

Methoden zur kontinuierlichen Performancemessung bei der agilen Entwicklung von Webanwendungen

- Diplomarbeit -

Patrick-Thomas Chmielewski
Institut für Informatik
Freie Universität Berlin

1. Warum ist Performance wichtig?
2. Was sind die Probleme klassischer Konzepte zur Performanceoptimierung?
3. Was wäre eine bessere Verfahrensweise?
4. Ergebnis

- Interaktion des Benutzers mit Webanwendung:
 - Seite aufrufen
 - Eingaben tätigen
 - Resultat abwarten

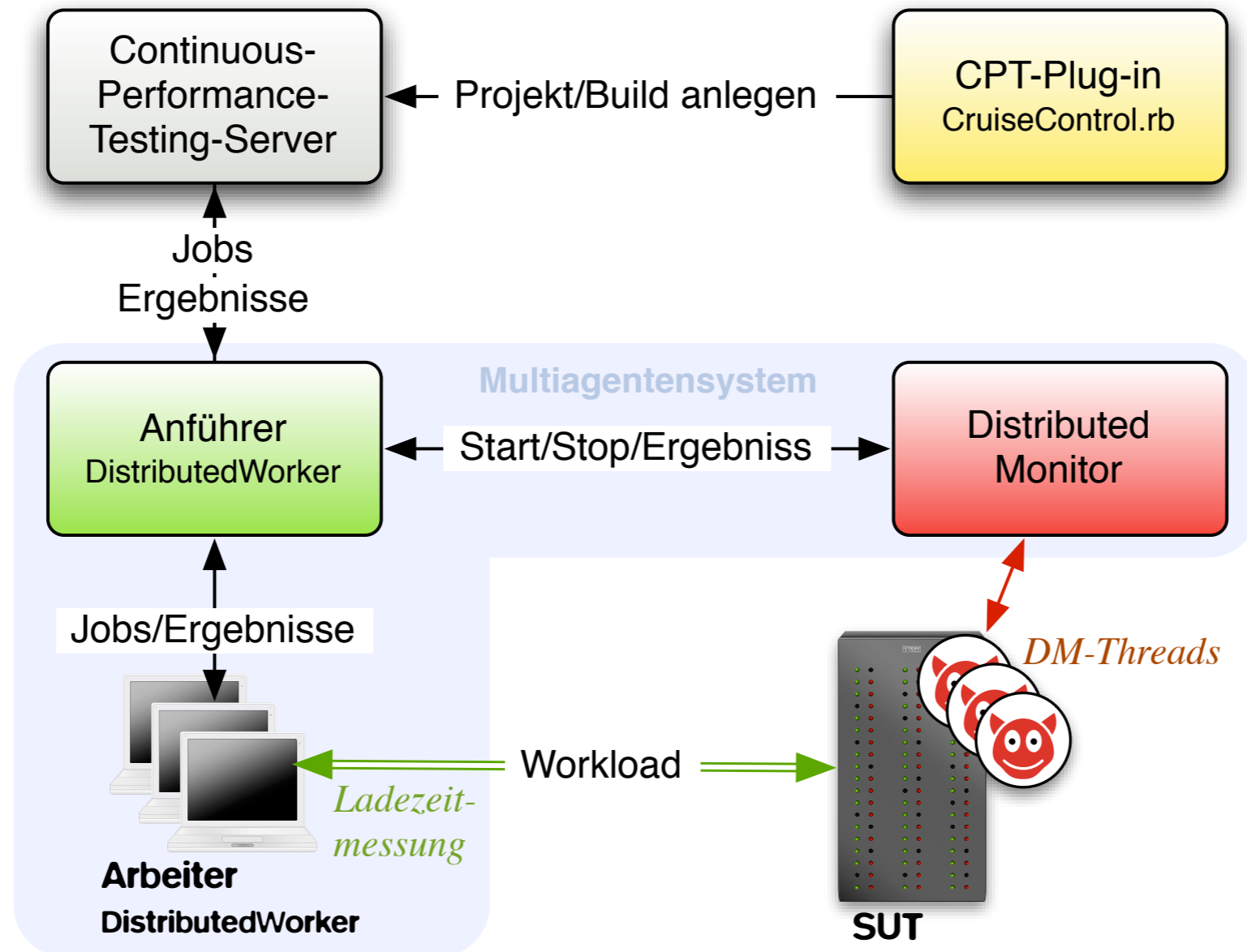
- Performance als Ladezeit
- lange Ladezeit führt zu Frustration, führt zu:
 - neg. Bewertung der Information
 - Abbruch von Transaktionen
 - negativen Assoziationen mit Marke
 - Sicherheitsbedenken
 - künftigen Meidung einer Seite

