Das BVL-Standard-Systemumfeld

Das standardisierte Systemumfeld für die Softwareentwicklung im BVL

Stand: 01.10.2020

Inhaltsverzeichnis

1 Ziel

Die Referatsgruppe Z3 betreibt im BVL sowohl Standard- als auch Individualsoftware. Um einen wartungsarmen und fehlerfreien Betrieb zu gewährleisten, muss das Zusammenspiel einer Vielzahl an Softwarekomponenten und -versionen sichergestellt werden. Hierunter fallen insbesondere Betriebssysteme, sogenannte Middleware wie Web-Applikationsserver oder die MACH-Software, Datenbankmanagementsysteme aber auch von Endanwendern verwendete Software wie Webbrowser oder Java-Laufzeitumgebungen.

Ein weiterer Einflussfaktor, welcher den Betrieb beeinflusst, ist die Beendigung des Softwarelebenszyklus einzelner Softwareprodukte. Dies kann beispielsweise durch die Terminierung des Supports seitens des Herstellers oder eine Insolvenz geschehen. Des Weiteren können neue Versionen wiederum die Aktualisierung anderer Teile der IT-Infrastruktur nach sich ziehen.

Aufgrund der begrenzten Ressourcen in Z3 und der hohen Komplexität der Aufgabenstellung ist es nicht möglich, sämtliche möglichen Kombinationen dieser Komponenten zu betreiben. Stattdessen wird in diesem Dokument ein Standardsystemumfeld definiert, welches für den Betrieb von Software seitens Z3 unterstützt wird. Dieses Systemumfeld ist für die Entwicklung und Freigabe von Individualsoftware bindend.

Der Support für eine Softwarekomponente (außer Betriebssystemen mit gesichertem Support seitens des Herstellers) kann maximal für ein Jahr ab der initialen Aufnahme definiert werden. Im Anschluss kann diese Komponente jährlich verlängert werden, insofern der Support seitens des Herstellers nicht in diesem Zeitraum aufgekündigt wird.

2 Geltungsbereich

BVL gesamt	☐
Organisationseinheit	☒ Z3

Die einheitlichen Vorgaben dieses Dokumentes gelten für die hausinterne Entwicklung und sollen auch bei einer externen Auftragsvergabe verwendet werden.

3 Für neue Anwendungen zu verwendende Versionen

Für Neuentwicklungen innerhalb des Supportzeitraums sind die in Tabelle 1 angegebenen Versionen der jeweiligen Software zu verwenden.

Tabelle 1 Die unterstrichene Software wird über die Paketverwaltung des Betriebssystems aktuell gehalten. Entsprechend gilt für den Support-Zeitraum des Herstellers jener des Betriebssystems.[1]

System	Software	Version	Ende Support (BVL)	Ende Support (Hersteller)
Betriebssysteme				
Server	CentOS	7	→	2024-06-30
Server	CentOS	8	→	2029-05
Server	RHEL	7	→	2024-06-30*
Server	RHEL	8	→	2029-05*
Client	Windows (64 Bit)	10 (1809)	→	2021-05-11*
Server	Windows Server	2012 R2	→	2023-10-10*
Server	Windows Server	2016 LTSC	→	2027-11-01*
Office-Anwendungen				
Client	Microsoft Office	2013**	→	2023-04-11*
Client	Microsoft Office	2016	→	2025-10-14
Client	Microsoft Office	2019**	→	2025-10-14
Client	The Document Foundation LibreOffice	6.x	n. a.	n. a.
Server	The Document Foundation LibreOffice	5	→	2024-06-30*
Server	The Document Foundation LibreOffice	6	→	2029-05*
Groupware und Mailing				
Server	Microsoft Exchange	2016	n. a.	n. a.
Server	Postfix	2.x	→	2024-06-30

