

Evaluierung des Webentwicklungsframeworks eGovWDF im Hinblick auf den Aspekt Validierung und die besonderen Anforderungen im Bereich des eGovernment

Walter Kern

**Lehrstuhl für Informationswissenschaft der
Universität Regensburg**

Juli 2009

Alle genannten und gegebenenfalls durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichnungsrechts und den Besitzrechten der jeweiligen Eigentümer. Allein aufgrund der bloßen Nennung kann nicht geschlossen werden, dass Markenzeichen nicht durch Rechte Dritter geschützt sind. Aus dem Fehlen des Zeichens (R) darf nicht geschlossen werden, dass ein Name oder Zeichen frei ist. Eine Haftung für ein etwaiges Fehlen des Zeichens (R) wird ausgeschlossen.

Abstract

Wie im Rahmen einer umfangreichen Evaluation gezeigt wurde, erfüllt keines der derzeit erhältlichen Frameworks zur Benutzereingaben-Validierung das sich im Schnittpunkt aus Webentwicklung, modernem Software-Engineering und eGovernment ergebende Anforderungsprofil. Auf Basis des genannten Anforderungsprofils wurde im Rahmen der Realisierung eines übergeordneten Rich Client-Webentwicklungsframeworks (eGovWDF – eGovernmental Web Development Framework) für den Einsatz im eGovernment-Bereich ein Validierungsteilframework realisiert, welches auf die vollständige Konformität zu diesen Anforderungen abzielt. Zielsetzung dieser Untersuchung ist nun die Prüfung des Grads der Anforderungserfüllung des eGovWDF-Teilframeworks bezüglich des in der oben genannten Evaluation spezifizierten Anforderungskatalogs.

Schlüsselbegriffe: Eingabevalidierung, Evaluation, Barrierefreiheit, Plausibilisierung, Validatoren, Software Engineering, eGovernment

Abstract (en)

As shown by a preliminary evaluation, there is no input validation framework that suffices the requirements at the intersection of web development, modern methods of software engineering and eGovernment. As a consequence, we extended our experimental rich client web development framework (eGovWDF – eGovernmental Web Development Framework) with an input validation sub framework that aims at complete compliance with the requirements mentioned above. It is the objective of this evaluation to examine the degree of compliance of the eGovWDF input validation framework with the catalog of requirements that was used in the preliminary evaluation.

Keywords: input validation, evaluation, Web accessibility, plausibility, validators, software engineering, eGovernment

Inhalt

1	EINLEITUNG	1
2	METHODIK DER EVALUATION	2
2.1	HYPOTHESE UND EVALUIERTE ASPEKTE	2
2.2	STATISTISCHE EVALUATIONSMETHODEN	4
2.2.1	METHODENWAHL	4
2.2.2	BEGRÜNDUNG DER METHODENWAHL	5
2.3	VISUELLE EVALUATIONSMETHODEN	6
3	EVALUIERUNG	8
3.1	BESCHREIBUNG	8
3.2	EVALUIERTE PRODUKTVERSION	9
3.3	ANFORDERUNGEN IM BEREICH VALIDIERUNGSLOGIK	9
3.3.1	UNTERSTÜTZUNG DEKLARATIVER VALIDIERUNGSLOGIK-FORMULIERUNG	9
3.3.2	ZENTRALE UND APPLIKATIONSLOKALE REGELBEREITSTELLUNG	10
3.3.3	WIEDERVERWENDBARKEIT (KOMBINIERBARKEIT, VERERBUNG, SKALIERUNG)	11
3.3.4	FLEXIBLE ERWEITERBARKEIT (PLUGIN-KONZEPTE)	12
3.3.5	DUALITÄT DER VALIDIERUNG (CLIENT- UND SERVERSEITIG)	13
3.3.6	BEL. BESCHREIBUNGSDATENFORMAT / PERSISTIERUNGSLOKATION	13
3.3.7	UMFANG AB WERK AUSGELIEFERTER VALIDIERUNGSLOGIK	13
3.3.8	GRAPHISCHE MODELLIERBARKEIT (TOOL-SUPPORT)	15
3.3.9	HOT-REMOTE-PROPAGIERUNG (UPDATE IM LAUFENDEN BETRIEB)	15
3.3.10	UNTERSTÜTZUNG VERSCHIEDENER FEHLERKRITIKALITÄTEN	15
3.3.11	GESAMTBEWERTUNG	15
3.4	ANFORDERUNGEN IM BEREICH VALIDIERUNGSINTEGRATION	16
3.4.1	EXPLIZITE INTEGRATIONSSCHICHT FÜR BELIEBIGE PLATTFORMEN/Frameworks	16
3.4.2	LOSE KOPPELUNG ZWISCHEN LOGIK UND VISUALISIERUNG	16
3.4.3	UNTERSTÜTZUNG DER VALIDIERUNG KOMPLEXER PLATTFORMSPEZ. KOMPONENTEN	16
3.4.4	GESAMTBEWERTUNG	17
3.5	ANFORDERUNGEN IM BEREICH VALIDIERUNGSERGEBNISVISUALISIERUNG	17
3.5.1	BARRIEREFREIHEIT	17
3.5.2	ASSISTIVER CHARAKTER	17
3.5.3	CLIENT- UND SERVERSEITIGE REPRÄSENTATION	18
3.5.4	FEHLERKRITIKALITÄTSVISUALISIERUNG	18
3.5.5	GESAMTBEWERTUNG	18
3.6	HORIZONTALE ANFORDERUNGEN	19
3.6.1	PLATTFORMNEUTRALITÄT	19
3.6.2	ZUSATZKOMPONENTENERFORDERNISSE	19
3.6.3	BERÜCKSICHTIGUNG VON SICHERHEITSASPEKTEN	19
3.6.4	SIMPLIZITÄT DER ANWENDUNG	20
3.6.5	GESAMTBEWERTUNG	20
4	AUSWERTUNG	21
4.1	BESCHREIBUNG	21
4.2	ÜBERBLICKS-ANALYSE	21
4.3	ANFORDERUNGSARTBEZOGENE ANALYSE	23

5 FAZIT	26
LITERATURVERZEICHNIS	27
ABBILDUNGSVERZEICHNIS	30
ANLAGENVERZEICHNIS	31

1 Einleitung

Im Rahmen der Studie „Evaluierung bedeutender Webframeworks im Hinblick auf den Aspekt Validierung und die Anforderungen im Bereich des eGovernment“ (vgl. Kern, 2008a) wurden die bedeutendsten und vielversprechendsten Webapplikationsframeworks im Hinblick auf die Erfüllung von Anforderungen aus der Domäne eGovernment im Bereich Validierung untersucht.

Wie Kern (2008a) feststellt, erfüllt keines der untersuchten Frameworks die vorliegenden Anforderungen in hinreichendem Maß, da selbst das Framework mit dem besten Testergebnis lediglich einen Gesamtanforderungserfüllungsgrad von 59% erreicht.

Aus diesem Grund wurde es in besagtem Dokument angeregt, ein Framework zu konzeptionieren, welches sich die Erfüllung der angeführten Anforderungen als Primärziel setzt. In diesem Zusammenhang ist eGovWDF¹ entstanden, dessen Teilframework zur Benutzereingabvalidierung versucht, einen möglichst hohen Grad an Anforderungsbefriedigung umzusetzen.

Zielsetzung dieser Arbeit ist damit nicht die funktionale und technische Beschreibung des Validierungsteilframeworks von eGovWDF, sondern die Untersuchung der Anforderungserfüllung von eGovWDF im Hinblick auf die in Kern (2008a) formulierten Anforderungen.

Als Definition von Validierung im Webkontext soll dabei im Rahmen dieser Untersuchung die in Kern (2008a: 15) formulierte Definition zugrunde gelegt werden:

Unter Validierung im Webkontext versteht man die automatisierte Prüfung von via HTTP übertragenen Benutzereingaben auf ihre syntaktische und semantische Korrektheit sowie deren assistive Visualisierung zur Fehleingabeneeliminierung.

¹ eGovWDF = eGovernmental Web Development Framework.

2 Methodik der Evaluation

2.1 Hypothese und evaluierte Aspekte

Die über diese Evaluation zu belegende Hypothese dieser Arbeit besteht darin, dass das Validierungsteilframework von eGovWDF, im späteren zur Vereinfachung kurz als eGovWDF bezeichnet, eine 100%-Anforderungserfüllung bezüglich der in Kern (2008a) formulierten Anforderungen erreicht.

Im Rahmen der Untersuchung soll äquivalent zu **Fehler! Verweisquelle konnte nicht gefunden werden.** eine Gliederung der zu prüfenden Anforderungen in die folgenden Bereiche erfolgen, um ein optimales Maß an Komparabilität zu erreichen, wobei für jeden Bereich eine Prüfung der Unterstützung der weiter unten genannten Subaspekte erfolgt:

- Validierungslogik
- Validierungsintegration
- Validierungsergebnisvisualisierung
- Horizontale Anforderungen

Der Aspekt Validierungslogik soll dabei folgende Subpunkte umfassen:

- Deskriptive/deklarative (vgl. Coenen, 1999; Ligeza, 2006) Form der Validierungsregel formulierung neben programmatischer Form
- Zentrale oder auch applikationslokale Bereitstellung und Nutzung
- Wiederverwendbarkeit (Kombinierbarkeit / Vererbung / Skalierbarkeit; vgl. Biddle und Tempero, 1996; Bondi, 2000; D'Antonio et al., 2004)
- Flexible Erweiterbarkeit (Software Architektur - vgl. Vogel et al. 2005; Plugin-Konzepte, Factory Pattern - vgl. Gamma et al. 1994)
- Dualität (client-/serverseitig; vgl. Moore 2002)
- Beliebiges Beschreibungsdatenformat / bel. Persistierungslokation
- Umfang ab Werk ausgelieferter Validierungslogik
- Graphische Modellierbarkeit
- Hot-Remote-Propagierung²
- Fehlerkritikalität³ (vgl. Bundesministerium des Innern 1997; Williams, 2004)

² Änderung von Validierungsregeln zur Laufzeit und automatische Aktualisierung nutzender Anwendungen

³ Unterstützung verschiedener Schweregrade an Fehlern

Die Validierungsintegration auf der anderen Seite stellt sich wie folgt dar:

- Integrationsschichtsupport für beliebige Plattformen / Frameworks
- Lose Koppelung (vgl. Pressman 1996) zwischen Logik und Visualisierung (client- u. serverseitig)
- Unterstützung für die Validierung komplexer⁴ plattformspezifischer Komponenten

Die Validierungsergebnisvisualisierung-Anforderungsuntersuchung besteht aus folgenden Teilaspekten:

- Barrierefreiheit (vgl. Bundesministerium des Innern 2002; Bayerisches Staatsministerium des Innern 2003; KBSt 2006; W3C 1999; Kern, 2008b)
- Assistiver Charakter: Eine Art kontextsensitive Hilfe (vgl. Lee 1987; Danielsen et al. 1993; Nielsen 2005)
- Client- und serverseitige Repräsentation (vgl. Moore 2002)
- Fehlerkritikalitätsvisualisierung (unterschiedliche Darstellung unterschiedlicher Fehlerkritikalität)

Unter dem Aspekt horizontale Anforderungen sollen schließlich folgende übergreifende Anforderungen betrachtet werden:

- Plattformneutralität (auch Plattformunabhängigkeit; vgl. KBSt 2006)
- Zusatzkomponentenerfordernisse (Einsatz von Plugins möglichst minimal⁵; vgl. KBSt 2006)
- Berücksichtigung von Sicherheitsaspekten (u. a. Datensicherheit; vgl. KBSt 2006)

Hinsichtlich einer detaillierteren Beschreibung dieser Aspekte soll auf Kern (2008a) verwiesen werden. Anzumerken ist, dass im Gegensatz zu Kern (2008a) und den darin vorgenommenen Framework-Dokumentations-gestützten Untersuchungen die Evaluierung von eGovWDF auf Grundlage des Expertenwissens des Architekten und Entwicklers beschrieben wird, weil aufgrund der Neuartigkeit von eGovWDF noch keine extensive schriftliche Dokumentation vorliegt. Gemeinsam ist der Evaluierung in Kern (2008a) und dieser Untersuchung die Verwendung von Framework-Funktionstests in Kombination mit einer visuellen Prüfung der sich bei Benutzung einer Komponente oder Funktionalität ergebenden (Web-)Oberfläche sowie eine Betrachtung des resultierenden ((X)HTML)-Quellcodes durch einen Experten.

⁴ Aus HTML-Basistags/Elementen zusammengesetzte Komponenten, vgl. Netherland und Ganguly 2002

⁵ Nach KBSt 2006 sollen nur Plugins der Positivliste (vgl. <http://www.kbst.bund.de/saga-plugins>) der KBSt eingesetzt werden.

Ebenso wird in beiden Evaluierungen die Client-Umgebung (Webbrowser des Clients) mit unterschiedlichem Funktionsumfang simuliert, indem bestimmte Funktionen testabhängig aktiviert bzw. deaktiviert werden. Exemplarisch sind die Deaktivierung von CSS, JavaScript und Farbe zu nennen, um die JavaScript-Unabhängigkeit (und damit die rein serverseitige Funktionsfähigkeit und Darstellung) sowie die Konformität zu den entsprechenden CSS- und Farbe-Unabhängigkeit fordernden BITV-Kriterien zu prüfen.

Um eine Vergleichbarkeit mit den Ergebnissen der Frameworkevaluierung nach **Fehler! Verweisquelle konnte nicht gefunden werden.** zu erreichen, soll in dieser Evaluierung eine zur besagten Untersuchung identische Bewertung und Gewichtung der oben genannten Anforderungen erfolgen. Daher wird für minimale Anforderungserfüllung ein Leistungswert von 0,0, für teilweise vorliegende Anforderungserfüllung ein Wert von 0,5 und für maximale Erfüllung ein Wert von 1,0 vergeben. Zusätzlich wird pro zu prüfendem Teilaspekt ein Gewichtungsfaktor spezifiziert, welcher über Produktbildung mit der Teilanforderungserfüllung in die Gesamtsummenbildung eingeht und Kern (2008a) entnommen wird.

2.2 Statistische Evaluationsmethoden

2.2.1 Methodenwahl

Analog zur Validierungsframework-Untersuchung in Kern (2008a) wird pro Framework, hier das eGovWDF-Validierungsframework, die Gesamtanforderungserfüllung auf Basis des gewichteten arithmetischen Mittels über alle Teilanforderungserfüllungswerte berechnet.

Grund für die Legitimität der Vereinfachung der Berechnung des Gesamterfüllungsgrads auf Basis der Formel für das arithmetische Mittel ist die Tatsache, dass der Maximalwert der Erfüllung pro Anforderung stets konstant 1 ist und dieser damit aus dem Nenner des Bruchs für die Gesamtanforderungsgrad-Bestimmung entfallen kann:

$$g_j = \frac{\sum_i r_i * a_{j,i}}{\sum_i r_i * m_i} \quad m_i = 1 \forall i \quad \frac{\sum_i r_i * a_{j,i}}{\sum_i r_i} = \overline{a_j}$$

g_j : Grad der Gesamtanforderungserfüllung des Frameworks j

$\overline{a_j}$: Gewichtetes arithmetisches Mittel aller Anforderungserfüllungswerte des Frameworks j

$a_{j,i}$: Erfüllungswert der Anforderung i bei Framework j

m_i : Maximal möglicher Erfüllungswert der Anforderung i; konstant 1

r_i : Relevanzfaktor der Anforderung i (Gewichtungsfaktor)

Um die qualitativen sowie quantitativen Unterschiede zwischen dem eGovWDF-Ansatz und den Benutzereingabe-Lösungen des aktuellen Stands der Technik im Detail, d.h. bezogen auf die einzelnen Validierungsanforderungen und bezogen auf die Validierungsanforderungsarten (Visualisierung, Validierungslogik, ...), zu analysieren, wird im Rahmen dieser Untersuchung zusätzlich eine Berechnung von Durchschnitt, Median, unterem und oberem Quartil sowie den Extremwerten vorgenommen.

