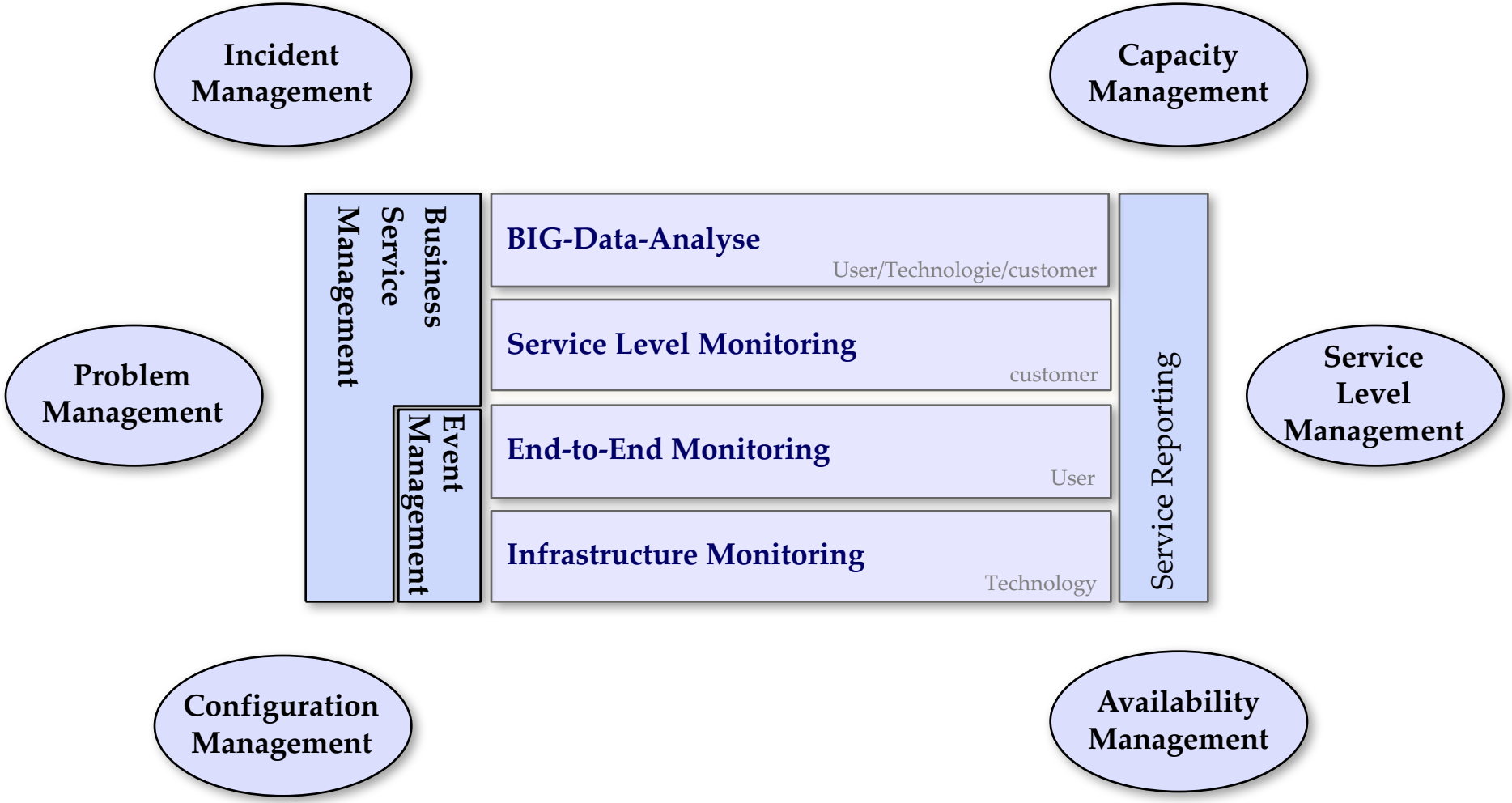
Munich, 4th May 2017

Who are we?

