

) OIO Kompass : Java Web-Frameworks)

Mannheim, 2011

Eine Studie zu den Hintergründen der Auswahl von Java Web-Frameworks.





Die Softcopy dieser Studie erhalten Sie über unsere Homepage unter Veröffentlichungen.

Wir werden wieder jedem Teilnehmer die Studie gezielt einzeln und vorab zukommen lassen.

Die neue Studie finden Sie unter der folgenden URL:

<http://www.oio.de/kompass/ueberblick-java-web-frameworks-vergleich/>

Ihr mobiler Client führt Sie mit einem Scan des QR-Codes direkt dorthin.

Inhaltsverzeichnis

Vorwort	2
Das Web: Stagnation, Evolution und Reaktion	4
Rad, das Web und Web-Frameworks	7
Reichweite der Studie	9
Alter der Entscheidungen	11
Web-Entwicklung ist Produkt-Entwicklung	12
Web-Frameworks wollen sorgfältig gewählt werden	16
Der IE steht nicht mehr alleine auf der Liste	19
Architekturwandel	21
Im Backend wenig Experimente	23
Vom Äusserem nicht blenden lassen	24
Erkenntnisse aus den Ressourcen- und Lastanforderungen	26
Die gewählten Web-Frameworks	28
Zuspitzung in den letzten zwei Jahren	32
Fazit	34



Vorwort

Orientation in Objects (OIO) ist das Expertenhaus für Softwareentwicklung mit Java und XML. Seit 1998 prüfen wir neue Technologien und Standards schon während ihrer Entstehung auf Potenzial und Marktfähigkeit und schulen und beraten unseren Kunden dementsprechend zukunftsicher bzw. wenden dieses Wissen in den Innovations-Projekten unserer Kunden praxisbezogen an.

Unser Unternehmen ist in drei Geschäftsbereiche gegliedert: Competence Center bietet nebst einem umfassenden Angebot von Seminaren und Coaching auch unabhängige Beratung und Gutachten zu technologischen Themen an. Software Factory realisiert und wartet schlüsselfertige Individual-Software, saniert auch fremdentwickelte Software und migriert bestehende Software nach Java Technologie. Object Rangers unterstützt Projektvorhaben bei Kunden vor Ort mit eingespielten Teams oder auch mit erfahrenen Softwareingenieuren in Schlüsselpositionen.

Unsere Ingenieure entwickeln Software in Kundenprojekten und schulen und beraten parallel mit ihrem speziellen Themenschwerpunkt im Competence Center, deshalb gilt: „Ihre Softwareentwickler sind auch Ihre Trainer und Berater“.

Seit über 10 Jahren beschäftigt sich OIO somit auch mit der Auswahl von passenden Java Webframeworks für die jeweiligen individuellen Kundenbedarfe.

Aus diesen Erfahrungen heraus haben wir ein Standard-Instrument für die jeweilige Kick-Off Veranstaltung einer solchen Auswahlentscheidung in Form eines Fragebogens entwickelt.

Im Jahr 2011 fiel der Entschluss, diesen Fragebogen erstmals in Form einer Online-Umfrage einer breiten Öffentlichkeit zu präsentieren und gleichzeitig nach den tatsächlich gewählten Java Web Frameworks zu fragen.

Die beiden Hauptziele dieser Umfrage waren:

- Die Korrelation unserer Fragen mit möglichen Antworten / Lösungsentscheidungen auf einer breiteren Basis zu überprüfen und somit neue Erkenntnisse für unsere Beratungstätigkeit zu sammeln
- Möglicherweise dabei gewonnenen Erkenntnisse der Öffentlichkeit zur Verfügung zu stellen und dann mit diesem Ergebnis als erstem Inhalt eine Folge- Umfrage auf eine noch breitere Basis zu stellen.

Zusätzlich zu unseren Entscheidungskriterien haben wir daher noch einige Fragen zu der im Rahmen der Framework-Entscheidung tatsächlich getroffenen Auswahl aufgenommen. Dieses Umfrageinstrument ermittelt somit die Grundlagen einer getroffenen Entscheidung für ein Java Webframework, sei es für ein einzelnes Projekt oder gar strategisch für einen Entwicklungsbereich.

Daher werden bei einem ausreichend breiten Teilnehmerkreis Korrelationen zwischen den Kriterien und tatsächlich im Java Markt getroffenen Framework Entscheidungen möglich.



Dirk M. Sohn ist einer der Gründer und Geschäftsführer der OIO Orientation in Objects GmbH - dem Expertenhaus für Softwareentwicklung mit Java und XML in Mannheim. Seit vielen Jahren leistet er Weiterbildungs-, Prozess- und Managementberatung in diesem technologischen Umfeld. Zu diesem Tätigkeitsfeld zählt auch die Erarbeitung von Studien als Grundlage für die Auswahl von Technologien.



Dipl.-Wirtsch.-Ing. (TH) **Papick G. Taboada** ist Entwickler, Berater, Softwarearchitekt. Schwerpunkte seiner Arbeit liegen in der Konzeption und Erstellung von Softwarearchitekturen im Java-EE-Umfeld mit Open-Source-Technologien. In den letzten Jahren hat er mehrere Projekte bei der Umsetzung von Web-Anwendungen durchgeführt oder beratend begleitet. In seiner jüngsten Vergangenheit beschäftigte er sich intensiv mit dem Google Web Toolkit. Dank verschiedener Vorträge auf namhaften Konferenzen und Veröffentlichungen in anerkannten Fachzeitschriften ist er in Deutschland als GWT Spezialist bekannt geworden.



Das Web: Stagnation, Evolution und Reaktion

Über einen erstaunlich langen Zeitraum haben sich Web- Technologien nicht bewegt. Aber wie lange war dieser Stillstand?

Die Stagnation.

Von 1992 bis 1999 hat sich die HTML-Spezifikation bis zur Version 4.0.1 entwickelt. Von 1999 stammt die HTTP 1.1 Spezifikation, die im Jahr 1998 veröffentlichte CSS-Spezifikation Level 2 wird bis heute von manchen Browsern nur mangelhaft unterstützt. Das ist Status Quo im Jahre 2011.

Wendet man den Blick in Richtung Java Web-Frameworks, so haben wir vom Jahr 2000 bis ca. 2007 das Web-Framework „Struts“ als De-Facto Standard beobachtet. Das MVC-Web-Framework wird bis heute noch in vielen Projekten eingesetzt. Im Jahr 2006 wurde das bis dato wenig verbreitete JavaServer Faces (JSF) in der Version 1.2 in die Java EE Spezifikation aufgenommen.

Das fehlende Komponentenmodell in Struts, die langsame Umsetzung neuerer JSF Versionen (die Version 2.0 stammt von Mitte 2009 und wird noch nicht von allen Anbietern unterstützt) sowie diverse anfängliche Kompatibilitäts- und Performanceprobleme haben in der Zeit ab 2008 Raum für die Verbreitung neuer Ansätze und Frameworks jenseits des Java Standards geschaffen.

So florierte die Framework-Landschaft in den letzten fünf Jahren prächtig, es gab Gewinner und Verlierer, beliebte und unbeliebte Ansätze. Auch andere Programmiersprachen als Java wurden im Java EE Umfeld salonfähig, so wurden auch Frameworks wie Rails und Grails bekannt.

Doch dann kam die Evolution.

Ajax- ein Akronym, das auf Technologien des letzten Jahrtausends basiert, hat die gesamte Web-Industrie durchgeschüttelt. Ob es nun Ajax oder das Web 2.0 war, oder ob einfach nur die Zeit reif war, ist heute schwer zu sagen. In Bezug auf die Entwicklung von Web-Anwendungen wurde JavaScript salonfähig.

Plötzlich wurden Anforderungen an Browser gestellt, die nicht nur Anwender, sondern auch Browser-Hersteller staunen ließen. Die Browser waren für den massiven Einsatz von JavaScript nicht optimiert. Die Hersteller haben schnell reagiert: So lieferte Firefox bald Release um Release zweistellige Performance-Gewinne. Als dann Safari, Chrome und Opera bei dem Rennen um Performance und Kompatibilität einstiegen, war ein neuer Browser-Krieg entfacht - allerdings anfänglich ohne den Internet Explorer. Das war der Anfang vom Ende der IE-Herrschaft, heute werden Firefox und Chrome in der Breite eingesetzt.

Die Browser der Zukunft werden noch mehr können. Bis dato wurden die Grenzen einer Web-Anwendung dem Entwickler sehr schnell sehr deutlich. Diese Grenzen sollten mit HTML5 fallen. Auch wenn die Spezifikation bis heute noch nicht fertiggestellt ist, so wird sie von modernen Browsern bereits größtenteils unterstützt.

In der jüngsten Vergangenheit hat zudem eine kleine Revolution stattgefunden: der Desktop hat ernsthafte Konkurrenz bekommen. Mobile Geräte wurden mit den Netbooks zum ersten Mal zum Verkaufsschlager, trotz kleiner Bildschirme und geringerer Leistung.