[1] Siehe hierzu http://mirror.centos.org/centos/7/os/x86_64/Packages/

System	Software	Version	Ende Support (BVL)	Ende Support (Hersteller)
Browser				
Mobile	Google et. al. Webkit (iOS, Android)	2+	n. a.	n. a.
Client	Google Chrome	80+	n. a.	n. a.
Client	Microsoft Edge[2,**]	44+	n. a.	n. a.
Client	Microsoft Internet Explorer[3]	11.x	n. a.	n. a.
Client	Mozilla Firefox	68+ (ESR)	n. a.	n. a.
Applikationsserver				
Server	Apache HTTP-Server	2.4	→	2024-06-30
Server	Apache Tomcat	7.0	→	2024-06-30
Server	JBoss Web Server	5.0	→	2024-06-30
Server	JBoss Web Server	5.1	→	n.a.
Server	JBoss EAP	6.4	→	2025-06*
Server	JBoss EAP	7.x	→	2029-05*
Laufzeitumgebungen und Entwicklungsplattformen				
Client	Apache JMeter	5.x	n. a.	n. a.
Client	Apache Maven[4]	3.x	n. a.	n. a.
Client	AT&T Labs Research GraphViz[5]	2.x	n. a.	n. a.
Client	Eclipse IDE[6]	x	n. a.	n. a.
Client	JBoss Arquillian[7]	1.6+	n. a.	n. a.

[2] Im März 2017 teilte der Hersteller Microsoft mit, dass die Weiterentwicklung des Internet Explorer eingestellt wird. Für künftige Fachanwendung wird als Ziel-Browser dessen designierter Nachfolger verlangt.

[3] Bis auf weiteres wird Microsoft Internet Explorer anstelle von Microsoft Edge im BVL eingesetzt. Edge ist für die Verwendung in der Bundesverwaltung bisher nicht freigegeben.

[4] Der Build einer Lieferung erfolgt auf einem Arbeitsplatzrechner. Hierzu wird Eclipse und deren mitgelieferte Maven-Version verwendet. Die Organisation der Quellcodes muss dem Maven Standard Directory Layout entsprechen, https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html

[5] Für die Visualisierung von Klassenhierarchien und Aufrufpfaden, siehe auch http://www.graphviz.org/

[6] Aufgrund des quartalsweisen Release-Zyklus wird auf eine Versionsangabe verzichtet, i.d.R. kommt die aktuellste Version im BVL zum Einsatz.

[7] Für Integrationstests. Maven-Artefakt: **org.jboss.arquillian.core:arquillian-core-parent**

System	Software	Version	Ende Support (BVL)	Ende Support (Hersteller)
Client	JBoss Arquillian Graphene[8]	2.3+	n. a.	n. a.
Client	JUnit[9]	5.6+	n. a.	n. a.
Client	Liquibase[10]	3.8+	n. a.	n. a.
Client	Maven-Plug-In Checkstyle[11]	3.x	n. a.	n. a.
Client	Maven-Plug-In Doxygen[12]	1.1.0	n. a.	n. a.
Client	Maven-Plug-In PMD[13]	3.13+	n. a.	n. a.
Client	Maven-Plug-In SpotBugs[14]	3.1+	n. a.	n. a.
Server	OpenJDK Java Development Kit[15]	8 LTS	→	2023-06
Server	OpenJDK Java Development Kit	11 LTS	→	2024-10
Client	OpenJDK Java Development Kit	8	→	2023-06
Client	OpenJDK Java Development Kit	11	→	2024-10
Client	OWASP Dependency Check[16]	5.3+	n. a.	n. a.
Client	PlantUML[17]	1.2020+	n. a.	n. a.
Client	SmartBear Software SoapUI[18]	5.5+	n. a.	n. a.
Datenbankmanagementsysteme				
Server	Oracle Database	19c SE1[19]	→	2024-04-30

[8] Für Frontend- und Akzeptanztests. Maven-Artefakt: **org.jboss.arquillian.graphene:graphene-webdriver**

[9] Für Modultests. Maven-Artefakt: **org.junit.jupiter:junit-jupiter-api**

[10] Als Datenbankschema-Manager. Maven-Artefakt: **org.liquibase:liquibase-maven-plugin**

[11] Siehe BVL_TA_04_9311_030_Checkstyle_Ruleset und BVL_TA_04_9311_030_Checkstyle_Suppressions. Maven-Artefakt: **org.apache.maven.plugins:maven-checkstyle-plugin**

[12] Für die Quellcodedokumentation. Maven-Artefakt: **com.soebes.maven.plugins:doxygen-maven-plugin**

[13] Siehe BVL_TA_04_9311_030_PMD_Ruleset. Maven-Artefakt: **org.apache.maven.plugins:maven-pmd-plugin**