- **Founding date:**
  14. Mai 2001
- **Legal form:**
  GmbH
- **Headquarters:**
  Cologne
- **Contact:**
  info@iteratio.com
- **Founder:**
  Hardy Düttmann
  Rolf Assenmacher

- **Core business**
  Systems management
  Automation
  Project management
- **Monitoring experience**
  Since 1997
  Tivoli Framework 2.0
  HP Openview NNM 5.0
  CA Unicenter TNG
- **First OMD/Check_MK Project**
  January 2011 (Version 0.44)

Business Field Monitoring



Agenda

- Overview End2End Monitoring – Terms and Definitions
- Overview End2End Monitoring – Active vs. Passive End2End Monitoring
- End2End Monitoring – Objectives and Addressees
- Overview Free End2End Monitoring - Tools
- End2End Monitoring of Web applications
- End2End Monitoring of GUI applications
- Combination of Web and GUI Application

Terms - Delimitation

End2End Monitoring mit Check_MK

End2End Monitoring

Synthetic Transaction

Active End2End Monitoring

Robotics

Real User Measurement

Application Response Measurement

Passives End2End Monitoring

Real User Monitoring

SLA Monitoring

Passive vs. Active E2E Monitoring

End2End Monitoring with Check_MK

Active E2E Monitoring

Simulation of end user behavior by creating test or robotic scripts

- Measurement 7 * 24 hours
- Simulation of End user behavior by robots
- Small number of measurements
- Same measurement of different locations
- Basis for service level monitoring

Passive E2E Monitoring

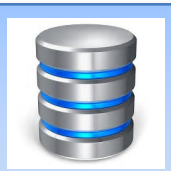
Determination of measured data on the basis of real user behavior

- Measurement only if user is active
- Real User measurement at protocol level
  - Use of sniffer or proxies in the network
  - Built-in interfaces in programming languages
- Large number of measurements

Objectives and addressees

Application Owner/ Customer Care Center

Operating / CCC /Application owner




Application Owner

- Early application error detection
- Changes in application behaviour

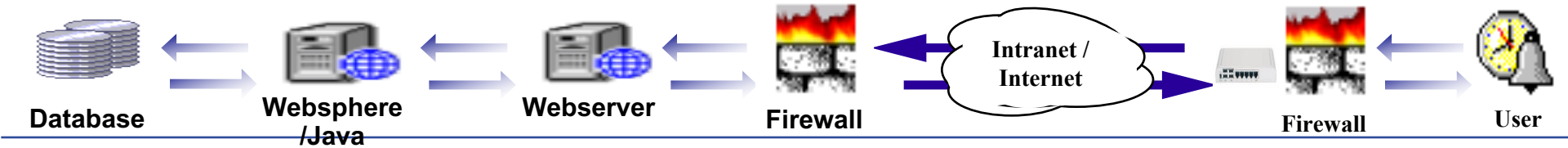
Customer Care Center

- Early Customer Information
- Application error tracking

Challenges

- Find out which subject area is responsible > there should be one or more component error for each End2End error

Use BI Aggregation of Check_MK !



ITeRATIO

Scenario

There should be a component error for each End2End error!!!

Component	**E2E**	
Fault	Ok	Component malfunction appears to have no effect on the end user
Fault	Fault	End user can not work. Cause probably reported as a component fault
Ok	Fault	End user can not work Reason unknown Monitoring Optimization needed !!!!

