

# Synergien aus Testautomatisierung und Lasttest

Vortrag im Rahmen des German Testing Day 2018

## Referent: Dirk O. Schweier

- **Erfahrungen**
  - Qualitätsmanagement
  - Testmanagement
  - Testautomatisierung
  - Trainer für ISTQB®- Seminare
- **Qualifikationen**
  - Fachinformatiker FR Anwendungsentwicklung
  - ISTQB® Certified Tester Full Advanced Level
  - Certified Consultant for NeoLoad



G. Muth Partners GmbH  
Borsigstraße 32  
65205 Wiesbaden

Tel: +49 6122 5981 22  
mobil: +49 160 96956905

[dirk.schweier@muthpartners.de](mailto:dirk.schweier@muthpartners.de)

## Gliederung

- Ausgangssituation und Herausforderung
- Testausführungswerkzeuge vs. Lasttestwerkzeuge
- Anforderungen
  - an eine Integration der Werkzeuge
  - an das Testausführungswerkzeug
  - an das Lasttestwerkzeug
- Umsetzung der Anforderungen
- Synergien aus dem kombinierten Einsatz
- Fragen aus dem Auditorium

## Ausgangssituation für LOHRFINK software engineering

- LOHRFINK entwickelt und vertreibt die Java-Applikation LF-ET zur Erstellung und Verwaltung von Entscheidungstabellen
- LF-ET wird mittels Testautomatisierung getestet
- Entscheidung:
  - LF-ET als SaaS anbieten
  - LF-ET als Webapplikation bereitstellen
  - Webapplikation mit WebSocket-Kommunikation
- Konsequenz: Anpassung der Teststrategie

The screenshot shows a web browser window displaying the LF-ET 2.1.5 application. The main content area features a table titled 'Konditionen für einen Finanzkauf'. The table has columns for conditions (B01-B04, A01-A04) and four scenarios (R01, R02, R03, R04). The LOHRFINK software engineering logo is visible in the bottom right corner of the application window.

	R01	R02	R03	R04
B01 PKW Modell	-	-	-	-
B02 Anzahlung	< 2.000	2.000-10.000	2.000-10.000	2.000-10.000
B03 Laufzeit der Finanzierung	-	24	24	36
B04 Rückgabe Altfahrzeug	-	J	N	-
A01 Nachlass von € 1.000	-	-	-	-
A02 Nachlass von 3% auf Listenpreis	-	-	X	-
A03 effektiver Jahreszins in %	-	3	3	4
A04 Fehlermeldung werfen	ANZ-01	-	-	-

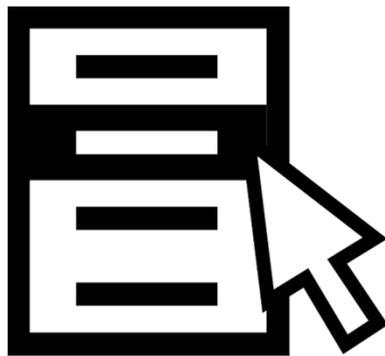
## Aufgabenstellung für LOHRFINK software engineering

- Funktionale Tests der Webapplikation
  - Kann die Webapplikation mit bestehenden Werkzeugen getestet werden?
- Herausforderung Lasttests für Webapplikation
  - Wie sehen Lasttests für die Webapplikation aus?
  - Lassen sich bestehende Testskripte wiederverwenden?