[14] Siehe BVL_TA_04_9311_030_SpotBugs_Suppressions. Maven-Artefakt: **com.github.spotbugs:spotbugs-maven-plugin**

[15] OpenJDK Life Cycle and Support Policy: https://access.redhat.com/articles/1299013

[16] Für Schwachstellenanalyse. Maven-Artefakt: **org.owasp:dependency-check-maven**

[17] Für die Visualisierung der Datenbankschemata und Dekomposition des Systems, siehe http://de.plantuml.com/

[18] Für Schnittstellentests von SOAP-basierten Web-Services.

[19] Siehe https://support.oracle.com/knowledge/Oracle%20Database%20Products/742060_1.html

System	Software	Version	Ende Support (BVL)	Ende Support (Hersteller)
Server	Oracle Database	19c SE2[20]	→	2024-04-30
Server	Oracle Database	19c EE[21]	→	2024-04-30
Server	P. Global Dev. Group PostgreSQL	9.2	→	2024-06-30[22]
Server	P. Global Dev. Group PostgreSQL	10[23]	→	2029-05
Server	Refractions Research PostGIS	2.4	n. a.	n. a.[24]
Server	MariaDB Foundation MariaDB	5.5	→	2024-06-30[25]
Server	MariaDB Foundation MariaDB[26]	10.3	→	2029-05[27]
Sonstiges				
Server	Git	1.8+	→	2024-06-30
Server	SAS	9.4	n. a.	n. a.
* Erweiterter Support durch den Hersteller				
** Wird im BVL nicht eingesetzt, bei Bereitstellung extern erreichbarer Diensten notwendig.				

Zur Vereinfachung der Wahl der richtigen Software-Versionen und den damit verbundenen technischen Rahmenbedingungen werden nachfolgend sogenannte „Software-Bundles“ definiert, die bestimmte Software-Kombinationen festlegen. Die jeweils aufgeführten Software-Versionen resultieren direkt aus dem Angebot der Paketverwaltung. Im Rahmen der Abstimmung zwischen dem AN und dem AG muss im Rahmen der Konzeptworkshops ein verbindliches Software-Bundle ausgewählt werden.

[20] Die Oracle Standard Edition Two steht nur nach Rücksprache mit Referatsgruppe Z3 zur Verfügung.

[21] Die Oracle Enterprise Edition soll nur dort verwendet werden, wo es für die Anwendung zwingend erforderlich ist. Die meisten Anwendungen des BVL sollen auf der Standard Edition betrieben werden.

[22] Versionierungsrichtlinien: https://www.postgresql.org/support/versioning/. Falls durch Red Hat die Pakete **postgresql*** länger unterstützt werden, gilt die EOL des Betriebssystems.

[23] PostgreSQL 10 on CentOS 7: https://tecadmin.net/install-postgresql-server-centos/

[24] PostGIS Support Matrix: https://trac.osgeo.org/postgis/wiki/UsersWikiPostgreSQLPostGIS. Die Installation von PostGIS erfordert i.d.R. den Einsatz zusätzlicher Repositorien. Es gelten die dortigen EOL.

[25] MariaDB general release maintenance periods: https://mariadb.org/about/maintenance-policy/. Falls durch Red Hat die Pakete **mariadb*** länger unterstützt werden, gilt die EOL des Betriebssystems.

[26] Installing MariaDB 10 on CentOS 7: https://mariadb.com/de/node/463

[27] Siehe Fußnote 25. Falls durch Red Hat die Pakete **mariadb*** länger unterstützt werden, gilt die EOL des Betriebssystems.

3.1 Software-Bundles

Um Einsatz, Betrieb und Pflege von Standardsoftware zu vereinfachen, werden vom BVL nur bestimmte Versionskombinationen als sogenannte Software-Bundles unterstützt, die nachfolgend definiert werden.

Hinweis: Unter Berücksichtigung der Produktzyklen und durchschnittlichen Projektlaufzeiten erfolgt die Definition von Software-Bundles im Zweijahresrhythmus. Eine jährliche Definition neuer Bundles kann durch das BVL leider nicht angeboten werden.