Beachtet werden sollte hierbei, dass sämtliche der oben genannten Kennzahlen - unabhängig von eGovWDF - für die in Kern (2008a) evaluierten Frameworks errechnet werden, da ansonsten eine Verfälschung der Kennzahlen auftritt. Beispielsweise würde der Wert der durchschnittlichen Anforderungserfüllung, der die aktuelle Situation ohne eGovWDF beschreiben soll, durch eine Berücksichtigung der Kennzahlen von eGovWDF verzerrt.

2.2.2 Begründung der Methodenwahl

Aufgrund der Primärzielsetzung des Aufzeigens des aktuellen Stands der Technik und der aktuellen Maximal-Anforderungserfüllung bei gegenwärtig verfügbaren Benutzereingabe-Validierungsframeworks, wurde in Kern (2008a) eine Beschränkung auf die Betrachtung der Maximal- und der Durchschnittsanforderungserfüllung der untersuchten Frameworks vorgenommen. Hierbei werden auch die unter Abschnitt 2.1 angesprochenen Gewichtungsfaktoren berücksichtigt, deren Multiplikation mit den Erfüllungsgradwerten der zugehörigen Anforderungen summarisch in die Berechnung der Gesamtanforderung einfließt. Der Grad der Gesamtanforderungserfüllung wird in dieser Evaluation zur Schaffung von Vergleichbarkeit zwischen dem eGovWDF-Ansatz und den in Kern (2008a) untersuchten Lösungen identisch zu Kern (2008a) bestimmt, indem ebenso das gewichtete arithmetische Mittel errechnet wird

(vgl. Abschnitt 2.2.1). Unabhängig davon ergeben sich für die Wahl der gewählten Evaluationsmethoden die im Folgenden genannten Gründe.

Die Gesamtanforderungserfüllung pro Framework wird grundsätzlich ermittelt, um eine Maßzahl für die Anwendbarkeit des jeweiligen Frameworks bei konkreten Projekten in der sich aus den Einzelanforderungen ergebenden Problemdomäne zu erhalten. Diese skalare Maßzahl ermöglicht damit auch die Ermittlung einer Framework-Rangfolge und allgemein den gesamtheitlichen Vergleich mehrerer Frameworks.

Über die Maßzahl der Durchschnittsanforderungserfüllung und damit des gewichteten arithmetischen Mittels über die Gesamtanforderungserfüllungswerte aller Frameworks soll festgehalten werden, welcher Gesamtanforderungserfüllungsgrad im Mittel erreicht wird, wodurch eine Aussage über den bei einer Schnittpfeldbetrachtung der Frameworks sich ergebende mittlere Anforderungserfüllung gegeben wird, was wiederum als Vergleichsgröße gegenüber den Gesamtanforderungserfüllungswerten der einzelnen Frameworks dienen kann.

Da das gewichtete arithmetische Mittel sehr anfällig für Extremwerte ist, eignet es sich nicht für die Bestimmung des Stands der Technik, sodass zusätzlich eine Berechnung des Medianwerts erfolgt, welcher als wesentlich störresistenter gilt und es in Kombination mit Minimum- und Maximumwerten erlaubt, starke Ausreißer nach oben (potentielle Lösungen über dem Stand der Technik) und unten klar davon abzugrenzen.

Ferner sollen oberes und unteres Quartil berechnet werden, um eine Einteilung in gesamtheitlich „gute Frameworks“ und „schlechte Frameworks“ vornehmen zu können.

Um neben der globalen eine feingranulare Bewertung auf Anforderungsartebene zu erhalten, werden die genannten Kennzahlen in der Untersuchung jeweils auch auf die zu Anforderungsarten gruppierten Teilmengen des Gesamtanforderungskatalogs angewendet.

2.3 Visuelle Evaluationsmethoden

Zur Visualisierung der Kennzahlen Maximal- und Durchschnittsanforderungserfüllung wurde angelehnt an Kern (2008a) eine Balken- bzw. Säulendiagrammdarstellung, welche die prozentuale Gesamt-Anforderungserfüllung jedes getesteten Frameworks aufzeigt, als probates Mit-

tel bestimmt. Damit wird auch die Komparabilität gefördert, da in Kern (2008a) die identische Darstellungsform gewählt wurde und somit ein Überblicks-Vergleich ermöglicht wird.

Zur weitergehenden Analyse ist die Entscheidung für eine Boxplot-Visualisierung (vgl. Tukey, 1977) gefallen, da diese es erlaubt, alle unter 2.2 beschriebenen Kenngrößen in einer Darstellung pro Anforderung bzw. Anforderungsart kompakt und die Komparabilität fördernd, zu visualisieren. Sowohl pro Anforderung als auch pro Anforderungsart wird der Boxplot-Visualisierung eine metrische Skala, genauer eine Intervallskala, zugrunde gelegt, da ein direkter Vergleich der erreichten Anforderungserfüllung möglich ist.

3 Evaluierung

3.1 Beschreibung

Bei eGovWDF handelt es sich um ein Framework mit der Zielsetzung, die besonderen Anforderungen und Belange (vgl. Kern, 2008a) in der Domäne eGovernment zu realisieren. Dabei besteht eGovWDF aus verschiedenen Teilframeworks (Abbildung 1) um die verschiedenen Themenfelder in diesem Umfeld abzubilden, wozu ein barrierefreies

- Validierungsframework,
- Rich-Client-(Komponenten-)Framework
- sowie ein Teilframework für assistive Benutzerführung zählen.

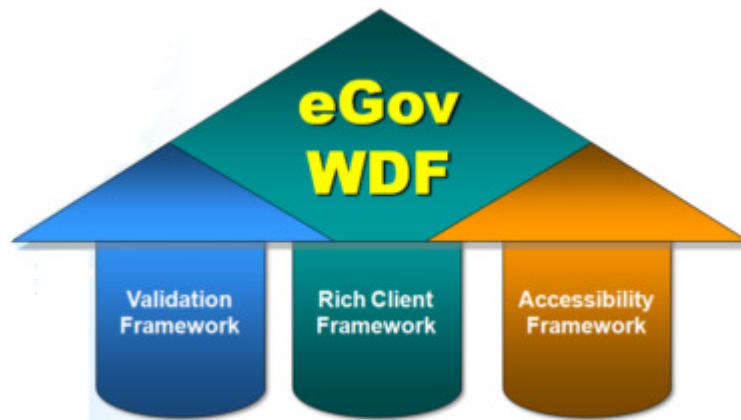


Abbildung 1: Teilframeworks von eGovWDF.

Da im Rahmen von eGovWDF Validierung als ein auf Mensch-Computer-Interaktion (vgl. Hewitt et al., 1992) basierender, gesamtheitlicher Prozess angesehen wird, werden nicht nur die reine Eingabewert-Plausibilisierung, sondern auch sämtliche assistiven Prozesse zur fehlerbereinigenden Benutzerführung von diesem Framework umfasst. Aus diesem Grund werden verschiedene Arten von Validierungslogik kapselnden Validator-Komponenten unterstützt:

- Plausibilisierende Validatoren (VerificationValidator-Komponenten)
- Normalisierende Validatoren (CorrectionValidator-Komponenten)
- Kompositionale Validatoren (CompositeValidator-Komponenten)
- Aktionale Validatoren (ActionValidator-Komponenten)

Plausibilisierende Validatoren prüfen zu validierende Werte gemäß dem gewählten Validator und dessen Konfiguration. Normalisierende Validatoren verändern die eingegebenen zu validierenden Werte dahingehend, dass eine konsistente Informationsrepräsentation entsteht, z.B.

indem Kommata in Punkte bei Währungsbeträgen umgewandelt werden und mehr als 2 Nachkommastellen abgeschnitten werden. Kompositionale Validatoren setzen sich aus beliebigen anderen Validatoren zusammen. Ihr Validierungsergebnis ergibt sich aus der logischen Verknüpfung der Validierungsergebnisse der untergeordneten Validatoren, z.B. besitzt ein logischer Und-Validator als Validierungsergebnis das Ergebnis der logischen Und-Verknüpfung der Validierungsergebnisse der untergeordneten Validatoren. Über bestimmte kompositionale Validatoren wie den IfValidator lässt sich zudem konditionale Validierung abbilden. Ferner können Feldabhängigkeiten über den ComponentReferenceValidator abgebildet werden. Generell können beliebig komplexe Validierungsszenarien durch die beliebige tiefe Verschachtelung auf Basis von kompositionalen Validatoren realisiert werden.

Zusätzlich ist anzumerken, dass Validatoren singulärer, zweidimensionaler oder benutzerdefinierter Natur sein können, was bedeutet, dass diese auf einzelne Werte, Wertlisten oder benutzerdefinierte Werttypen anwendbar sein können, wobei für jeden Validator die Anwendbarkeit über Annotationen in Form von .NET-Attributen vom Validator-Entwickler zu spezifizieren ist.

Die für die Untersuchung relevanten Details des Frameworks werden auf Basis der im Folgenden dargelegten Anforderungen vorgestellt.

3.2 *Evaluierte Produktversion*

Gegenstand dieser Untersuchung ist Version 1.0 der ASP.NET-Referenzimplementierung des Validierungsteilframeworks von eGovWDF.

3.3 *Anforderungen im Bereich Validierungslogik*

3.3.1 *Unterstützung deklarativer Validierungslogik-Formulierung*

Validierungslogik wird bei eGovWDF über parametrisierbare Validatoren repräsentiert, welche aufgabentypspezifische und gewöhnlich interdisziplinäre Validierungsfunktionalität kapseln. Beispielsweise werden ein Validator für Pflichtfeldvalidierungen und ein Validator für die Plausibilisierung basierend auf regulären Ausdrücken unterstützt. Validatoren und deren Parameter (z.B. Pattern-Parameter bei Validator für reguläre Ausdrücke) werden innerhalb

von Policies referenziert. Policies sind persistierte, in der Regel⁶ XML-basierte Konfigurationen (festgelegte Validatoren samt Parameterwerten) von referenzierten Validatoren und damit im Vergleich zu Validatoren auf einer höheren, eher nicht-technischen Abstraktionsebene angesiedelt. Als Beispiel kann hier eine *EmailAddressPolicy* genannt werden, die einen *RegexValidator* über dessen Typnamen referenziert und dessen Pattern-Parameter auf ein Muster für gültige E-Mail-Adressen festlegt.

```
<Policy name="EmailPolicy">
  <Validatortype="eGovWDF.Validation.Core.Validators.Common.RegexValidator"
    message="Invalid email address">
    <Param name="Pattern" value="([a-zA-Z0-9_-\.\,])@(\|[0-9]{1,3}\|[0-9]{1,3}\|[0-9]{1,3}\|)\|([a-zA-Z0-9-\.\,])\|[a-zA-Z]{2,4}\|[0-9]{1,3}\|\|?)"/>
  </Validator>
</Policy>
```

Da es sich bei den Policies zugrundeliegenden Validatoren um Klassen handelt, ist neben dieser deklarativen Validierungsdefinition auch eine imperative Implementierung möglich. Auch die in Abschnitt 3.1 angesprochene Möglichkeit beliebige Validatoren über kompositionale Validatoren (z.B. logischer Und-Validator, logischer Oder-Validator) beliebig tief zu verschachteln und selbst konditionale Validierung und Feldabhängigkeiten zu realisieren, lässt sich rein deklarativ oder alternativ imperativ beschreiben, was als vollständige Anforderungserfüllung bewertet wird.

Bewertungsfaktor: 1,0

3.3.2 Zentrale und applikationslokale Regelbereitstellung

Die Validierungsinformation kann zentral zur Verfügung gestellt werden, oder auch applikationslokal eingesetzt werden. Auch eine Kombination beider Mechanismen wird unterstützt, sodass man beispielsweise allgemeine Validierungsregeln, die für mehrere Anwendungen Gültigkeit besitzen, zentral zur Verfügung stellen und über Vererbungsregeln applikations-spezifische Regeln damit vermengen kann. Der Ort der Zur-Verfügungstellung, z.B. Dateisys-

⁶ Über ein im Rahmen des eGovWDF-Projekts neu entwickeltes Pattern, welches als *Dimension-Architektur* bezeichnet wird, wird ein generischer und flexibler, aber dennoch einheitlicher Zugriff auf Ressourcen unterschiedlichen Formats (z.B. XML, binär, Java-Properties-Datei-Format) und unterschiedlichem Speicherorttyp (z.B. Datenbank, Dateisystem, Cloud) ermöglicht.

tem oder Datenbank ist ebenfalls durch Anwendung eines Plugin-Patterns flexibel wählbar und austauschbar.

Bewertungsfaktor: 1,0

3.3.3 Wiederverwendbarkeit (Kombinierbarkeit, Vererbung, Skalierung)

Sämtliche Validatoren können miteinander kombiniert werden, indem diese in einer, einer GUI-Komponente zuordnungsfähigen, Policy festgelegt werden. Die Verknüpfung der Validatoren erfolgt dabei durch Unterordnung der Validatoren unter kompositionale Validatoren, z.B. einen AND- oder einen OR-Validator, wodurch boolesche Verknüpfungen der Validierungsregeln umsetzbar sind. Daneben besteht auch die Möglichkeit bedingte Validierungen vorzunehmen (IfValidator), wo wiederum sowohl die Bedingung als auch die Folge einem Validierungsteilbaum entsprechen.

Auch dem Aspekt Vererbung wird Rechnung getragen, indem Validierungsregeln auf verschiedene Ressourcen verschiedener Lokationstypen (z.B. Datenbank, Dateisystem, Netzwerk, Speicher) verteilt werden können. Diese verteilten Ressourcen, welche lediglich Validierungsregeln beherbergen, sogenannte Policy-Ressourcen, können von verschiedenen Definition-Ressourcen eingebunden werden, wobei eine Definition-Ressource die Zuordnung von Validierungsregeln zu Komponenten enthält. Beliebig viele solcher Definition-Ressourcen können dabei dann von einer Webapplikation über eine applikationslokale Configuration-Ressource eingebunden werden. Die in den verschiedenen Ressourcen definierten Validierungs- und Zuordnungsregeln können nun zusammengeführt werden, wobei eine feingranulare Steuerung des Verschmelzungsprozesses, des sog. Merging-Prozesses, unterstützt wird. Im Einzelnen werden folgende Merge-Aktionen, sog. Merge-Actions, für die Einbindung von Duplikatregeln unterstützt:

- First: Legt fest, dass die erste eingebundene Regel Anwendung findet.
- Last: Legt fest, dass die letzte eingebundene Regel Anwendung findet.
- Exception: Legt fest, dass eine auffangbare Ausnahme ausgelöst wird, falls ein Duplikat festgestellt wird.
- Merge: Fordert ein Merging der Regeln, die Details werden über folgende Optionen festgelegt:

- Keep: Legt fest, dass die Kindelemente beider Regeln Anwendung finden, aber bei Duplikaten in den Kindelementen, die Subregeln der ersten eingebundenen Ressource ausschlaggebend sind.
- Override: Legt fest, dass die Kindelemente beider Regeln Anwendung finden, aber bei Duplikaten in den Kindelementen, die Subregeln der letzten eingebundenen Ressource ausschlaggebend sind.
- Join: Führt ein echtes Joining der Regeln durch, was bedeutet, dass eine neue Regel berechnet wird, die die Intention beider Ausgangsregeln in Form der assoziierten Kindregeln beinhaltet.
- JoinException: Legt fest, dass die Kindelemente beider Regeln Anwendung finden, aber bei Duplikaten in den Kindelementen eine auffangbare Exception ausgelöst wird.