Dann wurde aus dem Nichts heraus ein ganz neuer Formfaktor auf den Radar der Web-Entwickler katapultiert: Smartphones und Tablets - und gerade im mobilen Umfeld hat der Internet Explorer keinen nennenswerten Marktanteil.



Die Reaktion

Die Veränderungen im Web-Umfeld ist auch in der Java Web-Framework Landschaft bemerkbar. Einerseits sind einige JavaScript Toolkits und neue Ansätze wie GWT und RAP in den letzten Jahren beliebt geworden, andererseits hat die standardisierte Einbindung von JavaScript in der JSF Spezifikation Einzug erhalten.

Wir befinden uns in einer Phase, in denen Web-Frameworks zunehmend auf die neuen Anforderungen in Bezug auf Benutzerfreundlichkeit, Wartbarkeit und Unterstützung mobiler Geräte reagieren. Ein Phase, in der sich bestehende Lösungen behaupten müssen und neue Web-Frameworks eine Chance bekommen, mit neuen Ansätzen zu punkten.

Klar ist nur eines: das Rennen ist noch lange nicht entschieden. Und gerade deshalb veröffentlichen wir jetzt diese Ergebnisse und möchten im Jahr 2012 nochmals mit Ihrer Hilfe nachlegen.

Rad, das Web und Web-Frameworks

Als Java Entwickler ist man nicht selten damit beschäftigt, Web-Anwendungen zu implementieren. Java ist allerdings teils Programmiersprache, teils Compiler und teils virtuelle Maschine, aber kein Teil Web-Framework. So wurde über Jahre hinweg Layer für Layer spezifiziert, Laufzeitumgebungen standardisiert, Programmier- und Komponentenmodelle definiert.

Das Grundproblem ist inzwischen verstanden, allerdings berechtigterweise mehrfach unterschiedlich gelöst worden. Deshalb gibt es nicht die eine allgemein gültige Web-Framework Lösung.

In Bezug auf die richtige Bereifung ist die Wahl auch nicht immer eindeutig: Michelin, Pirelli oder Goodyear - Winter, Sommer oder M&S? Welche Felgen? Reifen unterscheiden sich im Laufverhalten und haben sogar Auswirkung auf den Verbrauch des Fahrzeugs. Wer solche Fragen korrekt beantworten will, muss sein Fahrzeug, seinen Fahrstil und seine Fahrziele gut kennen.

Während in der Rad-Metapher der Reifenwechsel schnell durchgeführt werden kann, sieht es bei Web-Frameworks deutlich anders aus: Hier kann nicht schnell ein Boxenstopp eingelegt und das Web-Framework kurz ausgetauscht werden.

So fährt manch einer mit der falschen Bereifung durch die Gegend, weil der Umbau zu aufwändig oder kostspielig oder beides wäre.

Ähnlich wie in der Rad-Metapher haben wir in Bezug auf die Web-Frameworks eine Wahl zu treffen: das richtige generische Framework für die Entwicklung der eigenen Anwendung zu finden.



Keine einfache Entscheidung, denn auch hier beziehen sich einige der Entscheidungspunkte (oder die entsprechende Gewichtung) weniger auf das Framework und mehr auf das eigene Vorhaben.

Die Entscheidung ist stark bindend, denn oft unterscheiden sich Programmiermodell, Komponenten und Architektur derartig, dass nicht selten ein Wechsel im Aufwand einer Neuentwicklung gleichkommt.

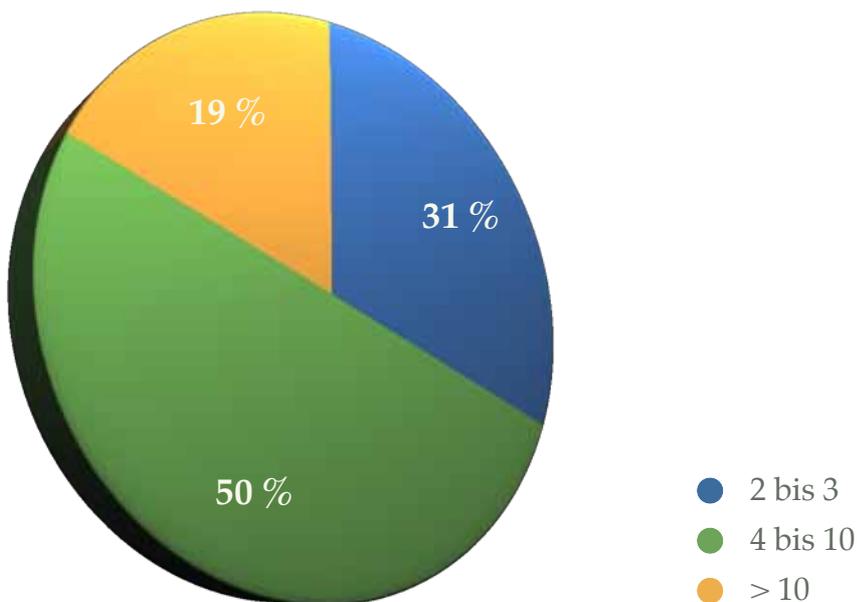
Wie wichtig die richtige Entscheidung den Teilnehmern der Studie war, wurde deutlich: Die meisten haben ausführliche Marktanalysen bis hin zur Umsetzung von Prototypen im Vorfeld der Entscheidung durchgeführt.

Und dennoch ist nicht jeder mit seiner Wahl zufrieden und sieht sich mit Problem konfrontiert, früher oder später, einen teuren Boxenstopp einlegen zu müssen.

Reichweite der Studie

Insgesamt haben 52 Teilnehmer an der Online-Befragung teilgenommen, viele davon stellvertretend für gleich mehrere Projekte oder viele Entwickler.

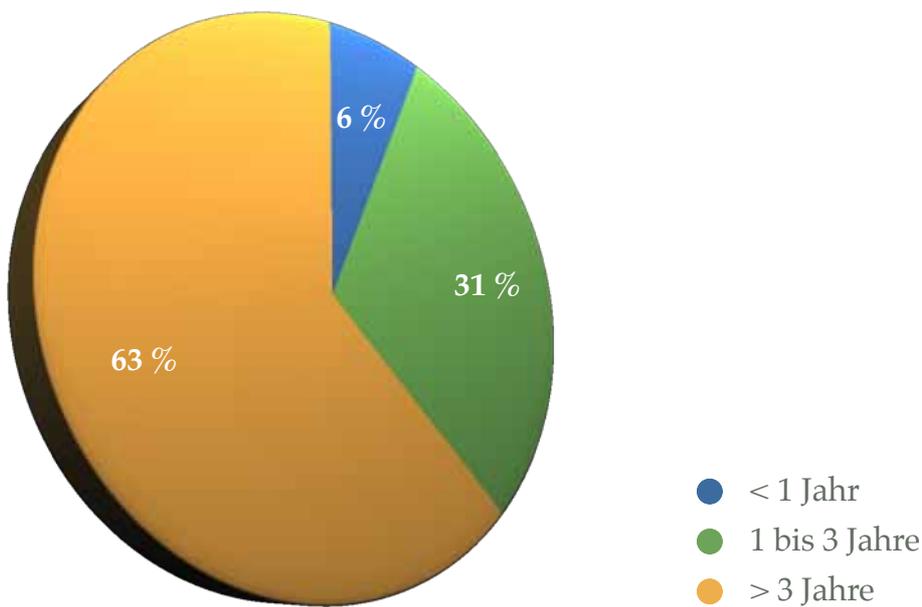
Betrachtet man die durchschnittliche Teamgröße (optionale Angabe), so gehen wir bei vorsichtiger Schätzung davon aus, dass mind. **250 Entwickler** von diesen Entscheidungen betroffen sind. Knapp ein Drittel der Teams bestand aus 2 bis 3, die Hälfte der Teams bestanden aus 4 bis 10 Entwicklern.



Anzahl der Entwickler im Client



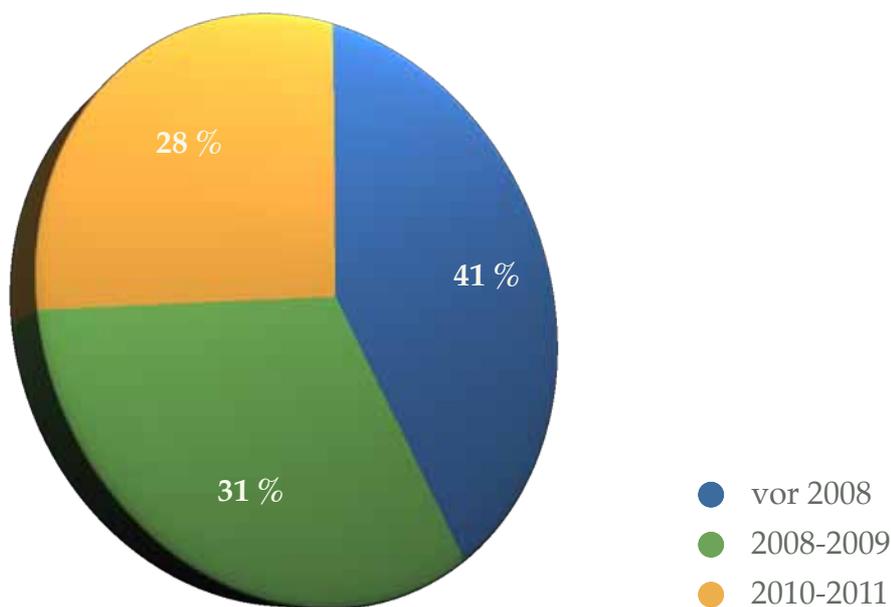
Die Anzahl der von der Entscheidung betroffenen Projekte war keine Pflichtangabe. Betrachtet man die angegebenen Antworten zum Thema Projekte und Planungshorizont, so kommen wir auf **knapp. 200 Projekte**, die unmittelbar von diesen Entscheidungen betroffen sind, und die überwiegende Mehrheit dieser Entscheidungen hatte einen Festlegungshorizont von über drei Jahren angegeben.



