msg systems ag

BG ETEM Intranet Präventionswerkzeuge

Nichtfunktionale Anforderungen

Version	1.4
Stand:	11.07.2014 15:37:00
Autor:	Achim.Mueller@msg-systems.com
Ablage:	70_Nichtfunktionale Anforderungen.docx
Umfang:	13 Seiten

Versionshistorie

Version	Beschreibung	Autor	Datum
0.1	Initial angelegt und erste Inhalte eingepflegt	Daniel Mager	03.04.2014
0.9	Überarbeitung und Fertigstellung für Freigabe	Achim Müller	17.04.2014
1.0	Freigabe zur Abnahme	Stefan Hofmaier	23.04.2014
1.1	Befundung durch BG ETEM	BG ETEM	16.05.2014

BG ETEM Intranet Präventionswerkzeuge
Nichtfunktionale Anforderungen

Review

Version	Datum	Teilnehmer

0.9	23.04.2014	Stefan Hofmaier

1 Inhaltsverzeichnis

2 Einleitung

2.1 Zweck

Dieses Dokument erfasst die Anforderungen an das System die nicht direkt einer fachlichen Funktionalität zugesprochen werden können („nichtfunktionale Anforderungen“).

2.2 Referenzen

Das Dokument nimmt Bezug auf den Abschnitt "1.2 Qualitätsziele" im Dokument "Intranet Präventionswerkzeuge-Systemarchitektur.pdf" in der Fassung 1.1 vom 28. März 2013.

Die dort aufgeführten Qualitätsziele

- Funktionalität
- Sicherheit
- Zuverlässigkeit
- Benutzbarkeit
- Effizienz
- Wartbarkeit
- Testbarkeit

werden im Folgenden aufgegriffen und beschrieben, wie diese erreicht werden sollen. Wo sinnvoll und möglich werden konkret messbare Ziele vorgegeben.

2.3 Abgrenzung

Funktionale Anforderungen werden in diesem Dokument nicht beschrieben. Die Funktionalität wird über die Vollständigkeit der Anwendungsfälle und deren Review durch den Auftraggeber gewährleistet.

Der Nachweis über den Funktionsumfang und die Korrektheit der Ausführung erfolgt über verschiedene Tests, die im Testkonzept beschrieben werden.

3 Sicherheit

3.1 Prämisse

Die Anwendung wird als Intranet-Anwendung betrieben und der Umfang der Schutzmaßnahmen an dieser Klassifizierung ausgerichtet. Die für die Risikoermittlung und Definition von Sicherheitsmaßnahmen relevanten Merkmale einer Intranet-Anwendung sind:

1. Die Anwendung, bzw. der (Web-)Server ist nicht über den öffentlichen Teil des Internets, sondern nur über ein eigenes Unternehmens-internes Netzwerk erreichbar.
2. Der Benutzerkreis ist nicht-öffentlich und umfasst lediglich Mitarbeiter des Unternehmens oder dritte Personen, die dem Unternehmen bekannt sind und als vertrauenswürdig und sicherheitstechnisch unkritisch angesehen werden.

3.2 Anforderungen an die Applikationssicherheit

Die Sicherheitsanforderungen an die Anwendung umfassen folgende Hauptaspekte:

1. Unberechtigte Personen sollen keinen Zugang zur Anwendung erhalten
2. Angemeldete Benutzer sollen keinen Zugang zu Daten und Dokumenten erhalten, für welche sie nicht autorisiert sind.

3.3 Sicherstellung der Applikationssicherheit

3.3.1 Benutzeridentifikation

Die Anwendung ist zugangsgeschützt. Dies bedeutet, dass sich Benutzer erst einmal „einloggen“, also mit einer gültigen Benutzername/Passwort-Kombination identifizieren müssen, bevor ihnen ein Zugriff auf Teile der Applikation (außerhalb der Login-Maske) gewährt wird.

Bei (Windows-)Arbeitsplätzen, an denen bereits ein Benutzer angemeldet ist, kann dieser im Sinne des Single-Sign-Ons die Anwendung ohne erneute Identifikation verwenden.

Als Authentifizierungsverfahren kommt die HTTP Basic Authentication vom Browser zum Server zur Anwendung. Dieses Authentifizierungsverfahren ist seit Jahren Standard im Internetumfeld und gilt in Kombination mit HTTPS als sicher. Die Authentifizierung erfolgt serverseitig gegen den konfigurierten LDAP Server.