Durch die hier genannten Merge-Optionen kann sowohl global als auch auf Einzelementebene eine feingranulare Vererbung erreicht werden.

Dem Aspekt Skalierung wird dadurch berücksichtigt, dass zum einen eine Separation der Belange Validierungsregelanbindung, Validierungsregeldefinition und Validierungsregelzuordnung auf verschiedene Ressourcen vorliegt. Zum anderen können die Informationen auf eine beliebige Anzahl entsprechender Definitions- und Zuordnungsressourcen, zusätzlich mit gemischtem Lokationstyp (z.B. Datenbank, Dateisystem, Cloud), verteilt werden.

Bewertungsfaktor: 1,0

3.3.4 Flexible Erweiterbarkeit (Plugin-Konzepte)

Es können neue Validatoren entwickelt werden, die sich nahtlos in das bestehende Rahmenwerk einfügen. Validatoren beschreiben sich dabei selbst (Introspektion) und sind automatisch von anderen Validatoren referenzierbar und damit kombinierbar. Daneben sind auch das Datenformat, der Ort und der Typ des Lokationsortes sowie eventuelle Zusatzparameter wie eine Verschlüsselung der Validierungsregeln durch Anwendung des eines neuartigen, auf der Separation einzelner Aspekte beruhenden Ansatzes, der Dimension-Architektur, völlig austauschbar.

Bewertungsfaktor: 1,0

3.3.5 Dualität der Validierung (client- und serverseitig)

Validierungsregeln müssen nur an einer Stelle deklarativ definiert werden. Es erfolgt automatisch eine transparente server- und clientseitige Umsetzung der Validierungsregeln. Ferner wird die Verständlichkeit für Framework-Entwickler verbessert, indem eine identische API für Client- und Serverseite angeboten wird.

Bewertungsfaktor: 1,0

3.3.6 Bel. Beschreibungsdatenformat / Persistenzlokation

Das Beschreibungsdatenformat und die Persistenzlokation sind durch Anwendung der Dimension-Architektur beliebig austauschbar.

Bewertungsfaktor: 1,0

3.3.7 Umfang ab Werk ausgelieferter Validierungslogik

Ab Werk werden Validatoren für alle relevanten Standard-Prüfsituationen ausgeliefert. Im Einzelnen handelt es sich um:

- Allgemein
 - RequiredValidator (Validator zur Pflichtfeldprüfung)
 - RegexValidator (Validator für reguläre Ausdrücke)
 - ListContentValidator (Prüfung der Elemente eines Containersteuerelements, z.B. ListBox-Komponenten, Radiobutton-Listen, Checkbox-Button-Listen, ...)
 - ListCountValidator (Prüfung d. Elementanzahl eines Containersteuerelements, z.B. ListBox-Komponenten, Radiobutton-Listen, Checkbox-Button-Listen, ...)
- Boolesche Validatoren
 - AndValidator
 - OrValidator
 - NotValidator
- Konvertierung
 - DateTimeValidator (Datums- und Uhrzeitformatvalidator)
 - FloatingPointNumberValidator (Validator für Gleitkommazahlen)
 - CurrencyValidator (Währungs-Validator)

- Fehlerkorrektur (Eingabewertnormalisierende Validatoren)
 - DateTimeCorrectionValidator
 - FloatingPointNumberCorrectionValidator
 - RegexCorrectionValidator
 - UpperCaseValidator
- Abhängigkeiten/Bedingungen
 - IfValidator
 - ComponentReferenceValidator
 - PolicyReferenceValidator
- Sicherheit
 - SQLInjectionValidator
 - XSSValidator
- Web
 - UrlValidator
 - EmailValidator
 - IpAddressValidator

Damit und über zusätzlich erhältliche Validator-Konfigurationen können die häufigsten Problemszenarien abgedeckt werden. Über Validatoren wie IfValidator sowie die logischen Validatoren And-Validator, Or-Validator und Not-Validator können komplexe, verschachtelte Prüfsituationen abgebildet werden. Auch im Hinblick auf die Fehlerkorrektur von Benutzereingaben besteht durch den RegexCorrectionValidator die Möglichkeit beliebige reguläre Quellenausdrücke auf beliebige reguläre Zielausdrücke abzubilden. Ein weiterer wichtiger Aspekt, der den Einsatzbereich der Validatoren erheblich erweitert ist die validatorabhängige Möglichkeit diese Validatoren sowohl für Einzelwert bereitstellende Oberflächenkomponenten, wie Textfelder, als auch für Listenkomponenten und benutzerdefinierte Komponenten gleichermaßen einzusetzen. Ferner wird eine Basis von Aktionsvalidatoren für Standardaktionen wie das Ein- und Ausblenden von Elementen mitausgeliefert.

Bewertungsfaktor: 1,0

3.3.8 Graphische Modellierbarkeit (Tool-Support)

Validierungsregeln können regulär über jeden XML-Editor formuliert werden. Daneben existiert ein explizit für den Zweck der Formulierung von Validierungsregeln geschaffener Designer, über welchen Validierungsregeln ohne tief-technisches Verständnis durch Non-IT-Menschen graphisch und über WYSIWYG formuliert werden können.

Bewertungsfaktor: 1,0

3.3.9 Hot-Remote-Propagierung (Update im laufenden Betrieb)

Die Hot-Remote-Propagierung wird über einen webbasierten Dienst realisiert. Dieser Dienst wird mit der jeweiligen Zielanwendung in Form eines IHttpHandler ausgeliefert. Über ein separates Tool, den ValidationDesigner, können Validierungsregeländerungen graphisch vorgenommen werden über per Mausklick in die laufenden Anwendung übernommen werden. Zur Authentifizierung wird SSL-verschlüsselt ein auf Anwendungsseite konfiguriertes Token übertragen.

Bewertungsfaktor: 1,0

3.3.10 Unterstützung verschiedener Fehlerkritikalitäten

Es werden verschiedene Fehlerkritikalitäten unterstützt, derzeit Warning und Error. Über „Embedded Keywords“⁷ kann das Framework um weitere Kritikalitäten erweitert werden.

Bewertungsfaktor: 1,0

3.3.11 Gesamtbewertung

Anforderung	Bewertung
Unterstützung deklarativer Validierungslogik-Formulierung	1,0
Zentrale und applikationslokale Regelbereitstellung	1,0
Wiederverwendbarkeit (Kombinierbarkeit, Vererbung, Skalierung)	1,0

⁷ Embedded Keywords erlauben die Unterbringung von bestimmten Token in Validierungstexten, die von einem modularen Präprozessor des Validierungsframeworks aufgrund ihrer besonderen Bedeutung herausgefiltert werden.

Flexible Erweiterbarkeit (Plugin-Konzepte)	1,0
Dualität der Validierung (client- und serverseitig)	1,0
Bel. Beschreibungsdatenformat / Persistenzlokalisation	1,0
Umfang ab Werk ausgelieferter Validierungslogik	1,0
Graphische Modellierbarkeit (Tool-Support)	1,0
Hot-Remote-Propagierung (Update im laufenden Betrieb)	1,0
Unterstützung verschiedener Fehlerkritikalitäten	1,0
Gesamtbewertung	1,0

3.4 Anforderungen im Bereich Validierungsintegration

3.4.1 Explizite Integrationsschicht für beliebige Plattformen/Frameworks

eGovWDF besitzt eine strikte Trennung zwischen Validierungslaufzeitumgebung, Validierungslogik, Validierungsvisualisierung und Validierungswerkzeugen. Sämtliche Aspekte sind völlig voneinander unabhängig und können ausgetauscht werden. Clientseitiger Code liegt zudem in einer externen Bibliothek vor, welche bei einer weiteren Implementierung, z.B. für JSF, wiederverwendet werden kann und gemäß „Unobtrusive JavaScript“ eine Erweiterung des serverseitig generierten HTML-Outputs vornehmen.

Bewertungsfaktor: 1,0

3.4.2 Lose Koppelung zwischen Logik und Visualisierung

Wie in Abschnitt 3.4.1 dargestellt, sind die Aspekte Logik und Visualisierung völlig voneinander separiert.

Bewertungsfaktor: 1,0

3.4.3 Unterstützung der Validierung komplexer plattformspez. Komponenten

Das Framework unterstützt grundsätzlich die Plausibilisierung plattformspezifischer, kompositionaler GUI-Komponenten. Konkret liegen beispielsweise Validatoren für List-Komponenten vor. Als Beispiel sind hier der ListContentValidator und der ListCountValidator

zu nennen, wobei ersterer eine inhaltliche Prüfung der zugeordneten Listenkomponente und letzterer eine Anzahlvalidierung vornimmt. Durch die realisierte Provider-Architektur können zusätzliche benutzerdefinierte Komponenten umgesetzt werden.

Bewertungsfaktor: 1,0

3.4.4 Gesamtbewertung

Anforderung	Bewertung
Explizite Integrationsschicht für beliebige Plattformen/Frameworks	1,0
Lose Koppelung zwischen Logik und Visualisierung	1,0
Unterstützung der Validierung komplexer plattformspezifischer Komponenten	1,0
Gesamtbewertung	1,0

3.5 Anforderungen im Bereich Validierungsergebnisvisualisierung

3.5.1 Barrierefreiheit

Für die Darstellung von Validierungsfehlern erfolgt eine Umsetzung in regulären, standardkonformen XHTML 1.0 Strict-Code. Bei entsprechender Konfiguration besteht auch die Option, XHTML 1.1 Code zu erzeugen. Ferner ist sämtliche Funktionalität sowohl client- als auch serverseitig implementiert, sodass die Validierungsfunktionalität unabhängig von der Verfügbarkeit von JavaScript verfügbar ist. Eine Prüfung der BITV-Kriterien sowie eine Browserkompatibilitätsanalyse (Internet Explorer 6+, Firefox 2+, Opera 9+, Safari 3.1+) haben ebenso ein uneingeschränktes Funktionieren des Validierungsframeworks bei den genannten, marktdominierenden Browsern aufgezeigt.

Bewertungsfaktor: 1,0

3.5.2 Assistiver Charakter

Die konkrete Darstellung von Validierungsfehlern hängt vom verwendeten Validierungsvisualisierungsframework ab, wobei die Referenz-Implementierung die Anzeige von feldbezogenen Fehlermeldungen inklusive einer Gesamtfehlerliste vorsieht. Jede Fehlermeldung hat

bei Klick darauf eine Fokussierung der fehlerverursachenden, zugeordneten Komponente zur Folge. Neben Fehlermeldungen werden auch feldbezogene, kontextsensitive Hilfen unterstützt. Zudem besteht Unterstützung für Screenreader-Benachrichtigungen und eine Unterstützung von Textbrowsern für Lynx, wodurch eine weitere Unterstützung von Screenreader erreicht wird.

Bewertungsfaktor: 1,0

3.5.3 Client- und serverseitige Repräsentation

Die konkrete Darstellung von Validierungsfehlern hängt vom verwendeten Validierungsvisualisierungsframework ab, wobei die Referenz-Implementierung eine Darstellung realisiert, die sowohl client- als auch serverseitig erfolgt.

Bewertungsfaktor: 1,0

3.5.4 Fehlerkritikalitätsvisualisierung

Wie unter Abschnitt 3.3.10 beschrieben, werden Fehlerkritikalitäten auf Seiten der Validierungslogik unterstützt. Die standardmäßig ausgelieferte Visualisierungsimplementierung berücksichtigt Warning- und Error-Kritikalitäten durch farbliche Unterscheidung. Error-Kritikalitäten verhindern den Abschluss der angeforderten Aktion, während Warning-Kritikalitäten keinen Einfluss auf den Prozessablauf besitzen. Ferner wird über „Embedded Keywords“ ein Erweiterungsmechanismus für Fehlerkritikalitäten unterstützt, sodass nicht vorgesehene Kritikalitäten auf einfache Art und Weise nachgerüstet werden können.

Bewertungsfaktor: 1,0

3.5.5 Gesamtbewertung

Anforderung	Bewertung
Barrierefreiheit	1,0
Assistiver Charakter	1,0
Client- und serverseitige Repräsentation	1,0

Fehlerkritikalitätsvisualisierung	1,0
Gesamtbewertung	1,0

3.6 Horizontale Anforderungen

3.6.1 Plattformneutralität

Es liegt Plattformneutralität vor, da herkömmlicher, XHTML 1.0 Strict-konformer HTML-Code erzeugt wird und keinerlei Plugins auf Clientseite benötigt werden. Die Konzeption von eGovWDF ist zudem plattformneutral, da nur plattformübergreifende Techniken und Konzepte benutzt worden sind. Die vorliegende Implementierung ist unter ASP .NET ablauffähig; weitere Implementierungen für andere Plattformen sind aufgrund des modularen Ansatzes möglich und angedacht. Durch die Separation von serverseitigen Komponenten und clientseitigem, unobtrusive JavaScript-Code wird eine Portierung auf andere Plattformen erheblich erleichtert.

Bewertungsfaktor: 1,0

3.6.2 Zusatzkomponentenerfordernisse

Es sind keinerlei Zusatzkomponentenerfordernisse in Form von Plugins erforderlich, da regulärer XHTML 1.0 Strict-konformer HTML-Code erzeugt wird.

Bewertungsfaktor: 1,0

3.6.3 Berücksichtigung von Sicherheitsaspekten

Es existieren sicherheitsspezifische Validatoren. Dabei handelt es sich im Einzelnen um einen XSSValidator zur Verhinderung von Crosssite-Scripting-Attacken und zum Anderem um einen SQLInjection-Validator zur Verhinderung von SQL-Injection-Attacken.

Bewertungsfaktor: 1,0

3.6.4 Simplizität der Anwendung

Eine wichtige Anforderung an das zu erstellende Validierungsframework besteht auch darin, die Benutzung des Validierungsframeworks so einfach wie möglich zu gestalten. Dies wird in eGovWDF dadurch erreicht, dass zum einen standardmäßig XML zur Beschreibung der Validierungsregeln verwendet wird und zum anderen ein Designertool zur graphischen, WYSIWYG-basierten Definition der Validierungsregeln, der sogenannte eGovWDF Designer, angeboten wird, welcher es auch Personen ohne tief-technische Kenntnisse ermöglicht, Validierungsregeln auf einem fachlichen Abstraktionsniveau zu modellieren.

