

JMeter offroad

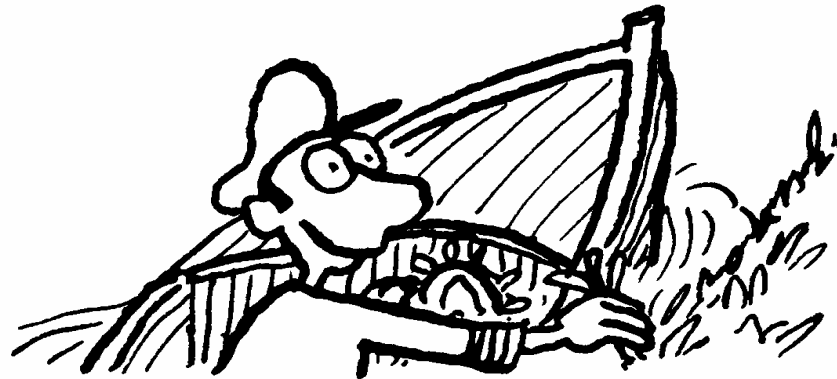
Performance- und Lasttests
nicht nur für Webanwendungen



Agenda

- Pleiten, Pech und Pannen - IT Projekte und Performance
- Basiswissen zu Last-/Performance-Tests
- JMeter Übersicht
- JMeter - Deep inside
- JMeter Offroad - Erweiterungen entwickeln
- Fazit / Zusammenfassung
- Weiterführende Links und Quellen

Pleiten, Pech und Pannen - IT Projekte und Performance



Pleiten, Pech und Pannen - IT Projekte und Performance

Sind oder waren Sie bereits in IT-Projekten, die

- Last-/Performance-Probleme hatten?
- aufgrund von Last-/Performance-Problemen gescheitert sind?

Hatten Sie in Ihren Projekten noch nie Last-/Performance-Probleme?

Wer arbeitet in einem Last-/Performance-Team mit > 1 Person?



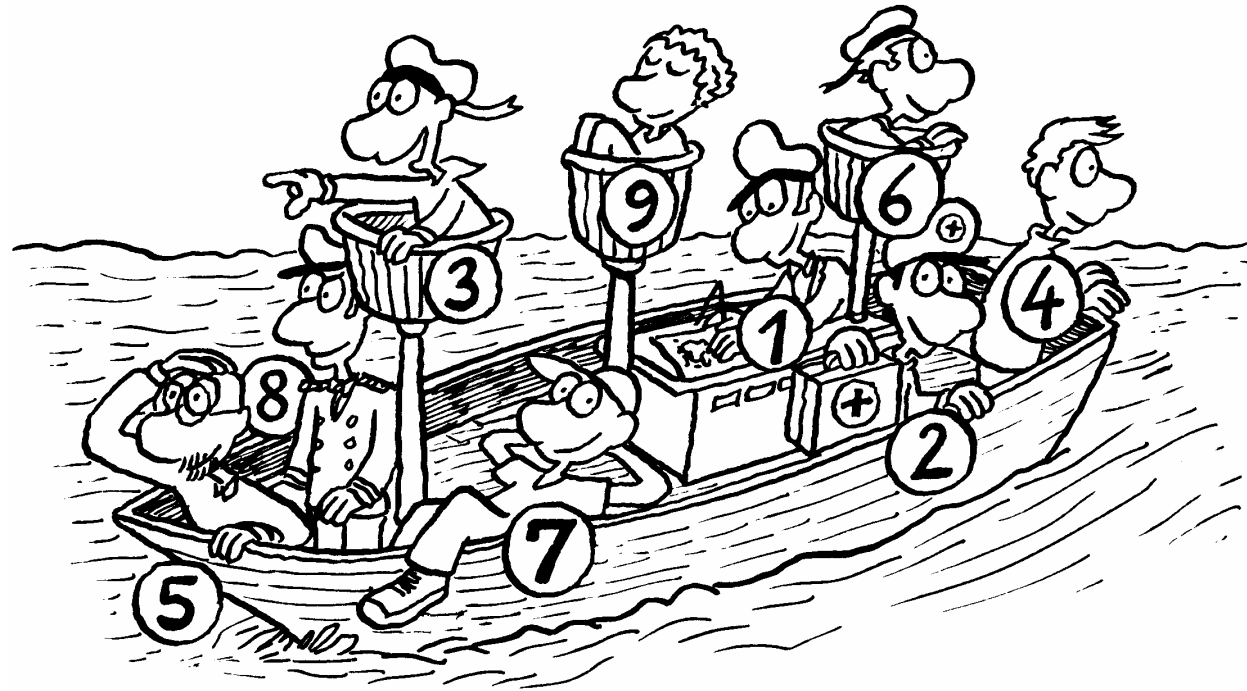
Pleiten, Pech und Pannen - IT Projekte und Performance

Mögliche Ursachen für zu spät entdeckte Performance-Probleme:

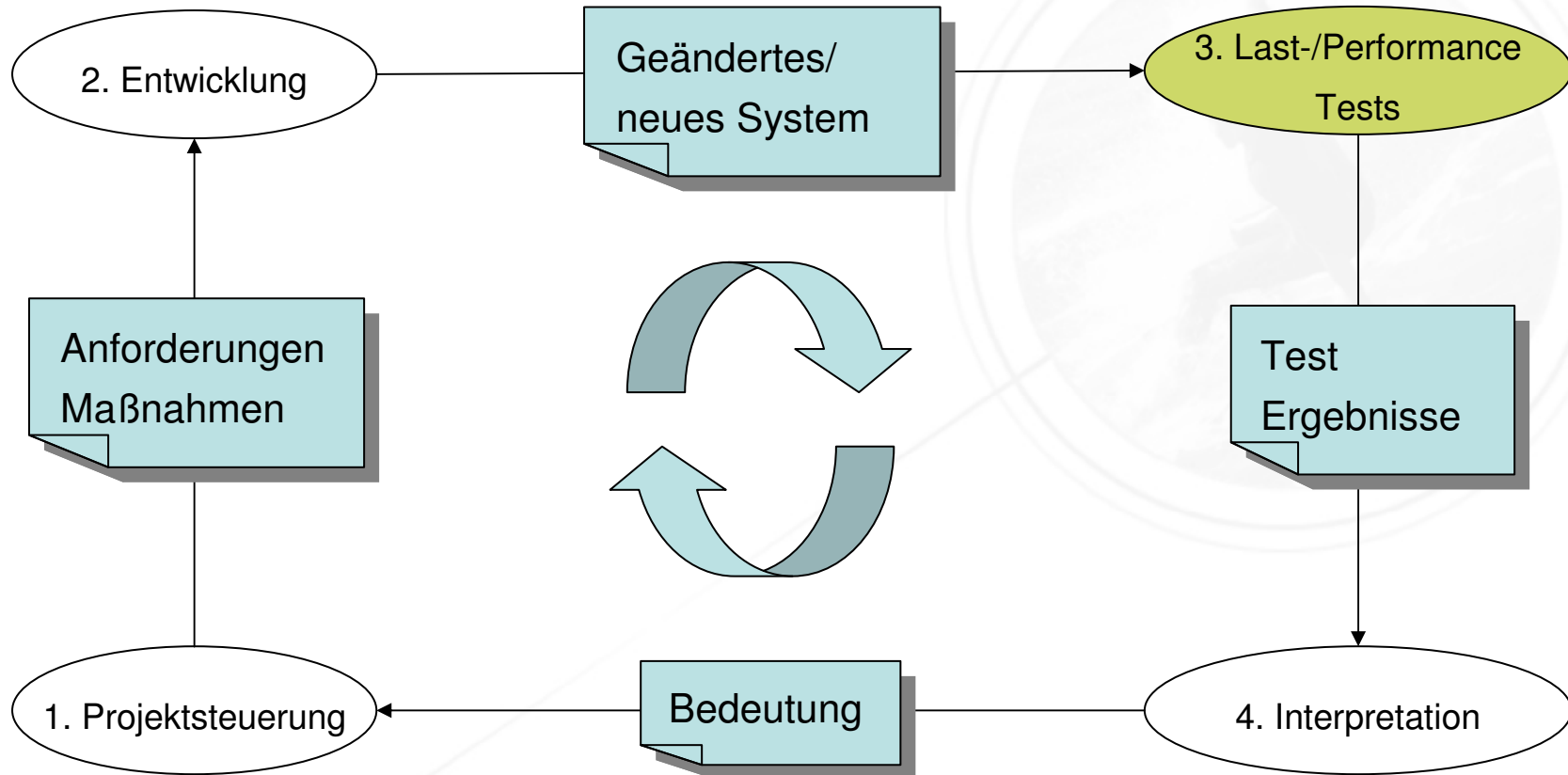
- Last-/Performance-Tests existieren nicht
- Kein Verantwortlicher für Last-/Performance-Tests
- Keine Dokumentation über Last- und Performance-Verhalten
- Keine Reaktion auf verändertes Last- und Performance-Verhalten
- Ungenügende Test-Hardware und Software, z.B.
 - Kein isoliertes Netzwerk
 - Ungenügende Netzwerk-Bandbreite
 - „Schwachbrüstige“ Rechner als Lasttreiber
- ...