Da das Passwort bei diesem Authentifizierungsverfahren praktisch im Klartext übertragen wird, ist eine Kombination mit HTTPS zwingend notwendig. SSL verschlüsselt die gesamte Kommunikation zwischen Browser und Server und sorgt dafür, dass kein Dritter die übertragenen Daten mitlesen oder manipulieren kann.

3.3.2 Autorisierung

Der Anwendung liegt ein Benutzer-, Rollen- und Rechtemodell zugrunde. Benutzern bzw. Rollen werden Berechtigungen für den Zugriff auf Bereiche der Anwendung, den Strukturbaum und bestimmten Dokumenten vergeben. Der Zugriff auf Bereiche und Dokumente ohne ausreichende Berechtigung wird technisch unterbunden (siehe Kapitel Error: Reference source not found).

3.3.3 Kommunikation zwischen Client und Server

Die Kommunikation zwischen Client und Server erfolgt über ein Netzwerk. Der Einsatz von SSL/TSL mit Zertifikaten (siehe Kapitel 3.3.1) sorgt für eine Verschlüsselung der übertragenen Daten und gewährleistet deren Unveränderlichkeit und die Identität des Servers.

3.3.4 Sicherheitsmaßnahmen am Client

Auch über eine gesicherte Verbindung können nicht valide oder sogar schädliche Daten in Richtung Serversystem gesendet werden.

Die Weiterverarbeitung von nicht validen Daten, kann zu korrupten Datensätzen in der Datenbank führen und schlimmstenfalls andere Datensätze in Mitleidenschaft ziehen.

In Benutzereingaben versteckter schadhafter Code (z. B. mittels Cross Side Scripting oder SQL-Injection) kann bei dessen Ausführung direkten Schaden anrichten oder dem Angreifer sensible Daten und Informationen offenbaren. Mittels SQL-Injection kann ein evtl. Angreifer direkt Befehle auf die Datenbank absetzen und so Daten manipulieren oder unberechtigt auslesen. Bei Cross Side Scripting (XSS) schleust der Angreifer schadhaften Code in die Webseite ein, der z. B. dazu dienen kann sensible Benutzereingaben wie Passwörter abzufangen und umzuleiten.

Beiden Szenarien wird durch die Verwendung von Validierungsmechanismen auf technischer wie fachlicher Ebene, sowie durch den Einsatz von robusten und erprobten Frameworks, die eine solche Gefahr bereits berücksichtigen, entgegen gewirkt. Insbesondere kommen folgende Verfahren zum Einsatz:

SQL Injection: Der Client fängt Anfrage-Parameter, die mit SQL-Steuerzeichen versehen sind durch Encoding ab.

Cross Site Scripting (XSS): Der Client fängt eingegebene HTML- bzw. Code-Steuerzeichen durch Encoding ab. Somit wird unter anderem auch ein Session Hijacking (in Verbindung mit SSL bzw. HTTPS) verhindert.

Überprüfung (Validierung): Der Client überprüft sämtliche Eingangs- und Ausgangsdaten, z. B. indem er ungefilterte Fehlerausgaben vom Backend unterbindet.

Länge bzw. Größe der Eingabedaten begrenzen. Alle Eingabedaten sind auf ihre Größe zu überprüfen, bevor sie verwendet oder kopiert werden, z. B. durch maxlength- Attribute an Eingabefelder.

3.3.5 Sicherheitsmaßnahmen am Backend

Das Backend ist über eine REST-Schnittstelle zugänglich. An dieser Stelle erfolgt eine zentrale Validierung sämtlicher übergebener Daten auf Korrektheit. Zusätzlich wurden bei der Auswahl der verwendeten Frameworks auch Sicherheitsaspekte berücksichtigt. Intranet Präventionswerkzeuge verwendet z. B. das Persistenzframework „Hibernate“, das als sicher gegenüber Angriffen mittels „SQL Injection“ gilt.

Dokumente (Dateien) werden in einem Dateisystem abgelegt, auf das der Application Server lesenden und schreibenden Zugriff hat. Die Nutzer des Systems haben auf dieses Dateisystem keinen direkten Zugriff. Die können ausschließlich über die Anwendung Intranet Präventionswerkzeuge auf diese Dokumente zugreifen. Die Rechtevergabe innerhalb der Anwendung regelt, welche Anwender welche Dokumente sehen können. Dokumente werden über den Webserver an den Client ausgeliefert und können dort angezeigt werden.