Bewertungsfaktor: 1,0

3.6.5 Gesamtbewertung

Anforderung	Bewertung
Plattformneutralität	1,0
Zusatzkomponentenerfordernisse	1,0
Berücksichtigung von Sicherheitsaspekten	1,0
Simplizität der Anwendung	1,0
Gesamtbewertung	1,0

4 Auswertung

4.1 Beschreibung

In diesem Abschnitt soll auf Basis der gewonnenen Daten ein Vergleich zwischen dem eGovWDF-Validierungsframework und den in Kern (2008a) untersuchten Validierungsframeworks erfolgen. Zunächst wird eine oberflächliche Analyse auf Basis der erreichten Gesamtanforderungserfüllung vorgenommen. Im nächsten Schritt soll eine anforderungsartbezogene Sichtung der Ergebnisse durchgeführt werden. Ferner wird eine Auswertung der Erfüllung der Einzelanforderungen vorgenommen (Anlage A1 bis einschließlich Anlage D3), welche jedoch vordergründig nicht dem Primärziel, der Bestätigung der in Kapitel 2 formulierten These, dient, sondern eine Datenbasis für zukünftige Analysen sowie eine Orientierung für eine bedarfsgesteuerte Erforschung der zugrundeliegenden Themenbereiche liefern soll.

4.2 Überblicks-Analyse

In Kern (2008a) wurde ein mittlerer Gesamtanforderungserfüllungsgrad von 51% und ein maximaler Gesamtanforderungserfüllungsgrad von 59% (beim Apache Commons Validator-Framework) festgestellt.

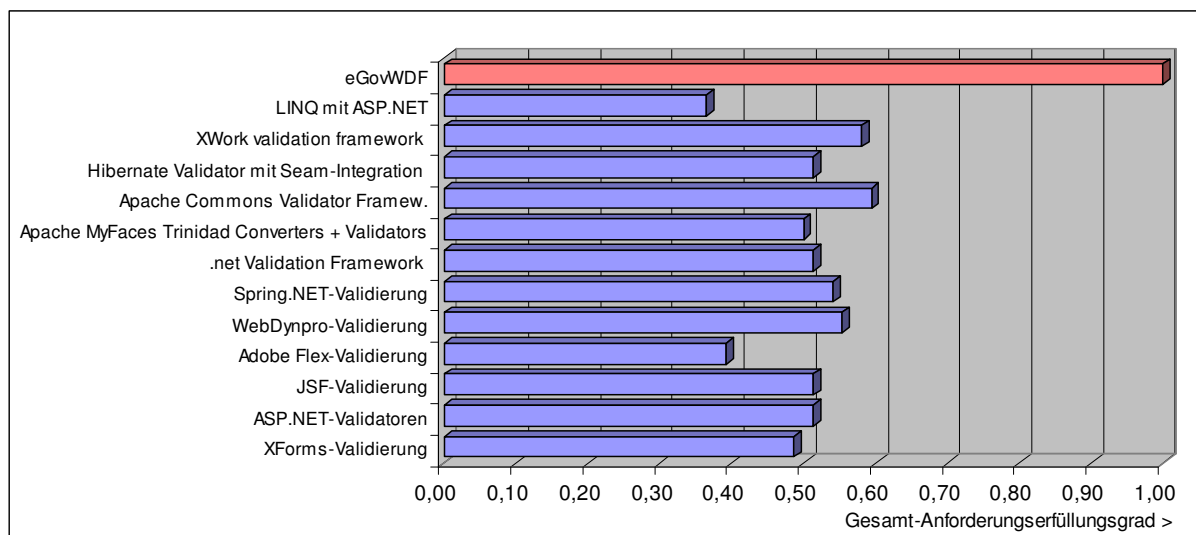


Abbildung 2: Vergleich der Gesamtanforderungserfüllung akt. Frameworks und eGovWDF.

Für eGovWDF ergibt sich bei Anwendung der Formel von Abschnitt 2.2 ein Gesamtanforderungserfüllungsgrad von 1,0, was 100% entspricht. Folglich liegt eine 100%ige Erfüllung

aller Anforderungen vor, was die zu untersuchende und in Abschnitt 2.1 formulierte Hypothese bestätigt und in Abbildung 2 sowie in Anlage E5 dargestellt ist.

Abbildung 3 und auch Anlage E6 verfeinern diese Darstellung, indem darin weitere statistische Kennzahlen zur sich aus der Betrachtung der verschiedenen Validierungsframeworks ergebenden Verteilung, visualisiert werden. Dabei werden zwei Boxen samt Whisker in einem Boxplot dargestellt, wobei die erste Darstellung die Verteilung der Gesamtanforderungserfüllung auf alle Frameworks, aber ohne eGovWDF, zeigt. Die zweite Grafik hingegen beinhaltet auch die Gesamtanforderungserfüllung von eGovWDF als Wert der Verteilung.

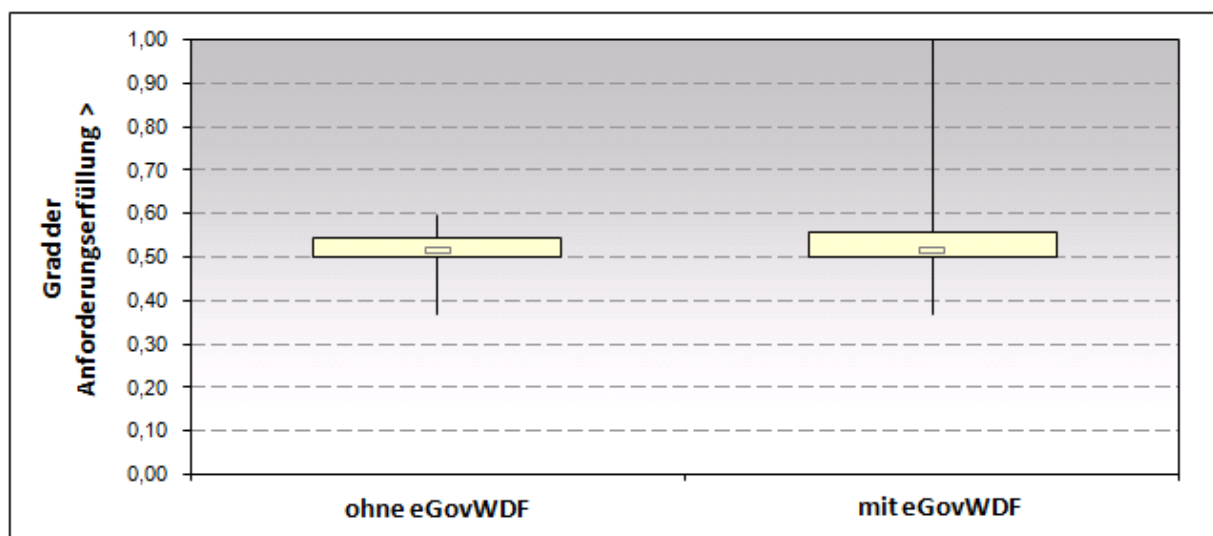


Abbildung 3: Verteilung der Gesamtanforderungserfüllung mit und ohne eGovWDF.

Die linke Box samt Whisker zeigt einen nur minimalen Abstand von oberem Quartil, unterem Quartil und Median. Zudem verdeutlichen die nach oben und unten zeigenden Whisker, dass zwar Ausreiser vorliegen, diese aber nur unwesentlich vom Median abweichen, welcher eine mittlere Gesamtanforderungserfüllung in Höhe von 0,51 (51 %) besagt. Daraus kann der Schluss gezogen werden, dass sich die aktuell verfügbaren Frameworks alle auf dem gleichen Niveau, dem Stand der Technik, bewegen und eine herausstechende, die besonderen Anforderungen im Bereich des eGovernment erfüllende Lösung, nicht vorliegt.

Bei Betrachtung der zweiten Box samt Whisker ist der Gesamtanforderungserfüllungsgrad von eGovWDF berücksichtigt. Wie aus der Darstellung erkennbar ist, sind Median und unteres Quartil unverändert, wobei aber eine minimale Erhöhung des oberen Quartils zu erkennen ist. Diese Ähnlichkeit zur ersten Darstellung verdeutlicht die Störunanfälligkeit der genannten statistischen Kennzahlen. Ausschlaggebend in dieser Darstellung ist jedoch, dass der obere

Whisker bis zum Maximalwert der Gesamtanforderungserfüllung, also 1,00 oder 100%, reicht. Der dabei visualisierte Extremwert des Whisker von 1,0 repräsentiert dabei die Gesamtanforderungserfüllung von eGovWDF und verdeutlicht damit, dass eGovWDF die oben angesprochene, fehlende, herausstechende und die besonderen Anforderungen im Bereich des eGovernment erfüllende Lösung über dem Stand der Technik, darstellt.

4.3 Anforderungsartbezogene Analyse

In diesem Bereich soll untersucht werden, in welchen als Validierungsarten bezeichneten thematischen Validierungsanforderungskategorien die Primär-Unterschiede zwischen der Anforderungserfüllung von eGovWDF und den aktuellen Lösungen vorliegen.

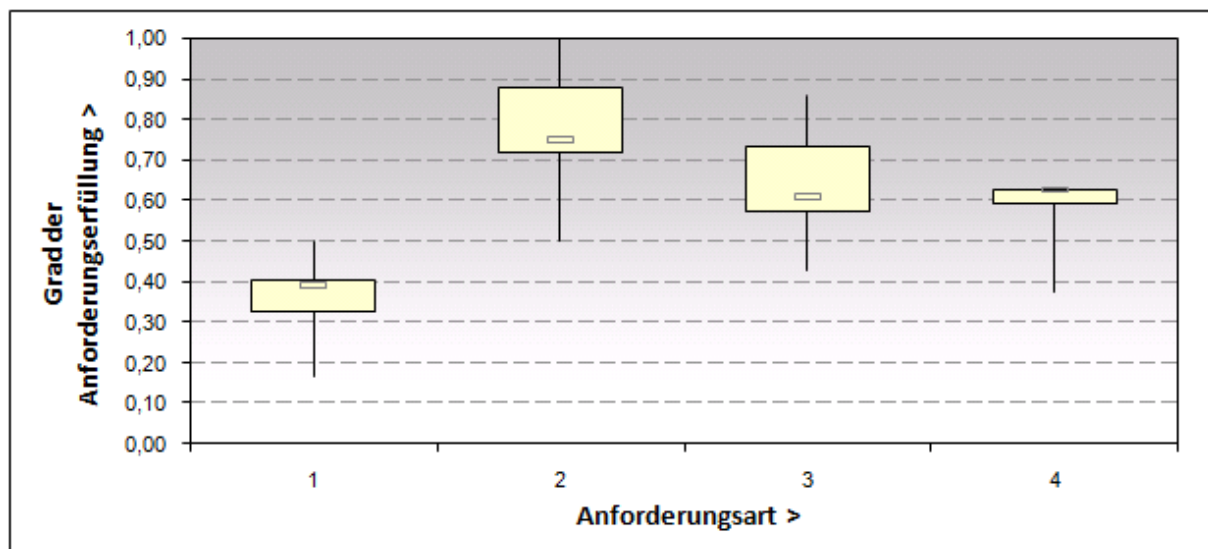


Abbildung 4: Boxplot-gestützte Betrachtung der Anforderungsarten-Anforderungserfüllung.

Wie der Boxplot in Abbildung 4 verdeutlicht (vgl. Anlage E2), werden Anforderungen der verschiedenen Anforderungsarten (1 = Validierungslogik, 2 = Validierungsintegration, 3 = Validierungsvisualisierung, 4 = Horizontale Anforderungen) unterschiedlich gut von den aktuell verfügbaren Validierungslösungen umgesetzt. Besonders im Bereich Validierungslogik, der als zentraler Bereich zu bewerten ist, besitzt der überwiegende Teil der Frameworks nur eine sehr geringe Anforderungserfüllung. Dies ist am Median sowie dem oberen und unterem Quartil erkennbar, da diese Werte im Boxplot alle im Bereich der 30%- und 40%-Grenze liegen. Auch die Betrachtung des oberen Whisker verdeutlicht dieses Bild, da auch Ausreißer lediglich maximal 50% erreichen. Ein wesentlich anderes Bild ergibt sich bei Betrachtung der Box für die Anforderungsart Validierungsintegration. Auch hier liegen Median, oberes und unteres Quartil relativ nahe beieinander. Jedoch sind die Werte wesentlich höher angesiedelt.

Zudem erreicht der obere Whisker den Wert 1,00. Dies bedeutet, dass ein Framework in dieser Anforderungsart volle Anforderungserfüllung erreicht. Für die Anforderungsart Validierungsvisualisierung und die Anforderungsart der horizontalen Anforderungen ergibt sich ein ähnliches Bild wie für den Bereich Validierungsintegration, wenngleich die genannten Kennzahlen lediglich im Bereich der 60%- und 70%-Grenze liegen. Zudem reicht der obere Whisker bei der Anforderungsart Validierungsvisualisierung zwar bis 86%; jedoch bei der Anforderungsart nur bis 63%. Insgesamt lässt sich also feststellen, dass bei allen Anforderungsarten, vor allem aber bei der zentralen Anforderungsart Validierungslogik, nur eine relativ geringe Anforderungserfüllung vorliegt. Wenngleich Ausreißer nach oben vorliegen und deutlich über dem Zentralwert liegende Anforderungserfüllungsgrade in Form von Whiskerwerten dem Boxplot entnommen werden können, ist kein besonders herausragendes Framework in den Bereichen Validierungslogik (zentraler Aspekt) und Horizontale Anforderungen zu identifizieren. Alleine in den Bereichen Validierungsintegration und Validierungsvisualisierung sind Frameworks mit signifikant besserem Anforderungserfüllungsgrad festzustellen. Die Feststellung dieser Frameworks kann über Abbildung 5 (Frameworks 1 bis 12) erfolgen, woraus man erkennen kann, dass im Bereich Validierungsintegration das Framework 6, was der Spring.NET-Validierung entspricht, maximale Anforderungsrealisierung erreicht. Wie man jedoch aus den Abbildungen 4 und 5 auch erkennen kann, liegt dieses hohe Maß der Anforderungserfüllung von Spring.NET nur im Bereich Validierungsintegration, aber nicht in den wesentlich wichtigeren Anforderungsarten wie der Validierungslogik vor, sodass insgesamt kein Framework auf dem Stand der Technik festgestellt werden kann, welches alle besonderen Anforderungen im Bereich des eGovernment entsprechend erfüllt. Im Vergleich dazu erreicht eGovWDF (Framework 13) bei allen Anforderungsarten eine vollständige Anforderungserfüllung.