## Point of Control u. Point of Observation unterschiedlich

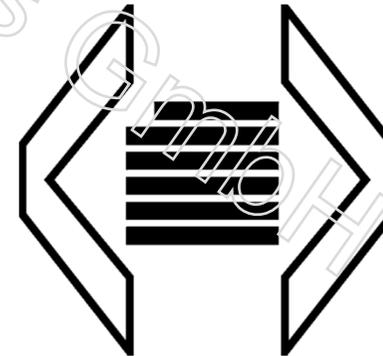
### Testausführungswerkzeuge

- Browserinstanz wird verwendet
- GUI-Objekte sind Point of Control
- GUI-Objekte sind Point of Observation



### Lasttestwerkzeuge

- Netzwerkkommunikation wird verwendet
- Requests sind Point of Control
- Responses sind Point of Observation



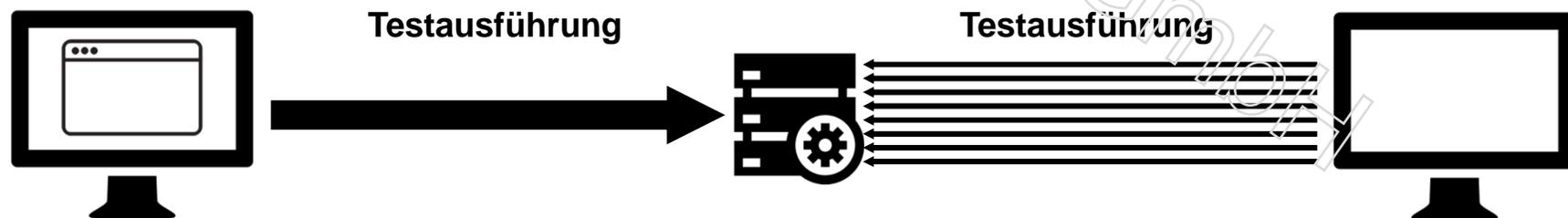
# Arbeitsweise hat Konsequenzen für die Testinfrastruktur

## Testausführungswerkzeuge

- Meist ein Rechner pro User erforderlich (Browserinstanz)
- Lizenzierung meist pro Instanz

## Lasttestwerkzeuge

- Ein Rechner kann viele User simulieren
- Lizenzierung meist pro Anzahl der zu simulierenden User



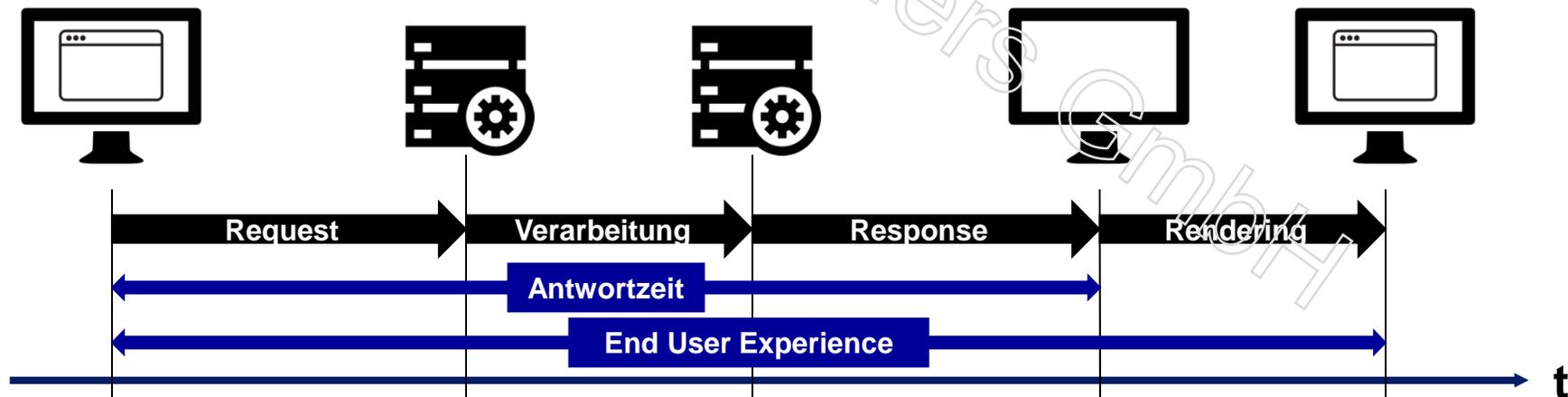
# Unterschiede in der Zeitmessung

## Testausführungswerkzeuge

- Dauer der Testdurchführung ist abhängig vom Erscheinen der erwarteten GUI-Objekten
- Damit kann die „End User Experience“ ermittelt werden