Basiswissen zu Last- / Performance-Tests

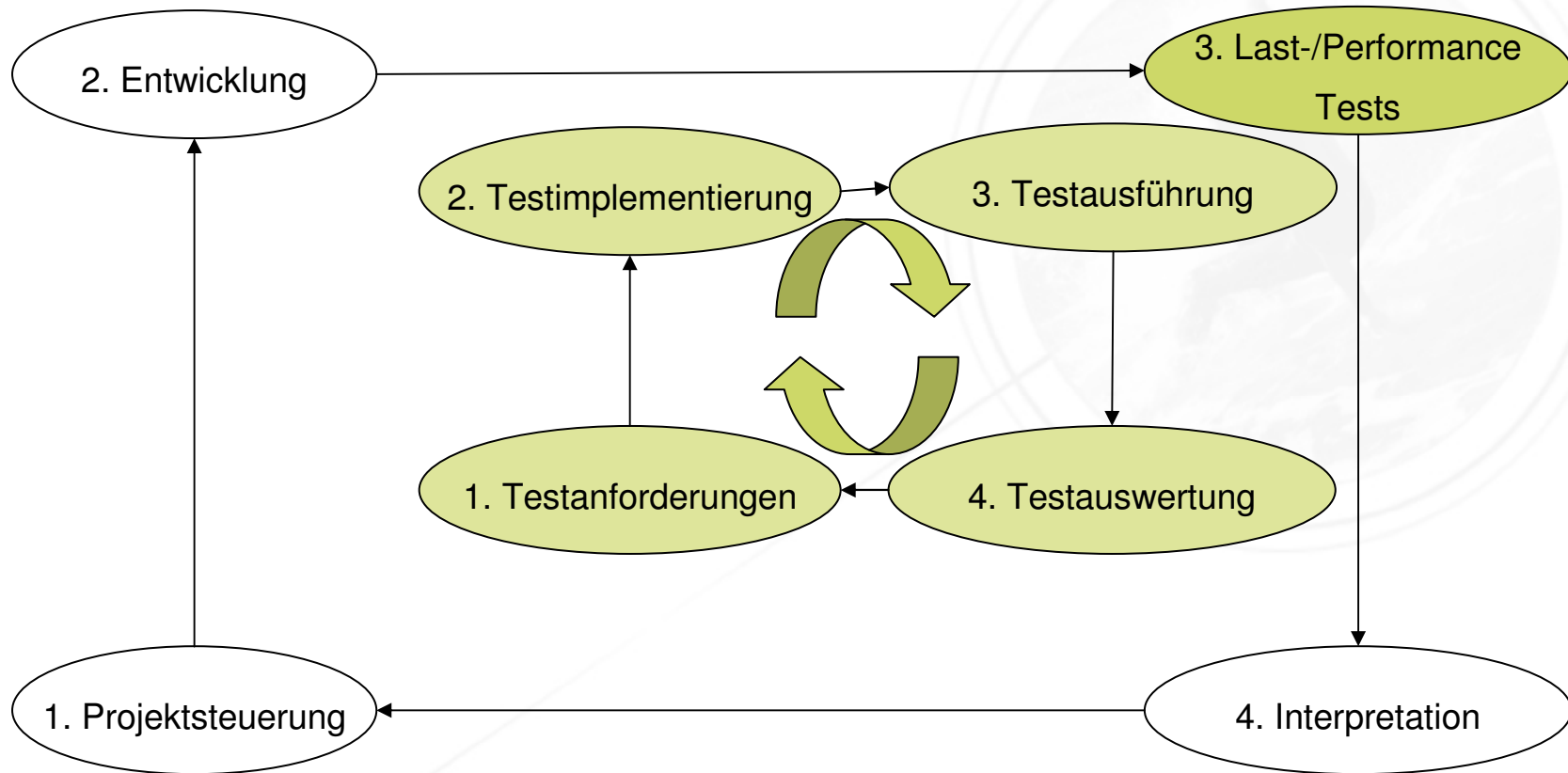


Basiswissen zu Last- / Performance-Tests

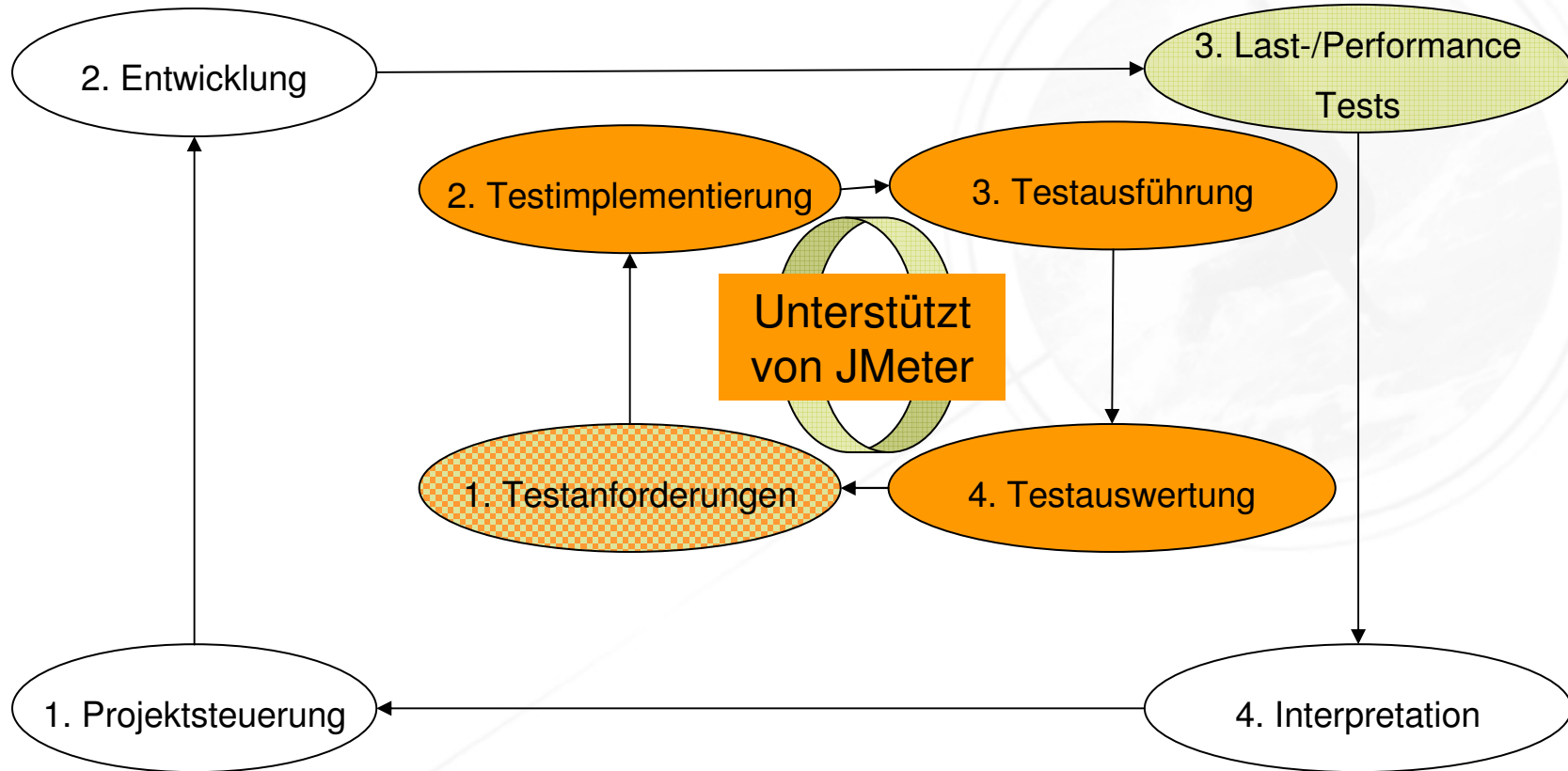


Quelle: Performance-Modellierung für Enterprise-Anwendungen, Matthias Bohlen

Basiswissen zu Last- / Performance-Tests

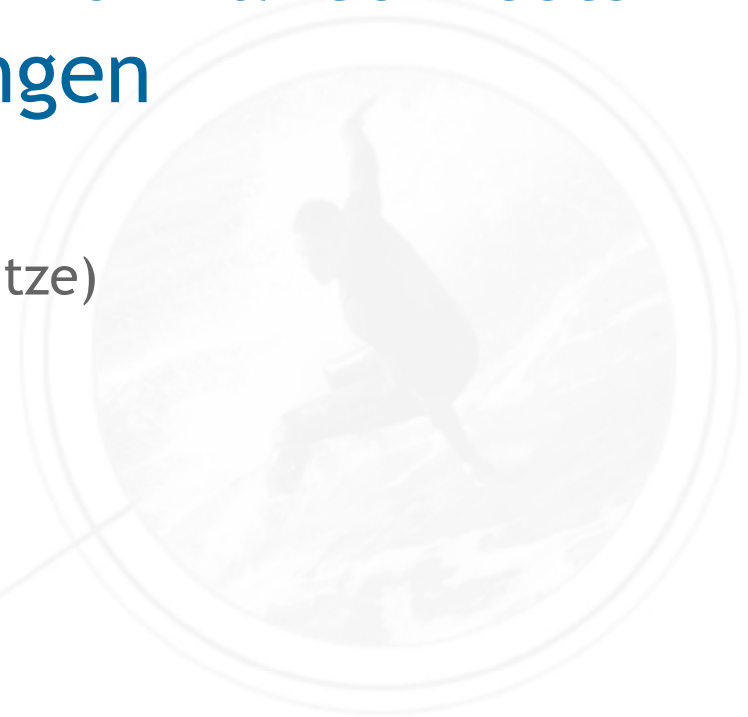


Basiswissen zu Last- / Performance-Tests



Basiswissen zu Last- /Performance-Tests Anforderungen

- Infrastruktur/Hardware
- Lastvarianten (Gleichzeitig, Aktiv, Spitze)
- Antwortzeiten und Durchsatz
- Weitere Modelle
 - Ablauf Wahrscheinlichkeiten
 - Denkzeiten/Timing

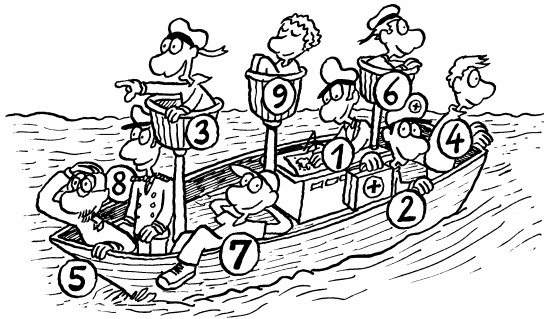


Basiswissen zu Last- /Performance-Tests

Rollenverteilung ;-)

Performance Team Skills

- Anforderungen
- Architektur
- Programmierung
- Testtheorie
- Testwerkzeuge
- Kommunikative Fähigkeiten

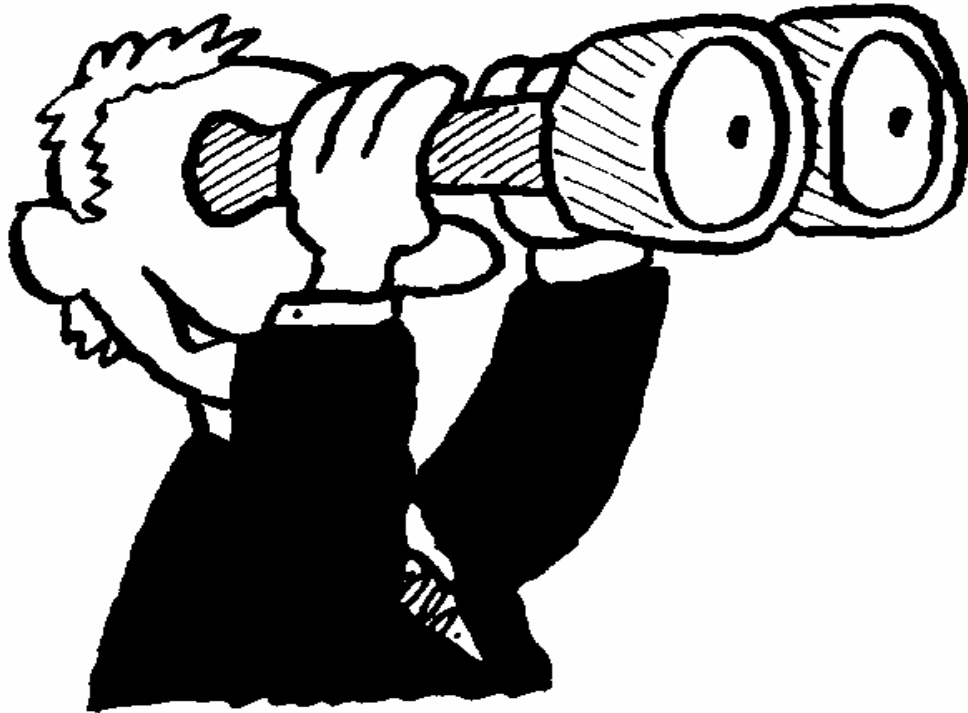


Performance Team Schnittstellen

- Management
- Marketing
- Architekten
- Entwickler
- Datenbank Administratoren
- System Administratoren
- Netzwerk Administratoren
- System Test, Qualitätsabteilung



JMeter Übersicht



JMeter Übersicht

- Gestern und heute
- GUI - antik, aber hilfreich
- Zentrale Elemente
 - ThreadGroup
 - Sampler
 - Controller
 - Listener
 - ...
- Gültigkeitsbereiche der Elemente
- Weiteres Wissenswertes



JMeter Übersicht

Gestern und heute

- Historie
