KITODO

KEY TO DIGITAL OBJECTS

Releaseplanung und IT-Sicherheit

Kitodo-Praxistreffen 2023

Releaseplanung und IT-Sicherheit

Kitodo verwendet „semantic versioning“, d.h. dreiteilige Versionsnummern in der Form **X.Y.Z**

- **X** -> Major Version; bedeutet „breaking changes“
- **Y** -> Minor Version; enthält neue Features
- **Z** -> Patch Level; behebt nur Fehler

- typischerweise wird nur die aktuellste Major Version („stable“) aktiv weiterentwickelt, um Doppelentwicklungen zu vermeiden
- für eine gewisse Übergangsfrist wird die vorherige Major Version („old stable“) jedoch weiterhin mit Fehlerbehebungen versorgt
- alle älteren Versionen werden nicht mehr unterstützt

Releaseplanung und IT-Sicherheit

Beispiel Kitodo.Presentation:

- Aktuelle Version ist 4.0.1 („Fehler in Major Version 4 behoben“)
- Nächstes Release ist 4.1.0 („neue Features in Major Version 4“)
- Folgendes Release ist 5.0.0 („unterstützt neue TYPO3-Version“)

- ab dem Release 5.0.0 gibt es für die Major Version 4 nur noch Fehlerbehebungen (4.1.1, 4.1.2, 4.1.3, …) und die Unterstützung für die Major Version 3 wird eingestellt
- generell Orientierung am Lebenszyklus von TYPO3, d. h. Unterstützung älterer Major Releases höchstens so lange wie die zugrundeliegende TYPO3-Version (LTS) noch unterstützt wird

Releaseplanung und IT-Sicherheit

Software Life Cycle

SECURITY.md

Security Policy

Supported Versions

The following versions of Kitodo.Presentation are currently being supported with ongoing feature development and/or security updates.

Version	With TYPO3	Active Development	Security Fixes
5.x	10 LTS + 11 LTS	☒	☒
4.x	9 LTS + 10 LTS	☐	☒
3.3.x	9 LTS	☐	☐
3.2.x	8 LTS + 9 LTS	☐	☐
3.1.x	8 LTS + 9 LTS	☐	☐
3.0.x	8 LTS	☐	☐
2.x	6 LTS + 7 LTS	☐	☐
1.x	4 LTS + 6 LTS	☐	☐

Reporting a Vulnerability

If you find a vulnerability please consider immediately reporting it to our release management team or by sending an email.

Releaseplanung und IT-Sicherheit

Konsequenzen für Kitodo:

- zeitnahe Aktualisierung auf neue Major Version angeraten (aus Sicherheitsgründen ohnehin sinnvoll)
- Behebung von Fehlern immer in „stable“ und „old stable“ (! Pull Requests für zwei Branches verpflichtend !)
- Feature-Entwicklungen immer in aktueller Major Version (auch wenn lokal ältere Version eingesetzt wird)

-> formale Regelung in Kitodo Coding Guidelines

Releaseplanung und IT-Sicherheit

Releasezyklen

- immer wieder wird der Wunsch nach regelmäßigen Releases geäußert und diskutiert
- Entwicklung verläuft jedoch i. d. R. nicht kontinuierlich, sondern projektgetrieben
- Zeitplan orientiert sich entsprechend an Projekterfordernissen und nicht an Release-Planung

-> regelmäßige Releases an festen Stichtagen daher unrealistisch

Releaseplanung und IT-Sicherheit

Releasezyklen

ABER: häufigere, kleinere Releases sind möglich, erfordern jedoch mehr Engagement der Community

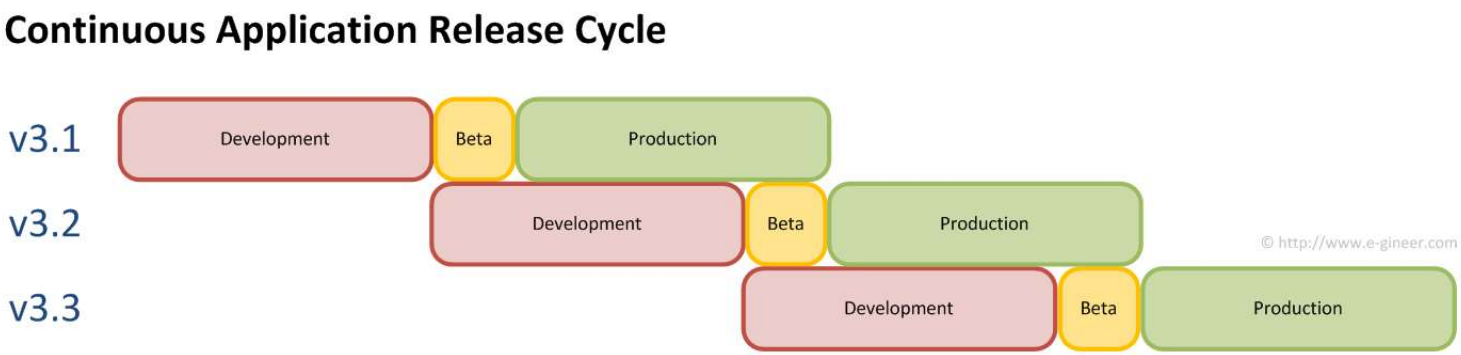
Aktuelle Probleme:

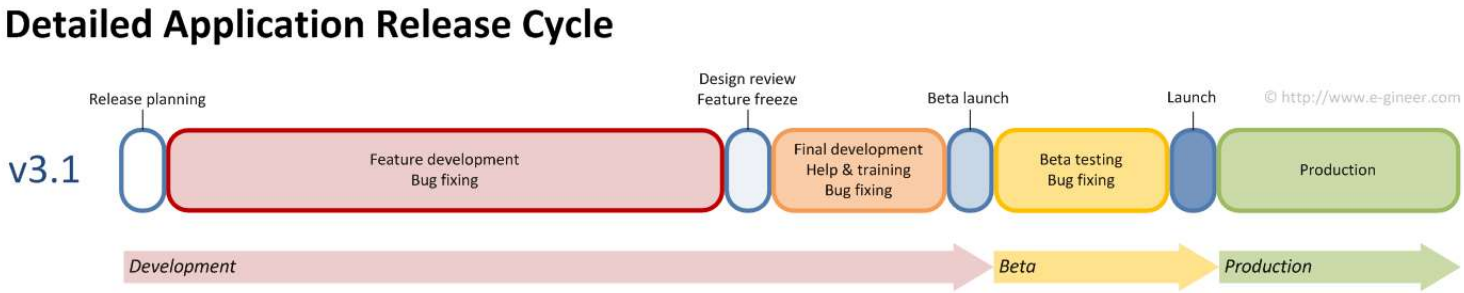
- ~~weit mehr Bug Reports und Pull Requests als Reviews und Tests -> dadurch „Stau“ beim Release Management und hohe Aufwände für Testing~~
- ~~häufiger Feature-Entwicklungen als Fehlerbehebungen -> dadurch keine notwendige Stabilisierung für Release~~

Vielen Dank nach Dresden, Hamburg und Mannheim!

Releaseplanung und IT-Sicherheit

Releasezyklen





Releaseplanung und IT-Sicherheit

Releasezyklen

Konsequenzen für Kitodo:

- kürzerer Verlauf von der Entwicklung bis zum Release
- übersichtlichere Änderungen, einfachere Aktualisierung
- nach Feature Freeze haben Fehlerbehebungen Priorität gegenüber neuen Features
- zeitnahes Testing und Bugfixing durch die Community nötig (Beta Releases und Release Candidates)
- Branch Management wird etwas komplexer, aber gute Unterstützung durch Git/GitHub

Releaseplanung und IT-Sicherheit

Releasezyklen

Aktueller Stand:

- Feature Freeze für Version 5.0
- derzeit Fehlerbehebung und „Feinschliff“
- danach Veröffentlichung eines Beta Release zum Testen

- Entwicklungsende für Version 4.x
- derzeit Vorbereitung eines letzten Feature Release 4.1.0
- danach nur noch Minor Releases für Fehlerbereinigung

Releaseplanung und IT-Sicherheit

Sicherheitslücke in Kitodo.Production: Stichwort „log4shell“

- 2021 entdeckte schwere Sicherheitslücke in der Java-Bibliothek „log4J“
- Kitodo.Production betroffen durch veraltete Dependencies
- zeitnah behoben in Releases 3.4.1 und 3.3.3

Sicherheitslücken in Kitodo.Presentation

- TYPO3-EXT-SA-2020-015 (CSS)
- TYPO3-EXT-SA-2022-001 (SSRF)
- zeitnah behoben in Releases 2.3.2, 3.1.2, 3.2.3 und 3.3.4

Releaseplanung und IT-Sicherheit

Maßnahmen zur Reduzierung von Sicherheitsrisiken

- Einsatz von Static Code Analyzers (Codacy, CodeQL, Dependabot, PHPStan)
- Regelmäßige Sicherheitsprüfung durch TYPO3 Security Team (für Kitodo.Presentation)
- Einhaltung hoher Programmierstandards via Coding Guidelines
- Report von Sicherheitslücken durch Nutzende direkt an Release Management (via GitHub oder E-Mail security@kitodo.org)
- Professioneller Umgang mit Sicherheitslücken (Dokumentation, Kommunikation, Security Advisory, zeitnahe Fehlerbehebung)

Releaseplanung und IT-Sicherheit

Fragen?

Vielen Dank für Ihre Aufmerksamkeit!

Fragen? Fragen!

Oder später:

Kitodo.Presentation: Sebastian Meyer
sebastian.meyer@openculturecons ulting.com
Kitodo.Production: Arved Solth
solth@effective-webworks.de

KITODO