## Lasttestwerkzeuge

- Messung von Antwortzeiten der Requests
- Monitoring des System unter Test



## Anforderungen an eine integrierte Lösung

- Entwicklung von Lasttestskripten
  - Wiederverwendung bestehender Skripte
  - Import von Skripten in das Lasttestwerkzeug
  - Wiedererkennung der Skripte in den Werkzeugen
- Durchführung von Lasttests
  - Bereitstellung der Metriken aus Testautomatisierung in Lasttestwerkzeug
  - Messung der End User Experience

## Anforderungen an das Testausführungswerkzeug

- Kommandozeilenparameter
  - zum Start von Testskripten
  - zur Übergabe von variablen Werten
- Konzept zur Integration individueller Aktionen in die Testablaufsteuerung
- Schnittstelle zur Ermittlung von Metriken aus der Testdurchführung
  - Status des aktuellen Testfalls (passed / failed)
  - Dauer der Ausführung des aktuellen Testfalls
  - Fehlermeldungen und Schweregrad

## Anforderungen an das Lasttestwerkzeug

- Schnittstelle zur Übertragung externer Metriken
- Schnittstelle zur Steuerung von Aufnahmen zum Import
- Funktion für eine Zusammenführung von Skripten nach Neuaufnahme

## Werkzeugauswahl für Lohrfink nach Evaluierung

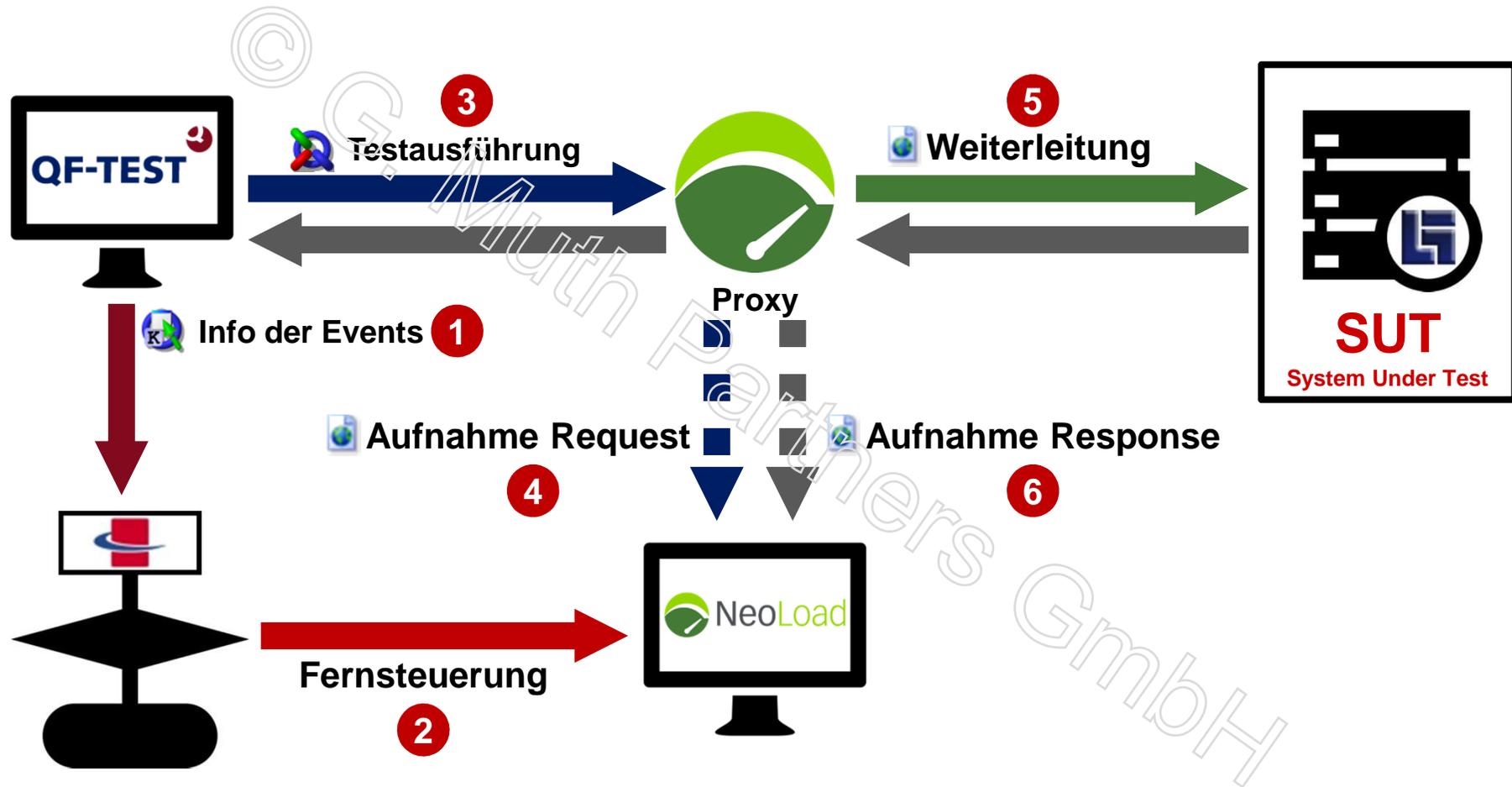
### QF-Test

- 👍 Unterstützung für Java und Web
- 👍 Kommandozeilenparameter
- 👍 Listener-Konzept erlaubt Individualisierung im Testablauf
- 👍 Runtime API liefert Metriken
- 👍 Plug In-Konzept erlaubt Integration externer Java Libs