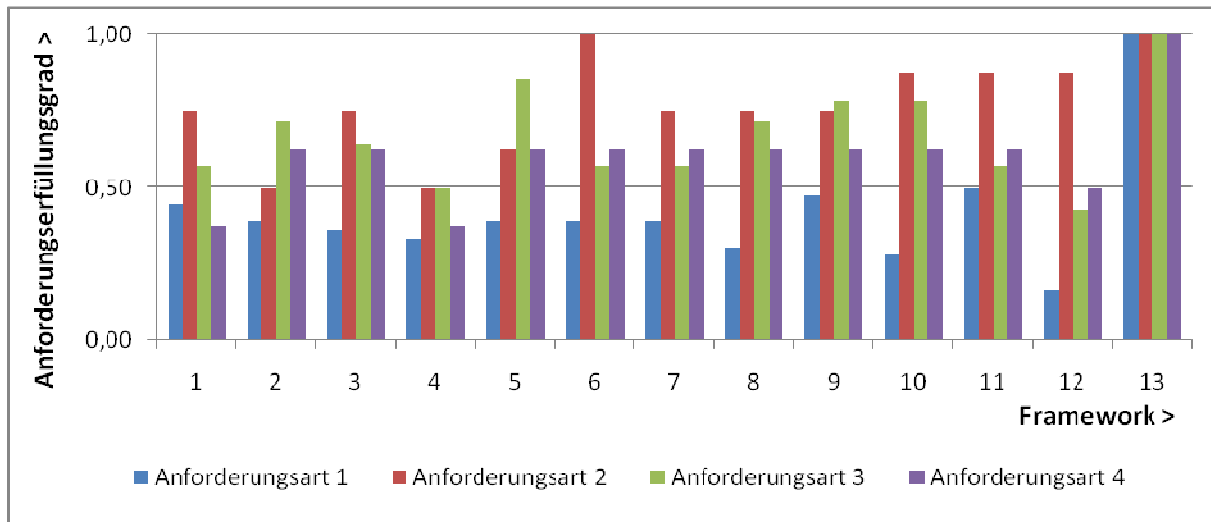


Abbildung 5: Anforderungsarten-Anforderungserfüllungsgrade jedes Frameworks.

Legende:

- Framework 1: XForms-Validierung
- Framework 2: ASP.NET-Validatoren
- Framework 3: JSF-Validierung
- Framework 4: Adobe Flex-Validierung
- Framework 5: WebDynpro-Validierung
- Framework 6: Spring.NET-Validierung
- Framework 7: .net Validation Framework
- Framework 8: Apache MyFaces Trinidad Converters + Validators
- Framework 9: Apache Commons Validator Framework
- Framework 10: Hibernate Validator + Seam-Integration
- Framework 11: XWork validation framework
- Framework 12: LINQ mit ASP.NET
- Framework 13: eGovWDF
- Anforderungsart 1: Validierungslogik
- Anforderungsart 2: Validierungsintegration
- Anforderungsart 3: Validierungsvisualisierung
- Anforderungsart 4: Horizontale Anforderungen

Details zu den Anforderungserfüllungsgradwerten der verschiedenen Frameworks, auch bezogen auf die Einzelanforderungen, können den Anlagen A1 bis E6 entnommen werden.

5 Fazit

Zusammengefasst kann festgehalten werden, dass sowohl bei Betrachtung der Gesamtanforderungserfüllung als auch der Anforderungserfüllungswerte auf Anforderungsartbasis keines der in Kern (2008a) untersuchten Frameworks dazu in der Lage ist, die besonderen Anforderungen im Bereich des eGovernment zu erfüllen. Auch zeigt die Boxplot-getriebene Analyse unter Abschnitt 4.3, dass keines der in Kern (2008a) untersuchten Frameworks sich vom Stand der Technik abhebt. eGovWDF erreicht jedoch einen sehr hohen Prozentsatz an Gesamtanforderungsbefriedigung, was daraus resultiert, dass eGovWDF auf Basis der in Kern (2008a) beschriebenen Anforderungen entwickelt worden ist. Die Gesamtanforderungsbefriedigung von eGovWDF in Höhe von 100% bedeutet dabei, dass eGovWDF sämtliche Einzelanforderungen jeder Anforderungsart vollständig erfüllt, wobei eGovWDF ferner als wesentlich über dem Stand der Technik bewertet werden kann, wenn die statistischen Werte des Vergleichs aller in Kern (2008a) beschriebenen Frameworks betrachtet werden (vgl. Abschnitt 4.3).

Literaturverzeichnis

[Bayerisches Staatsministerium des Innern 2003] Bayerisches Staatsministerium des Innern (Hrsg.): Bayerische Barrierefreie Informationstechnik-Verordnung (BayBITV). Juni 2003

[Biddle und Tempero 1996] Biddle, Robert ; Tempero, Ewan: Explaining inheritance: a code reusability perspective. In: SIGCSE '96: Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education. New York, NY, USA : ACM, 1996, S. 217–221

[Bondi 2000] Bondi, André B.: Characteristics of scalability and their impact on performance. In: WOSP '00: Proceedings of the 2nd international workshop on Software and performance. New York, NY, USA : ACM, 2000, S. 195–203

[Bundesministerium des Innern 1997] Bundesministerium des Innern (Hrsg.): Entwicklungsstandard für IT-Systeme des Bundes (*EStdIT*). Teil 3: Handbuchsammlung Sicherheit und Kritikalität. 1997. – V-Modell 97

[Bundesministerium des Innern 2002] Bundesministerium des Innern (Hrsg.): *Barrierefreie Informationstechnik-Verordnung (BITV)*. Juli 2002. – URL <http://bundesrecht.juris.de/bundesrecht/bitv/gesamt.pdf>

[Coenen 1999] Coenen, Frans: Characteristics of declarative programming languages. 1999. – URL <http://www.csc.liv.ac.uk/~frans/OldLectures/2CS24/declarative.html>
– Letzter Abruf am 27.05.2008

[D'Antonio et al. 2004] D'Antonio, S. ; Esposito, M. ; Romano, S. P. ; Ventre, G.: Assessing the scalability of component-based frameworks: the CADENUS case study. In: *SIGMETRICS Perform. Eval. Rev.* 32 (2004), Nr. 3, S. 34–43

[Danielsen et al. 1993] Danielsen, Rita L. ; Farrand, A. B. ; Wolfe, Susan J.: Searching for help vs. having it handed to you: the relative advantages of index-accessed help and context-sensitive help. In: *CHI '93: INTERACT '93 and CHI '93 conference companion on Human factors in computing systems*. New York, NY, USA : ACM, 1993, S. 147–148

[Gamma et al. 1994] Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John: *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Massachusetts : Addison Wesley, 1994

[Hewitt et al. 1992] Hewitt, T. ; Baecker, R. ; Card, S. ; Carey, T. ; Gasen, J. ; Mantei, M. ; Perlman, G. ; Strong, G. ; Verplank, W: *ACM SIGCHI Curricula for Human-Computer Interaction*. 1992

[Kern 2008a] Kern, Walter: *Validierungsunterstützung in Frameworks zur Webentwicklung: Eine Evaluierung der bedeutendsten Webentwicklungsframeworks im Hinblick auf den Aspekt Validierung im Kontext von eGovernment*. Vdm Verlag Dr. Müller, 9 2008

[Kern 2008b] Kern, Walter: Web 2.0 - End of Accessibility? Analysis of Most Common Problems with Web 2.0 Based Applications Regarding Web Accessibility. In: *International Journal of Public Information Systems* 02 (2008), Nr. 2, S. 131–154

[KBSt 2006] KBSt (Hrsg.): SAGA 3.0: Standards und Architekturen für E-Government-Anwendungen. Oktober 2006. – URL http://www.cio.bund.de/cae/servlet/contentblob/83726/publicationFile/5152/archiv_saga_3_0_download.pdf – Letzter Zugriff am 26.06.2009

[Lee 1987] Lee, W.: ? : a context-sensitive help system based on hypertext. In: *DAC '87: Proceedings of the 24th ACM/IEEE conference on Design automation*. New York, NY, USA : ACM, 1987, S. 429–435

[Ligeza 2006] Ligeza, Antoni: *Logical Foundations for Rule-Based Systems. (Studies in Computational Intelligence)*. 2nd ed. Springer, Berlin, 3 2006.

[Moore 2002] Moore, Anthony: *ASP.NET Validation in Depth*. 2002. – URL [http://msdn2.microsoft.com/en-us/library/aa479045\(printer\).aspx](http://msdn2.microsoft.com/en-us/library/aa479045(printer).aspx). – Letzter Abruf am 09.01.2008

[Netherland und Ganguly 2002] Netherland, Wynn ; Ganguly, Rupak: *Asp.Net Custom Controls*. Greenwich, CT, USA : Manning Publications Co., 2002

[Nielsen 2005] Nielsen, Jakob: *Jakob Nielsen's Alertbox, September 19, 2005: Forms vs. Applications*. 2005. – URL <http://www.useit.com/alertbox/forms.html>. – Letzter Abruf am 12.01.2008

[Pressman 1996] Pressman, Roger S.: *Software Engineering: A Practitioner's Approach*. Fünfte Auflage. McGraw-Hill Higher Education, 1996

[Tukey 1977] Tukey, John W.: *Exploratory Data Analysis (Addison-Wesley Series in Behavioral Science)*. 1. Addison Wesley, 1977

[Vogel et al. 2005] Vogel, Oliver ; Arnold, Ingo ; Chughtai, Arif ; Völter, Markus: *Software-Architektur. Grundlagen - Konzepte - Praxis*. Spektrum Akademischer Verlag, 2005

[W3C 1999] W3C (Hrsg.): *Web Content Accessibility Guidelines 1.0*. 1999. – URL <http://www.w3.org/TR/WAI-WEBCONTENT/>. – Letzter Abruf am 14.06.2008

[Williams 2004] Williams, Jeff: *Input Validation*. AppSec 2004. 2004. – URL http://www.owasp.org/images/3/37/AppSec2004-Jeff_Williams-Input_Validation.ppt. – Letzter Abruf am 01.02.2009

Abbildungsverzeichnis

Abbildung 1: Teilframeworks von eGovWDF.....	8
Abbildung 2: Vergleich der Gesamtanforderungserfüllung akt. Frameworks und eGovWDF	21
Abbildung 3: Verteilung der Gesamtanforderungserfüllung mit und ohne eGovWDF.....	22
Abbildung 4: Boxplot-gestützte Betrachtung der Anforderungsarten-Anforderungserfüllung	23
Abbildung 5: Anforderungsarten-Anforderungserfüllungsgrade jedes Frameworks.....	25

Anlagenverzeichnis

Im Folgenden sind die Detailergebnisse der Evaluierung von eGovWDF angeführt. Dabei gilt folgende übergreifende Legende, falls nicht anders angegeben:

- Framework 1: XForms-Validierung
- Framework 2: ASP.NET-Validatoren
- Framework 3: JSF-Validierung
- Framework 4: Adobe Flex-Validierung
- Framework 5: WebDynpro-Validierung
- Framework 6: Spring.NET-Validierung
- Framework 7: .net Validation Framework
- Framework 8: Apache MyFaces Trinidad Converters + Validators
- Framework 9: Apache Commons Validator Framework
- Framework 10: Hibernate Validator + Seam-Integration
- Framework 11: XWork validation framework
- Framework 12: LINQ mit ASP.NET
- **Framework 13: eGovWDF**

Zusätzlich werden anlagenspezifische Legendendaten unter der jeweiligen Anlage vermerkt.

Hinsichtlich der „Summe der Anforderungserfüllung“ ist anzumerken, dass sich diese aus dem Produkt der Anforderungen und der zugehörigen Relevanzfaktoren (Gewichtungsfaktoren) ergeben. Der (kumulative, stets fett gedruckte) Anforderungserfüllungsgrad berechnet sich hingegen aus der Umrechnung der absoluten „Summe der Anforderungserfüllung“ in einen relativen Wert.

Falls nicht explizit anders angegeben, berücksichtigen die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum sowie das arithmetische Mittel der Anforderungserfüllung stets alle in Kern (2008a) untersuchten Frameworks **mit Ausnahme von eGovWDF**. Diese Festlegung dient der Verhinderung von Messwertverzerrungen, da der aktuelle Stand der Technik unabhängig von eGovWDF als Vergleichsgrundlage für die Analyse von eGovWDF berechnet werden soll. Bei der graphischen Gegenüberstellung der verschiedenen Frameworks zum Vergleich der jeweiligen Anforderungserfüllungsgrade, wird eGovWDF jedoch miteingezeichnet, um einen direkten Vergleich zu erhalten. Die Ermittlung

von Median, oberes Quartil, unteres Quartil, Minimum und Maximum erfolgt neben der Berechnung für jede einzelne Anforderung auch bezogen auf die Anforderungsart (Validierungslogik, Validierungsintegration, Validierungsvisualisierung, horizontale Anforderungen). Diese anforderungsartbezogenen Kennzahlen können bei den validierungsartbezogenen Tabellenwerken jeweils in der rechtesten Spalte in den entsprechenden Zeilen der genannten Kennzahlen abgelesen werden.

Anlage A1: Validierungslogikanforderungen-Tabellenwerk

Diese Anlage zeigt die Erfüllung der Validierungslogik-Anforderungen durch die unten angeführten Frameworks.

Anforderung > ----- Framework v	1	2	3	4	5	6	7	8	9	10	Summe Anforderungser- füllung	Anforde- rungser- füllungs- grad
Relevanzfaktor	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,50		
XForms-Validierung	0,50	0,00	0,50	0,50	0,50	0,00	1,00	1,00	0,50	0,00	4,00	0,44
ASP.NET-Validatoren	0,00	0,00	0,50	1,00	1,00	0,00	0,50	1,00	0,00	0,00	3,50	0,39
JSF-Validierung	0,00	0,00	0,50	1,00	0,50	0,00	0,50	1,00	0,00	0,50	3,25	0,36
Adobe Flex-Validierung	0,00	0,00	0,50	1,00	0,50	0,00	0,50	1,00	0,00	0,00	3,00	0,33
WebDynpro-Validierung	0,00	0,00	0,50	0,50	1,00	0,00	0,50	1,00	0,00	1,00	3,50	0,39
Spring.NET-Validierung	1,00	0,00	0,50	1,00	0,50	0,00	0,50	0,00	0,00	0,00	3,50	0,39
.net Validation Framework	1,00	0,50	0,00	1,00	0,50	0,00	0,50	0,00	0,00	0,00	3,50	0,39
Apache MyFaces Trinidad Converters + Validators	0,00	0,00	0,00	1,00	1,00	0,00	0,50	0,00	0,00	0,50	2,75	0,31
Apache Commons Validator Framew.	1,00	0,00	0,00	1,00	0,50	0,50	1,00	0,00	0,00	0,50	4,25	0,47
Hibernate Validator + Seam-Integration	0,00	0,00	0,00	1,00	0,50	0,00	0,50	0,50	0,00	0,50	2,50	0,28
XWork validation framework	1,00	0,50	0,50	1,00	0,50	0,00	1,00	0,00	0,00	0,00	4,50	0,50
LINQ mit ASP.NET	0,00	0,00	0,50	0,00	0,50	0,00	0,00	1,00	0,00	0,00	1,50	0,17
Median	0,00	0,00	0,50	1,00	0,50	0,00	0,50	0,75	0,00	0,00	-	0,39
Oberes Quartil	1,00	0,00	0,50	1,00	0,63	0,00	0,63	1,00	0,00	0,50	-	0,40
Maximum	1,00	0,50	0,50	1,00	1,00	0,50	1,00	1,00	0,50	1,00	-	0,50
Minimum	0,00	0,00	0,00	0,00	0,50	0,00	0,00	0,00	0,00	0,00	-	0,17
Unteres Quartil	0,00	0,00	0,00	0,88	0,50	0,00	0,50	0,00	0,00	0,00	-	0,33
eGovWDF	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	9,00	1,00

