4. Juni 2014

Sicherheit in der SW-Entwicklung und beim Betrieb einer Identity und Access Management Lösung

Best Practice Standardportal 2.0

Inhalt

Foto: G. Pesendorfer

- Herausforderungen für LFRZ
- Betrieb sicherheitskritischer Infrastruktur
- Sichere SW-Entwicklung / Beispiel Standardportal
- Portalverbund
- Vorgaben für sichere SW-Entwicklung
- Zusammenfassung

Wer sind wir?

- Gerold Pesendorfer
  - Produkt- und Projektmanager / E-Government
  - Im LFRZ seit 1996
  - Leitung Standardportal

Foto: LFRZ

Foto: Daniela Korb, © Peter Pichler

- Peter Pichler
  - Software Architekt / Experte Portalverbund
  - Portalentwicklung seit 2003
  - Technische Leitung Standardportal

Herausforderungen für LFRZ

Foto: G. Pesendorfer

- LFRZ als Serviceprovider für seine Kunden (BMLFUW, AMA, BKA, BBG, Rechnungshof)

(1) Betrieb von sicherheitskritischen Komponenten und Anwendungen z.B. Portalsysteme, CMS und Websites

- LFRZ in der Rolle Software-Hersteller

(2) Entwicklung des Standardportals als zentraler SW-Komponente für den österreichischen Portalverbund

Betrieb sicherheitskritischer Infrastruktur (I)

Foto: Maik Schwertle / pixelio.de

- Anforderungen
  - Schutzziele der Informationssicherheit - Verfügbarkeit, Vertraulichkeit und Integrität
  - Compliance mit Gesetzen (GmbHG, DSG 2000, ...)
  - Vorgaben aus Portalverbund (PVV, Portalrevision)
  - Service Level Agreements
  - Bedrohungen durch Angriffe und Schadsoftware

Betrieb sicherheitskritischer Infrastruktur (II)

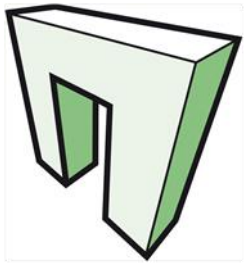
Foto: Joerg Trampert / pixelio.de

- Maßnahmen
  - Einführung eines ISMS
  - Regelmäßige Schwachstellenanalyse
  - Sicherheitsüberprüfungen eigener Systeme
  - Awareness durch Schulungen
- Kooperation mit Partnern und Gremien
  - GovCERT Austria
  - Arbeitsgruppen (AG-IZ)
  - Security Workshops für Kunden

ISO 27001 Zertifizierung

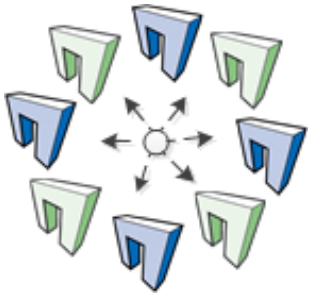
- LFRZ seit April 2014 zertifiziert für Betrieb & SW-Entwicklung
- Projekt wurde mit Security Research als Partner umgesetzt
- Projektleitung Georg Melzer (Security Manager)
- Ziel ist kontinuierliche Verbesserung (PDCA)

Sichere SW-Entwicklung / Beispiel Standardportal



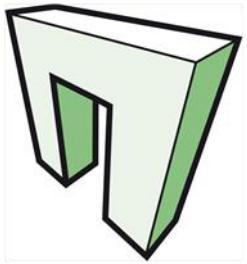
- Referenzimplementierung für österr. Portalverbund
- Eigentümer BM.I und LFRZ GmbH
- Lizenznehmer sind Organisationen und Unternehmen der öffentlichen Verwaltung
- Gemeinsame Weiterentwicklung über LA und User-Group
- Standardportal 2.0 - Implementierung von PVP2 (SAML)

Was ist der Portalverbund?



- Zentraler Baustein für IT-Sicherheit im österr. E-Government
- Identity u. Access Management für öffentliche Verwaltung
- Benutzerverwaltung erfolgt in verteilten Portalen – nicht in den Anwendungen
- Regelt den organisationsübergreifenden Zugang zu zentralen IT-Anwendungen (z.B. ZMR)

Projekt Standardportal 2.0



- Erweiterung der Software für Verwendung mit internationalem Standard (SAML 2.0)
- Start Anfang 2012 – Fertigstellung Mitte Juni 2014

✓ Standardportal als SAML Identity und Service Provider

✓ Konverterfunktion für unterschiedlicher Protokollversionen

✓ Überprüfung und Zertifizierung nach ÖNORM A 7700

Austrian Standards plus Certification

Zertifikat

Nr. W 000011

Die Austrian Standards plus GmbH stellt der

Land-, forst- und wasserwirtschaftliches Rechenzentrum GmbH

Dresdnerstraße 89, AT 1200 Wien

dieses Zertifikat aus.