Objectives and addressees

Service Management

Measurement of Business Services
Basis for service level management

Service Management

- Customer Reports and Real time information
  - Quality Assurance
  - Post Processing
  - Outage explanations

Challenges

- Erroneous measurements
- Single Measurement errors
  - Network or Client problems
  - Measurement Runaways
- Handle Application Updates
  - The Monitoring team is often informed last

- Use Check_MK service times
- Use BI Aggregation
  - Correlation of multiple testclients (locations, versions)
- Use Check_MK corrective functions
  - Annotations and subsequent downtimes (Version 1.4xx)

There are several open source tools for E2E monitoring

Tools

Alvix

Al'exa

Sakuli

Casper JS

Auto it

Boomerang

Selenium

Sikuli

Robotics

Webinject

sahi

Tools for End2End Testing

Monitoring of Web Applications

- Check_http
- Check_webinject

- Seleninum
- Sahi
- AutoIT

Monitoring of GUI Applications

- AutoIT
- Sikuli

Combination of Web and GUI Applications

- Sakuli (Consol Labs)

Check_http , nt_check_http.ps1

Check_http, nt_check_http(Powershell Skript), ...

- Easy http/https access to a web page (incl. redirects)
- Evaluation of HTTP status codes
- Can Monitor Connection Times and Page Sizes
- Can search for strings and regular expressions in Header an Content
- Can Check Certificates
- Works with http auth and partially also with form auth (post)

Limitations

- Calling up a single web page (possibly with redirect and login)
- No real end2nd test simulation

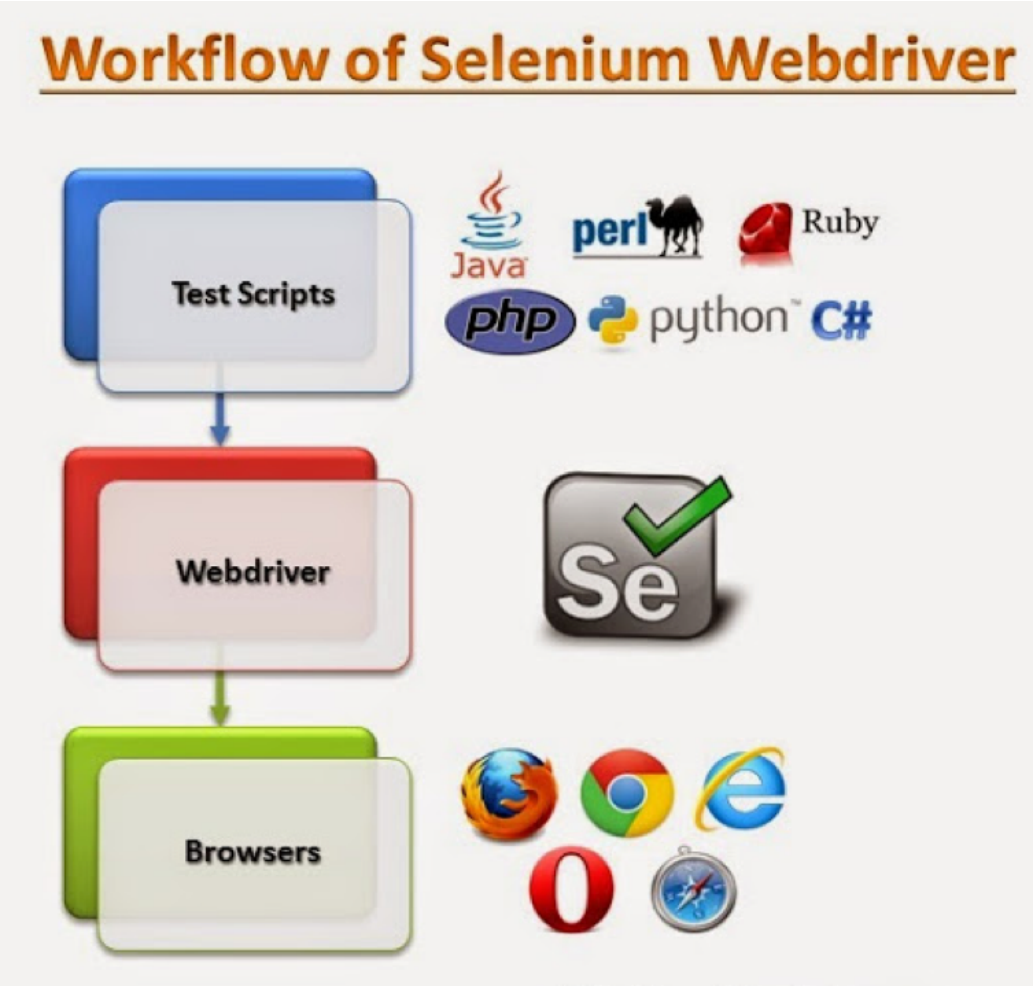
check_webinject (webinject project)