 - Version 1.0 von Stefano Mazzocchi (02/99)
 - Aktuelle Version 2.1.1 von Michael Stover (10/05)
 - Release in halbjährigem Abstand
- Ziele / Zweck
 - Ursprünglich: Test Performance von Apache JServ
 - Simulation von Lastszenarien
 - Messung von Performance
- Open Source
 - unter <http://jakarta.apache.org/jmeter/>
 - 811 Klassen, ~ 58 Kilo NCSS

JMeter Übersicht

GUI - antik, aber hilfreich

The screenshot displays the Apache JMeter GUI. The left pane shows a test plan tree with the following structure:

- Test Plan
 - Thread Gruppe
 - ContactSite
 - 2 x
 - HTTP Header Manager
 - HTTP Request Default Einstellungen
 - /de/kontakt/formular.htm**
 - /cgi-bin/mailmanager.pl
 - Browser-derived headers
 - HTML Parameter Mask
 - HTTP User Parameter Modifier
 - User Parameter
 - HTTP HTML Link Parser
 - Spider
 - HTTP Header Manager
 - HTTP Request Default Einstellungen
 - 10 x
 - Recording Controller
 - View Results Tree

The right pane shows the configuration for the selected **HTTP Request**:

HTTP Request

Name: /de/kontakt/formular.htm

Web Server

Server Name or IP:

Port Number:

HTTP Request

Protocol: http Method: GET POST

Path: /de/kontakt/formular.htm Redirect Automatically Follow Redirects Use KeepAlive

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
-------	-------	---------	-----------------

Send a File With the Request:

Filename:

Parameter Name:

MIME Type:

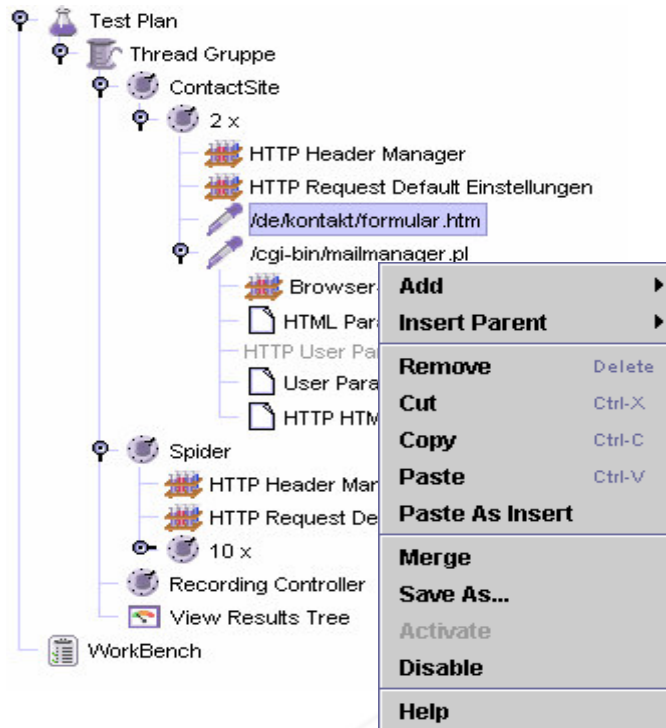
Optional Tasks

Retrieve All Embedded Resources from HTML Files Use as Monitor



JMeter Übersicht

GUI - antik, aber hilfreich



- Linke Seite
 - Testplan Baumdarstellung
 - Cut - Copy - Paste
 - Kontextsensitives Menu
 - Merge von Testplänen
 - Speicherung von Teilen
 - Immer nur ein Testplan



JMeter Übersicht

GUI - antik, aber hilfreich

- Rechte Seite - Detailansicht zur
 - Spezifikation von Testelementen
 - Anzeige von Testergebnissen

HTTP Request

Name: /de/kontakt/formular.htm

Web Server

Server Name or IP:

Port Number:

HTTP Request

Protocol: http Method: GET POST

Path: /de/kontakt/formular.htm Redirect Automatically Follow Redirects Use KeepAlive

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

Send a File With the Request:

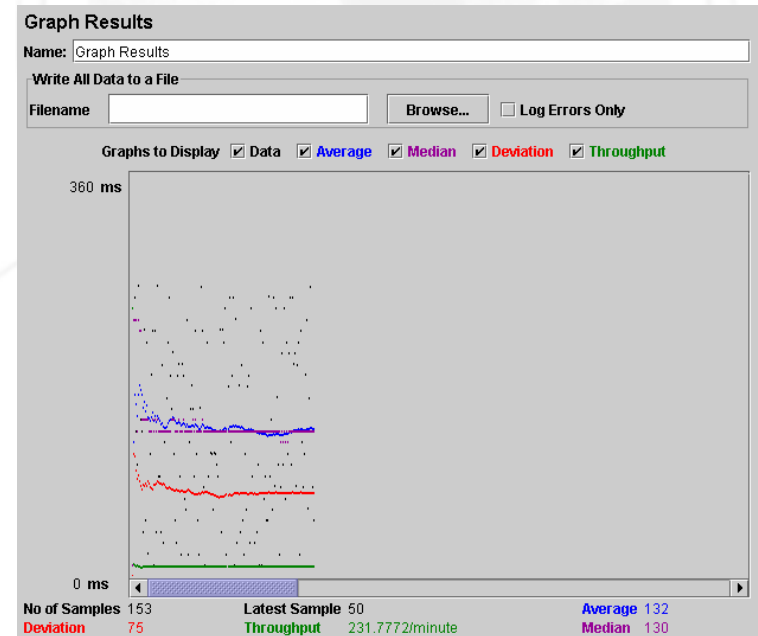
Filename:

Parameter Name:

MIME Type:

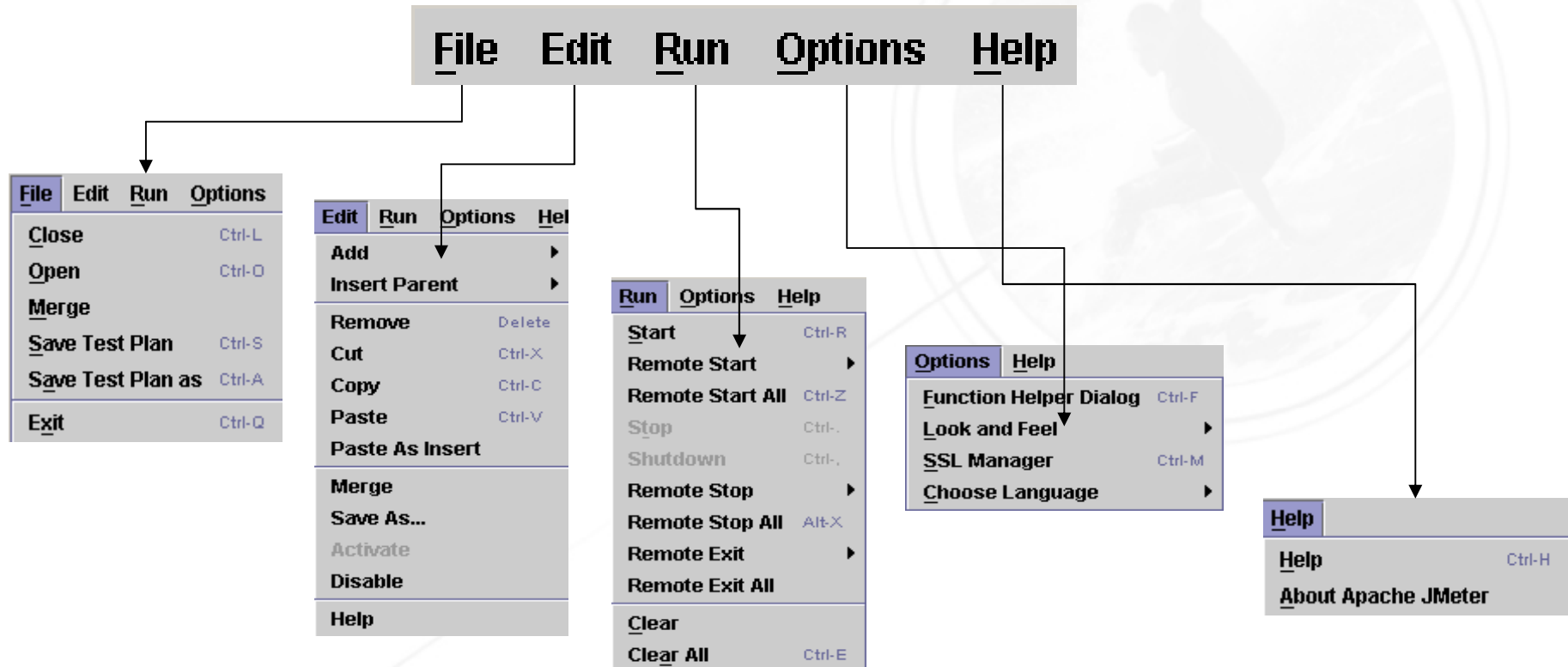
Optional Tasks

Retrieve All Embedded Resources from HTML Files Use as Monitor



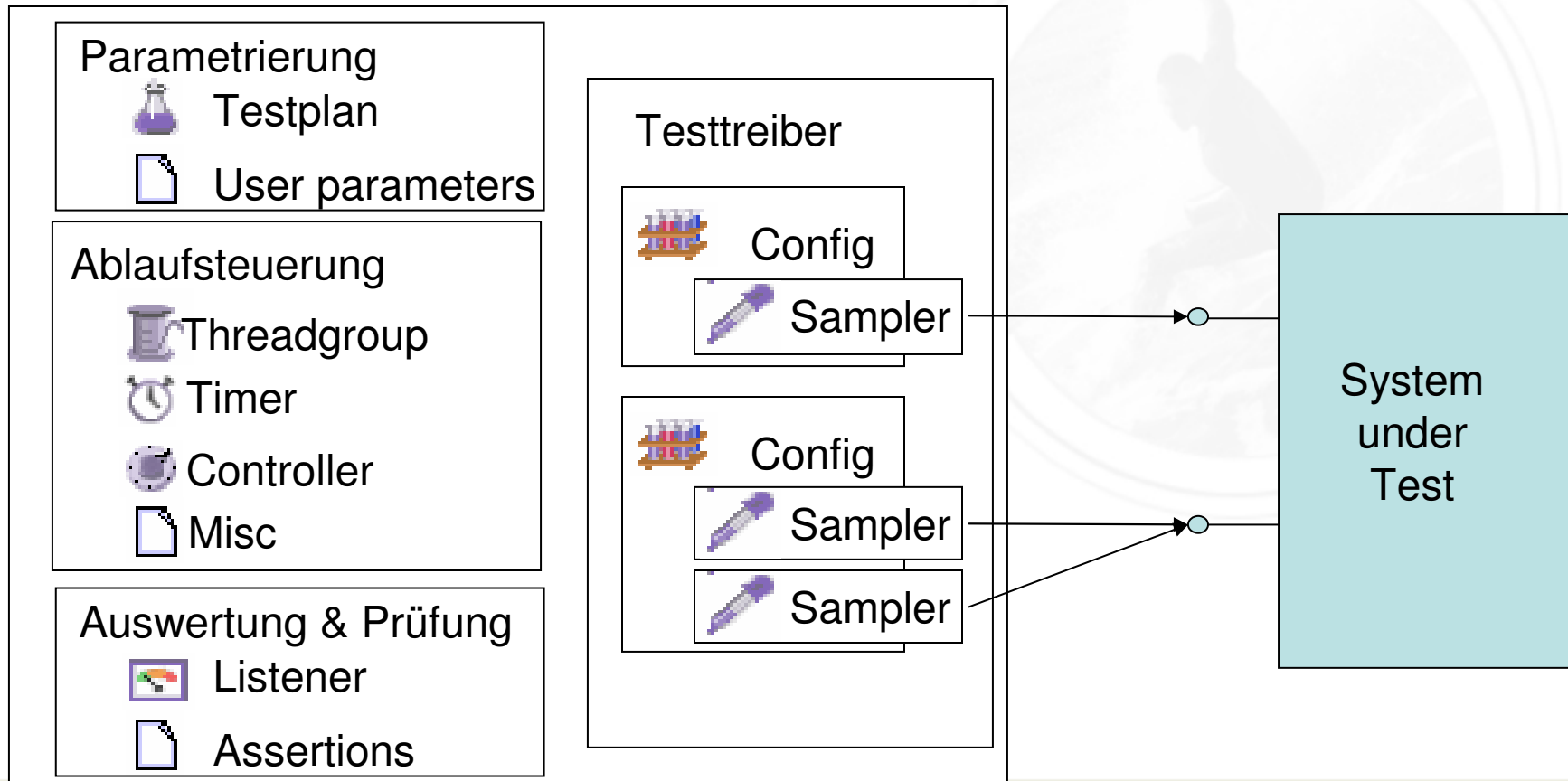
JMeter Übersicht

GUI - antik, aber hilfreich



JMeter Übersicht

Zentrale Elemente



JMeter Übersicht

Zentrale Elemente - Threadgroup



- Steuerung des Rampups
- Parallelisierung des Testablaufs
- Wiederholanzahl
- Scheduling
- Fehlerverhalten

Thread Group

Name:

Action to be taken after a Sampler error

Continue Stop Thread Stop Test

Thread Properties

Number of Threads:

Ramp-Up Period (in seconds):

Loop Count: Forever

Scheduler

Scheduler Configuration

Start Time:

End Time:

Duration (seconds):

Startup delay (seconds):

JMeter Übersicht

Zentrale Elemente - Sampler



- Ausführung einer Anfrage
 - *HTTP*
 - SOAP, Webservice
 - LDAP
 - JDBC
 - Java Requests (Test!)
 - TCP
 - Beanshell, BSF
 - (Tomcat Access Log)
 - Junit
 - JMS
 - Mail Reader

The screenshot shows the 'HTTP Request' configuration dialog in JMeter. It includes fields for Name, Web Server (Server Name or IP, Port Number), Protocol, Method (GET/POST), Path, and checkboxes for Redirect Automatically, Follow Redirects, and Use KeepAlive. There is a table for 'Send Parameters With the Request' with columns for Name, Value, Encode?, and Include Equals?. Below that are fields for 'Send a File With the Request' (Filename, Parameter Name, MIME Type) and 'Optional Tasks' (Retrieve All Embedded Resources from HTML Files, Use as Monitor).

Name:	Value	Encode?	Include Equals?



JMeter Übersicht

Zentrale Elemente - Config



- Parametrierung mehrerer Sampler gleichen Typs
 - *HTTP*
 - JDBC
 - LDAP
 - TCP
 - Java
- Definition von Variablen
- Manager
 - HTTP Authorization (Simple)
 - HTTP Headers
 - HTTP Cookies

HTTP Request Defaults

Name:

Protocol:

Server Name or IP:

Path:

Port Number:

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?



JMeter Übersicht

Zentrale Elemente - Controller



- Ablauf und Durchsatzsteuerung
 - Ablauf (wie oft, wann, etc.)
 - Durchsatz
 - Zeitlich
- Strukturierung (Simple, Include)
- Parametrierung (Zufälligkeit)
- Unterstützte Typen
 - If, While, *For Each*, Switch
 - Throughput,
 - Module, Simple
 - Random, Random Order
 - Interleave, OnlyOnce

ForEach Controller

Name: ForEach Controller

Input variable prefix: input

Output variable name: forEachInput

Add "_" before number ?

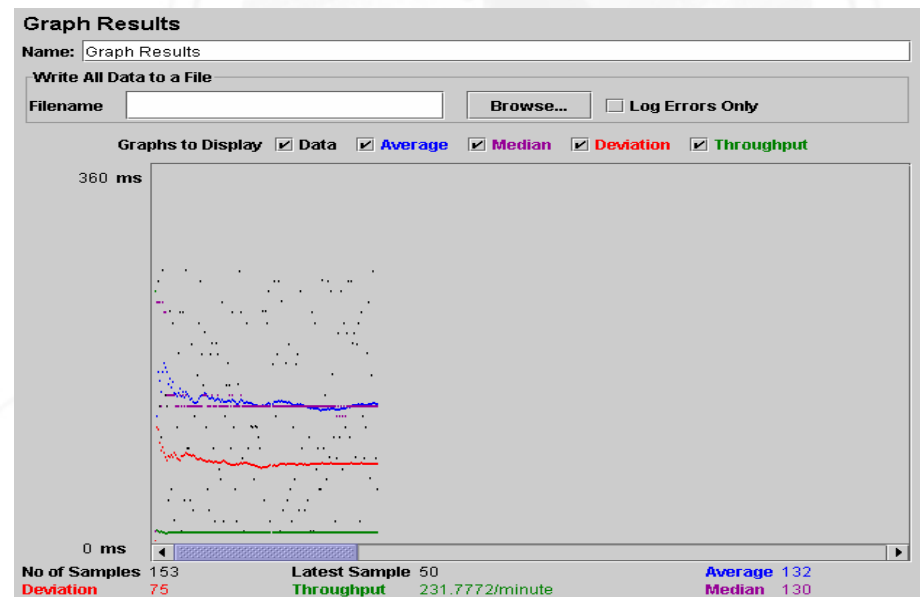


JMeter Übersicht

Zentrale Elemente - Listener/Visualizer



- Zusammenfassung von Ergebnissen in
 - tabellarischen Ansichten
 - grafischen Ansichten
 - Dateien
- Monitoring Tomcat (ab 5.0.19)
- Email-Benachrichtigung
- Unterstützte Typen
 - Aggregate Report
 - View Results in Table
 - Graph Results
 - View Results Tree
 - Simple Data Writer
 - Mailer
 - Assertion Results



JMeter Übersicht

Zentrale Elemente - Timer



- Steuerung
 - Durchsatz
 - Zeitlich
- Parametrierung (Zufälligkeit)
- Unterstützte Typen
 - Constant Timer
 - Uniform Random Timer
 - Gaussian Random Timer
 - Synchronizing Timer
 - *Constant Throughput Timer*

Constant Throughput Timer

Name:

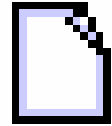
Delay before each affected sampler:

Target throughput (in samples per minute):



JMeter Übersicht

Zentrale Elemente - Assertions



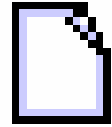
- Überprüfung der durchgeführten Sampler auf
 - Inhalt
 - Dauer
 - Größe
 - „Wohlgeformtheit“
 - Identität
- Unterstützte Typen
 - Response
 - Beanshell (größte Freiheit ;-)
 - Duration
 - MD5Hex
 - HTML Tidy Prüfung
 - Size
 - XML, *XPath*

The screenshot shows the configuration window for an XPath Assertion in JMeter. The window title is "XPath Assertion". The "Name" field contains "XPath Assertion". Under the "XML Parsing Options" section, there are four unchecked checkboxes: "Tolerant XML/HTML Parser", "Use Namespaces", "Validate XML", and "Ignore Whitespace". The "XPath Assertion" section contains a text field with the XPath expression "//ok[3]" and a "Validate" button. Below the text field, there is an unchecked checkbox labeled "True if nothing matches".