Dieses Zertifikat bestätigt die Konformität folgender Webapplikation

Bezeichnung der geprüften Applikation: "Standardportal", Version 2.0, Build 155
Datum der Prüfung: 2014-05-23.
Der Umfang des Zertifikates exkludiert eingesetzte Frameworks und Bibliotheken.

mit den Bestimmungen der

ÖNORM A 7700:2008-12

Datum der Ausstellung: 2014-05-26

Dieses Zertifikat gilt ausschließlich für die obig bezeichnete Versionsnummer der Webapplikation.

Austrian Standards plus Certification ist eine Zertifizierungsstelle gemäß EN 45011.

Die Austrian Standards plus GmbH ist eine 100 %ige Tochtergesellschaft des Austrian Standards Institute.

EN ISO 9001 zertifiziert durch SQS.

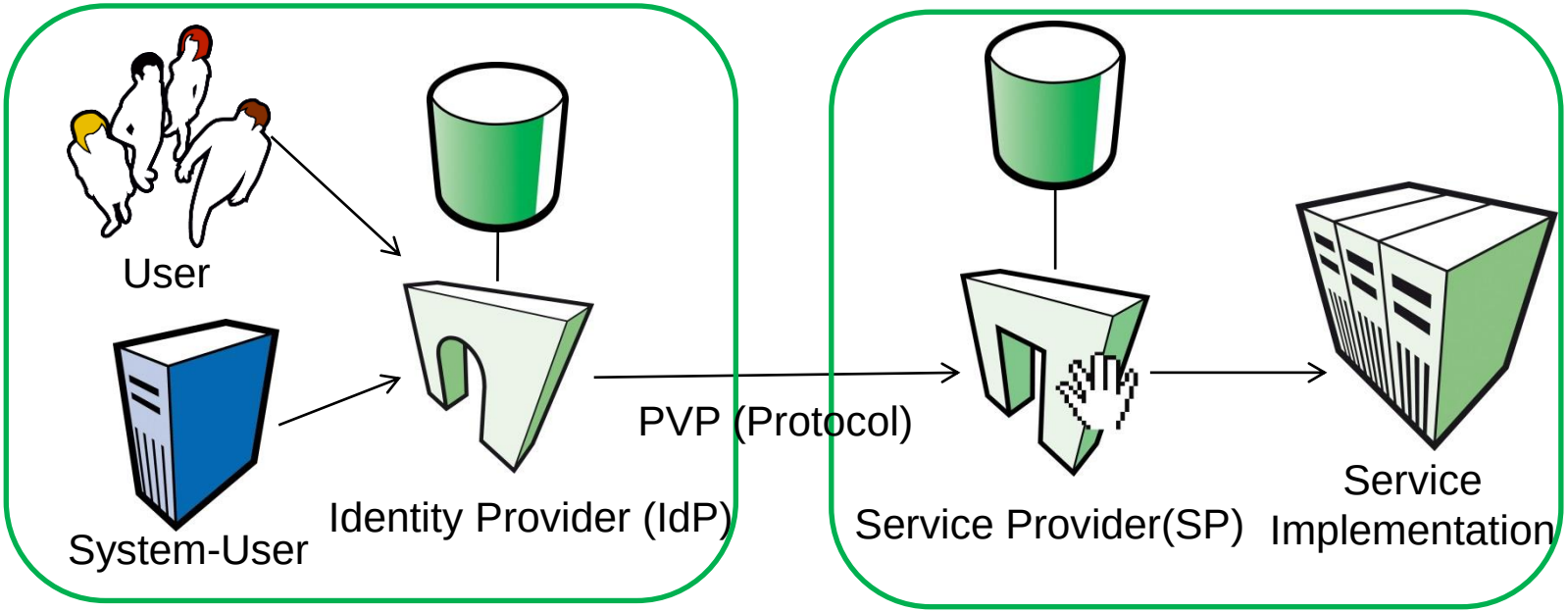
Dieses Zertifikat berechtigt zur Führung des Konformitätszeichens:

CERTIFIED NCERT WEBSITE ÖNORM A 7700

www.as-plus.at/certification

Dr. Peter Jonas
Director Certification

PVP Standardportal – Identity Provider (IdP, Stammportal) und Service Provider (SP, Anwendungsportal)

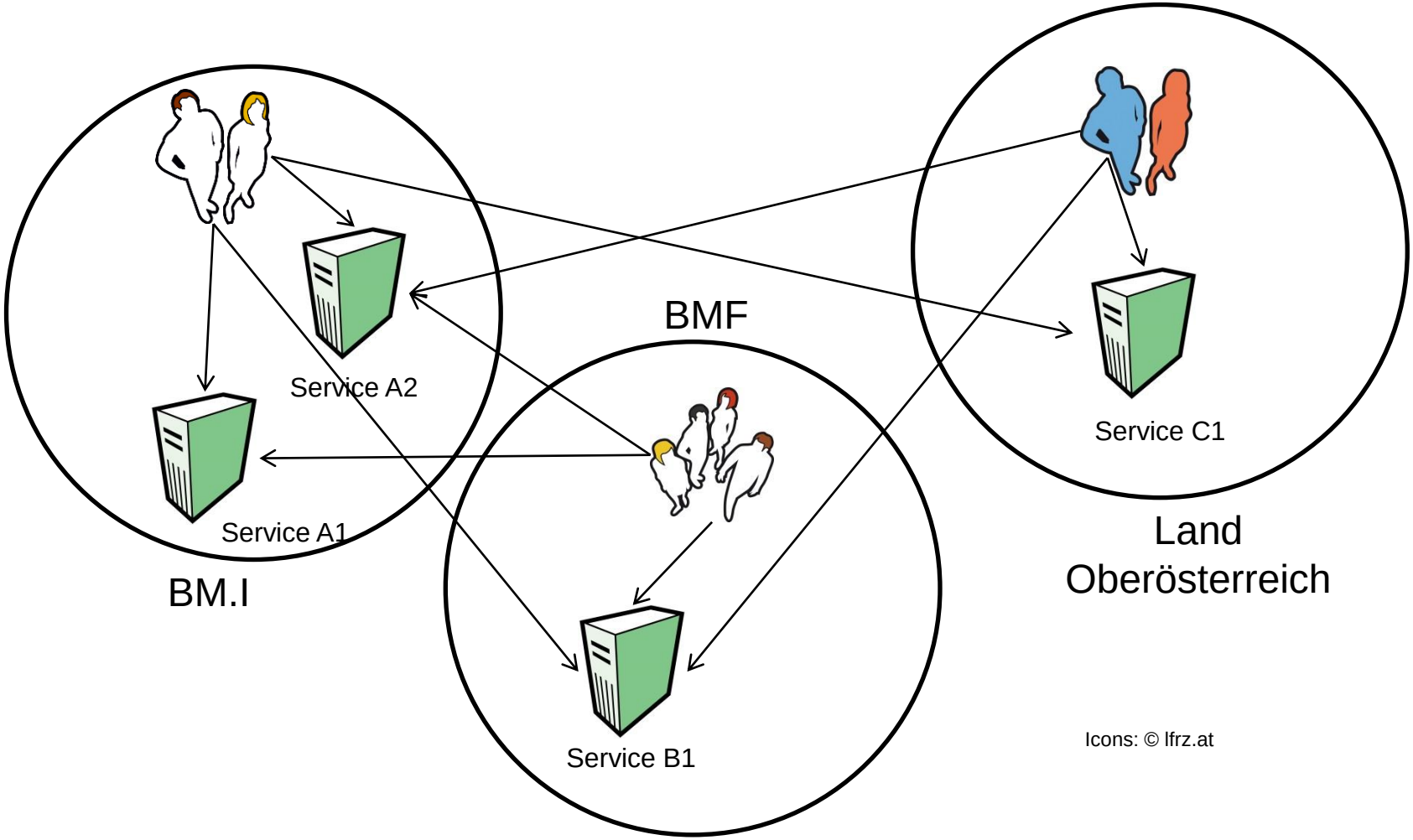


Stammorganisation nutzt E-Government-Services

Anwendungsverantwortliche Organisation bietet Services für andere Behörden an

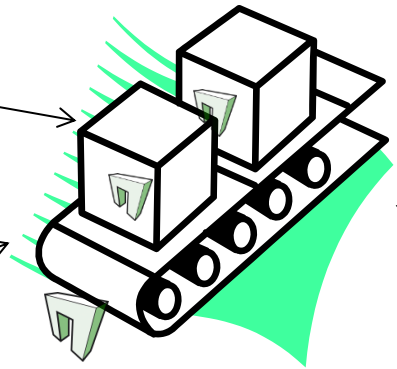
Quelle lfrz.at

Organisationsübergreifende Servicenutzung



Sicherheitsrelevante Vorgaben SW-Entwicklung

Interne
Sicherheitsüberprüfungen



Software-Entwicklung
Standardportal

ÖNORM A 7700

Inputs aus Security-
Checks von Kunden

Externe Audits (A 7700)

Grafiken von:
owasp.org,
a7700.org,
lfrz.at, Microsoft
Office Clip Art

Open Web Application Security Project

Quelle: owasp.org,

Open Source Projekt zur Entwicklung von Vorgaben für die sichere Software-Entwicklung.

Vielzahl von positiven Guidelines (= wie macht man es richtig – nicht was darf man nicht machen).

Problem ist die Vielzahl der Vorgaben / OWASP Projekte

https:\\www.owasp.org

Bespiel: OWASP 2013 TOP 10 List (Platz 1-4)

A1-Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2-Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
A3-Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4-Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

Quelle: owasp.org,

Beispiel: OWASP TOP1-2013 - Injection

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability EASY	Prevalence COMMON	Detectability AVERAGE	Impact SEVERE	Application / Business Specific