3.4 Abgrenzung

Die Anwendung wird als Intranet-Applikation von einem eingeschränkten Personenkreis verwendet bzw. ist nur für diesen erreichbar. Die beschriebenen Security-Standards und Best Practices sind für diese Risikostufe angemessen und ausreichend. Gezielte und mit krimineller Energie durchgeführte „Angriffe von innen“, z. B. durch die eigene Belegschaft werden hierbei nicht explizit berücksichtigt.

4 Zuverlässigkeit

Die Applikation wird im Dauerbetrieb als Mehrbenutzeranwendung eingesetzt. Daraus ergibt sich ein hoher Anspruch an die Zuverlässigkeit und Verfügbarkeit der Gesamtanwendung. Durch den Einsatz von im Enterprise-Umfeld weit verbreiteten, erprobten und robusten Technologien wird die Anwendung diesem Anspruch gerecht.

- Relationale Datenbanksysteme wie PostGreSQL sind seit Jahrzehnten erfolgreich als Quasi-Standard für die Speicherung von strukturierten Daten im Einsatz.
- JPA/Hibernate als Werkzeug für die objektrelationale Abbildung dient als standardisiertes und zuverlässiges Bindeglied zwischen der in Java implementierten Business-Logik und dem Datenbanksystem. Es abstrahiert die Komplexität der Abbildungsmechanismen und macht den fehlerträchtigen manuellen Zusammenbau von SQL-Statements nahezu überflüssig. Die Bündelung von Datenmanipulationen innerhalb von Transaktionen schützt die Integrität der Datenbankeinträge.
- Java EE-zertifizierte Enterprise Application Server unterstützen die Erstellung von robusten, strukturierten, skalierbaren und zuverlässigen Anwendungen und unterstützen die Anwendung durch ein eingebautes Transaktionshandling. Dadurch bleibt die Datenbank auch bei evtl. auftretenden Fehlern konsistent.
- Der Einsatz von REST-Schnittstellen hilft dabei die Kommunikation zwischen Client und Server effizient zu gestalten, d.h. es werden nur so viele Anfragen gestellt wie zur Arbeit am aktuellen Anwendungsteil notwendig sind. Dies verhindert unnötige Belastungen des Application Servers. Der zustandslose Ansatz dieser Architektur bindet weniger Ressourcen auf der Serverseite und erhöht dessen Verfügbarkeit.
- Zum Datenaustausch zwischen Client und Server wird das JSON-Format verwendet. Dieses ist nicht nur schlanker als XML, sondern per JavaScript auf der Clientseite auch ohne spezielle Transformationen verwendbar.

4.1 Richtwerte

Die Anwendung wird als zuverlässig angesehen, wenn sie dem jeweiligen Anwender in 95% der Arbeitszeit zur Verfügung steht.

4.2 Abgrenzung

Aussagen über die Zuverlässigkeit der Anwendung beziehen sich auf die Bestandteile der Anwendung sowie deren Zusammenspiel und nicht auf Bestandteile der Umgebung, in welche diese eingebettet ist. Ausfälle oder Einschränkungen in der Funktionalität, welche z. B. auf

- Infrastruktur-bedingte Ausfälle von Systemkomponenten,
- zu geringe Bandbreite im Netzwerk, zu geringe Rechenleitung oder unzureichender Speicherplatz (Haupt- und Massenspeicher) des Serversystems,
- zu geringe Rechenleistung oder unzureichender Speicherplatz (Haupt- und Massenspeicher) der Clientsysteme,
- Verwendung nicht unterstützter Browser oder auf
- Funktionseinschränkungen durch falsch konfigurierte Sicherheitseinstellungen oder Firewalls

zurück zu führen sind, sind von den in „4 Zuverlässigkeit“ beschriebenen Eigenschaften nicht abgedeckt.

Ebenso zählen Wartungsarbeiten am Netzwerk, an den Servern und den Arbeitsplatzrechnern nicht zu einer Ausfallzeit der Anwendung. Vorher angekündigte Zeiträume, in denen Updates der Anwendung aufgespielt oder Änderungen an der Datenbank vorgenommen werden, sind ebenso hiervon ausgenommen.