- Steigt die Ladezeit...
- ... so fällt die Zufriedenheit
- ... so fällt der Umsatz/Gewinn!

- Problem bei klassischem Vorgehen:
 - Performancetests oft erst spät
 - Kosten der Optimierung steigen mit der Zeit
 - Auswirkungen von Codeänderungen werden nicht direkt erfasst

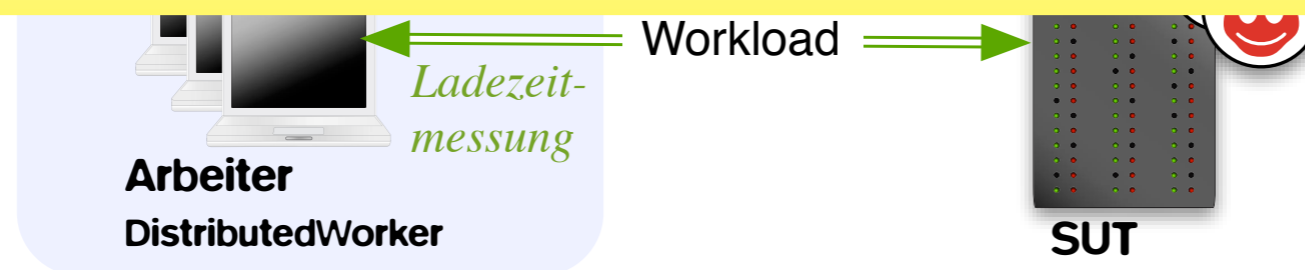
- Lösungsmethode:
 - Performancetests früh und fortdauernd
 - schon während der Entwicklung
 - häufige Performanceoptimierung durch ständige Refaktorisierung
 - Speicherung der Ergebnisse zur Erkennung von Tendenzen

- *ContinuousPerformanceTesting*:
 - zentraler Server mit GUI
 - Multiagentensystem
 - Integration in *CruiseControll.rb*



```

Scenario: Add T-Shirt into the cart and increment quantity
  Given 3 stress clients
  When I go to spree
  And I follow "Clothing"
  And I follow "Ruby on Rails Jr. Spaghetti"
  And I press "Add To Cart"
  And I would like to measure the loadtime
  And I type in "100000" for "order_line_item_attributes_\d+_quantity"
  And I press "Update"
  And I'd like to remember the preceding loadtime
  And it should load within 5 seconds
  And it should return elapsed loadtime
    
```



Projektübersicht

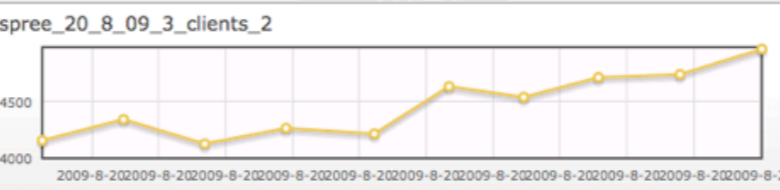
ContinuousPerformanceTesting

Projects Builds Scenarios Worker

Listing projects

« Previous 1 2 3 4 5 Next »