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.	Attacker sends simple text-based attacks that exploit the syntax of the targeted interpreter. Almost any source of data can be an injection vector, including internal sources.	Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss or corruption, lack of accountability, or denial of access. Injection can sometimes lead to complete host takeover.	Consider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified, or deleted. Could your reputation be harmed?

Quelle: owasp.org,

Überblicksinformationen (s.o)
– gefolgt von Detailinfos zum Verständnis (kein Screen-Shot)

Beispiel: OWASP TOP1-2013 - Injection

How Do I Prevent 'Injection'?

Preventing injection requires keeping untrusted data separate from commands and queries.

1. The preferred option is to use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface. Be careful with APIs, such as stored procedures, that are parameterized, but can still introduce injection under the hood.
2. If a parameterized API is not available, you should carefully escape special characters using the specific escape syntax for that interpreter. OWASP's ESAPI provides many of these escaping routines.
3. Positive or "white list" input validation is also recommended, but is not a complete defense as many applications require special characters in their input. If special characters are required, only approaches 1. and 2. above will make their use safe. OWASP's ESAPI has an extensible library of white list input validation routines.

References

OWASP

- OWASP SQL Injection Prevention Cheat Sheet
- OWASP Query Parameterization Cheat Sheet
- OWASP Command Injection Article
- OWASP XML eXternal Entity (XXE) Reference Article
- ASVS: Output Encoding/Escaping Requirements (V6)
- OWASP Testing Guide: Chapter on SQL Injection Testing

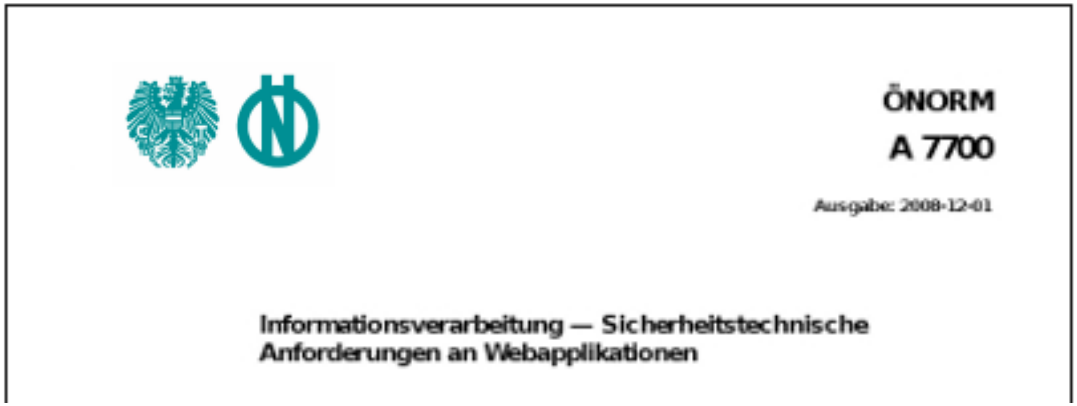
External

- CWE Entry 77 on Command Injection
- CWE Entry 89 on SQL Injection
- CWE Entry 564 on Hibernate Injection

Konkrete Vorgaben um sicherzustellen, dass Software nicht anfällig ist.