	2016	2018	2020
Betriebssysteme	RHEL 7 CentOS 7	RHEL 7 CentOS 7	RHEL 8 CentOS 8
Laufzeitumgebungen	OpenJDK 7 Java EE 6	OpenJDK 8 Java EE 7	OpenJDK 11 Jakarta EE 9
Applikationsserver	JBoss EAP 6.4 Apache Tomcat 7	JBoss EAP 7.x JBoss Web Server 5.0 Apache Tomcat 7	JBoss EAP 7.x JBoss Web Server 5.1*
Datenbankmanagement	Oracle DB 12 MariaDB 5.5	Oracle DB 12 MariaDB 5.5 PostgreSQL 9.2	Oracle DB 19c MariaDB 10 PostgreSQL 10
* Ab RHEL 8 wird der Tomcat nur noch in Verbindung mit dem JBoss Web Server angeboten.			

4 Konfiguration und Verwendungshinweise

Für die Konfiguration der nachfolgenden Maven-Plug-Ins stellt der AN entsprechende Konfigurationsdateien bereit, die in Kombination mit dem Build-Werkzeug verwendet werden müssen. Im Einzelnen sind dies:

- **Checkstyle_Ruleset.xml**

  Legt die Checkstyle-Module fest, die für die Einhaltung der Quellcoderichtlinien verwendet werden sollen und konfiguriert diese.

- **Checkstyle_Suppressions.xml**

  Legt den Ausschluss einiger Regeln für bestimmte Quellcode-Dateien fest. So gelten die Quellcoderichtlinien nicht für automatisch generierten Quellcode oder Ressourcen wie CSV-, JSON-, DTD-Dateien etc.

- **SpotBugs_Suppressions.xml**

  Unterdrückt die statische Codeanalyse durch SpotBugs bei automatisch generierten Quellcodedateien.

- **PMD_Ruleset.xml**

  Legt die geltenden PMD-Kategorien fest und konfiguriert diese. Dieser Regelsatz gilt für neuere PMD-Versionen.[28]

- **PMD_Ruleset_Legacy.xml**

  Legt die geltenden PMD-Kategorien fest und konfiguriert diese. Dieser Regelsatz gilt für ältere PMD-Versionen.

4.1 Einhaltung der Quellcoderichtlinien

Der Quellcode wird nach den „Code Conventions for the Java TM Programming Language Revised April 20, 1999“ erstellt. Für die Einhaltung der Konventionen muss ein Maven-Plug-In für Checkstyle verwendet werden. Eine entsprechende Konfiguration wird durch den AG bereitgestellt. Folgendes Maven-Plug-In wird für die Einbindung empfohlen:

- **org.apache.maven.plugins:maven-checkstyle-plugin**

[28] Siehe hierzu https://pmd.github.io/pmd-6.0.1/pmd_release_notes.html

4.2 Einhaltung der Richtlinien zur Authentifizierung und Autorisierung

Die Authentifizierung und Autorisierung muss über das zentrale Identitätsmanagement des AG erfolgen. Hierzu wird vom AG die Standardsoftware Red Hat SSO eingesetzt, die sowohl die Authentifizierung als auch Autorisierung von Benutzern implementiert.[29] Die Konfiguration von Benutzerrollen erfolgt durch den AG auf Grundlage der abgestimmten und verbindlichen Regelungen des Pflichtenheftes bzw. des Installations- und Betriebshandbuches. Das Rechte- und Rollenkonzept der Fachanwendung wird Im Rahmen der Konzeptworkshops zwischen dem AN zwischen dem AG festgelegt.

4.3 Einhaltung der Protokollierungsrichtlinien

Bei der Protokollierung wird grundsätzlich zwischen der fachlichen und technischen Protokollierung unterschieden. Während die fachliche Protokollierung im Kontext der fachlichen Anforderungen im Rahmen von Konzeptworkshops zwischen dem AN und dem AG definiert wird, gelten für die technische Protokollierung allgemeine Vorgaben, die nachfolgend beschrieben werden.