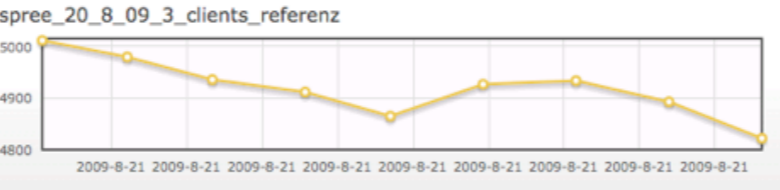
spree_20_8_09_3_clients_2



2009-8-20 2009-8-20 2009-8-20 2009-8-20 2009-8-20 2009-8-20 2009-8-20 2009-8-20 2009-8-20 2009-8-20

Show details...

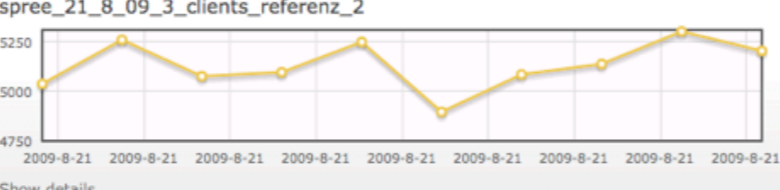
spree_20_8_09_3_clients_referenz



2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21

Show details...

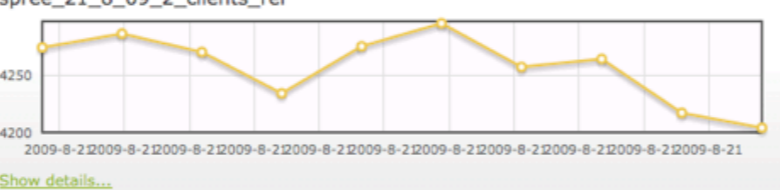
spree_21_8_09_3_clients_referenz_2



2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21

Show details...

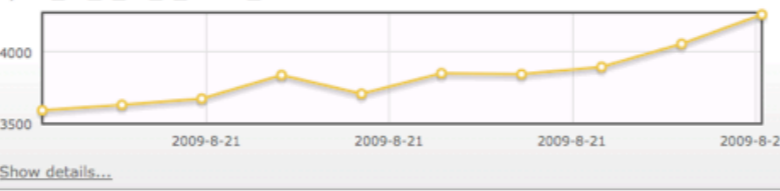
spree_21_8_09_2_clients_ref



2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21

Show details...

spree_21_8_09_2_clients_1



2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21 2009-8-21


Show details...

« Previous 1 2 3 4 5 Next »

[New project](#)

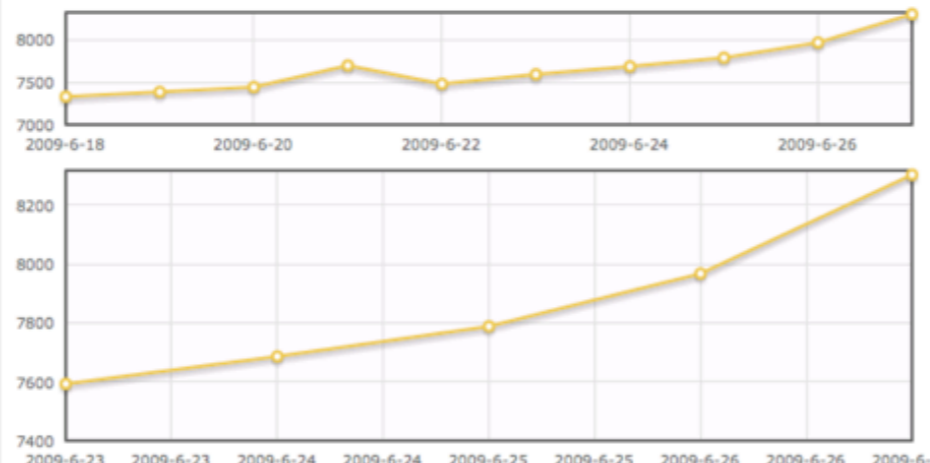
© 2009 Infopark AG XHTML | CSS | Design by Kulor