5 Benutzbarkeit

Für die Anwendung **Intranet Präventionswerkzeuge** sind folgende Ziele in Richtung Benutzbarkeit gefordert:

- Sie muss intuitiv nutzbar sein und den Benutzer bei der Interaktion unterstützen
- Sie soll für den Benutzer attraktiv sein

Um diese Ziele zu erfüllen werden im Rahmen der Projektabwicklung Usability-Optimierungen durch Einbeziehung der Fachanwender des Pilotkunden durchgeführt.

Dabei orientieren wir uns an den in der ***EN ISO Norm 9241, Teil 110 und 210*** verankerten Vorgaben. Diese beschreiben unter anderem die folgenden Aspekte:

- Aufgabenangemessenheit – geeignete Funktionalität, Minimierung unnötiger Interaktionen
- Selbstbeschreibungsfähigkeit – Verständlichkeit durch Hilfen / Rückmeldungen
- Lernförderlichkeit – Anleitung des Benutzers, Verwendung geeigneter Metaphern, Ziel: minimale Erlern-Zeit
- Steuerbarkeit – Steuerung des Dialogs durch den Benutzer
- Erwartungskonformität – Konsistenz, Anpassung an das Benutzermodell
- Individualisierbarkeit – Anpassbarkeit an Bedürfnisse und Kenntnisse des Benutzers
- Fehlertoleranz – Das System reagiert tolerant auf Fehler oder ermöglicht eine leichte Fehlerkorrektur durch den Benutzer

Der Nutzungskontext des Anwenders wird dabei ganzheitlich betrachtet. Das bedeutet, dass sich die Arbeitsumgebung und die Geschäftsprozesse in die Anwendung eingliedern müssen.

So lässt sich eine Anwendung schaffen, die den Anwender optimal unterstützt und so besser dazu beiträgt, einen Mehrwert für den Kunden zu schaffen.

6 Effizienz

6.1 Zeitverhalten

Die Anwendung soll dem Anwender einen reibungslosen Arbeitsfluss ermöglichen. Dazu gehört, dass diese auf Benutzeraktionen in einem angemessenen Zeitrahmen reagiert und neue oder geänderte Benutzeroberflächen schnell und effizient aufbaut und anzeigt.

6.2 Richtwerte

Ob eine Anwendung als flüssig oder langsam empfunden wird, hängt vom Unter- bzw. Überschreiten von bestimmten Schwellwerten ab. Für die Applikation werden folgende Richtwerte angesetzt:

Klassifizierung:	Ablauf:	Richtwert:
Öffnen eines neuen Dialogs	Nach einer Benutzeraktion wird ein neuer Dialog geladen und angezeigt	< 3 Sekunden in 90% der Fälle
Speichern von Eingaben	Nach der Eingabe von Daten betätigt der Benutzer die Schaltfläche *Speichern.* Er erhält daraufhin eine Fehlermeldung oder eine Meldung oder wird auf eine Folgemaske weitergeleitet.	< 4 Sekunden in 90% der Fälle
Laden/Hochladen eines Dokuments < 1024kb	Der Benutzer fordert ein Dokument zur Übertragung an den Browser an (Download), dessen Größe 1024kb nicht übersteigt.	< 3 Sekunden in 90% der Fälle

6.3 Abgrenzung

Einen bedeutenden Einfluss auf die benötigte Zeitspanne hat die Netzwerkverbindung. Beim Laden von größeren Datenmengen, z. B. bei Dokumenten stellt diese sogar den größten Einflussfaktor dar. Den oben genannten Zahlen liegt die Annahme zugrunde, dass ein Netzwerk mit mind. 10Mbit-Nenn-Rate vorhanden ist.

Überbelastung, Routing über Teilstrecken mit niedrigerer Nenn-Rate oder eine hohe Grundlast und Peaks sind durch die Anwendung nicht beeinflussbar.