Die eingesetzten Applikationsserver des AG werden so konfiguriert, dass Deployment-spezifische Konfigurationen der Protokollierung ignoriert werden (per-deployment logging).[30] Es ist daher unzulässig, Konfigurationsdateien wie die folgenden zu verwenden:

- **logging.properties**
- **jboss-logging.properties**
- **log4j.properties**
- **log4j.xml**
- **jboss-log4j.xml**

Für die Ausgabe von Protokolldateien muss das Framework „JBoss Logging“ verwendet, welches über das offizielle Maven Repository bezogen werden kann:

- **org.jboss.logging:jboss-logging**

[29] Siehe https://github.com/redhat-developer/redhat-sso-quickstarts/tree/7.2.x/app-jee-html5

[30] Siehe https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.2/html/configuration_guide/logging_with_jboss_eap#about_per_deployment_logging

Aus Gründen der Effizienz sind Referenzen auf einen Logger stets als statische Ressource anzufordern:[31]

```
import org.jboss.logging.Logger;
private static final Logger LOGGER = Logger.getLogger(HelloWorld.class);
LOGGER.errorf("Configuration file <%s> not found.", CONFIG_FILE);
```

Für die unterschiedlichen Stufen bei der Protokollierung gelten die nachfolgenden Festlegungen.[32] Hierbei ist insbesondere zu beachten, dass pbD gesetzlichen Löschfristen gemäß der EU-DSGVO unterliegen und eine besondere Sorgfalt bei deren Verarbeitung erfordern. Grundsätzlich ist bei der technischen Protokollierung im Wirkbetrieb (ab Stufe **INFO**) von der Ausgabe pbD in Protokollnachrichten abzusehen. Die Ausgabe von Fehlerauszügen (stacktraces) ist nur zum Zwecke der detaillierten Fehleranalyse zulässig, also maximal bis zur Stufe **TRACE**.

Stufe	pbD	Stacktrace	Beschreibung
FATAL	☐	☐	Kritisches Ereignis, welches den Betrieb des Systems unmöglich macht (betriebsverhindernd).
ERROR	☐	☐	Problematisches Ereignis, das bei der Verarbeitung einer konkreten Aktion oder in einem Systemteil zu Störungen führt, das Gesamtsystem jedoch nicht beeinträchtigt (betriebsbehindernd).
WARN	☐	☐	Problematisches Ereignis, das bei der Verarbeitung einer konkreten Aktion von den vorgegebenen Parametern abweicht. Eine kontextbezogene Abgrenzung zur Stufe **ERROR** liegt im Ermessen des Entwicklers.
INFO	☐	☐	Ereignisse, die über reguläre Aktionen und den Lebenszyklus von Diensten informieren. Diese Stufe wird primär dazu verwendet, den geregelten Start, den aktuellen Zustand und den geregelten Stopp eines Systems zu beurteilen.
DEBUG	■	☐	Ereignisse, die zusätzliche Informationen über reguläre Aktionen und den Lebenszyklus von Diensten liefern. Diese Stufe wird primär dazu verwendet, während des Testbetriebes die Fehlerdiagnose zu vereinfachen.

TRACE	■	■	Ereignisse, die der Feinanalyse von regulären Aktionen und dem Lebenszyklus von Diensten dienen. Aufgrund der zu erwartenden Anzahl von Protokollnachrichten muss vor der Ausgabe der Protokollnachricht geprüft werden, ob diese Stufe konfiguriert wurde:
			`if (LOGGER.isTraceEnabled()) {`
			`  LOGGER.tracef("Loading file <%s>.", CONFIG_FILE);`

[31] Siehe https://access.redhat.com/documentation/en-us/jboss_enterprise_application_platform/6/html/development_guide/add_logging_to_an_application_with_jboss_logging

[32] Siehe https://docs.jboss.org/process-guide/en/html/logging.html

Stufe	pbD	Stacktrace	Beschreibung
			}

4.4 Versionierung des Datenbankschemas

Die Definition des Datenbankschemas muss unabhängig von einem herstellerspezifischen SQL-Dialekt erfolgen und im Sinne der Produktpflege versioniert werden. Für die Versionierung des Datenbankschemas muss der Schema-Manager Liquibase verwendet werden.[33] Folgendes Maven-Plug-In wird für die Einbindung empfohlen:

- **org.liquibase:liquibase-maven-plugin**

4.5 Erstellung der Quellcodedokumentation

Zur Erzeugung der Dokumentation muss ein Maven-Plug-In für Doxygen verwendet werden. Empfohlen wird zusätzlich der Einsatz von GraphViz, so dass Doxygen automatisch Visualisierungen der Klassenhierarchie und Aufrufpfade erzeugt. Folgendes Maven-Plug-In wird für die Einbindung empfohlen:

- **com.soebes.maven.plugins:doxygen-maven-plugin**

4.6 Einbindung der statischen Quellcodeanalyse

Für die statische Quellcodeanalyse müssen die Werkzeuge PMD und SpotBugs verwendet werden. Entsprechende Konfigurationen werden durch den AG bereitgestellt. Folgende Maven-Plug-Ins werden für die Einbindung empfohlen:

- **org.apache.maven.plugins:maven-pmd-plugin**
- **com.github.spotbugs:spotbugs-maven-plugin**

4.7 Einbindung der Schwachstellenprüfung

Für die statische Schwachstellenprüfung der Abhängigkeiten der Anwendung muss das dafür entwickelte OWASP-Werkzeug verwendet werden. Folgende Abhängigkeit muss hierzu eingebunden werden:

- **org.owasp:dependency-check-maven**

[33] Siehe hierzu http://www.liquibase.org/quickstart.html

4.8 Kodierung und Zeichensätze

Um den Einsatz der XÖV-Standards[34] inklusive den SAGA-Vorgaben zu genügen, muss der zur Verfügung stehende Zeichenvorrat eingeschränkt werden. Hierzu hat die KoSIT einen Zeichenvorrat auf Grundlage von UTF-8 definiert[35], der die lateinischen Buchstaben und wenige Sonderzeichen vorsieht. Die Definition dieser Untermenge[36] wird in Form eines regulären Ausdrucks definiert[37], der bei jeder Zeichenoperation als syntaktische Validierung zum Einsatz kommen muss:

**(([	-

 -~¡-¬®-ćĊ-ěĞ-ģĦ-ıĴ-śŞ-ūŮ-žƏƠ-ơƯ-ưƷǍ-ǔǞ-ǟǤ-ǰǴ-ǵǺ-ǿȘ-țȞ-ȟȪ-ȫȮ-ȳəʒḂ-ḃḊ-ḋḐ-ḑḞ-ḡḤ-ḧḰ-ḱṀ-ṁṄ-ṅṖ-ṗṠ-ṣṪ-ṫẀ-ẅẌ-ẓẞẠ-ầẪ-ẬẮ-ềỄ-ồỖ-ờỤ-ỹ€])|(M̂|N̂|m̂|n̂|D̂|d̂|J̌|L̂|l̂))***

Bei der Verarbeitung von Zeichenketten muss die Prüfung auf unzulässige Zeichen auf der Applikationsebene via Bean Validation erfolgen, siehe Anhang: String-Validierung via Bean Validation.

34 http://www.xoev.de/sixcms/detail.php?gsid=bremen02.c.738.de

35 http://www.xoev.de/sixcms/detail.php?gsid=bremen83.c.4813.de

36 http://xoev.de/latinchars/1_1/latinchars.pdf

37 http://xoev.de/latinchars/1_1/datatypes/latinchars.xsd

5 Mitgeltende Unterlagen

Dokument
Checkstyle_Ruleset.xml
Checkstyle_Suppressions.xml
SpotBugs_Suppressions.xml
PMD_Ruleset.xml
PMD_Ruleset_Legacy.xml

6 Änderungen an diesem Dokument

- Das Dokument wurde in der Version 4.0 den aktuellen Entwicklungen entsprechend angepasst:
    - Aktualisierung der Software-Bundles. Das BVL verfolgt eine direkte Migration von Oracle DB 12c auf Oracle DB 19c.
    - Hinzufügen der Implementierungshinweise zur Authentifizierung und Autorisierung.
    - Hinzufügen der Implementierungshinweise zur technischen Protokollierung.
    - Hinzufügen der Implementierungshinweise zur Versionierung des Datenbankschemas.
    - Aktualisierung der Versionstabelle und Supportzeiträume.
    - Definition von Software-Bundles zur Reduzierung der kombinatorischen Komplexität bei den Versionen der eingesetzten Standardsoftware.
- Das Dokument wurde in der Version 3.0 den aktuellen Entwicklungen entsprechend angepasst:
    - Auswertung der Support-Matrizen der Hersteller von Standardsoftware im BVL.