Horizont der Festlegung

Alter der Entscheidungen

Entscheidungen in der IT fallen von Jahr zu Jahr vor einem anderen Hintergrund, so dass die Teilnehmer auch um die Angabe des Zeitpunkts ihrer Entscheidungen gebeten wurden.



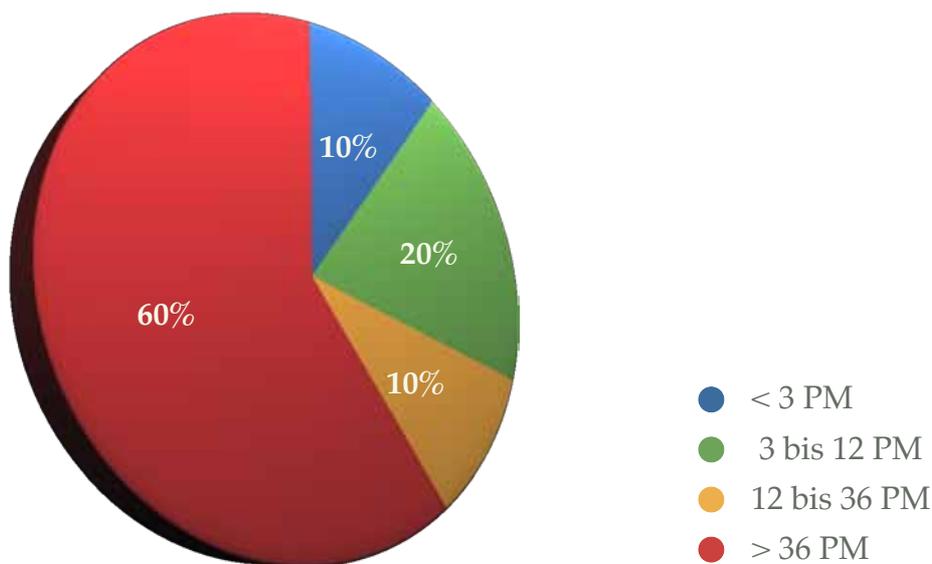
Entscheidungsjahr

Die angegebenen Antworten bezogen sich auf einen recht langen Zeitraum. In den letzten zwei Jahren waren 28%, in den letzten vier Jahren dann 59% der Antworten anzutreffen. In dem Abschnitt „Zuspitzung in den letzten zwei Jahren“ wird gezielt ein Blick auf die Entscheidungen der letzten zwei Jahre geworfen.



Web-Entwicklung ist Produkt-Entwicklung

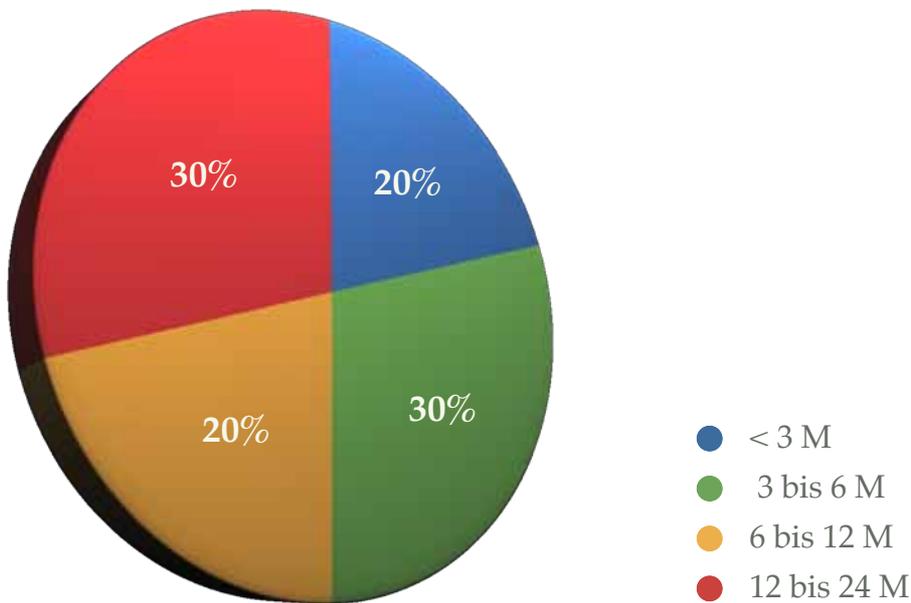
Letztlich hat nicht nur ein technologischer Wandel stattgefunden, es haben sich auch die Anforderungen an „Die typische Web-Anwendung“ in den letzten Jahren geändert. Sie ist im Vergleich zu früher größer im Umfang, komplexer im Aufbau und hat nicht selten kritische Aufgaben übernommen. Dies erkennt man nicht zuletzt auch an der Investitionsgröße:



Größe des Projekts

Betrachtet man die in der Studie angegebenen Projektgrößen, so waren die Teilnehmer der Studie überwiegend an der Umsetzung größerer Projekte beteiligt: 60% der Befragten sehen ihren Projektaufwand über 36 Personen-Monate liegen.

Betrachtet man die Dauer der Erstentwicklung, so waren lediglich 30% der Projekte nicht unterhalb eines Jahres fertiggestellt.



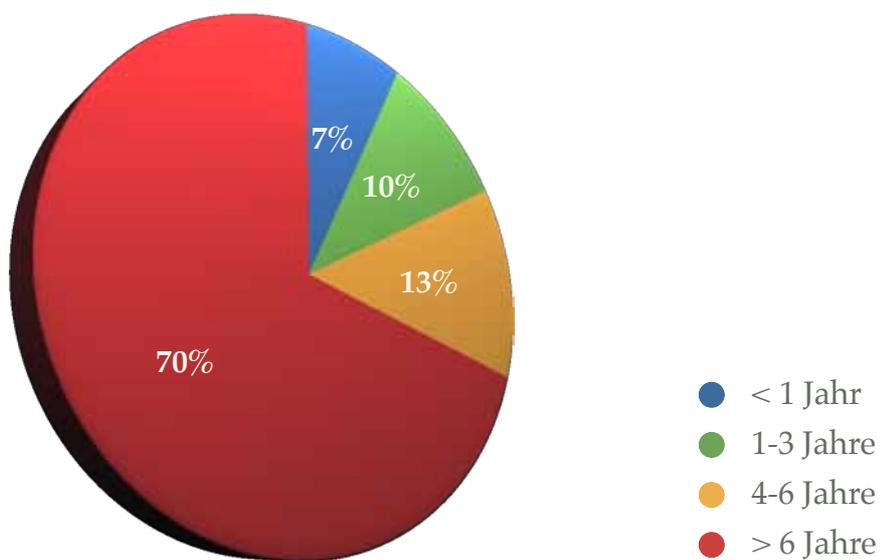
Dauer der Erstentwicklung

Die frühere Wartungsstrategie bei Web-Anwendungen war nicht selten die Neuentwicklung. Diese Strategie ist, angesichts der Komplexität heutiger Anwendungen, kaum noch durchsetzbar.