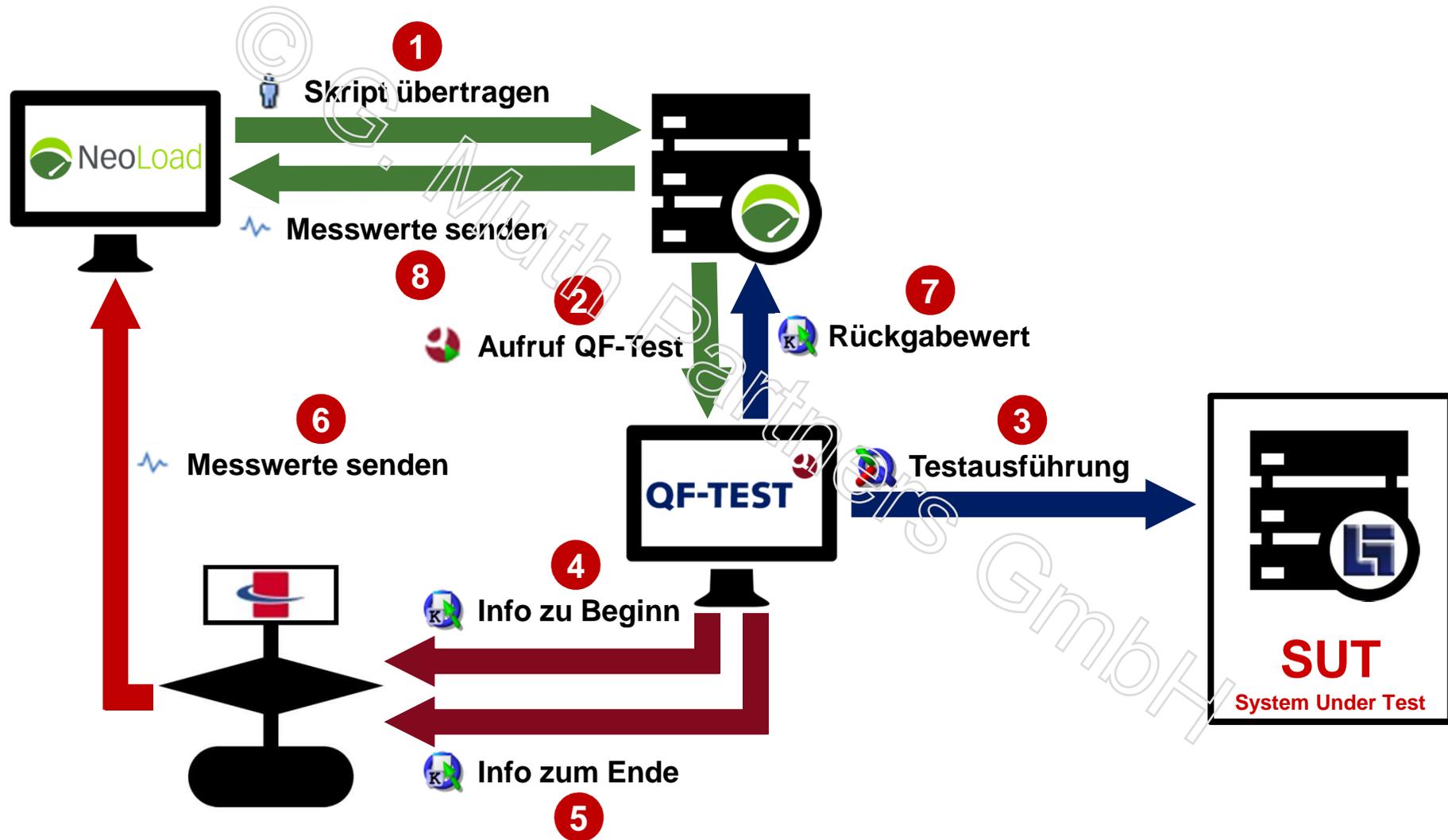
### NeoLoad

- 👍 Unterstützung von WebSockets
- 👍 API zur Übertragung von Metriken
- 👍 API zur Fernsteuerung der Entwicklungsumgebung
- 👍 Plug In-Konzept für spezielle Aktionsknoten zum Aufruf externer Testskripte
- 👍 Script Maintenance führt bestehende Skripte und erneute Aufnahme zu einem Skript zusammen