Quelle: owasp.org

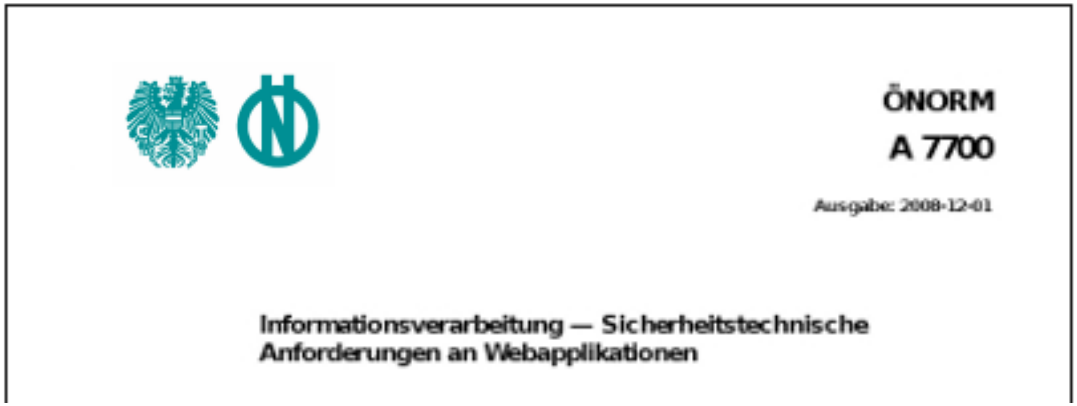
ÖNORM A 7700


ÖNORM
A 7700
Ausgabe: 2008-12-01
Informationsverarbeitung — Sicherheitstechnische Anforderungen an Webapplikationen

- Österreichische Norm für sichere Webanwendungen
- Es ist möglich Versionsstände einer Software nach dieser Norm zu zertifizieren.
- Kompakte – aber sehr abstrakte Definition (Seit 2008 unverändert)
- Aufwand Erstzertifizierung Standardportal >> 100.000 €

Quelle: http://www.a7700.org/

ÖNORM A 7700


ÖNORM
A 7700
Ausgabe: 2008-12-01
Informationsverarbeitung — Sicherheitstechnische Anforderungen an Webapplikationen

Inhaltsverzeichnis:

- Architektur der Webapplikation
- Datenspeicherung und Datentransport
- Konfigurationsdaten
- Authentifizierung
- Authentifizierungsmethoden
- Passwörter
- Autorisierung
- Sitzungen
- Separierung durch Sitzungen
- Qualitätskriterien für Sitzungen
- Behandlung von Benutzereingaben
- Dateigenerierung
- Speichermanagement
- Einbinden von Ressourcen
- Behandlung von Datenausgaben
- Hintergrundsysteme
- System- und Fehlermeldungen
- Kryptographie

Quelle: http://www.a7700.org/

ÖNORM A 7700 – Beispiel für eine Vorgabe

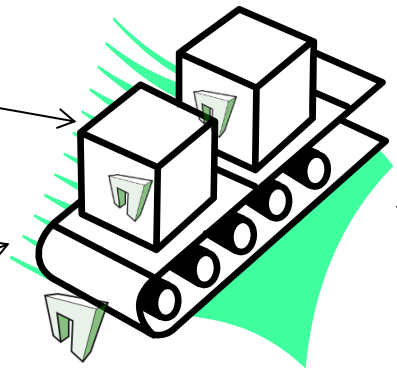
…

10 System und Fehlermeldungen

System- und Fehlermeldungen sowie Kommentare dürfen keine Informationen an die Benutzer ausgeben, die in weiterer Folge für Angriffe auf das System genutzt werden können