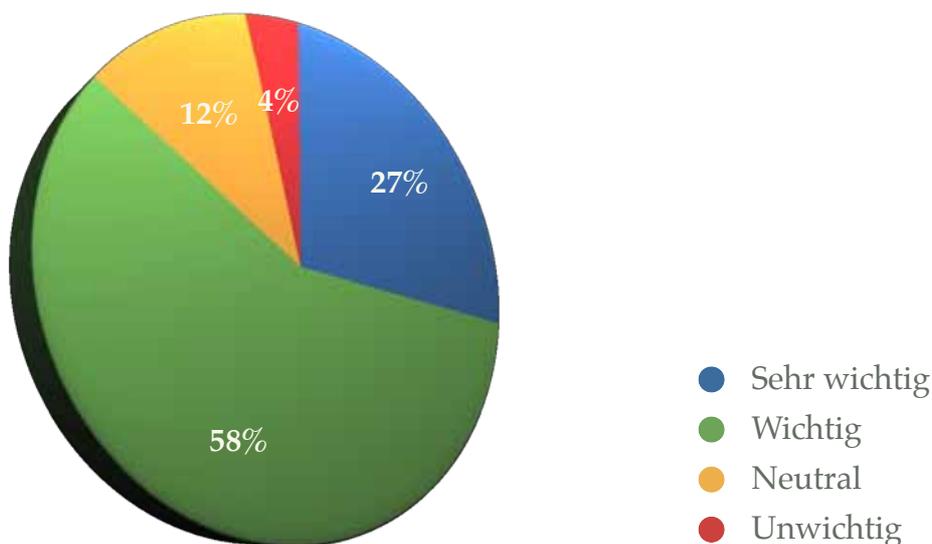
Die angegebene geschätzte Produktlebenszeit lag entsprechend weit über den Gültigkeitshorizont der Entscheidung und über der angegebenen Entwicklungszeit.

Bei 70% der Angaben lag diese geschätzte Produktlebenszeit sogar über 6 Jahren.



Produktlebenszeit

Folglich haben sich auch die Anforderungen in Bezug auf Software-Engineering verschärft: Wartung und Pflege spielen heute eine wesentliche Rolle in der Web-Entwicklung: 85% stufen „Refactoring über mehrere Schichten der Anwendung“ als wichtige Anforderung an Ihre Java Web Technologie ein.

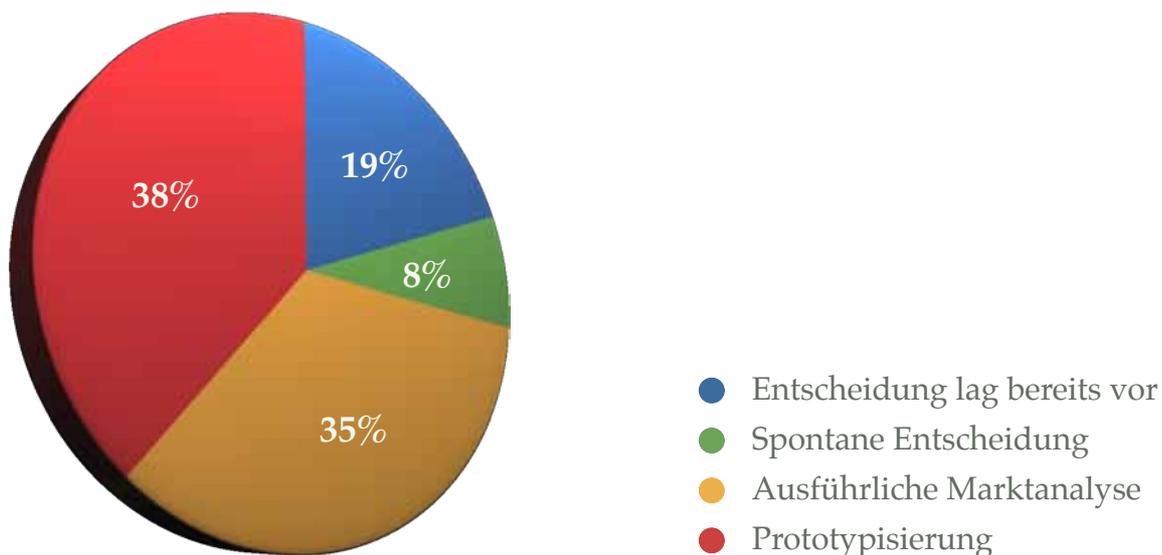


Refactoring über mehrere Schichten der Anwendung



Web-Frameworks wollen sorgfältig gewählt werden

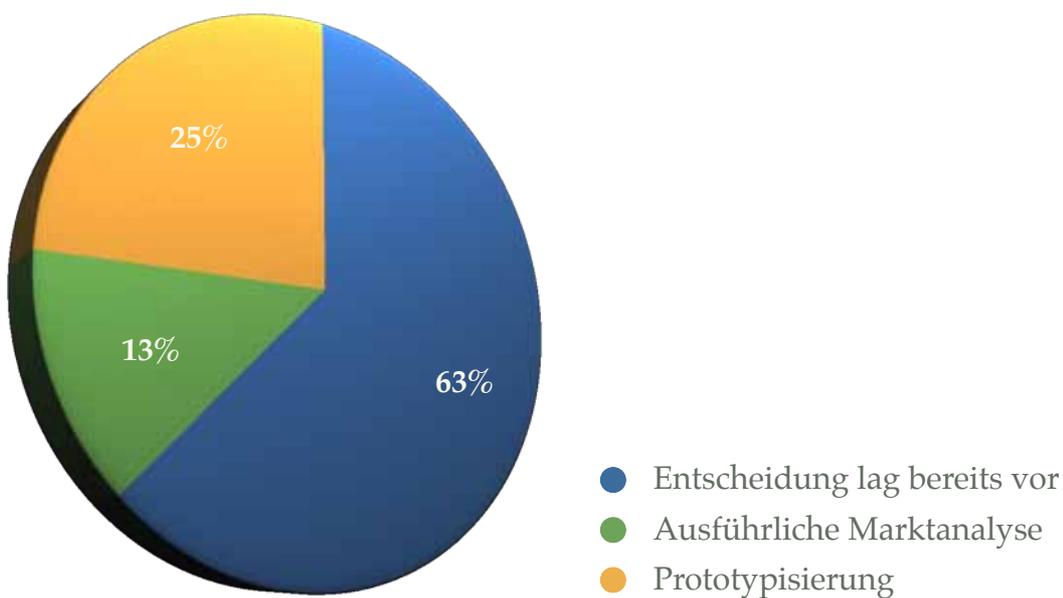
Bei der Wahl des Web-Frameworks wurde nur selten eine spontane Entscheidung getroffen. Bei 73% der Teilnehmer wurde entweder eine ausführliche Marktanalyse oder eine Prototypisierung im Vorfeld vorgenommen.



Wie wurde das Web-Framework ausgesucht?

Wie wichtig die sorgfältige Auswahl des Frameworks ist, konnten wir an der Zufriedenheit der Teilnehmer mit ihrem Framework erkennen.

Bei fast jedem fünften Teilnehmer lag allerdings die Entscheidung bereits vor und wurde nicht team bzw. projektbezogen getroffen. Betrachtet man ausschließlich die unzufriedenen Teilnehmer, so wurde hier bei 63% der Fälle weder eine Prototypisierung noch eine ausführliche Anforderungs- oder Marktanalyse durchgeführt.



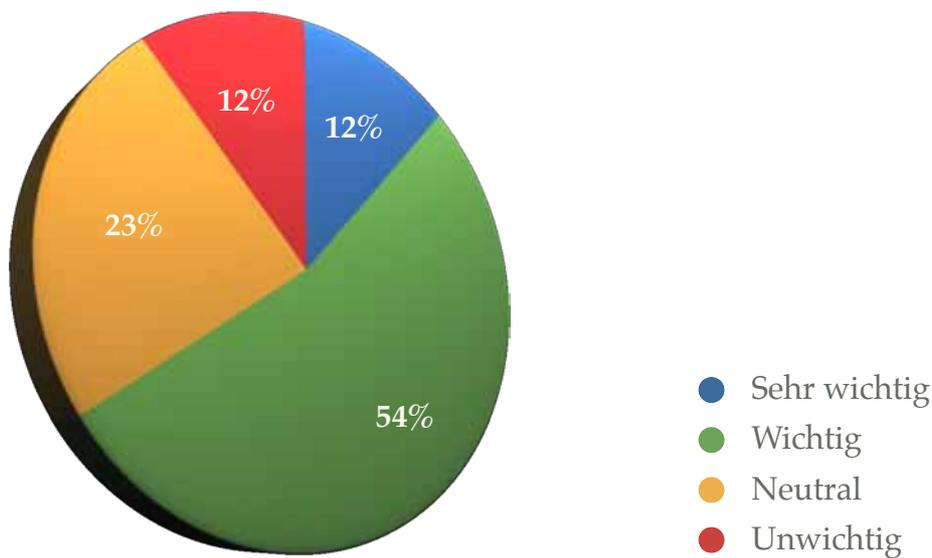
Wie wurde das Framework bei den unzufriedenen Teilnehmern ausgewählt?

Ganze 77% der Teilnehmer ist die Anbieter-Unabhängigkeit wichtig, sogar 84% der Teilnehmer gaben an, dass der Einsatz quelloffener Produkte wichtig sei bzw. sogar eine grosse Rolle bei der Entscheidung spielen würde.



Leicht im Widerspruch zu der Anbieter-Unabhängigkeit ist es 45% der Teilnehmer wichtig, Herstellergarantie zu bekommen. Welches Kriterium letztendlich im Entscheidungsprozess eine höhere Gewichtung bekommt, ist der Studie nicht zu entnehmen.

Die Konformität zu Java EE Spezifikationen spielte interessanterweise nur für 35% eine Rolle, während die Konformität zu Web Standards für 66% wichtig war.