- Tool for the automated testing of web applications (Perl)
- Thresholds for individual steps (Warning, Critical)
- XML-like configuration file for testcases
- Works on HTTP (S) GET protocols or POST requests
- Performance measurements for total runtime and individual steps
- Is shipped with Check_MK (OMD)

limitations

- No GUI rendering (no javascript, VB script, etc)
- No automatic redirects -> new testcase for the landing page
- Runtime = real runtime in the browser -> approximation value
- Only warning threshold (globaltimeout) for total run time
- No recorder (possible use of a proxy meaningful -> proxomitron (no https)
- Time-intensive search for HTML elements for HTTP (S) Request (Hidden Fields, Session ID, ...)

Selenium Web Driver



Pros:

- Open Source
- Large community
- Very good documentation even for language bindings
- Works on multiple os (java)
- Works on multiple browsers
- Multiple scripting languages
- Works headless (phantomJS or xvfb)
- Cons:
  - Usage of Xpath
  - Need of wait statements for making ajax work

Selenium Web Driver API

Locating elements:
- find_element_by_id
- find_element_by_name
- find_element_by_xpath
- find_element_by_link_text
- find_element_by_partial_link_text
- find_element_by_tag_name
- find_element_by_class_name
- find_element_by_css_selector
- .....

Actions :
- _click, _double_click ...
- _send_keys
- _move_to_element
- .....

Wait statements:
- Implicit Waits
  - Waiting on DOM objects
- Explicit Waits
  - Waiting on conditions

Page navigation, history and popup handling:
- _switch_to_window
- _switch_to_frame
- _switch_to_alert
- .....

Selenium Integration in Check_MK (Nagios) 2/2

Nagios Plugins

- check_selenium, check_selenium (Perl), check_selenium.py and many more

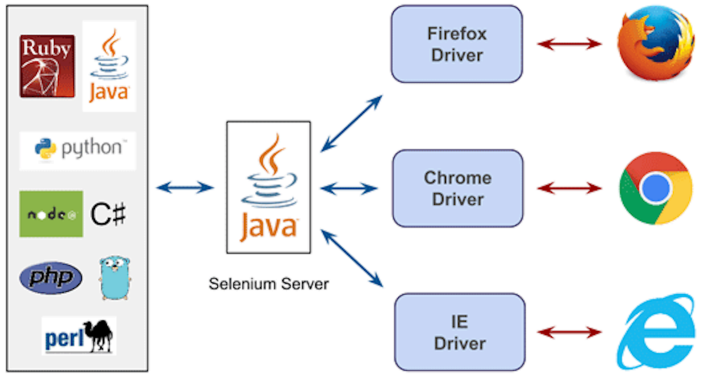
!!!All of them have limited functionality!!!!

Selmon

- Classical active check
- https://github.com/objectified/selmon
- Python Library
- Easy Performance values for individual steps
- Easy integration of conditional test
- Integrated Picture Tests
- Needs selenium server running

```
def run(self):
    driver = self.driver
    with benchmark(self.nagios_message, 'open_homepage', warning=5, critical=10):
        driver.get("https://zammadsrv.hdudomain.local/")
```

```
with test(self.nagios_message, 'ITeratio Logo present'):
    self.verify(driver.find_elements_by_class_name('company-logo'))
```