Quelle: ÖNORM A 7700

SW-Entwicklung



Software-Entwicklung

Interne Sicherheitsüberprüfungen

ÖNORM A 7700

Inputs aus Security-Checks von Kunden

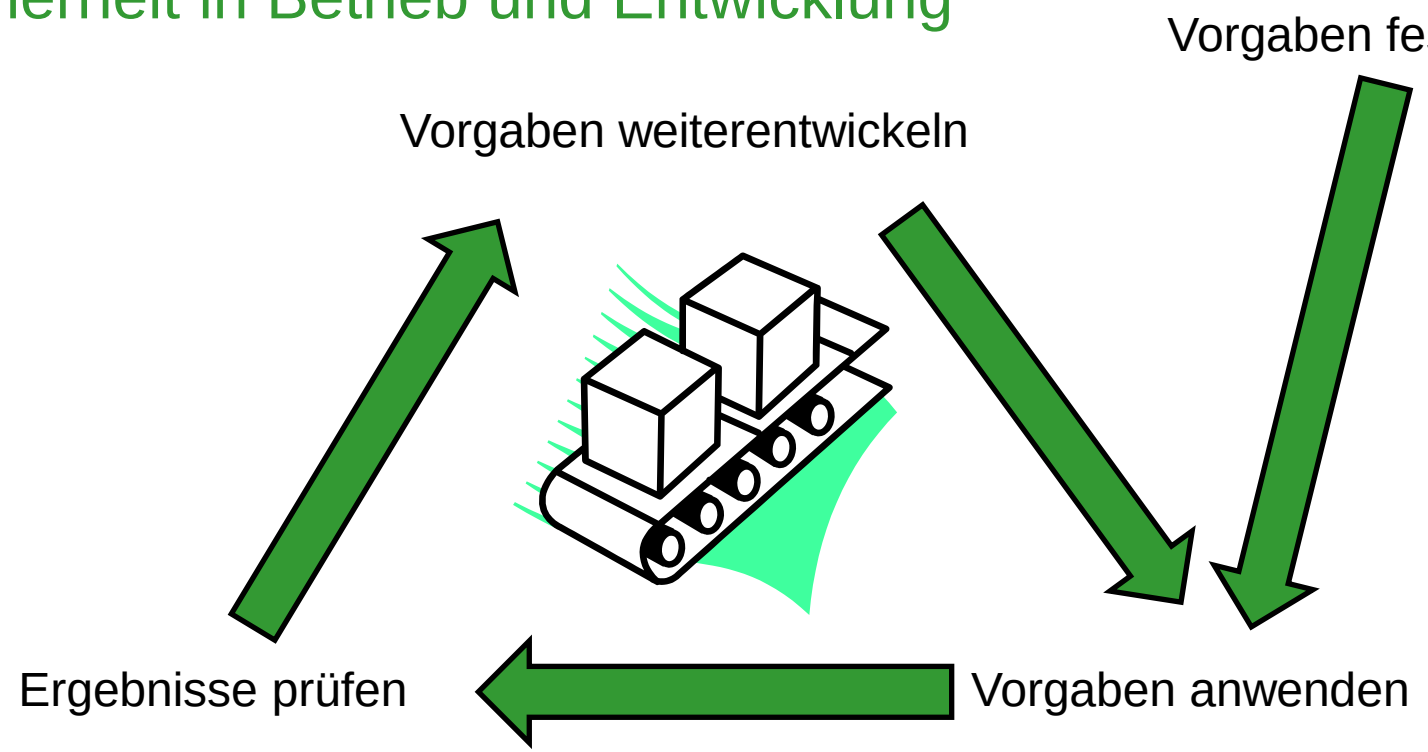
Externe Audits (A 7700)

Grafiken von: owasp.org, a7700.org, lfrz.at, Microsoft Office Clip Art

Zusammenfassung

- Es gibt ausreichend sicherheitstechnische Vorgaben und Normen (z.B. OWASP, ÖNORM A 7700)
- Vielzahl von Vorgaben und Entdeckung immer neuer Angriffsarten führt dazu, dass kein Software-Entwicklungs-Team alle aktuellen sicherheitstechnischen Vorgaben kennt.
- Security-Reviews helfen die Qualität der eigenen Software-Entwicklung abschätzbar zu machen und die Software-Entwicklungsqualität zu optimieren.

Sicherheit in Betrieb und Entwicklung



Grafiken von:
Microsoft Office,
LFRZ

Links & E-Mail-Adressen

LFRZ
http://www.lfrz.at

Portalverbund
http://reference.e-government.gv.at/Portalverbund.577.0.html

OWASP
https://www.owasp.org/index.php/Main_Page

ÖNORM A7700
http://www.a7700.org/

gerold.pesendorfer@lfrz.at
peter.pichler@lfrz.at