Konformität zu Web Standards (IETF, W3C)

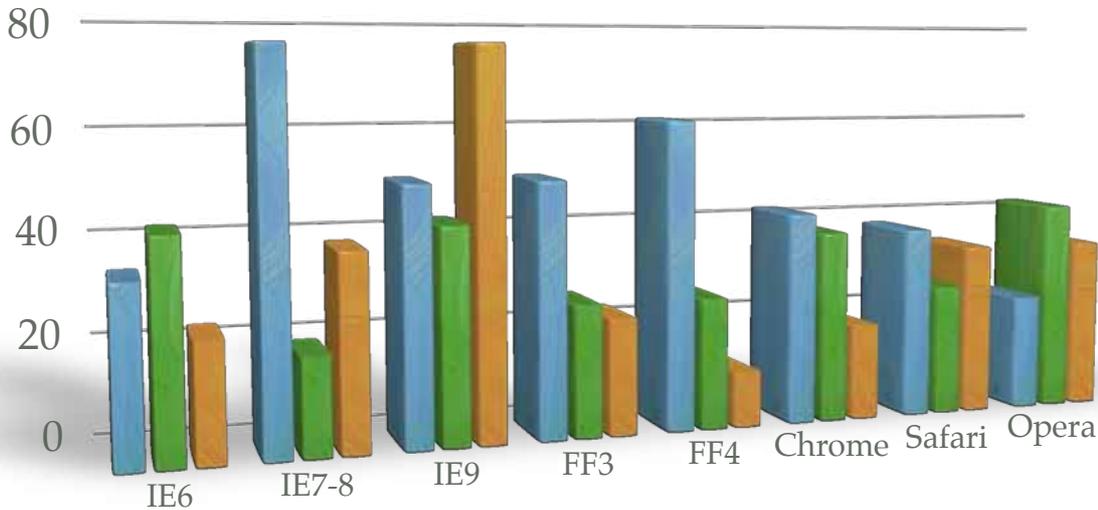
Noch eindeutiger war die Haltung in Bezug auf Marktgängigkeit und Größe der Community: 85% der Teilnehmer war es wichtig bis hin zu „spielt eine grosse Rolle“.

Der IE steht nicht mehr alleine auf der Liste

Die marktüblichen Browser-Nutzungsstatistiken zeigen, welche Browser von den Endanwendern in einer bestimmten Periode bevorzugt verwendet wurden. Je nach Art der Erhebung findet eine mehr oder weniger deutliche Abkehr vom Internet Explorer statt.

Diesen Trend können wir (nur) insofern bestätigen, dass die Kompatibilität zu anderen Browsern auch verpflichtend eingestuft wurde.

- Pflicht
- Angestrebt
- Gar nicht auf dem Radar



Browseranforderungen



Eine klare Abkehr vom Internet Explorer ist jedoch auf dem Anforderungsradar der Java Web Entwickler nicht zu erkennen:

Bei 42% der Teilnehmer wurde die Internet Explorer 6 Kompatibilität angestrebt, bei 35% wurde sie nicht nur angestrebt, sondern war sogar Pflicht.

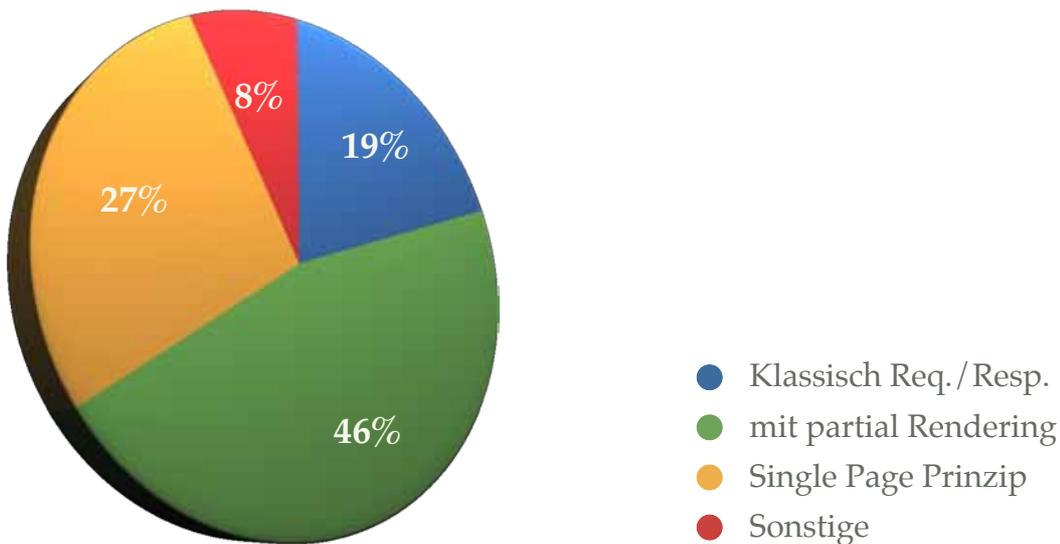
Bei 77% der Teilnehmer war die Kompatibilität zu dem Internet Explorer in den Versionen 7-8 Pflicht, bei weiteren 19% war sie angestrebt. Ähnlich auch das Bild in Bezug auf die Version 9 des Internet Explorers.

Interessanterweise war die Kompatibilität zu Firefox bei 62% der Teilnehmer Pflicht, bei weiteren 27% war diese zudem noch angestrebt. Überraschend konnte festgestellt werden, dass die Pflicht-Kompatibilität zu Chrome und Safari deutlich über der 30% - Marke lag.

Architekturwandel

Mit dem Einsatz einer Single Page Prinzip Architektur findet unter Umständen eine Abkehr von JCP diktierten Technologien bis hin zu dem Erlernen einer neuen Programmiersprache statt.

Lediglich 19% der Teilnehmer haben angegeben, eine klassische Request-Response basierte Anwendung zu implementieren, so dass wir im Rahmen der Studie bereits von dem Beginn eines Architekturwandels bei Web-Anwendungen sprechen können: Bereits 46% der Teilnehmer haben durch gezielten Einsatz von JavaScript teilweise Funktionalität in den Browser verlagert.



Grundsätzliche Angaben zu der Architektur der Webanwendung

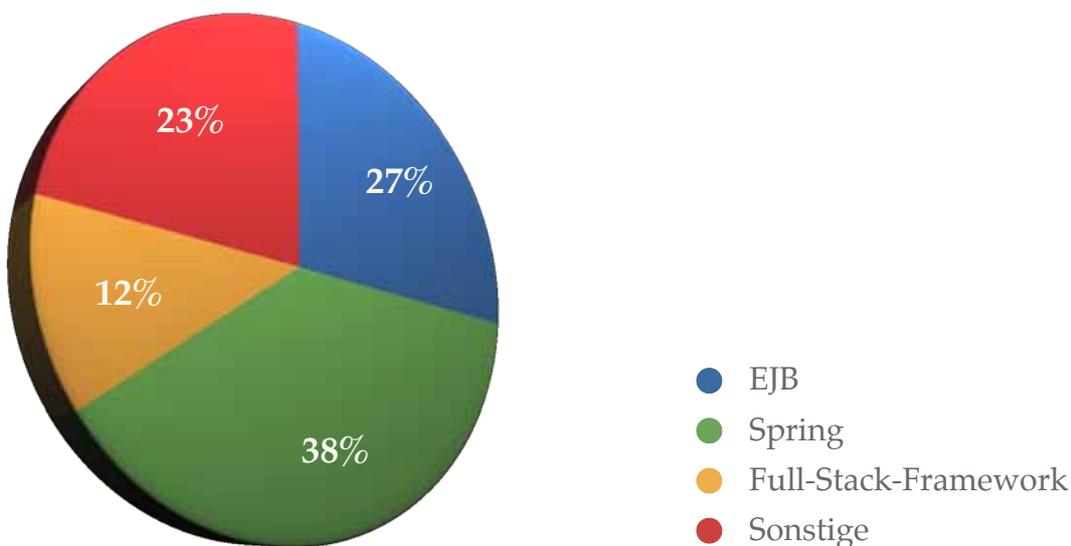


27% der Teilnehmer haben angegeben, bereits eine RIA im Single-Page-Prinzip umzusetzen, hier hat ein Paradigmen- bzw. ein Architekturwechsel schon stattgefunden: von der klassischen Request-Response basierten Web-Anwendung bis hin zu dem anderen Extrem, der sogenannten Rich-Internet-Anwendung (RIA).

Im Backend wenig Experimente

Im technischen Kontext haben wir schließlich noch die Frage nach der Backend-Technologie gestellt. Wenig überraschend haben Spring und EJB basierte Lösungen 65% des Kuchens belegt.

Rechnet man noch die Full-Stack-Lösungen wie zum Beispiel Grails und Roo hinzu, so haben die heute marktüblichen Backend-Ansätze einen Anteil von 77%.