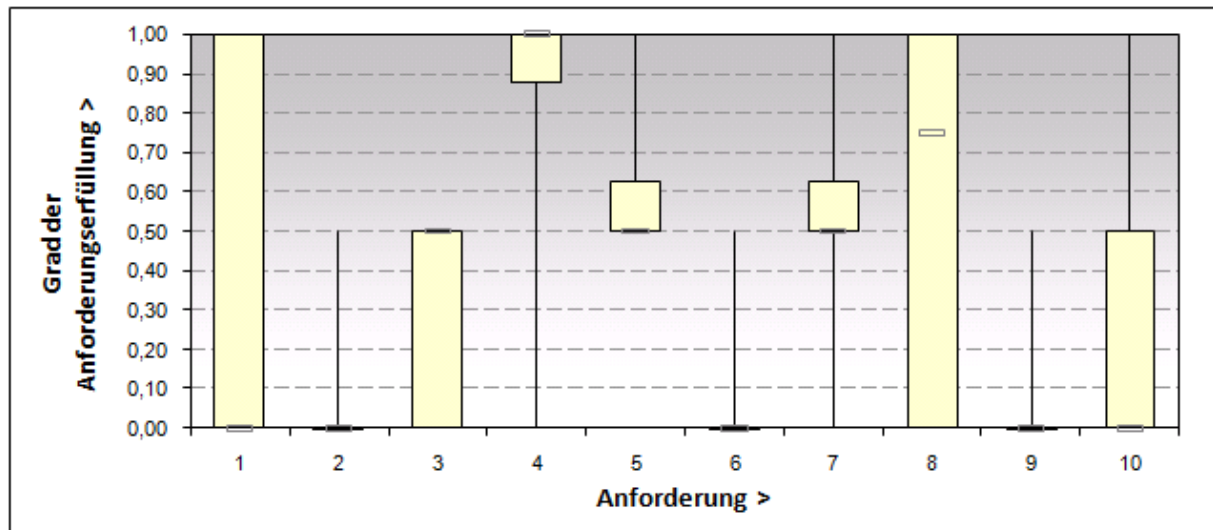
Legende:

- Anforderung 1: Unterstützung deklarativer Validierungslogik-Formulierung
- Anforderung 2: Zentrale und applikationslokale Regelbereitstellung
- Anforderung 3: Wiederverwendbarkeit (Kombinierbarkeit, Vererbung, Skalierung)
- Anforderung 4: Flexible Erweiterbarkeit (Plugin-Konzepte)
- Anforderung 5: Dualität der Validierung (client- und serverseitig)
- Anforderung 6: Bel. Beschreibungsdatenformat / Persistenzlokation

- Anforderung 7: Umfang ab Werk ausgelieferter Validierungslogik
- Anforderung 8: Graphische Modellierbarkeit (Tool-Support)
- Anforderung 9: Hot-Remote-Propagierung (Update im laufenden Betrieb)
- Anforderung 10: Unterstützung verschiedener Fehlerkritikalitäten

Anlage A2: Validierungslogikanforderungen-Boxplot

Diese Anlage visualisiert die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum pro Validierungslogik-Anforderung. Dabei visualisiert eine Box samt zugehöriger Whisker-Linien die Erfüllung der zugehörigen Validierungslogik-Anforderung durch die in Kern (2008a) angeführten Validierungsframeworks.

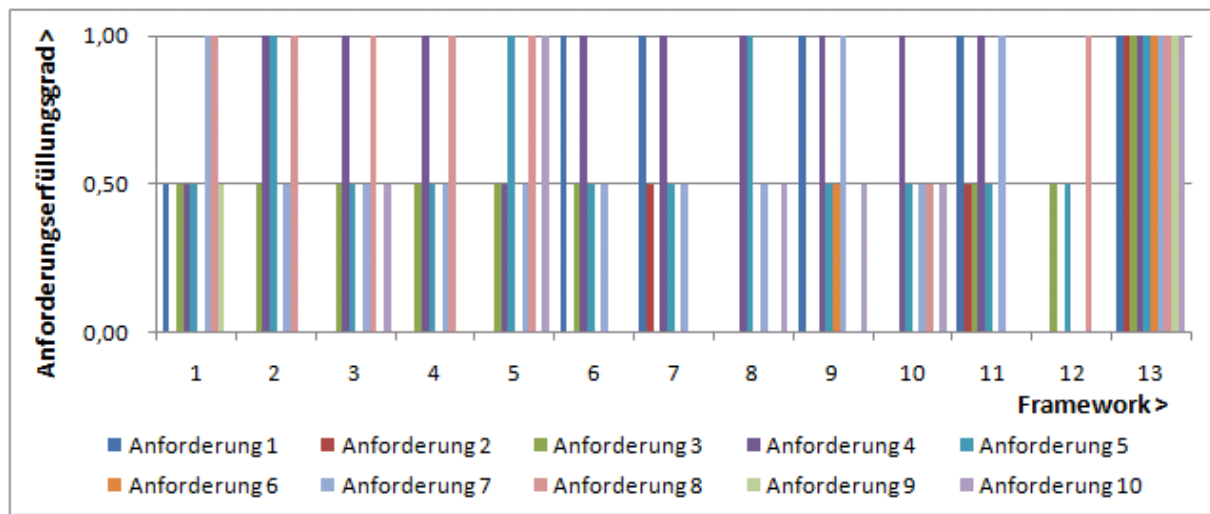


Legende:

- Anforderung 1: Unterstützung deklarativer Validierungslogik-Formulierung
- Anforderung 2: Zentrale und applikationslokale Regelbereitstellung
- Anforderung 3: Wiederverwendbarkeit (Kombinierbarkeit, Vererbung, Skalierung)
- Anforderung 4: Flexible Erweiterbarkeit (Plugin-Konzepte)
- Anforderung 5: Dualität der Validierung (client- und serverseitig)
- Anforderung 6: Bel. Beschreibungsdatenformat / Persistenzlokation
- Anforderung 7: Umfang ab Werk ausgelieferter Validierungslogik
- Anforderung 8: Graphische Modellierbarkeit (Tool-Support)
- Anforderung 9: Hot-Remote-Propagierung (Update im laufenden Betrieb)
- Anforderung 10: Unterstützung verschiedener Fehlerkritikalitäten

Anlage A3: Validierungslogik-Framework-Anforderungserfüllung

Diese Anlage visualisiert für jedes Framework (nummeriert von 1 bis 13) den Erfüllungsgrad der Validierungslogik-Anforderungen.



Legende:

- Anforderung 1: Unterstützung deklarativer Validierungslogik-Formulierung
- Anforderung 2: Zentrale und applikationslokale Regelbereitstellung
- Anforderung 3: Wiederverwendbarkeit (Kombinierbarkeit, Vererbung, Skalierung)
- Anforderung 4: Flexible Erweiterbarkeit (Plugin-Konzepte)
- Anforderung 5: Dualität der Validierung (client- und serverseitig)
- Anforderung 6: Bel. Beschreibungsdatenformat / Persistenzlokation
- Anforderung 7: Umfang ab Werk ausgelieferter Validierungslogik
- Anforderung 8: Graphische Modellierbarkeit (Tool-Support)
- Anforderung 9: Hot-Remote-Propagierung (Update im laufenden Betrieb)
- Anforderung 10: Unterstützung verschiedener Fehlerkritikalitäten

Anlage B1: Validierungsintegrationanforderungen-Tabellenwerk

Diese Anlage zeigt die Erfüllung der Validierungsintegration-Anforderungen durch die unten angeführten Frameworks.

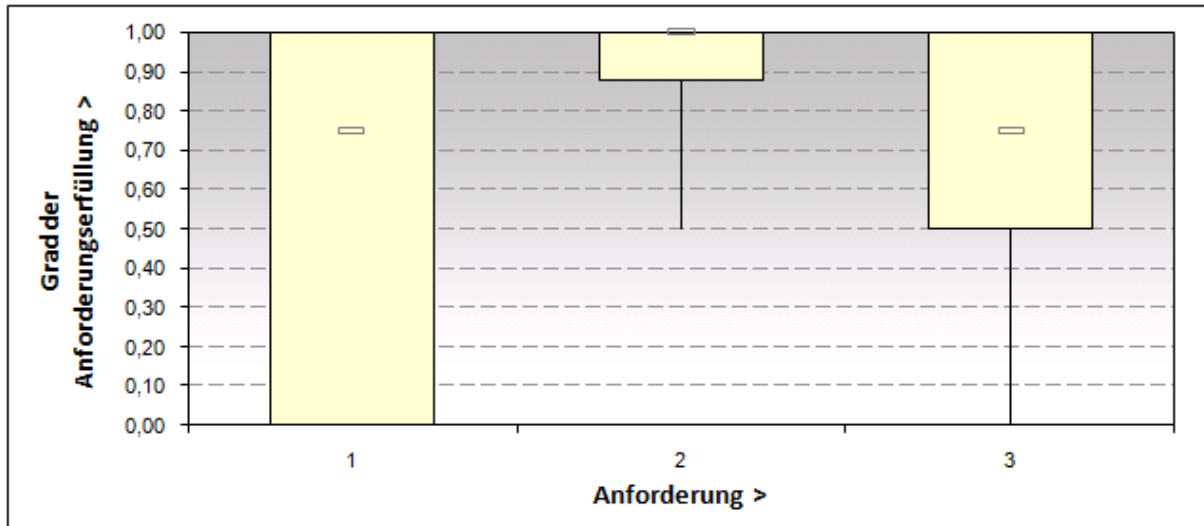
Anforderung > ----- Framework v	1	2	3	Summe Anforde- runge- erfüllung	Anforde- rungeer- füllungs- grad
Relevanzfaktor	0,50	1,00	0,50		
XForms-Validierung	1,00	0,50	1,00	<u>1,50</u>	0,75
ASP.NET-Validatoren	0,00	0,50	1,00	<u>1,00</u>	0,50
JSF-Validierung	0,00	1,00	1,00	<u>1,50</u>	0,75
Adobe Flex-Validierung	0,00	0,50	1,00	<u>1,00</u>	0,50
WebDynpro-Validierung	0,00	1,00	0,50	<u>1,25</u>	0,63
Spring.NET-Validierung	1,00	1,00	1,00	<u>2,00</u>	1,00
.net Validation Framework	0,50	1,00	0,50	<u>1,50</u>	0,75
Apache MyFaces Trinidad Converters + Validators	0,00	1,00	1,00	<u>1,50</u>	0,75
Apache Commons Validator Framew.	1,00	1,00	0,00	<u>1,50</u>	0,75
Hibernate Validator mit Seam-Integration	1,00	1,00	0,50	<u>1,75</u>	0,88
XWork validation framework	1,00	1,00	0,50	<u>1,75</u>	0,88
LINQ mit ASP.NET	1,00	1,00	0,50	<u>1,75</u>	0,88
Median	0,75	1,00	0,75	-	0,75
Oberes Quartil	1,00	1,00	1,00	-	0,88
Maximum	1,00	1,00	1,00	-	1,00
Minimum	0,00	0,50	0,00	-	0,50
Unteres Quartil	0,00	0,88	0,50	-	0,72
eGovWDF	1,00	1,00	1,00	<u>2,00</u>	1,00

Legende:

- Anforderung 1: Explizite Integrationsschicht für beliebige Plattformen/Frameworks
- Anforderung 2: Lose Koppelung zwischen Logik und Visualisierung
- Anforderung 3: Unterstützung der Validierung komplexer plattformspez. Komp.

Anlage B2: Validierungsintegrationanforderungen-Boxplot

Diese Anlage visualisiert die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum pro Validierungsintegration-Anforderung. Dabei visualisiert eine Box samt zugehöriger Whisker-Linien die Erfüllung der zugehörigen Validierungsintegration-Anforderung durch die in Kern (2008a) angeführten Validierungsframeworks.

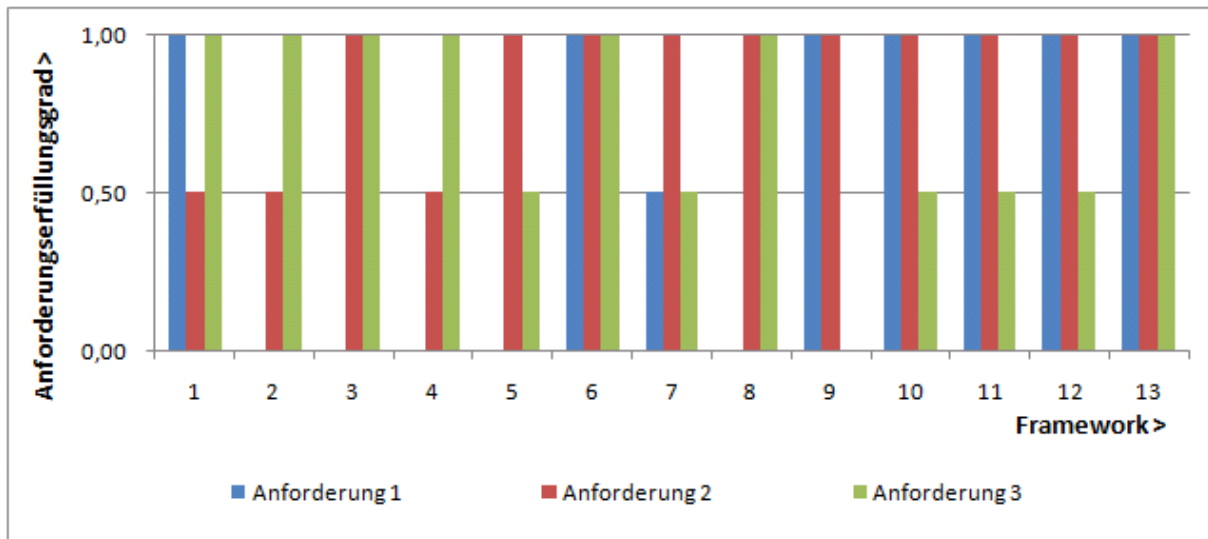


Legende:

- Anforderung 1: Explizite Integrationsschicht für beliebige Plattformen/Frameworks
- Anforderung 2: Lose Koppelung zwischen Logik und Visualisierung
- Anforderung 3: Unterstützung der Validierung komplexer plattformspez. Komp.

Anlage B3: Validierungsintegration-Framework- Anforderungserfüllung

Diese Anlage visualisiert für jedes Framework (nummeriert von 1 bis 13) den Erfüllungsgrad der Validierungsintegration-Anforderungen.



Legende:

- Anforderung 1: Explizite Integrationsschicht für beliebige Plattformen/Frameworks
- Anforderung 2: Lose Koppelung zwischen Logik und Visualisierung
- Anforderung 3: Unterstützung der Validierung komplexer plattformspez. Komp.

Anlage C1: Validierungsvisualisierungsanforderungen-Tabellenwerk

Diese Anlage zeigt die Erfüllung der Validierungsvisualisierung-Anforderungen durch die unten angeführten Frameworks.