Buildsauflistung



Projects Builds Scenarios Worker

Total Performance



« newer 1 2 older »

Listing builds

dfcb54465e4cefde4279938edab1979a57d2d817 Sat Jun 27 00:00:00 2009 +0000 Show details...
8da74d8fd5bffb4d629671f14b16f4d2c556f480 Fri Jun 26 00:00:00 2009 +0000 Show details...
d4f4c1f5e74a1d2cfb0f1d70969e146914b39600 Thu Jun 25 00:00:00 2009 +0000 Show details...
667a13a7b6aeaa605dbca016baa3a9d0e61c937c Wed Jun 24 00:00:00 2009 +0000 Show details...
302749476e3c58518deb0aa7fcd5b315a7e2b033 Tue Jun 23 00:00:00 2009 +0000 Show details...

« newer 1 2 older »

[New build](#)

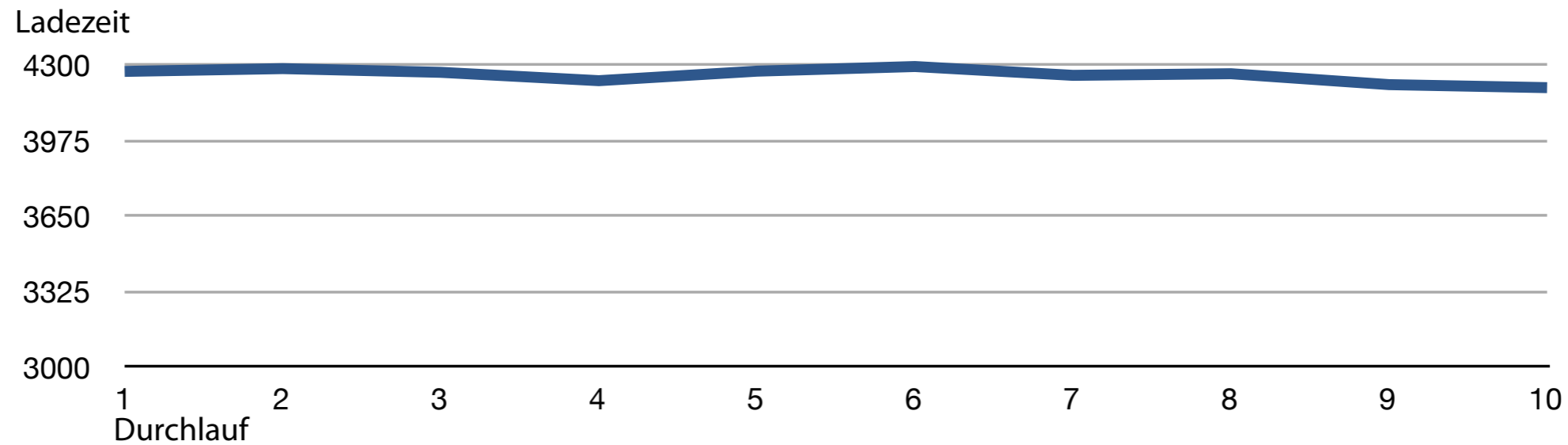
© 2009 Infopark AG XHTML | CSS | Design by Kulor