Backend-Technologie

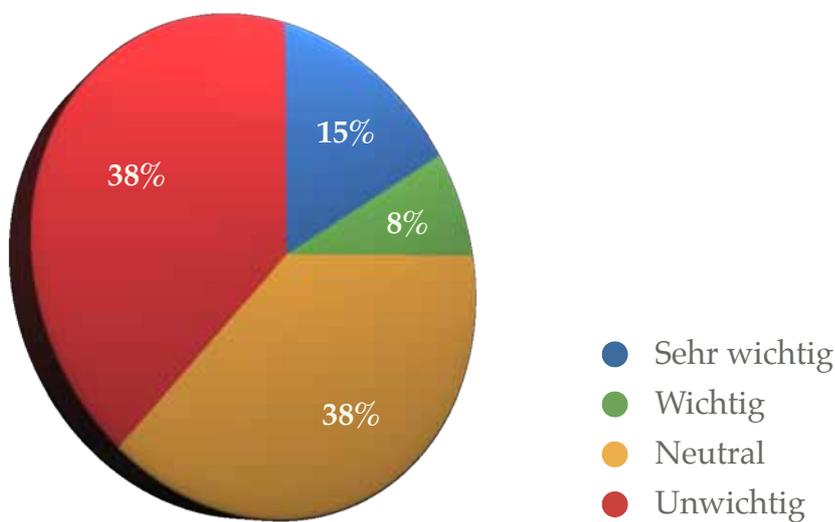
Der hohe Anteil von 23% an „sonstigen Lösungen“ ist sehr interessant: Unter „sonstigen“ Lösungen sind sowohl Fremdsysteme als auch hauseigene Lösungen untergekommen.

Java als Technologie wird aus dieser Perspektive den Ruf als Integrationslösung gerecht.



Vom Äusserem nicht blenden lassen

Lediglich 23% der Teilnehmer sind ansprechende UI Komponenten wichtig bis sehr wichtig. Ganze 38% der Teilnehmer ist diese Anforderung sogar unwichtig.



Ansprechende UI Komponenten

Aber für exakt die Hälfte der Teilnehmer ist die Tastaturbedienung eine wichtige bis sehr wichtige Anforderung. Komplexe client-seitige Validierungen werden nur von 43% der Teilnehmer als wichtig bis sehr wichtig eingestuft, Anwendungsdialoge in vorgegebenen Reihenfolgen (bspw. Wizards) werden von 71% als wichtig bis sehr wichtige Anforderung gesehen. Die Anforderung nach der Fähigkeit zu Popups für Meldungen, gezielte Dateneingabe und Datenanzeige wurde von 88% der Teilnehmer als wichtig bis sehr wichtig eingestuft.

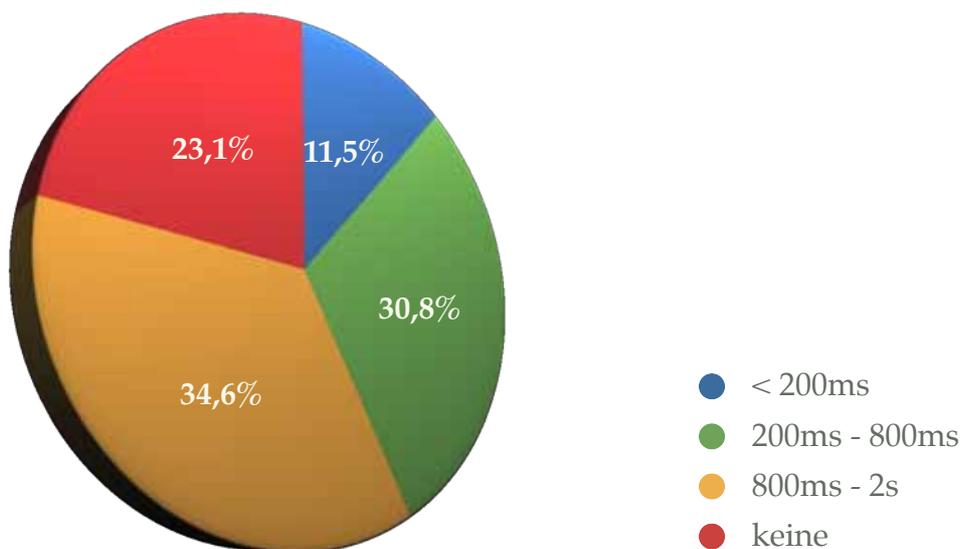
Das Wechseln von Themes in der Anwendung wurde nicht als wichtige Anforderung eingestuft (nur 31%) und die Anpassbarkeit an die eigenen CI Anforderungen wurde von 53% der Teilnehmer als wichtig bis sehr wichtig eingestuft.



Erkenntnisse aus den Ressourcen- und Lastanforderungen

Web-Frameworks unterscheiden sich nicht nur in der Architektur und den graphischen Komponenten, sondern können in Bezug auf Lastanforderungen sehr unterschiedlich ausfallen.

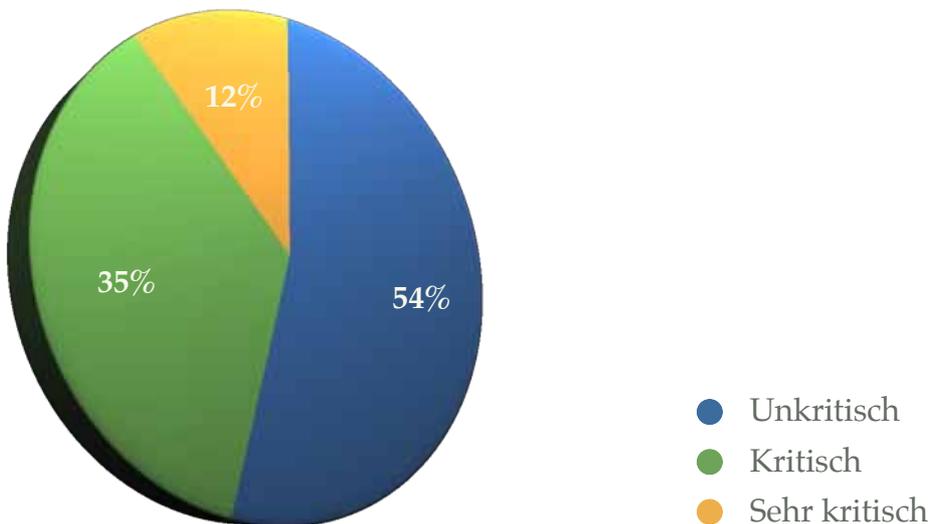
Bei den Teilnehmern der Studie konnte ein recht breites Spektrum in Bezug auf den Spitzenwert der gleichzeitigen Sitzungen festgestellt werden, wobei lediglich 31% der Teilnehmer mehr als 100 gleichzeitige Sitzung angegeben haben.



Vorgegebene Antwortzeiten

In Bezug auf die vorgegebene Antwortzeit (Latenz) haben 42% der Teilnehmer Antwortzeiten unterhalb 800ms gefordert. Immerhin ein Fünftel der Teilnehmer haben offen zugegeben, hier über keinerlei Anforderungen zu verfügen.,

Die verbrauchte Netzwerkbandbreite wurde von 54% der Teilnehmer als unwichtig eingestuft, und lediglich 12% der Teilnehmer sehen den Verbrauch als sehr kritisch an.



Wie kritisch ist die Netzwerkbandbreite?

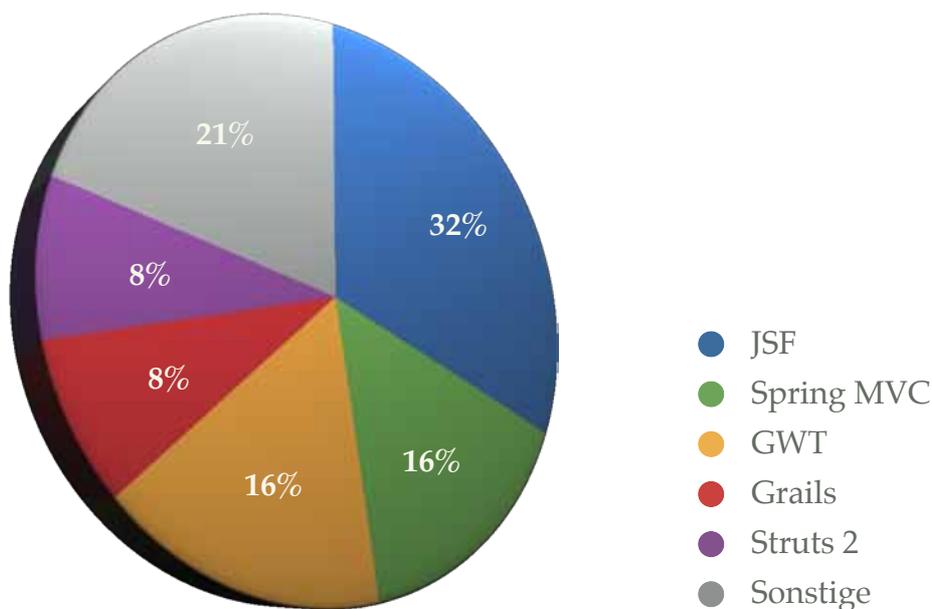
Anders sieht es mit der Serverlast aus. Hier haben 47% der Teilnehmer angegeben, den Ressourcenverbrauch auf dem Server kritisch einzustufen.