JMeter Übersicht

Zentrale Elemente - Preprocessors



- Benutzerabhängige
 - Parameterwerte
 - Zähler
- Parametrierung von Requests bzgl.
 - Linkadressen
 - Werten von Eingabemasken
 - Session Ids
- Unterstützte Typen
 - User Parameters
 - *Interaktiv*
 - Aus XML Datei
 - HTML Link Parser
 - HTML Parameter Mask
 - HTTP URL Rewriting Modifier

User Parameters

Name:

Update Once Per Iteration

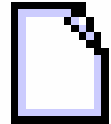
Parameters

Name:	User_1	User_2	User_3
username	Martin	Christine	Andreas
visitedPages	5	15	25



JMeter Übersicht

Zentrale Elemente - Postprocessors



- Auswertung und Speicherung von Ergebnissen
- Steuerung der Testfortführung
- Unterstützte Typen
 - *Regular Expression Extractor*
 - Result Status Action Handler
 - Save Responses to file
 - Generate Summary Results

Regular Expression Extractor

Name: Regular Expression Extractor

Response Field to check: Body Headers

Reference Name: whileCondition

Regular Expression: Failed

Template:

Match No. (0 for Random):

Default Value: false



JMeter Übersicht

Zentrale Elemente - Sonstiges



- *HTTP Proxy Server - Recording*
- Workbench - JMeter „Scrapbook“

The screenshot shows the configuration window for an HTTP Proxy Server in JMeter. The window title is "HTTP Proxy Server". The "Name" field is set to "HTTP Proxy Server". The "Port" is set to "8080". There are three checkboxes: "Capture HTTP Headers" (checked), "Set Keep-Alive" (checked), and "Add Assertions" (unchecked). The "Target Controller" is set to "Use Recording Controller". The "Grouping" is set to "Do not group samplers". There are two sections for patterns: "Patterns to Include" and "Patterns to Exclude", each with an "Add" and "Delete" button. At the bottom, there are "Start", "Stop", and "Restart" buttons.

JMeter Übersicht

Gültigkeitsbereich der Elemente

- Folgende Elemente betreffen Vaterknoten und all seine Kinder
 - Config Elements
 - Pre-/Postprocessors
 - Assertions
 - Timers
 - Listeners

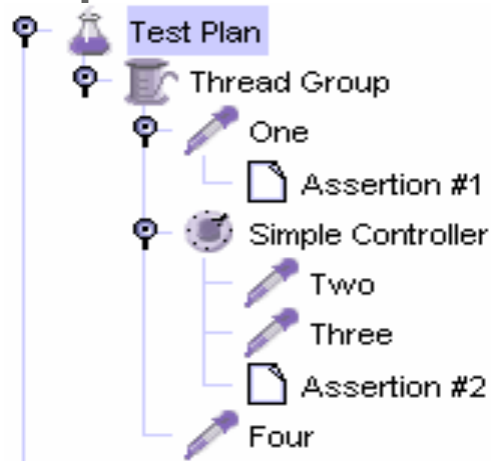


Sampler One
Sampler Two
Sampler Three
Sampler Four

JMeter Übersicht

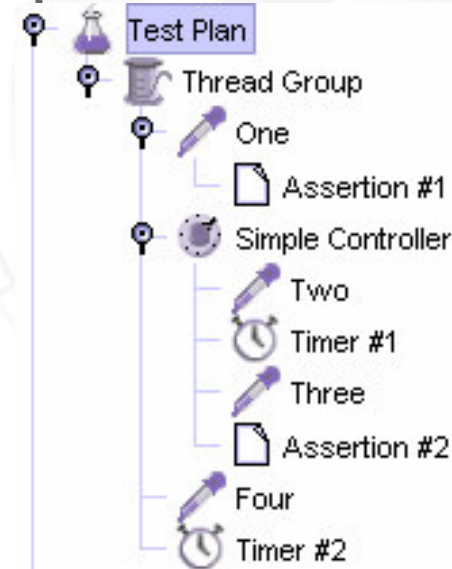
Gültigkeitsbereich der Elemente

- Beispiel 1



One mit A #1
Two mit A #2
Three mit A #2
Four

- Beispiel 2



One mit A #1 + T #2
Two mit A #2 + T #1 + T #2
Three mit A #2 + T #1 + T #2
Four mit T #2



JMeter Übersicht

Weiteres Wissenswertes

- Dateiformat
 - XML (mit Suffix jmx)
 - Elementtypen node, element, collection, property
- Ausführungsmöglichkeiten
 - Interaktiv
 - BatchMode
 - Verteilt über mehrere Rechner
 - Ant TaskDef