7 Wartbarkeit

Man unterscheidet zwischen korrektiver, perfektionierender und adaptiver Wartung:

- korrektive Wartung: die Beseitigung von Fehlern
- perfektionierende Wartung: Verbesserung von Eigenschaften wie etwa der Leistungsfähigkeit oder der Wartbarkeit. Darunter fällt insbesondere die Anpassung des Entwurfs oder der Implementierung durch Reengineering (Software), Refactoring usw.
- adaptive Wartung: Anpassung der Software an veränderte technische Bedingungen der Umgebung (vgl. IEEE 610.12-1990 und ISO/IEC 12207)

Die Eigenschaft „Wartbarkeit“ lässt sich weiter herunterbrechen auf Basis-Eigenschaften. Im Folgenden eine Zusammenstellung der Basis-Eigenschaften inklusive einer kurzen Erläuterung.

- Analysierbarkeit (korrektiv)

  Aufwand, um Mängel oder Ursachen von Versagen zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen.

- Stabilität (korrektiv, perfektionierend, adaptiv)

  Wahrscheinlichkeit des Auftretens unerwarteter Wirkungen von Änderungen.

- Prüfbarkeit (korrektiv, perfektionierend, adaptiv)

  Aufwand, der zur Prüfung der geänderten Software notwendig ist.

- Erweiterbarkeit (perfektionierend, adaptiv)

  Kann das System leicht und ohne große Seiteneffekte erweitert werden

- Deploybarkeit (korrektiv, perfektionierend, adaptiv)

  Wie leicht lässt sich die Software in eine andere Umgebung übertragen? – Eignung der Software, von der Umgebung in eine andere übertragen werden zu können. Umgebung kann organisatorische Umgebung, Hardware- oder Software-Umgebung sein.

Diese Basis-Eigenschaften werden bei Intranet Präventionswerkzeuge durch folgende Maßnahmen sichergestellt:

- Analysierbarkeit
  - *Fehlebehandlungs- und Logging-Konzept, um Fehler nachvollziehen zu können*
  - *Produktionsnahe Testumgebung, in der Fehler nachgestellt werden können*
- Stabilität
  - *Die fachlichen Module werden unabhängig voneinander entwickelt. Es bestehen lediglich Abhängigkeiten zu den Basisfunktionen wie Benutzerverwaltung und Drucken.*
  - *Anwendung der Design-Prinzipien der Modularisierung und losen Koppelung*
  - *Fehlertolerante Anbindung der Backend-Systeme*
  - *Adäquate Behandlung von Fehlern, um unkontrollierte Abbrüche zu vermeiden*
- Prüfbarkeit
  - *Zu jeder Komponente werden von den Entwicklern automatisierte Tests auf Modulebene geschrieben. Diese laufen während des täglichen Builds automatisiert ab. Das Vorgehen bei der Erstellung der Tests wird im Entwicklerhandbuch beschrieben.*
- Erweiterbarkeit
  - *Anwendung der Best Practices bei der Implementierung:*

        - *Trennung von Implementierung und Schnittstelle*
        - *Separation of Concerns*
        - *Lose Kopplung*
        - *KISS-Prinzip (**K**eep **I**t **S**hort and **S**imple)*
        - *Prinzipien des objektorientierten Designs*
        - *Serviceorientierte Architektur mit REST*
    - *Überwachung der Einhaltung der Best Practices durch Code Reviews*
- Verteilbarkeit
    - *Die Anwendung wird so konzipiert, dass die gleiche Binärversion in alle Umgebungen verteilt werden kann. Die Anpassung an die jeweilige Zielumgebung erfolgt durch Konfigurationsdateien mit möglichst wenigen Einstellungen*
    - *Es wird das Standardformat für Java EE-Anwendungen (EAR) verwendet*

8 Testbarkeit

Das Testen der Anwendung Intranet Präventionswerkzeuge erfolgt grundsätzlich auf verschiedenen Stufen:

- Komponententest
- Integrationstest
- Abnahmetest

Der Komponententest wird von den Entwicklern implementiert und automatisch bei jedem Build-Prozess durchlaufen. Er dient als Basis bei der Implementierung neuer Funktionen, aber auch der Optimierung von vorhandenen Funktionalitäten. So werden evtl. Fehler innerhalb der Komponenten schnell gefunden bzw. von vorneherein unterbunden.

Der Integrationstest wird auf der Testumgebung durchgeführt und testet die Zusammenarbeit voneinander unabhängiger Komponenten. Er wird im Regelfall manuell durchgeführt und ist im Testkonzept mit entsprechenden Testfällen beschrieben.

Schließlich wird der Abnahmetest von der BG ETEM durchgeführt.

9 Offene Punkte

Nr.	Beschreibung	Verantwortlich