Die gewählten Web-Frameworks

Vorab: Die Teilnehmer der Studie haben angegeben, zu 73% mit der Auswahl des Frameworks zufrieden zu sein.

Das Spektrum der letztendlich gewählten Web-Frameworks ist recht breit aufgestellt. Die häufigste Entscheidung wurde im Zusammenhang mit der JSF Spezifikation getroffen, insgesamt 32% der Teilnehmer haben JSF gewählt.



Eingesetztes Webframework

An zweiter Stelle kommen Spring MVC und GWT mit jeweils 16%.

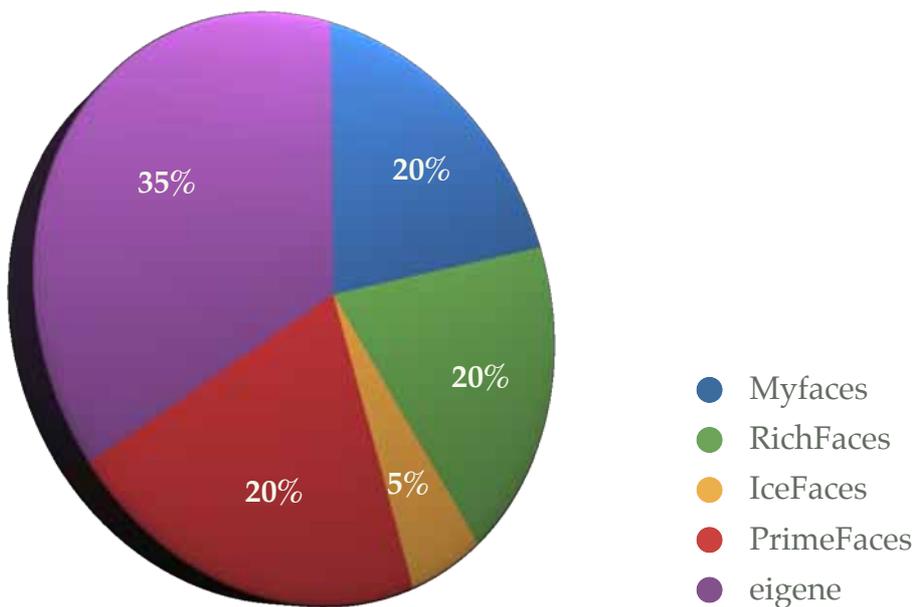
Java Web-Frameworks 2011

Die heute als marktüblich angesehenen Web-Frameworks im Java Umfeld JSF, Spring MVC und GWT belegen 62% der Entscheidungen. Grails, ein auf Groovy basierendes Framework, kommt mit 16% auf den 4. Platz, obwohl die Entwickler eine neue Programmiersprache lernen müssen (Groovy).

Interessanterweise stammen mit Spring MVC und Grails 32% der eingesetzten Frameworks aus dem Hause SpringSource bzw. VMWare.

Die Fragmentierung im Java Web-Framework Umfeld wird noch deutlicher, wenn die üblicherweise im Zusammenhang mit JSF eingesetzten Komponentenbibliotheken hinzugezogen werden.

Ganze 35% der Teilnehmer, die sich für den Einsatz von JSF entschieden haben, haben sich auch für die Entwicklung von eigenen UI Komponenten entschieden.



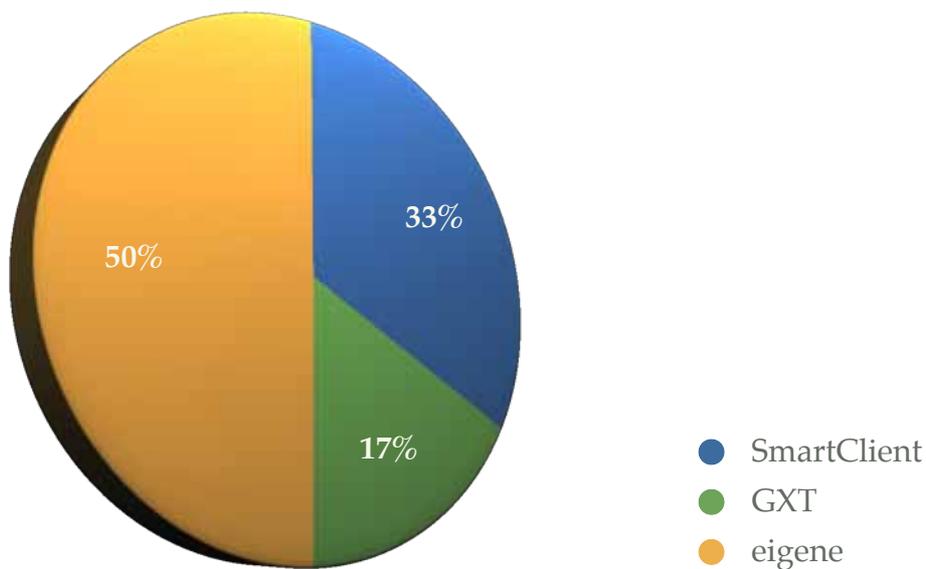
JSF Komponentenbibliotheken



Im Google Web Toolkit Umfeld können auch Fremdbibliotheken mit UI Komponenten eingesetzt, bzw. eigene Komponenten umgesetzt werden.

Hier haben sich die Hälfte der Teilnehmer für die Umsetzung eigener Komponenten entschieden.

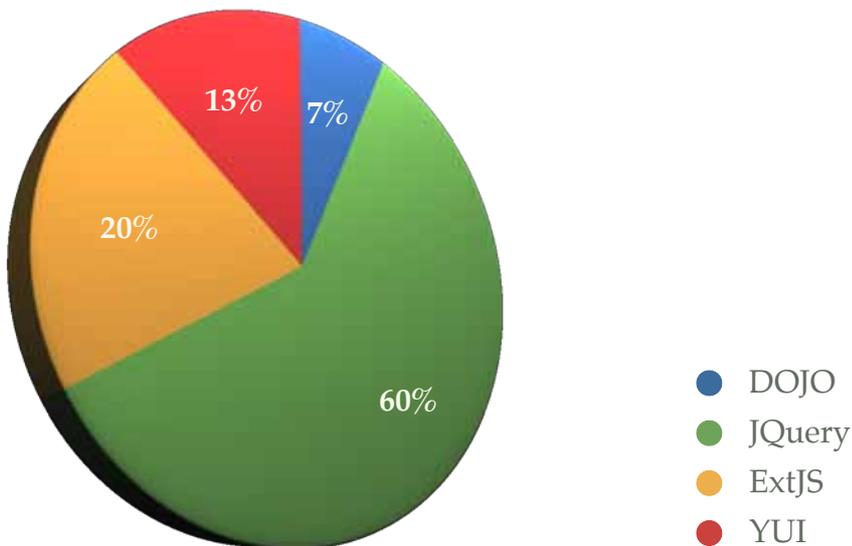
SmartClient wurde bei 33% und GXT bei 17% der GWT einsetzenden Teilnehmern ausgewählt.



GWT Komponentenbibliotheken

Bei Spring MVC und Grails wird technisch keine UI Komponentenbibliothek vorgeschrieben bzw. mitgeliefert, so dass eine weitere Entscheidung für die eigentliche Frontend-Technologie notwendig wird. Auch bei dem Einsatz eigener JSP- bzw Servlet basierter Lösungen muss diese Entscheidung getroffen werden.

In diesen Fällen wird nicht selten eine UI Komponentenbibliothek (Widgets) auf JavaScript Ebene eingesetzt (sogenannte Toolkits). Die meisten Teilnehmer, insgesamt 60%, haben sich für jQuery entschieden.