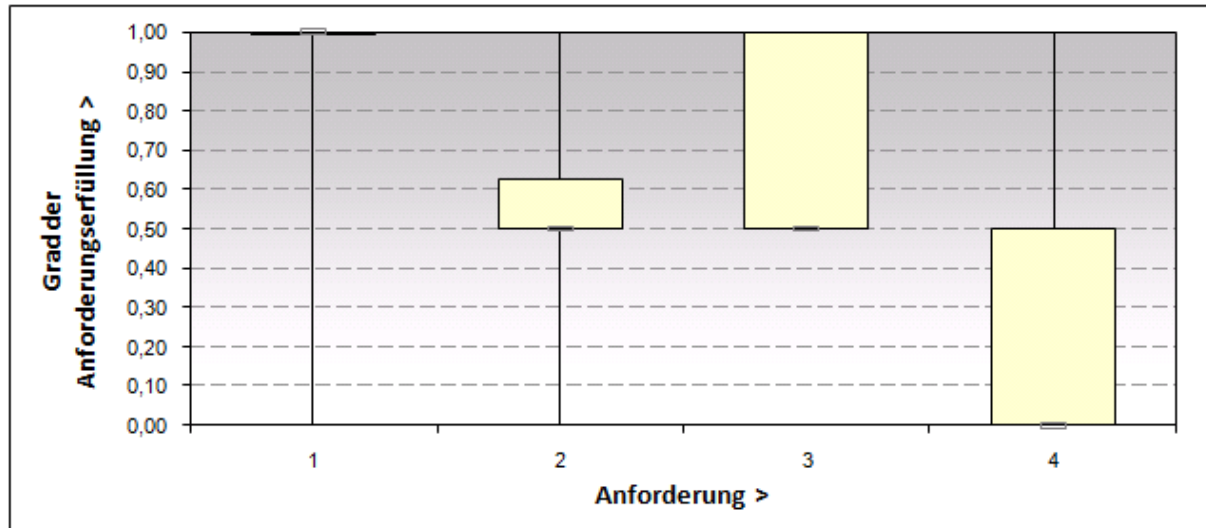
Anforderung > ----- -- Framework v	1	2	3	4	Summe Anforde- rungser- füllung	Anforde- rungser- füllungs- grad
Relevanzfaktor	1,00	1,00	1,00	0,50		
XForms- Validierung	1,00	0,50	0,50	0,00	<u>2,00</u>	0,57
ASP.NET- Validatoren	1,00	0,50	1,00	0,00	<u>2,50</u>	0,71
JSF- Validierung	1,00	0,50	0,50	0,50	<u>2,25</u>	0,64
Adobe Flex- Validierung	0,00	1,00	0,50	0,50	<u>1,75</u>	0,50
WebDynpro- Validierung	0,50	1,00	1,00	1,00	<u>3,00</u>	0,86
Spring.NET- Validierung	1,00	0,50	0,50	0,00	<u>2,00</u>	0,57
.net Validation Fra- mework	1,00	0,50	0,50	0,00	<u>2,00</u>	0,57
Apache MyFaces Trinidad Conver- ters + Validators	1,00	0,50	1,00	0,00	<u>2,50</u>	0,71
Apache Commons Validator Framew.	1,00	0,50	1,00	0,50	<u>2,75</u>	0,79
Hibernate Validator mit Seam- Integration	1,00	1,00	0,50	0,50	<u>2,75</u>	0,79
XWork validation framework	1,00	0,50	0,50	0,00	<u>2,00</u>	0,57
LINQ mit ASP.NET	1,00	0,00	0,50	0,00	<u>1,50</u>	0,43
Median	1,00	0,50	0,50	0,00	-	0,61
Oberes Quartil	1,00	0,63	1,00	0,50	-	0,73
Maximum	1,00	1,00	1,00	1,00	-	0,86
Minimum	0,00	0,00	0,50	0,00	-	0,43
Unteres Quartil	1,00	0,50	0,50	0,00	-	0,57
eGovWDF	1,00	1,00	1,00	1,00	<u>3,50</u>	1,00

Legende:

- Anforderung 1: Barrierefreiheit
- Anforderung 2: Assistiver Charakter
- Anforderung 3: Clientseitige und serverseitige Repräsentation
- Anforderung 4: Fehlerkritialitäts-Visualisierung

Anlage C2: Validierungsvisualisierungsanforderungen-Boxplot

Diese Anlage visualisiert die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum pro Validierungsvisualisierung-Anforderung. Dabei visualisiert eine Box samt zugehöriger Whisker-Linien die Erfüllung der zugehörigen Validierungsvisualisierung-Anforderung durch die in Kern (2008a) angeführten Validierungsframeworks.

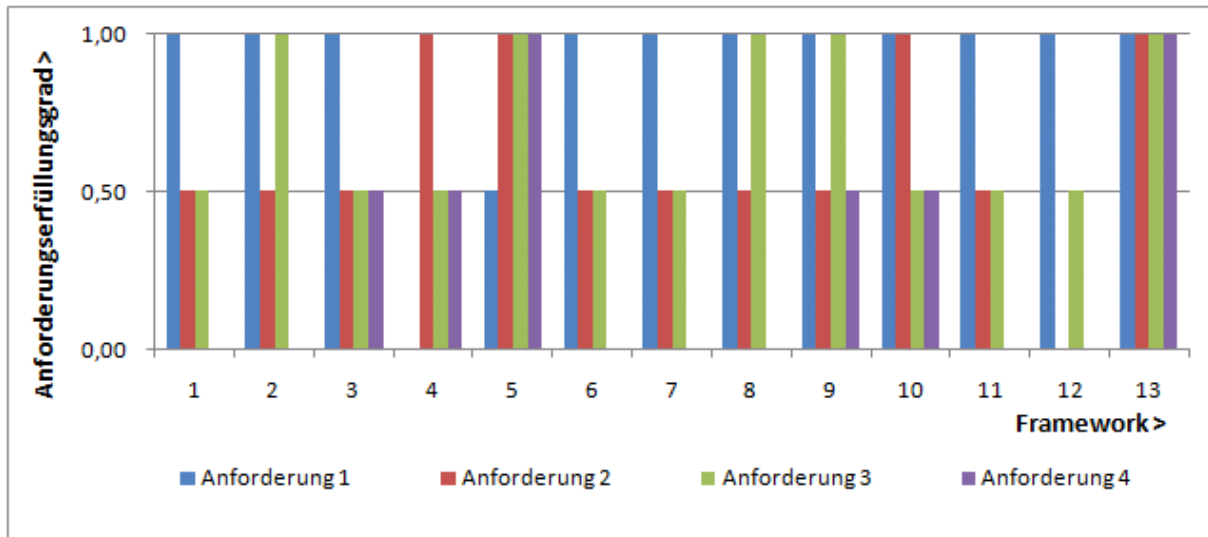


Legende:

- Anforderung 1: Barrierefreiheit
- Anforderung 2: Assistiver Charakter
- Anforderung 3: Clientseitige und serverseitige Repräsentation
- Anforderung 4: Fehlerkritikalitäts-Visualisierung

Anlage C3: Validierungsvisualisierung-Framework- Anforderungserfüllung

Diese Anlage visualisiert für jedes Framework (nummeriert von 1 bis 13) den Erfüllungsgrad der Validierungsvisualisierung-Anforderungen.



Legende:

- Anforderung 1: Barrierefreiheit
- Anforderung 2: Assistiver Charakter
- Anforderung 3: Clientseitige und serverseitige Repräsentation
- Anforderung 4: Fehlerkritikalitäts-Visualisierung

Anlage D1: Horizontale Validierungsanforderungen-Tabellenwerk

Diese Anlage zeigt die Erfüllung der horizontalen Validierungsanforderungen durch die unten angeführten Frameworks.

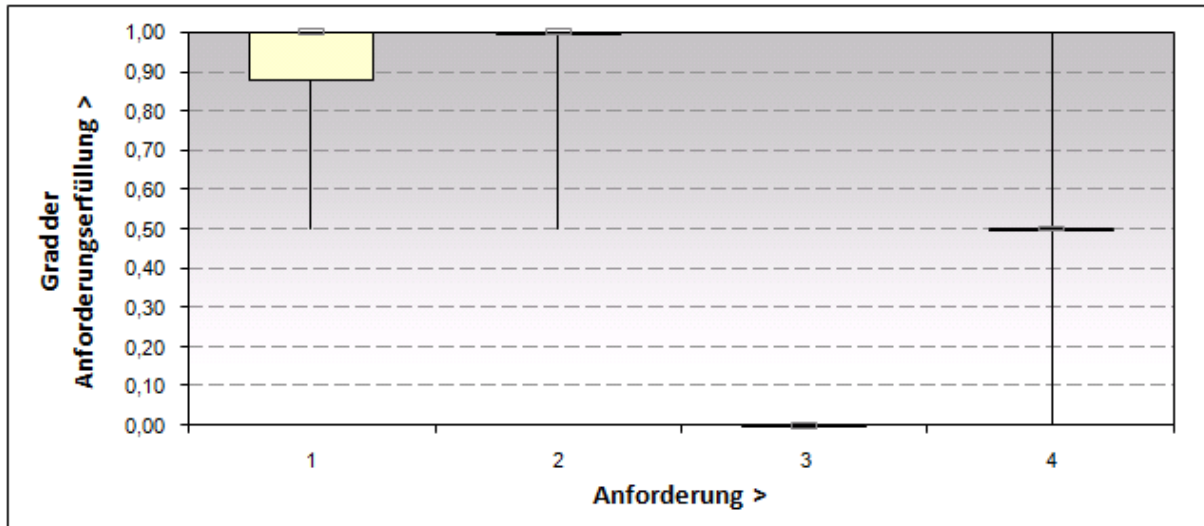
Anforderung > ----- -- Framework v	1	2	3	4	Summe Anforderungser- füllung	Anforde- rungser- füllungs- grad
Relevanzfaktor	1,00	1,00	1,00	1,00		
XForms-Validierung	0,50	0,50	0,00	0,50	<u>1,50</u>	0,38
ASP.NET-Validatoren	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
JSF-Validierung	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
Adobe Flex-Validierung	0,50	0,50	0,00	0,50	<u>1,50</u>	0,38
WebDynpro-Validierung	0,50	1,00	0,00	1,00	<u>2,50</u>	0,63
Spring.NET-Validierung	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
.net Validation Framework	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
Apache MyFaces Trinidad Converters + Validators	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
Apache Commons Validator Framew.	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
Hibernate Validator mit Seam-Integration	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
XWork validation framework	1,00	1,00	0,00	0,50	<u>2,50</u>	0,63
LINQ mit ASP.NET	1,00	1,00	0,00	0,00	<u>2,00</u>	0,50
Median	1,00	1,00	0,00	0,50	-	0,63
Oberes Quartil	1,00	1,00	0,00	0,50	-	0,63
Maximum	1,00	1,00	0,00	1,00	-	0,63
Minimum	0,50	0,50	0,00	0,00	-	0,38
Unteres Quartil	0,88	1,00	0,00	0,50	-	0,59
eGovWDF	1,00	1,00	1,00	1,00	<u>4,00</u>	1,00

Legende:

- Anforderung 1: Plattformneutralität
- Anforderung 2: Zusatzkomponentenerfordernisse
- Anforderung 3: Berücksichtigung von Sicherheitsaspekten
- Anforderung 4: Simplizität der Anwendung

Anlage D2: Horizontale Validierungsanforderungen-Boxplot

Diese Anlage visualisiert die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum pro horizontaler Validierungsanforderung. Dabei visualisiert eine Box samt zugehöriger Whisker-Linien die Erfüllung der zugehörigen horizontalen Validierungs-Anforderung durch die in Kern (2008a) angeführten Validierungsframeworks.

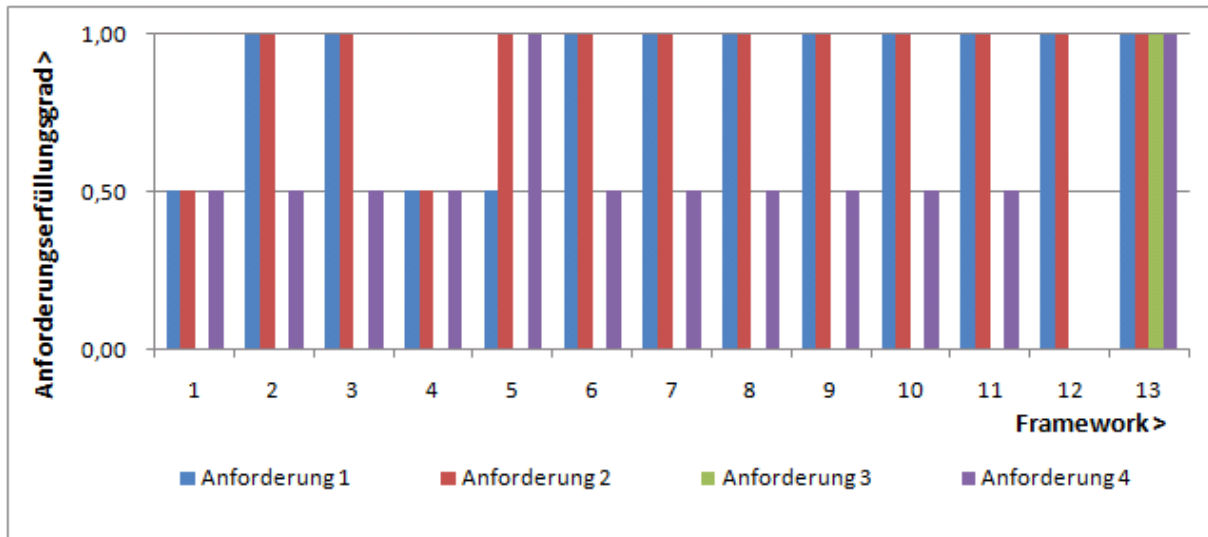


Legende:

- Anforderung 1: Plattformneutralität
- Anforderung 2: Zusatzkomponentenerfordernisse
- Anforderung 3: Berücksichtigung von Sicherheitsaspekten
- Anforderung 4: Simplizität der Anwendung

Anlage D3: Horizontale Anforderungen-Framework- Anforderungserfüllung

Diese Anlage visualisiert für jedes Framework (nummeriert von 1 bis 13) den Erfüllungsgrad der horizontalen Anforderungen.



Legende:

- Anforderung 1: Plattformneutralität
- Anforderung 2: Zusatzkomponentenerfordernisse
- Anforderung 3: Berücksichtigung von Sicherheitsaspekten
- Anforderung 4: Simplizität der Anwendung

Anlage E1: Validierungsartanforderungen-Tabellenwerk

Diese Anlage zeigt die Erfüllung aller Validierungsanforderungen durch die unten angeführten Frameworks. Dabei findet eine Aggregation nach Validierungsart (Validierungslogik, Validierungsintegration, Validierungsvisualisierung, horizontale Anforderungen) statt, d.h. es wird anstelle der Betrachtung jeder Einzelanforderung der Validierungsarten nur die Gesamtanforderungserfüllung jeder Validierungsart betrachtet. Wichtig ist hierbei, dass die Anforderungserfüllungswerte hier bereits die Relevanzfaktoren (Relevanzdivisor-Werte) beinhalten, sodass die „Summe Anforderungserfüllung“ sich jeweils aus der simplen Summation der links davon stehenden Werte ergibt. Unabhängig davon, dass bei den einzelnen Anforderungserfüllungswerten (z.B. 4,00 bei XForms-Validierung und Framework 1) Absolutwerte vorliegen, werden die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Maximum und Minimum stets auf Basis des relativen Anforderungserfüllungsgrads berechnet.