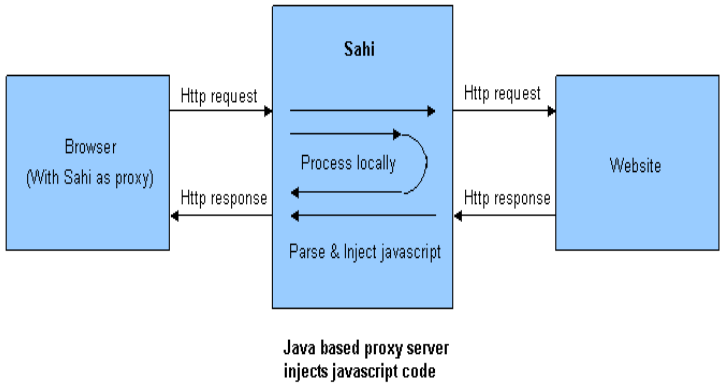
Sahi Functionalities

Functionalities:

- Open Source (sahipro with extended apis and functions)
- Sahi javascript for recording and executing actions
- Intelligent Browser Object detections
  - Implicit smart identification of elements even across frames and Iframes (no xpath)
  - Can also find element close to other searches (ex. Textbox near the "Username" label)
- Automatically waits for web pages to load
- Runs on each os where java runs
- The same script can be used for any browser
- core element: sahi proxy

sahiproxy:

- Recorder: records all requests in sahiscript(java script)
- Playback via injection of sahiscript in browser
- Works headless with phantomjs or xvfb



Sahi APIS

Accesor and Relation API: identify DOM-objects in the browser

- Base for Browser action api
- Typical are _link, _textbox, _password, _datebox .....
  - Identified by indizes, names, classes, etc.
  - Regex can be used
- Sahi does not use xpath for locating elements
  - Uses relation statements like _near, _in, _under,_above,
  - Use multiple statements for robust testcases

Action Apis:

- _click, _double_click, _right_click ...
- _mousover, _mousedown, _mouseup
- _check, _uncheck
- _setvalue, _keydown,_keyup ... Playback via injection of sahiscript in browser
- Multiple _touch action apis

AssertationApis:

- _assertExists, _assertNotExists
- _assertVisible, _assertNotVisisble
- _assertEqual
- _assertContainsText
- .....

Sahi Check_MK integration

Check_sahipro

- Bash script
- Duration Thresholds and performance measurement only for total run time.

Sahi2OMD (Consol Labs)

- VBS script (Windows Only)
- Send_nsca or db connect

Sakuli (Console Labs) Check_MK Integration

- Integration takes place via forwarder modules
  - gearman, Icinga2, Database
  - Check_mk (since January 2017)
- Thresholds on whole testcase
- Thresholds on each step
- Automatic exception detection (test object missing)
- Uses Check_MK Spool Directory

AutoIT

Functionalities:

- Automate windows workflows
- Own Scripting Language(basic similar)
- COM and Powershell Support
- Can us DLL and Windows API Functions
- Simulation of mouse and keyboard actions
- Interaction with Windows windows and Standard Controls
- Interaction with Registry, clipboard, process control and und Windows/DOS Files incl. Execution
- Integrated IDE(SciTE)

Pros:

- Expandable via User Defined Functions
  - AutoIt3
    - Internet Explorer
    - Excel
    - Word
    - SQLite
  - Community UDF's
    - Firefox
    - Opera
    - Powerpoint
    - TeamViewer...
    - …. Up to 100 more

Web Application E2E Monitoring 1/3

Feature \ Product	Selenium	AutoIt	Sahi	check_webinject
Browser-Support	Internet Explorer, Firefox, Chrome, Safari, ...	Internet Explorer, Firefox, Opera	Internet Explorer, Firefox, Chrome, Safari, Opera, ...	No browser support , works on protocol level
Platform	Linux, Windows, Mac	Windows	Linux, Windows, Mac	Independent(Perl-Skript/Modules)
Testcases (Description Language)	Java, Python, Ruby, C#, Perl, ...	Similar to BASIC	Sahiscript (JavaScript)	XML-format
Creation of test cases	Selenium IDE or anny other IDE e.g. Eclipse	SciTe or any other IDE	Sahi Driver, Eclipse or any other Java IDE	Arbitrary Editor