Szenarioauflistung

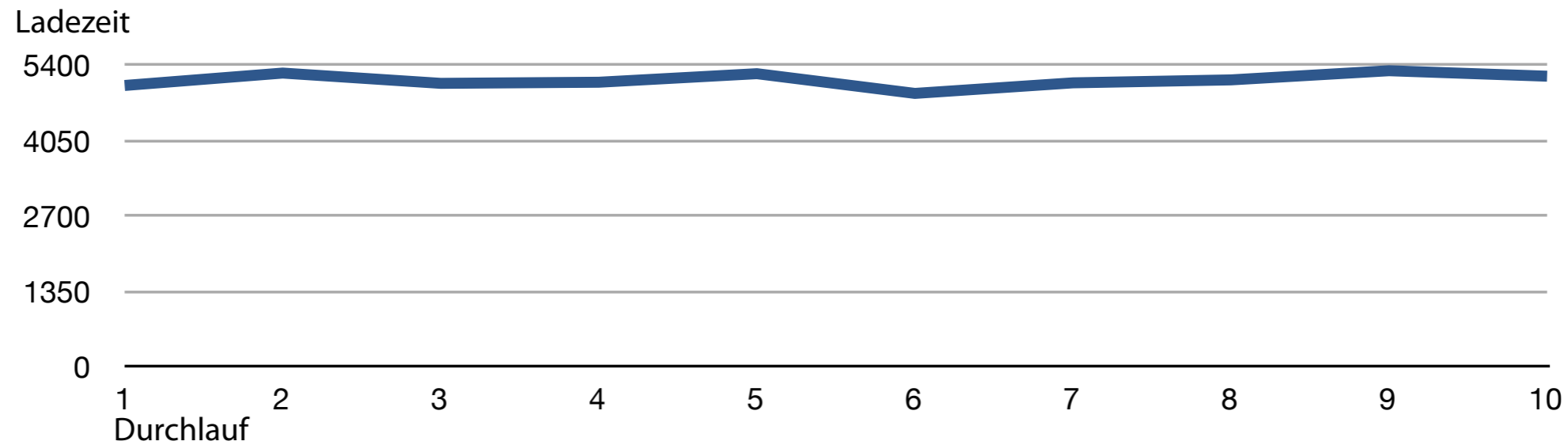


- *Evaluierung: Quelloffener Webshop „Spree“*
 - Referenzmessung
 - simulierte kontinuierliche Messung
 - zwei bzw. drei Stressclients
 - fünf Szenarien
 - 25 Durchläufe je Revision
 - 10 Revisionen