  Wesentliche inhaltliche Änderungen:
    - Hinzufügen des Kapitels „Mitgeltende Unterlagen“, welches die vordefinierten Konfigurationsdateien für die Werkzeuge in Kapitel 4.8 aufführt.
    - Ergänzung der Anforderungen der KoSIT.
    - Hinzufügen des Anhanges „String-Validierung via Bean Validation“.
    - Hinzufügen der Konfigurationshinweise zu den Quellcoderichtlinien.
    - Hinzufügen der Konfigurationshinweise zur Quellcodedokumentation.
    - Hinzufügen der Konfigurationshinweise zur statischen Quellcodeanalyse.
    - Hinzufügen der Konfigurationshinweise zur Schwachstellenprüfung.

Anhang: String-Validierung via Bean Validation

<KositStringValidator.java>

```java
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.validation.ConstraintValidator;
import javax.validation.ConstraintValidatorContext;
import de.bund.bvl.sample.utils.Configuration;

/**
 * Führt die Validierung einer Zeichenkette gegen den regulären Ausdruck der KoSIT durch.
 *
 * @author <a href="mailto:Z32@bvl.bund.de">BVL, Referat Z32</a>
 * @version 13.10.2016
 */
public class KositStringValidator implements ConstraintValidator<KositString, String> {
    /**
     * Die Kompilierung des KoSIT-Pattern erfolgt aus Performancegründen nur einmalig beim Deployment.
     */
    private static final Pattern PATTERN = Pattern.compile(Configuration.Validation.Pattern.KOSIT);

    @Override
    public void initialize(final KositString constraint) {
        // Eine Initialisierung des Validator ist nicht notwendig.
    }

    @Override
    public boolean isValid(final String value, final ConstraintValidatorContext context) {
        boolean isValid = false;

        if (value == null) {
            isValid = true;
        } else {
            final Matcher matcher = PATTERN.matcher(value);
            isValid = matcher.matches();
        }

        return isValid;
    }
}
```

<KositString.java>

```java
import static java.lang.annotation.RetentionPolicy.RUNTIME;

import java.lang.annotation.Documented;
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.Target;
import javax.validation.Constraint;
import javax.validation.Payload;

/**
 * Die annotierte {@code CharSequence} muss dem regulären Ausdruck der KoSIT genügen, der den Zeichenvorrat von UTF-8 auf
 * lateinische Zeichen einschränkt.
 * <p/>
 * {@code null} wird als gültig akzeptiert.
 *
 * @author <a href="mailto:Z32@bvl.bund.de">BVL, Referat Z32</a>
 * @version 13.10.2016
 */
@Target({ ElementType.METHOD, ElementType.FIELD, ElementType.ANNOTATION_TYPE, ElementType.CONSTRUCTOR, ElementType.PARAMETER })
@Retention(RUNTIME)
@Documented
@Constraint(validatedBy = { KositStringValidator.class })
public @interface KositString {
    /**
     * Die Fehlermeldung, die ausgegeben wird, wenn die Zeichenkette ungültige Zeichen enthält.
     */
    String message() default "Die Zeichenkette enthält ungültige Zeichen, siehe <http://xoev.de/latinchars/1_1/latinchars.pdf>.";

    /**
     * Die Gruppe, auf die sich die Validierungsregel bezieht.
     */
    Class<?>[] groups() default {};

    /**
     * Der mit der Validierungsregel assoziierte Payload.
     */
    Class<? extends Payload>[] payload() default {};
}
```

<Address.java>

```java
import java.io.Serializable;
import javax.persistence.Entity;
import javax.validation.constraints.NotNull;
import de.bund.bvl.sample.model.validation.KositString;

/**
 * Entitätenklasse für Adressdatensätze.
 *
 * @author <a href="mailto:Z32@bvl.bund.de">BVL, Referat Z32</a>
 * @version 22.09.2016
 */
@Entity
public class Address implements Serializable {
    /**
     * ID für die Passivierung von Entitäten.
     */
    private static final long serialVersionUID = 1L;

    /**
     * Pflichtfeld. Die Stadt der Postadresse des Kunden.
     */
    @NotNull
    @KositString
    private String city;
}
```