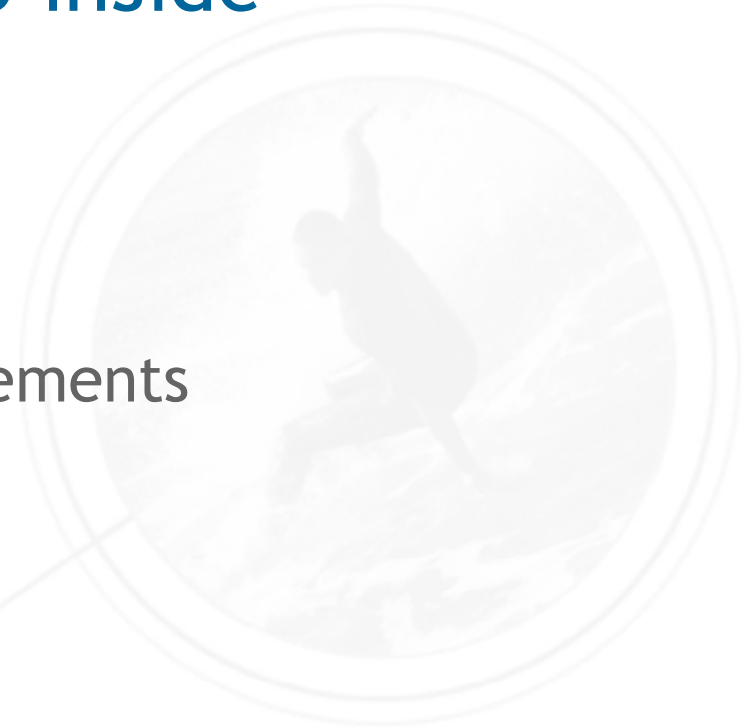


JMeter - Deep inside

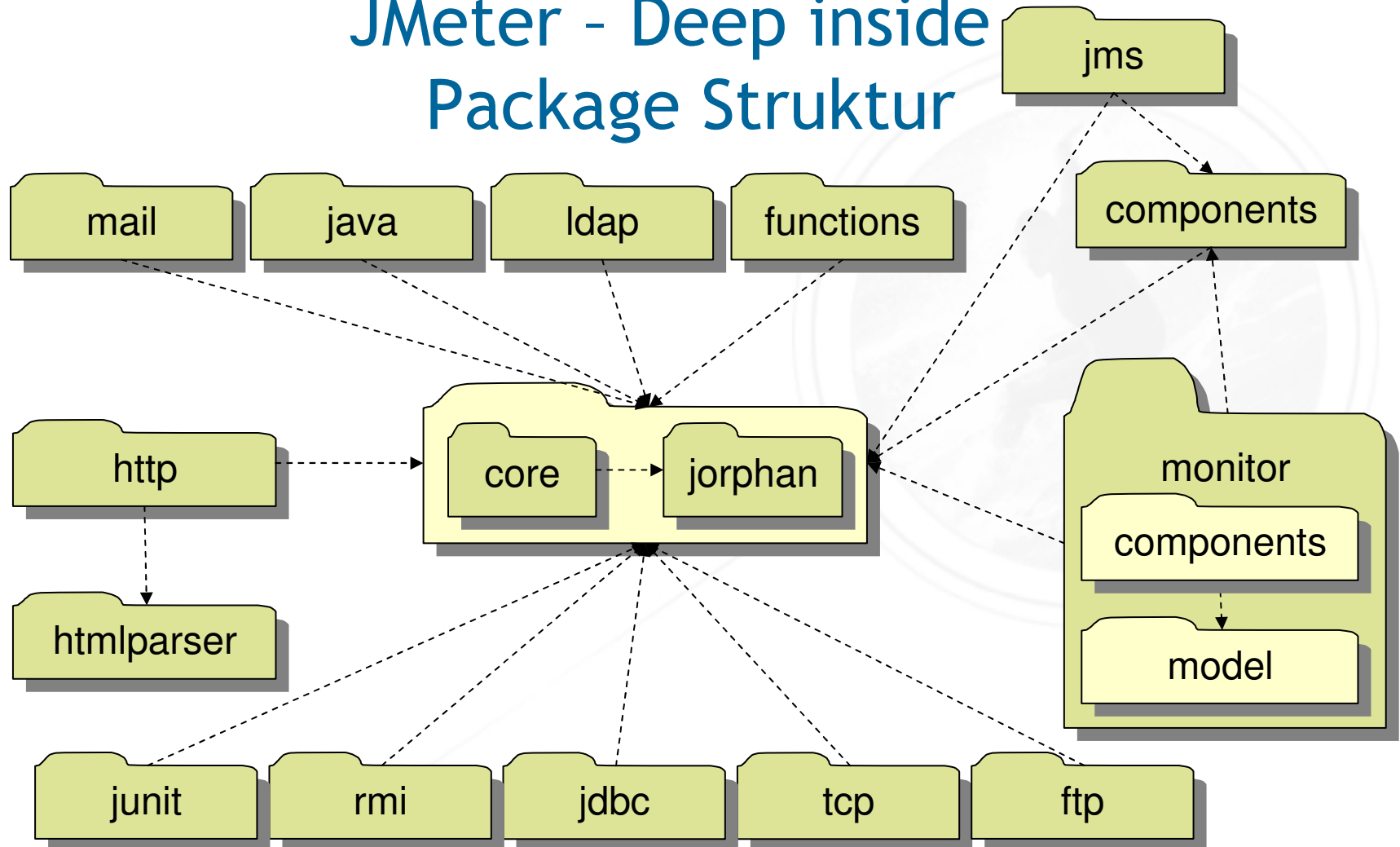


JMeter - Deep inside

- Package Struktur
- Wichtige Interfaces
- Zusammenspiel GUI und TestElements



JMeter - Deep inside Package Struktur



JMeter - Deep inside Zentrale Interfaces

- GUI

JMeterGUIComponent

...

Abstrakte Basisklassen

*Abstract*Gui*

...

AbstractVisualizer

...

Verbleibende zu implementierende Interfaces

von JMeterGUIComponent

TestElement createTestElement()
modifyTestElement(TestElement)
[configure(TestElement)]
String getLabelResource()

Visualizer

add(SampleResult ())
boolean isStats()

Clearable

clear()



JMeter - Deep inside

Zusammenspiel GUI und Testelemente

JMeterGUIComponent

`void createTestElement();`
Erzeugt zugeordnetes Testelement aus GUI

`void modifyTestElement(TestElement)`
Aktualisiert Model mit Werten aus GUI

`void configure (TestElement)`
Aktualisiert GUI mit Werten aus Model

JMeter - Deep inside

Zentrale Interfaces und Klassen

- Model

TestElement

...

Abstrakte Basisklassen und konkrete Implementierungen

Abstract...

...

ResultCollector

...

Verbleibende zu implementierende Interfaces

Preprocessor

void process()

Sampler

SampleResult sample(...)

Controller

void initialize
boolean isDone
Sampler next()
...

Postprocessor

void process()

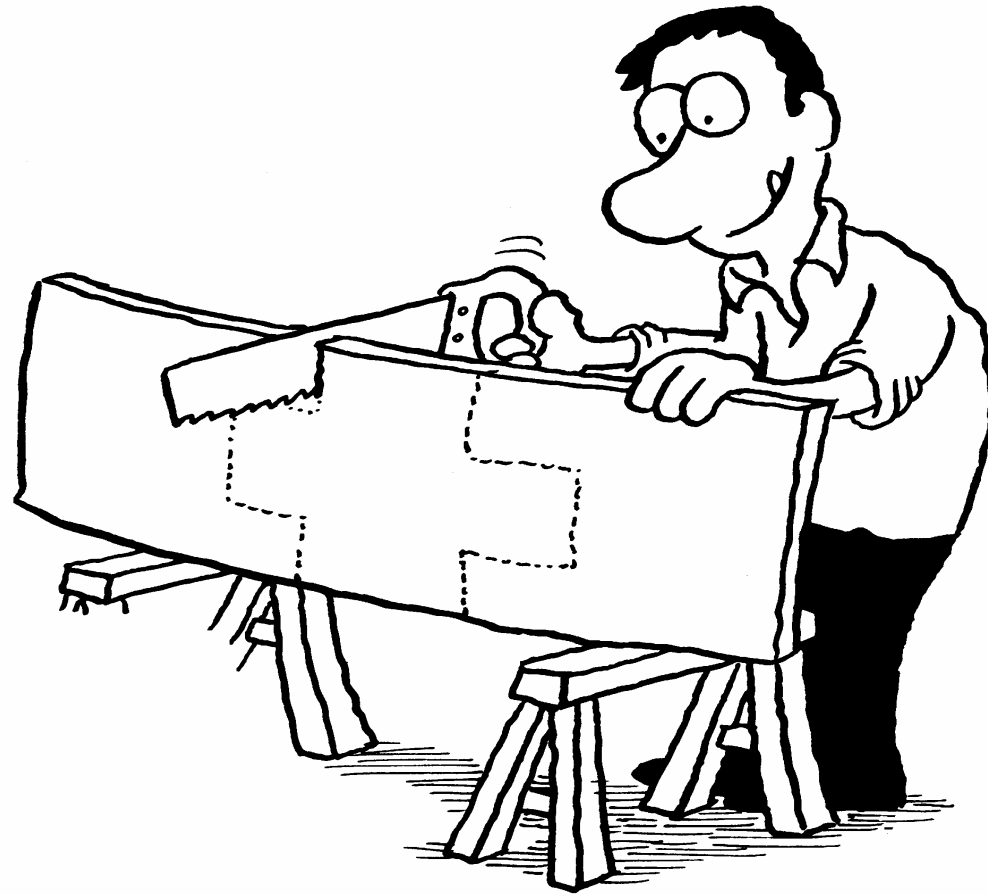
Timer

long delay()

Reicht meist
aus 😊



JMeter Offroad - Erweiterungen entwickeln



JMeter Offroad - Erweiterungen entwickeln

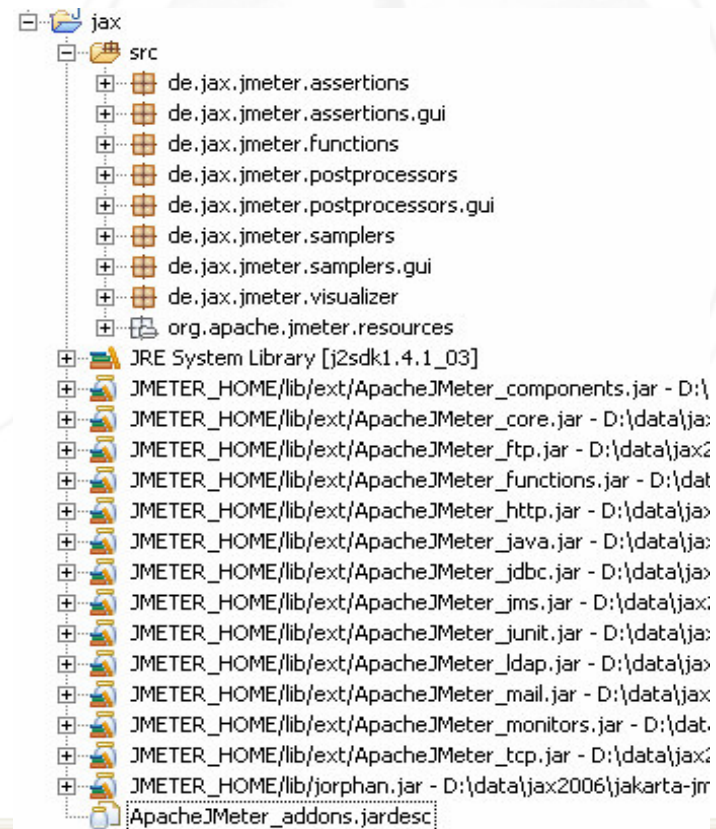
- Vorbereitende Maßnahmen
- Entwicklung eigener JMeter Elemente in Eclipse
- Im Projekt entwickelte JMeter Plugins