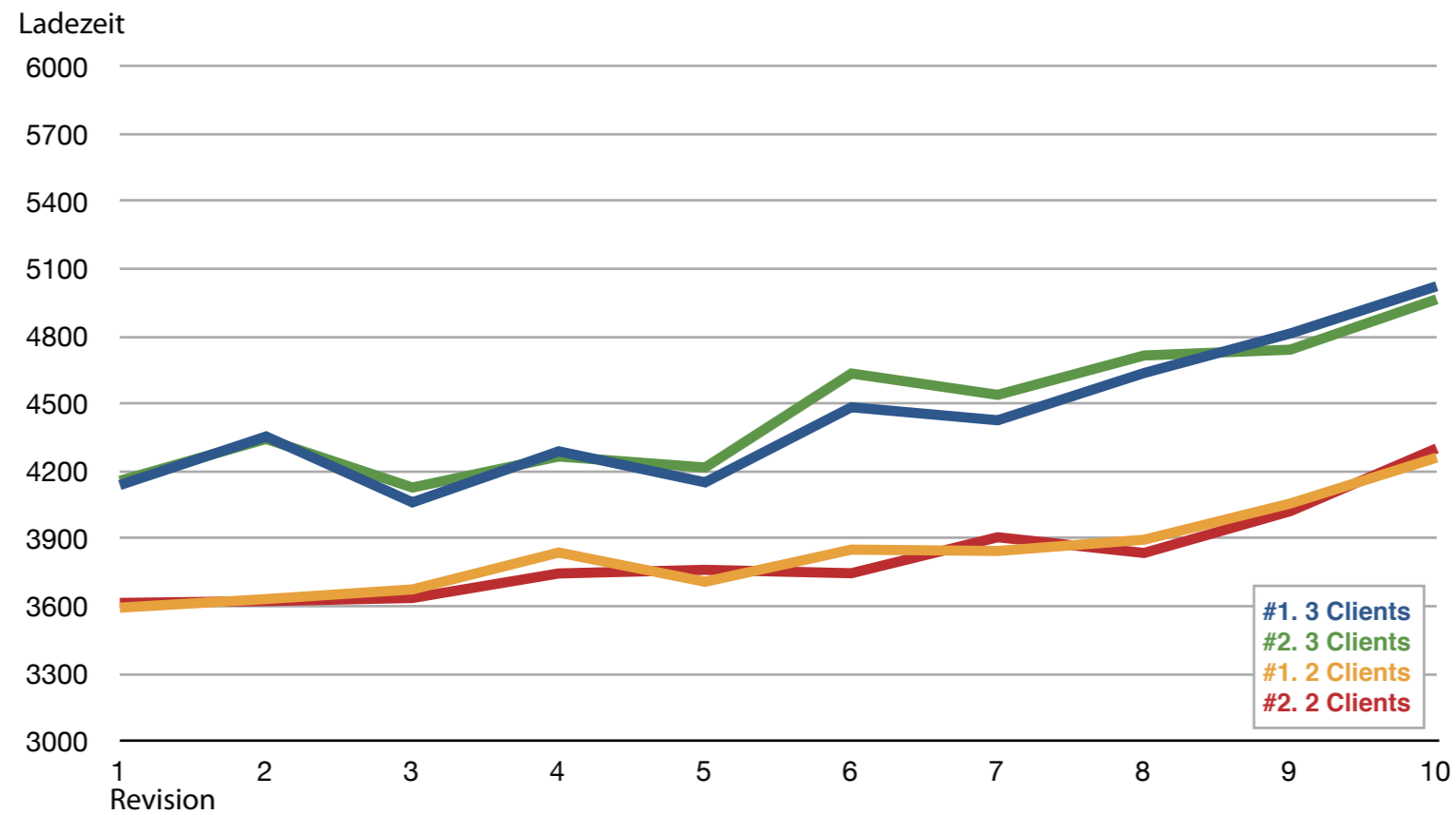
zwei Stressclients



drei Stressclients

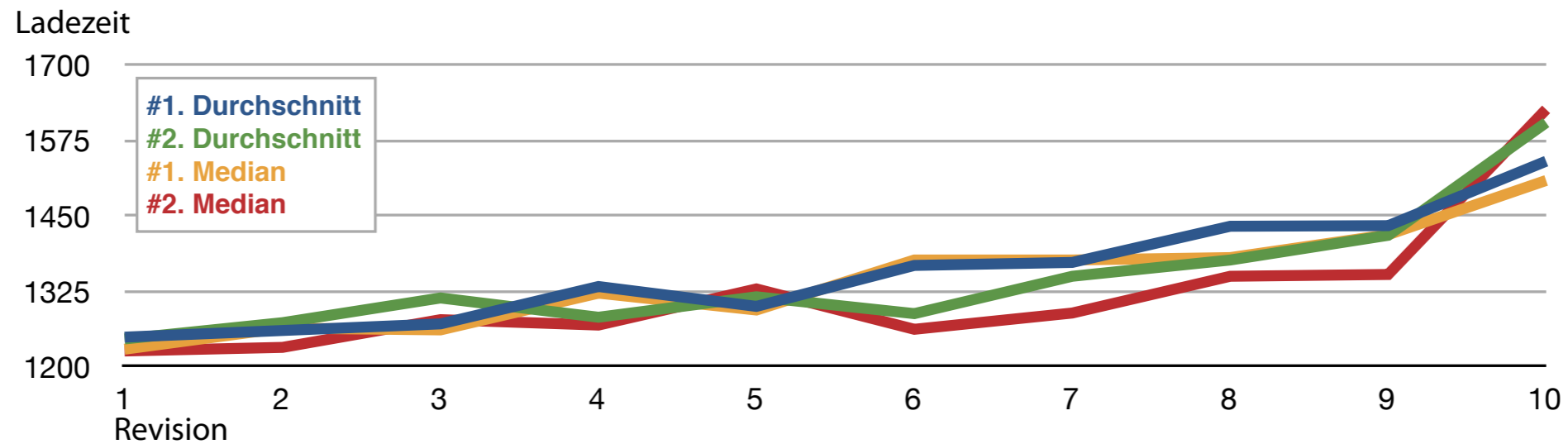


Gesamtperformance simulierte kontinuierliche Messung

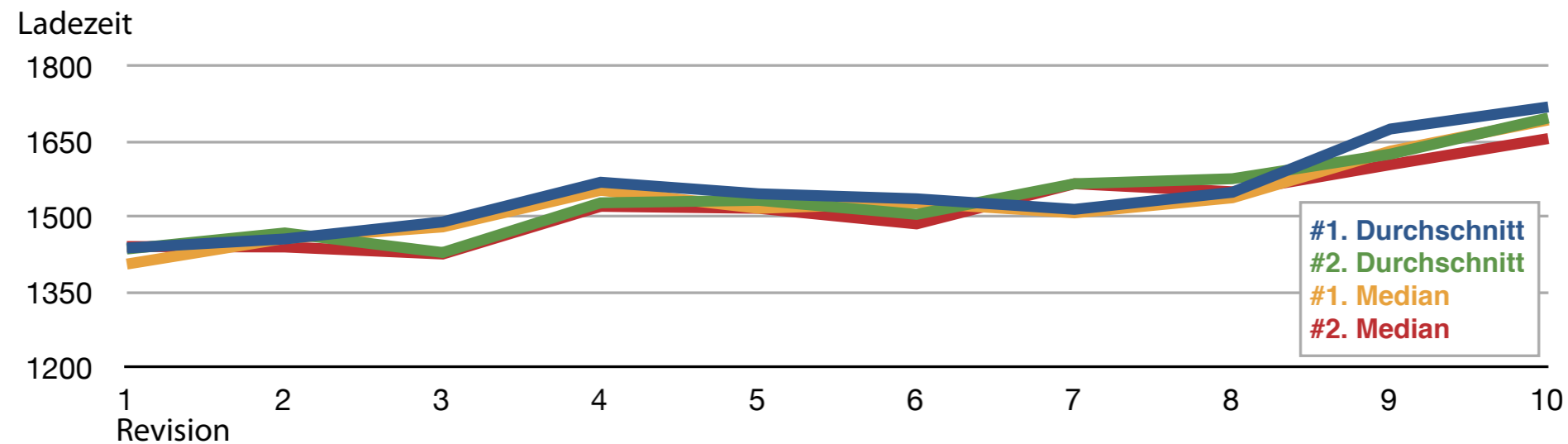


Problematische Szenarien

scenario add t-shirt to cart and continue shopping



scenario browse through categories



- *ContinuousPerformanceTesting-Prototyp...*
- ... liefert stabile Messergebnisse
- ... zeigt Tendenzen hinsichtlich der Performance auf
- ... erlaubt kontinuierliche Performancetests im Anschluss an Integrationstests

- *Performance wichtig wegen Zufriedenheit + Umsatz*
- Klassisches Vorgehen zu spät + oft kostspielig
- Besser: kontinuierliche Performancetests schon während der Entwicklung
- *Continuous Performance Testing*-Prototyp zeigt frühzeitig Tendenzen auf

- Wie wird kontinuierlicher Performancetest von Entwicklern aufgenommen?
- Verbessert sich in Folge die Performance der Webanwendung?
- Wie mit dynamischer Prozessortaktung umgehen?

Fragen?

Danke!