JS Toolkits

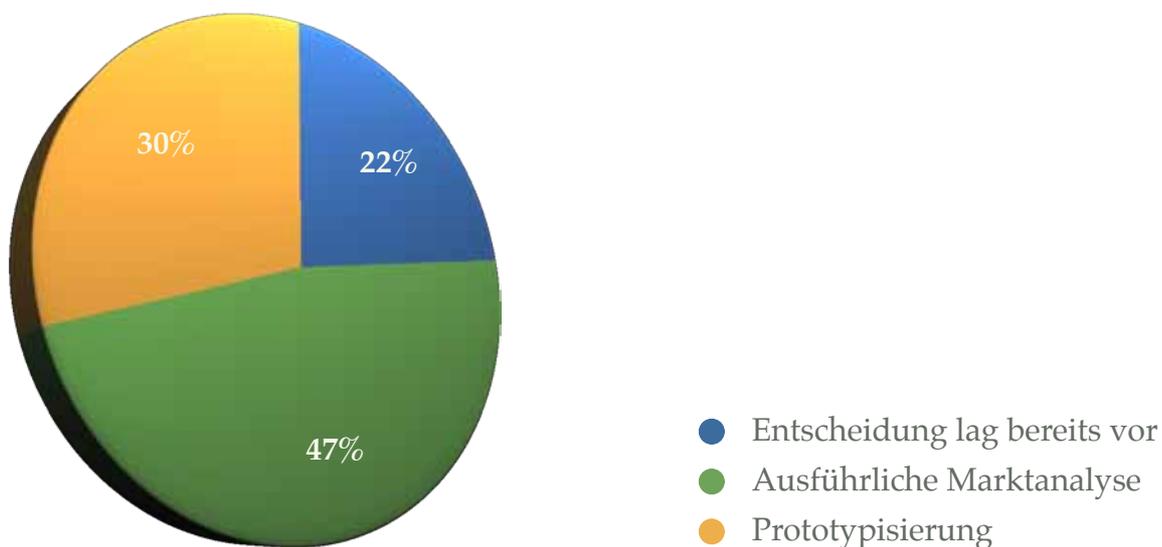


Zuspitzung in den letzten zwei Jahren

Nach der Gesamt-Betrachtung aller angegebenen Ergebnisse möchten wir hier noch einige Trends aus der eingeschränkten Betrachtung der jüngeren Projekte (innerhalb der letzten 2 Jahre) ableiten.

Besonders interessant ist hier die Zuspitzung auf einen reinen Zweikampf zwischen JSF und GWT bei der Auswahl der Frameworks:

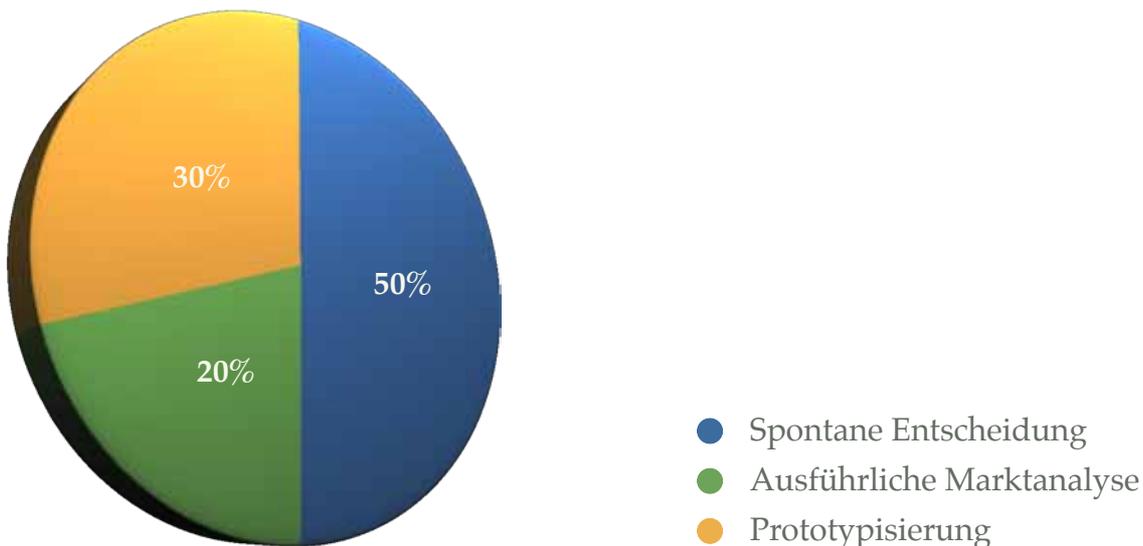
Für JSF hat sich niemand spontan entschieden. Hingegen lag bei fast einem Viertel der Vorhaben diese Entscheidung bereits als gegeben vor. Wir nehmen an, dass dies aus der Bedeutung von JSF als politischer „Standard“ resultieren mag. Bei der großen Mehrheit der Teilnehmer, die sich für den Einsatz von JSF entschieden haben, wurde diese Entscheidung durch ausführliche Marktanalysen bis hin zu einer Prototypisierung vorgeschaltet.



JSF - Entscheidung in den letzten 2 Jahren

Bei den GWT Entscheidungen hingegen handelte es sich bei jeder zweiten Entscheidung um eine spontane Entscheidung mit einer sehr kurzen Marktanalyse. Nie war die Entscheidung vorgegeben. GWT scheint insofern deutlich mehr vom Team selbst gewählt zu werden, und einer Machbarkeit des Vorhabens mit GWT auch deutlich ungeprüfter vertraut zu werden.

Eine ausführliche Marktanalyse bis hin zu einer Prototypisierung hat es bei der zweiten Teilnehmerhälfte gegeben.



GWT - Entscheidung in den letzten 2 Jahren

Interessanterweise sind alle Teilnehmer, die angegeben haben sich für GWT entschieden zu haben, mit der Entscheidung zufrieden.

Bei den JSF Entscheidungen ist jeder zweite mit seiner Wahl unzufrieden.



Dabei stammt allerdings interessanterweise die Mehrheit der unzufriedenen Teilnehmer aus der Gruppe, bei der die Framework Entscheidung Projekt-extern bereits vorlag.

Fazit

Konsolidierung erkennbar?

Auf der Basis unserer Daten aus dem Zeitraum Ende 2009 bis Ende 2011 deutet sich zur Zeit ein Zweikampf an:

Wir können eine deutliche Zunahme der Auswahl von JSF und GWT dokumentieren.

Beides sind komponentenbasierte Ansätze, was wir für keinen Zufall halten. Genau diese Web-Frameworks bieten darüber hinaus einen hohen Investitionsschutz in Bezug auf Java Know-How und Java-Tooling. Aus Sicht eines Java Entwicklers gibt es aus dieser Perspektive keinen Verlierer, nur eine grundsätzliche Architekturentscheidung.

Unzufriedenheit... mit Vorgaben oder mit Technologie?

Eine im Vergleich erhöhte Unzufriedenheit bei der Verwendung von JSF ist durchaus dokumentierbar. Die konkrete Nennung der Gründe umfasst Testbarkeit, Typisierung, Debugging und Refactoring über mehrere Schichten der Anwendung. Diese Probleme sind Technologie-immanent und werden nicht einfach behoben werden können. Mit entsprechenden Tools können jedoch die Hürden geebnet werden, das liegt allerdings nicht ausschließlich in den Händen der JSF Komponentenanbieter.

Relativierend bleibt anzumerken, dass in unserer Datenbasis die Unzufriedenheit mit JSF gerade dort deutlich erhöht war, wo die Entscheidung vorgegeben war. Dies muss nicht, mag aber bedeuten, dass die Probleme hier auch außerhalb der Technologie, bspw. in der Technologiewahl oder gar in politischen Fragestellungen zu suchen sind.

Der IE6...

Die Unterstützung des IE6 war in unserer Studie durchaus noch gefordert. Wir sind gespannt, ob in unserer Folgestudie der Methusalem unter den Browsern vom Forderungs-Tableau verschwunden sein wird. Zum Zeitpunkt der Veröffentlichung dieser Studie zeigt der Microsoft eigene Countdown <http://www.ie6countdown.com> noch 8% an.

Was die aktuellen Versionen der Browser betrifft, hat der IE seine Alleinstellung verloren. Wir können eine gleichverteilte Nachfrage nach annähernd allen Browsern dokumentieren. Das wird die Nachfrage nach standkonformen Browser oder nach Cross-Browser Technologien steigern.

Das heute von morgen ist das morgen von heute

In den Daten der letzten Jahre lassen sich die Unterstützung von mobilen Clients und von HTML 5 als neue Anforderungen bereits vorfinden. In diesem Bereich möchten wir unsere Folgestudie gezielt erweitern.



Und wie geht es weiter?

Wir danken vielmals allen bisherigen Teilnehmern, die diese Studie erst möglich gemacht haben.

Wir hoffen, Sie mit der Veröffentlichung dieser Studie neugierig auf weiterführende Ergebnisse gemacht zu haben. Deshalb wollen wir mit einer verschlankten Befragung eine nochmals deutlich breitere Datenbasis gewinnen.

Im Speziellen möchten wir den Fokus auf den erkannten Zweikampf JSF und GWT und die erkannten Trends HTML5 und Unterstützung von mobilen Clients stärken. Bei GWT interessieren uns zusätzlich noch die Frage der Nutzung fertiger Komponenten sowie die weitere Ausrichtung auf Enterprise Anforderungen wie Validierung und Data Binding. Bei JSF wollen wir speziell die Verbreitung der Komponentenbibliotheken in der neuen Ära der JSF 2 Komponenten im Auge behalten.



Dirk M. Sohn

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

Web: <http://www.oio.de>
Mail: info@oio.de

Telefon: +49 621 71839-0



Papick G. Taboada

pgt technology scouting GmbH

Bachstr. 11
76185 Karlsruhe

Web: <http://pgt.de>
Mail: info@pgt.de

Telefon: +49 721 9573625