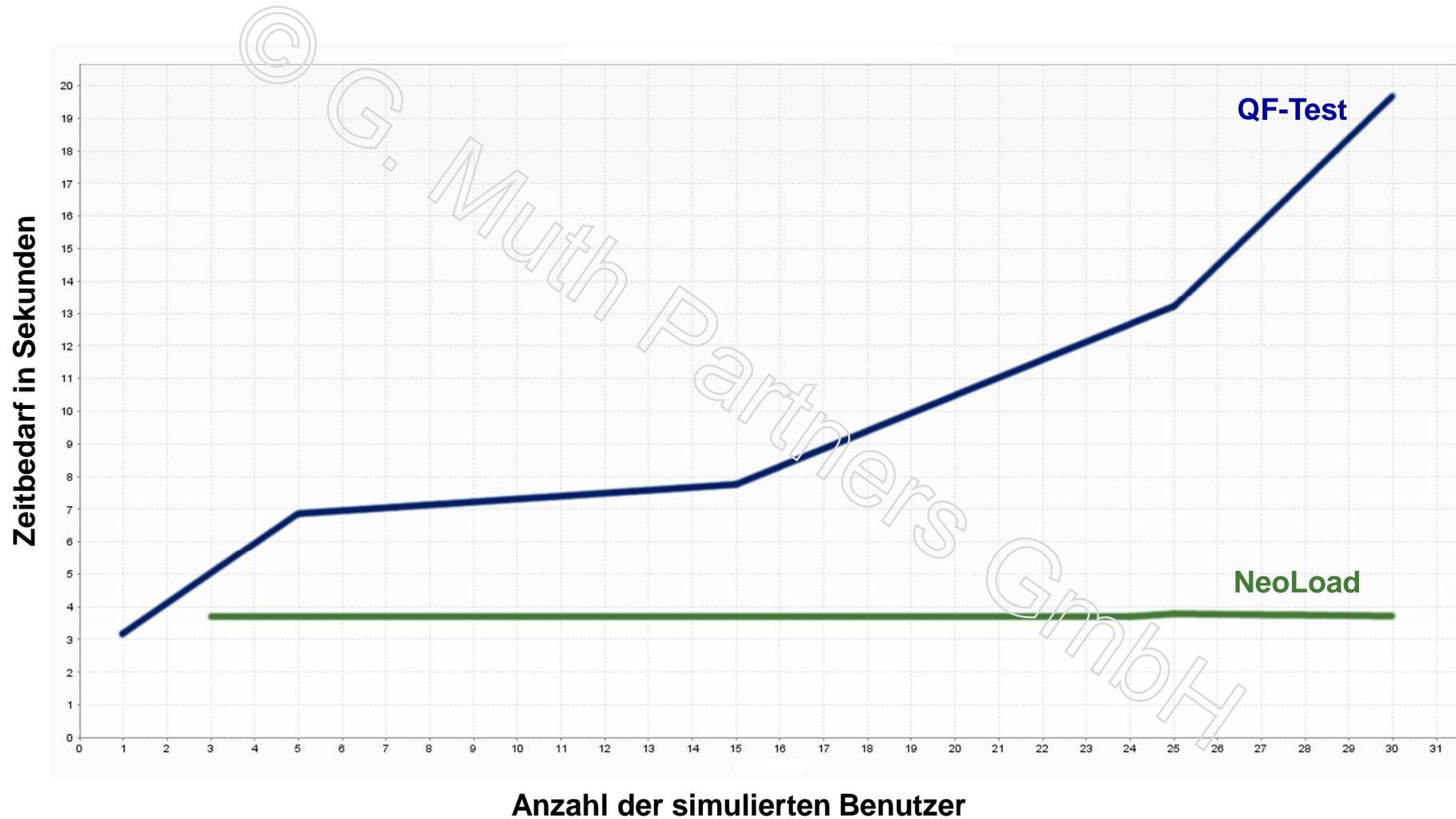
# Import von funktionalen Tests in das Lasttestwerkzeug



# Ablauf des Lasttest mit Testausführungswerkeug

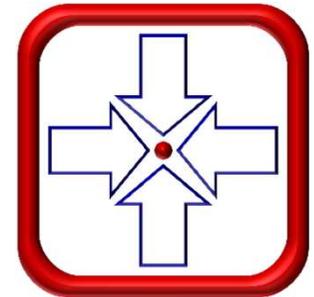


# End User Experience für eine Aktion des Skripts



## Testautomatisierung und Lasttest kombinieren

- Synergien aus der Kombination der Werkzeuge
  - Reduzierung der Aufwände durch Wiederverwendung
  - Messung der End User Experience als starkes Akzeptanzkriterium
  - Single Point of Truth durch Übertragung von Metriken
- Kritische Erfolgsfaktoren zur Erreichung der Synergien
  - Schnittstelle zur Übertragung von Metriken
  - Schnittstelle zur automatisierten Aufnahme von Lasttestskripten



## Fazit von LOHRFINK software engineering

„Dank der Integration zwischen QF-Test und NeoLoad waren wir in der Lage die Serverspezifikationen zu ermitteln, die unsere Kunden brauchen und die mit Blick auf die Kosten wirtschaftlich sind.

Durch Import der Testskripte und Script Maintenance von NeoLoad können unsere Testautomatisierer bestehende Skripte aus QF-Test einfach in NeoLoad übernehmen.“

## ... noch Fragen?