Anforderung > ----- Framework v	1	2	3	4	Summe Anforderungserfüllung	Gesamt-Anforderungserfüllungsgrad
Relevanzdivisor	9,00	2,00	3,50	4,00		
XForms-Validierung	4,00	1,50	2,00	1,50	<u>9,00</u>	0,49
ASP.NET-Validatoren	3,50	1,00	2,50	2,50	<u>9,50</u>	0,51
JSF-Validierung	3,25	1,50	2,25	2,50	<u>9,50</u>	0,51
Adobe Flex-Validierung	3,00	1,00	1,75	1,50	<u>7,25</u>	0,39
WebDynpro-Validierung	3,50	1,25	3,00	2,50	<u>10,25</u>	0,55
Spring.NET-Validierung	3,50	2,00	2,00	2,50	<u>10,00</u>	0,54
.net Validation Framework	3,50	1,50	2,00	2,50	<u>9,50</u>	0,51
Apache MyFaces Trinidad Converters + Validators	2,75	1,50	2,50	2,50	<u>9,25</u>	0,50
Apache Commons Validator Framew.	4,25	1,50	2,75	2,50	<u>11,00</u>	0,59
Hibernate Validator mit Seam-Integration	2,50	1,75	2,75	2,50	<u>9,50</u>	0,51
XWork validation framework	4,50	1,75	2,00	2,50	<u>10,75</u>	0,58
LINQ mit ASP.NET	1,50	1,75	1,50	2,00	<u>6,75</u>	0,36
Median	0,39	0,75	0,61	0,63	-	0,51
Oberes Quartil	0,40	0,88	0,73	0,63	-	0,54
Maximum	0,50	1,00	0,86	0,63	-	0,59

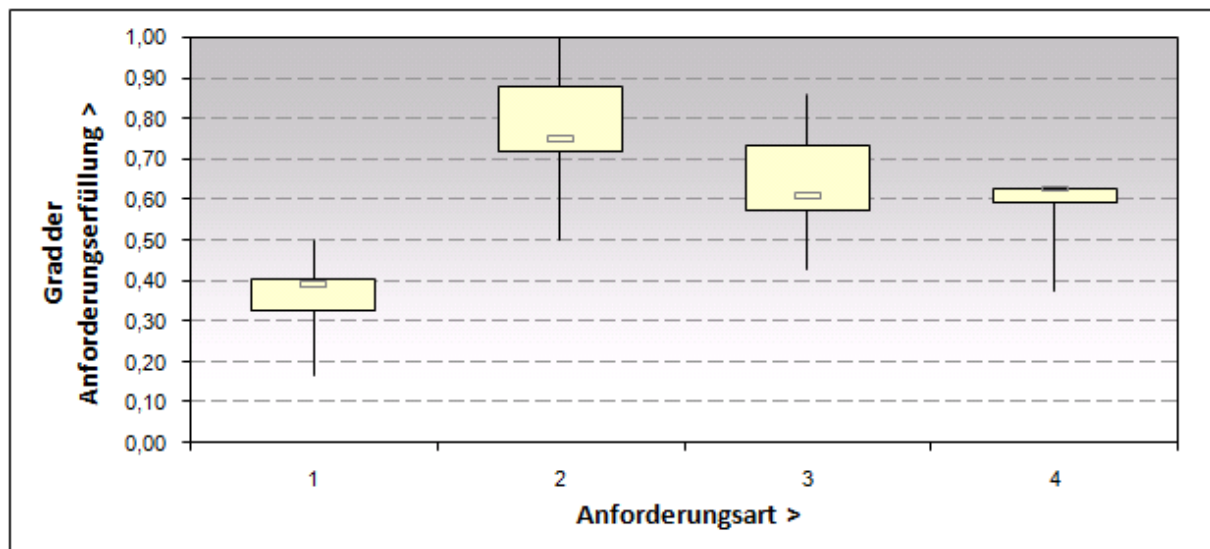
Minimum	0,17	0,50	0,43	0,38	-	0,36
Unteres Quartil	0,33	0,72	0,57	0,59	-	0,50
eGovWDF	9,00	2,00	3,50	4,00	18,50	1,00

Legende:

- Anforderungsart 1: Validierungslogik
- Anforderungsart 2: Validierungsintegration
- Anforderungsart 3: Validierungsvisualisierung
- Anforderungsart 4: Horizontale Anforderungen

Anlage E2: Validierungsartanforderungen-Boxplot

Diese Anlage visualisiert die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum pro Validierungsart. Dabei visualisiert eine Box samt zugehöriger Whisker-Linien die Erfüllung der zugehörigen Validierungsart-Anforderung (aggregiert auf Basis der zur Validierungsart gehörigen Einzelanforderungen) durch die in Kern (2008a) angeführten Validierungsframeworks.

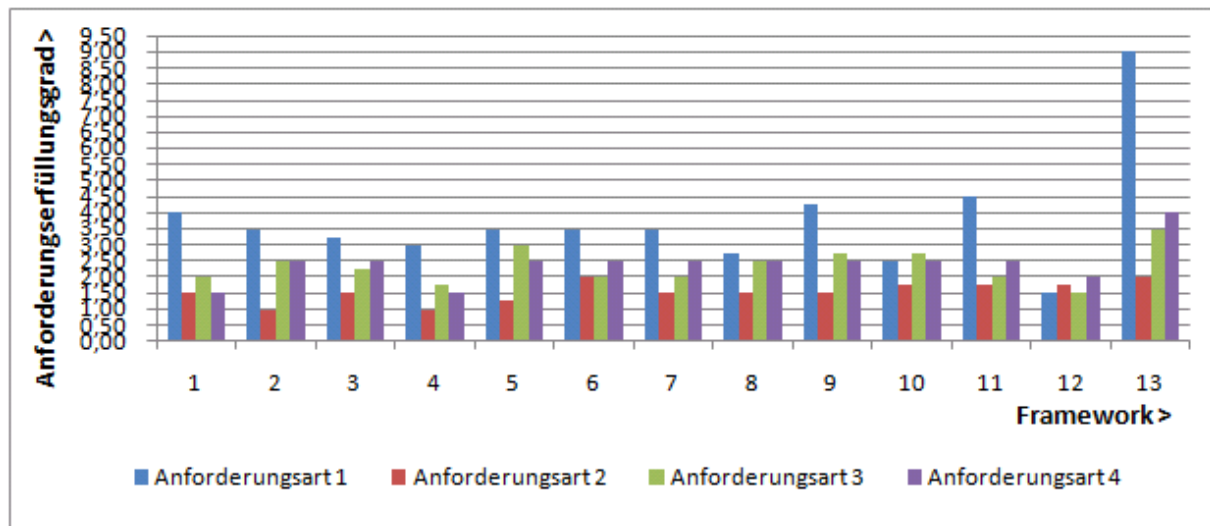


Legende:

- Anforderungsart 1: Validierungslogik
- Anforderungsart 2: Validierungsintegration
- Anforderungsart 3: Validierungsvisualisierung
- Anforderungsart 4: Horizontale Anforderungen

Anlage E3: Absolute Anforderungsart-Framework-Anforderungserfüllung

Diese Anlage visualisiert für jedes Framework (nummeriert von 1 bis 13) den absoluten Wert der Anforderungserfüllung für jede Anforderungsart.

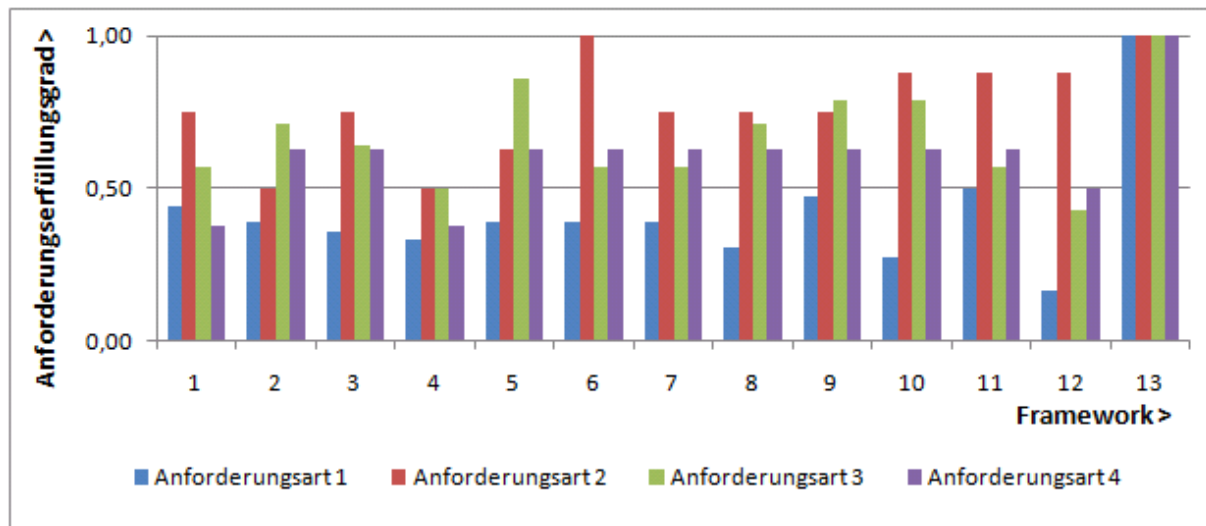


Legende:

- Anforderungsart 1: Validierungslogik
- Anforderungsart 2: Validierungsintegration
- Anforderungsart 3: Validierungsvisualisierung
- Anforderungsart 4: Horizontale Anforderungen

Anlage E4: Relative Anforderungsart-Framework-Anforderungserfüllung

Diese Anlage visualisiert für jedes Framework (nummeriert von 1 bis 13) den relativen Wert der Anforderungserfüllung (Anforderungserfüllungsgrad) für jede Anforderungsart.

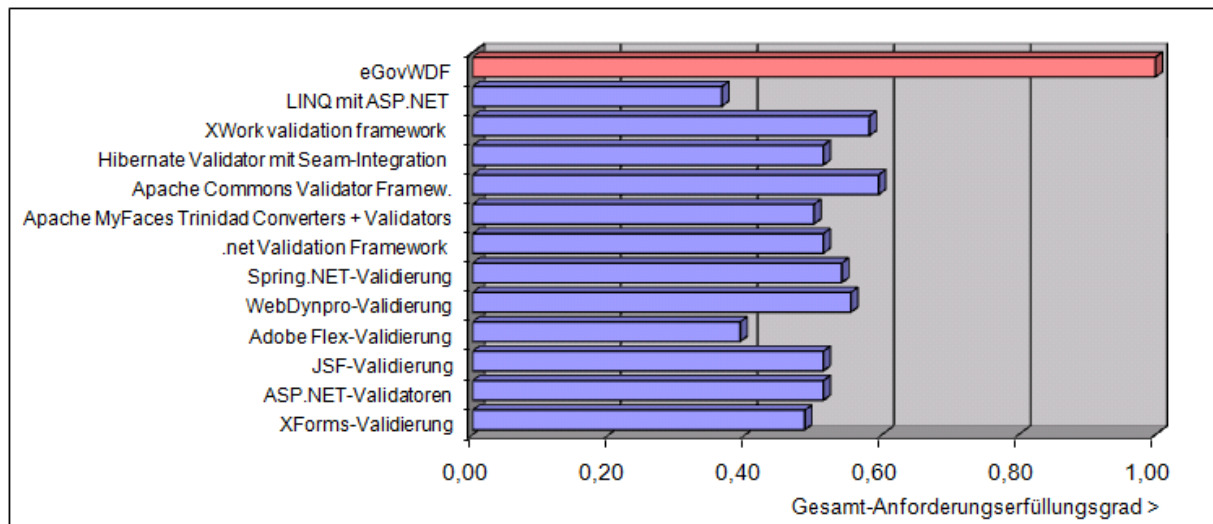


Legende:

- Anforderungsart 1: Validierungslogik
- Anforderungsart 2: Validierungsintegration
- Anforderungsart 3: Validierungsvisualisierung
- Anforderungsart 4: Horizontale Anforderungen

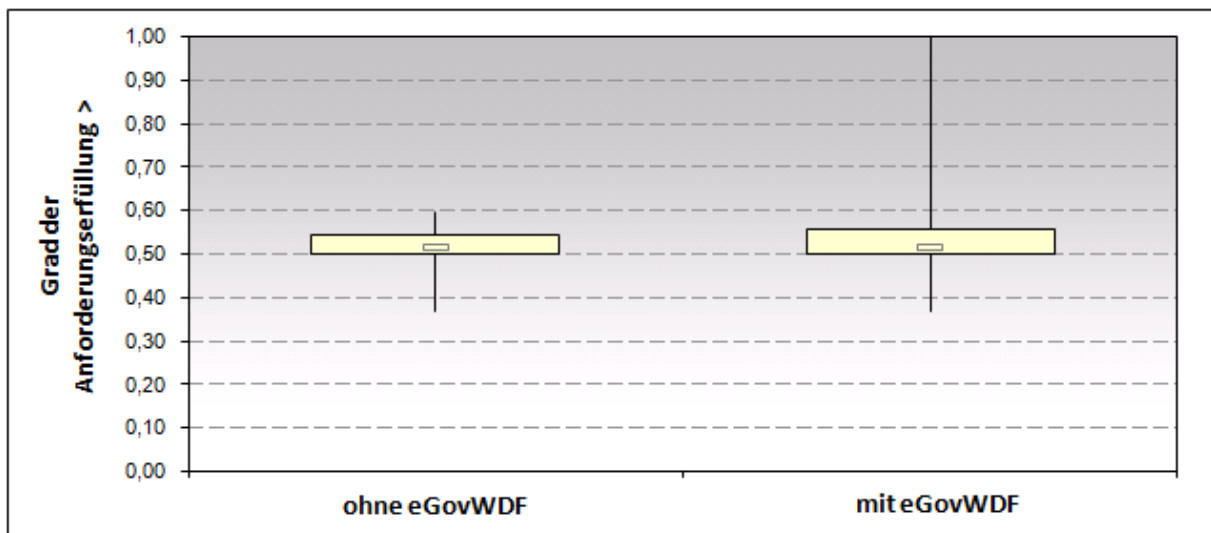
Anlage E5: Gesamtanforderungserfüllungsvergleich- Balkendiagramm

Diese Anlage stellt die Gesamtanforderungserfüllungsgrade von eGovWDF und der in Kern (2008a) getesteten Frameworks für einen Überblicksvergleich gegenüber.



Anlage E6: Gesamtanforderungserfüllungsvergleich-Boxplot

Diese Anlage visualisiert die statistischen Kennzahlen Median, oberes Quartil, unteres Quartil, Minimum und Maximum bezogen auf die Gesamtanforderungserfüllungsgrade von eGovWDF und der in Kern (2008a) getesteten Frameworks. Dabei werden die statistischen Kennzahlen im Fall der Nicht-Berücksichtigung und im Fall der Berücksichtigung von eGovWDF ermittelt und dargestellt.



Detailergebnisse:

	ohne eGovWDF	mit eGovWDF
Median	0,51	0,51
O. Quartil	0,54	0,55
Maximum	0,59	1,00
Minimum	0,36	0,36
U. Quartil	0,50	0,50

Legende:

- Anforderungsart 1: Validierungslogik
- Anforderungsart 2: Validierungsintegration
- Anforderungsart 3: Validierungsvisualisierung
- Anforderungsart 4: Horizontale Anforderungen