Web Application E2E Monitoring 2/3

Feature \ Product	Selenium	AutoIt	Sahi	check_webinject
Execution of Testcases	Registered and logged on desktop user			Background
Execution with locked desktop	Yes (no GUI control possible)	Yes (As long as no direct GUI control is necessary)	Yes (no GUI control possible)	Yes
Check_MK / Nagios Plugin	Yes	Yes	Yes	Yes
Check_MK integration	MRPE/NRPE / Nagios Plugin, NSCA, Spool Directory local check (no Windows, Session 0 Problem)			Classical Nagios Plugin

Web Application E2E Monitoring 3/3

Feature \ Produkt	Selenium	AutoIt	Sahi	check_webinject
Automatically waits for web page loading (ajax elements)	„no“ User wait statements needed	no	yes	Not relevant, HTTP Status Codes (No GUI)
Test case Recording	yes, export in different language bindings possible	no	yes	No (use proxy to identify actions)

!!!What you should do!!!

Use robust selectors

Use multiple Test Clients

Use multiple Locations

Use Check_MK BI Aggregations

!!!What you should not do!!!

Don‘t test this with your Icloud Account!!!!!!

Testing Xing works!

Sikuli Functionalities

- Java Applications
- Script Languages
  - Python (Jython), Ruby, JavaScript, RobotFramework
- Robotic Control of GUI Aplications (fat clients)
- Identify Image Patterns (Screenshots) on the display (OpenCV)
  - Screenshot-based tool
    - Screenshots must be created manually
  - Dependent on
    - ScreenSize(resolution)
    - Presentation form (e.g. colors, background image, etc.
    - Localization(Language)
    - Specific Application (Firefox, IE, ...)
- Execute Mouse an Keyboard actions
- Can search for more objects in the region
- Can extract text by OCR (Tesseract) (BETA)

Comparison

Gui Applications

Feature \ Product	AutoIt	Sikuli
Testcases (description language)	Similar to BASIC	JavaScript, Python, Ruby
GUI-Control and Comparison	- Standard Controls - Usage of DLL and Windows API functions - keyboard and mouse actions	- Pixel-/Screenshot comparison - OCR / Text Recognation - keyboard and mouse actions
Integrationin Check_MK	MRPE/NRPE over Nagios Plugin, NSCA, local Check,Spool Dir	
Creation of Testcases	Scripting in IDE necessary	Scriping in IDE necessary
Execution of Testcases	Registered and logged on desktop User	Registered and logged on desktop User
Execution with locked screen	Yes, if no mouse or keyboard actions are executed	No, since no text input is possible via control methods..
Platform	Windows	Linux, Windows, Mac

Sakuli (Consol Labs)

- Sakuli (Sahi + Sikuli)
- Combines the best out of both tools
  - Sahi for Web Testing
  - Sikuli for native rich client GUI content
- One Test for HTML generated PDF Documents, Flash, Citrix ...
- Special integration of various tools by Forwarder Files
  - Gearman
  - Database(MySQL): Storage of measured data
  - Icinga 2
  - Check_MK
- Core Components
  - Starter, Test Definitions, Forwarder Modules

Sakuli (Consol Labs)

- Check_MK Integration

  - Sakuli Check_MK Forwarder generates Files in Check_MK Spool Directory
  - Number prefixes can be configured for data freshness
  - Performance Data and thresholds can be set for the whole testcase and for each step
  - Integrated detection of Testing(comparison) errors

Summary

- All of the tools can be integrated in Check_MK
- Use them together with
  - Element/Component monitoring (360° View)
  - BI
  - Reports
  - Service Times
  - Performance Graphs
- Creation of robust test cases
  - Error handling
  - Multiple element validation,
- Security
  - If possible, run with a locked desktop
  - Use virtualization
  - Location based security
- Ensure smooth test case operation
  - Auto Recovery
    - Intercepting e.g. Memory leaks of the controlled application (browser)
    - Check state of the last testcase, before starting new one

Wir stellen ein!!!

karriere@iteratio.com