JMeter Offroad - Erweiterungen entwickeln

Vorbereitende Maßnahmen

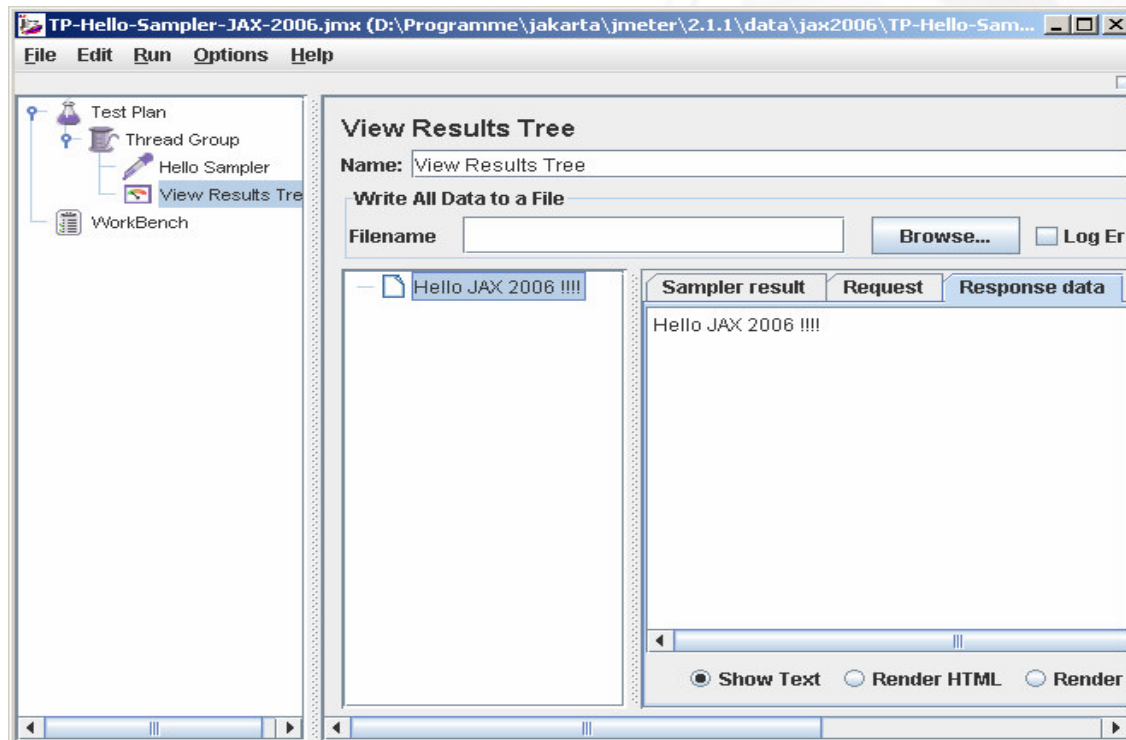
- Eclipse Projekt anlegen, ApacheJMeter*.jar und jorphan.jar einbinden
- Kopie der Resource-Dateien aus src\core\org\apache\jmeter\resources* für eigene Resource-Erweiterungen
- Extrahierte Resources aus ApacheJMeter_core.jar entfernen
- Deployment der *.class und geänderten Ressourcen als ApacheJMeter_addons.jar
- Kopie der ApacheJMeter_addons.jar in JMeter-Laufzeitumgebung nach <JMETER_HOME>\lib\ext



JMeter Offroad - Erweiterungen entwickeln

Entwicklung eines neuen Samplers

- Eintauchen in den Code und Präsentation in JMeter ...



JMeter Offroad - Erweiterungen entwickeln

Im Projekt entwickelte JMeter Plugins

- JMX Sampler
- HL7 Sampler
- CommandLine / BatchSampler
- AntScript*Sampler
- SSH / SFTP Sampler
- Comment Controller
- XPath Postprocessor (analog Regular Expression Extractor)
- Regular Expression Counter
- HL7Assertion
- FormattedActualDateFunction
- ...

Fazit / Zusammenfassung



Fazit / Zusammenfassung

JMeter Pros und Cons

- ☺
 - Grundausstattung an Visualizern, Sampler, Controller, etc.
 - Produktivitätskurve durch grafische Testplan-Erstellung
 - Erweiterungsfähigkeit
 - Java Bibliotheksunterstützung
 - Planübergreifenden Includes seit Version 2.1
- ☹
 - Hoher Ressourcenverbrauch
 - Keine Verwaltung mehrerer Pläne
 - Kein HTTPS Recording

Fazit / Zusammenfassung

Weiterführende Links und Quellen (I)

- Bücher
 - Performance Analysis for Java Web Sites, Stacy Joines, Ruth Willenbourg and Ken Hygh, Addison Wesley, 2003
 - J2EE Performance Testing, Peter Zadrony, Expert Press, 2002
 - J2EE-Entwicklung mit Open-Source-Tools, Martin Backschat / Stefan Edlich, Spektrum Verlag, 2004 (ein paar Seiten)
- Artikel
 - JMeter Tutorial unter <http://wiki.apache.org/jakarta-jmeter/JMeterLinks>



Fazit / Zusammenfassung

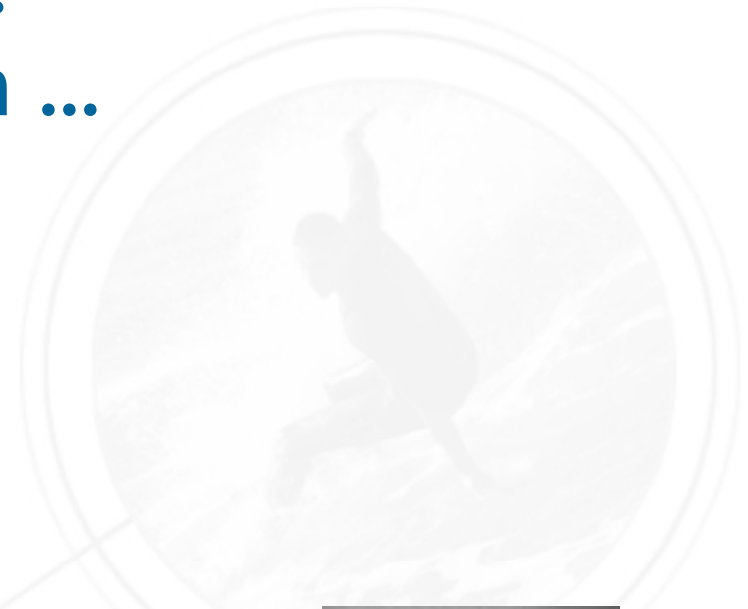
Weiterführende Links und Quellen (II)

- Links
 - JMeter Website <http://jakarta.apache.org/jmeter>
 - Grinder Website <http://grinder.sourceforge.net/>
 - Software Performance Engineering <http://www.perfeng.com/>
 - <http://www.testingreflections.de>
 - Vergleich zwischen OpenSource Werkzeugen Grinder und JMeter http://www.infomar.de/de/consulting/Hochseetauglich_mit_Opensource_JAX2005.pdf

Fragen? Aber gern ...



Martin Heider
Infomar software
mh@infomar.de
www.infomar.de



Michael Burger
Sepp Med GmbH
michael.burger@seppmed.de
www.seppmed.de

