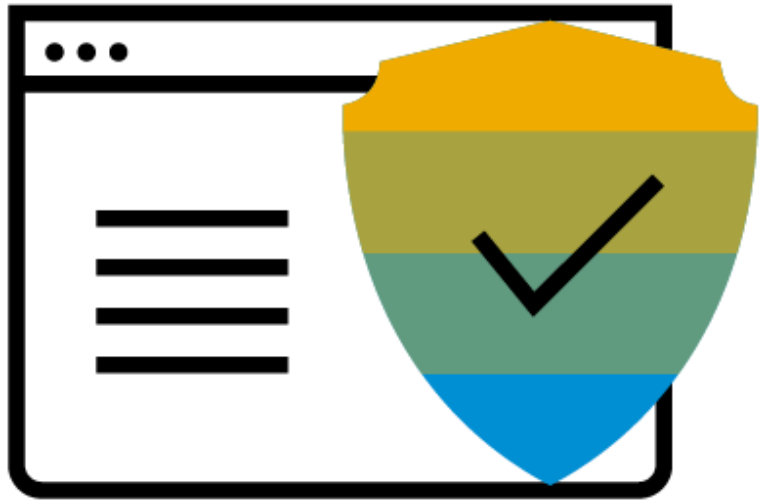
# Copyright- und Lizenz-Compliance
## bei SAP-Open Source-Projekten



Sebastian Wolf, SAP
29. September 2022

Public

THE BEST RUN SAP

# Hauptrisiken bei eigenen Open Source-Projekten/Beiträgen

Veröffentlichung von Betriebsgeheimnissen

Konkurrenz zu eigenen Produkten

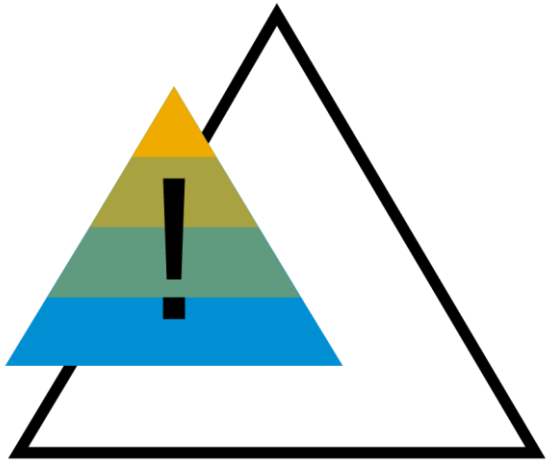
Verstoß gegen Exportkontrollrichtlinien

- Beispiele: Verschlüsselungsalgorithmen, Dual-Use-Software

Verwässerung der eigenen Marke

Sicherheitsprobleme

- Schadensersatzforderungen, negative Publicity



**Probleme mit Urheberrechten, Patenten und Lizenzen**

- Kopieren/Modifikation von fremdem Code ohne (korrekte) Urheberangabe
- Nutzung von Copyleft-Code ohne Übernahme von Copyleft
- Versehentliche Gewährung einer Patentlizenz

# Risikomanagement für Copyright und Lizenzen: Je früher, desto besser!

## Shift-Left nicht nur für DevOps!

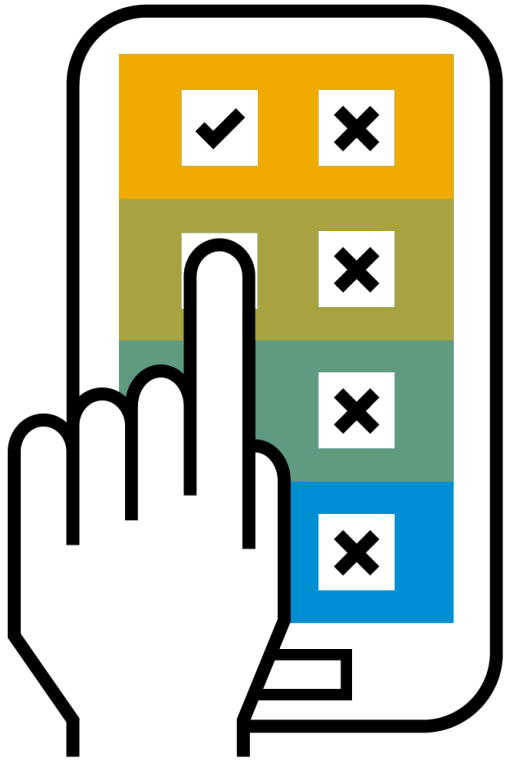
- Risiken und Herausforderungen so früh wie möglich angehen
- Auch relevant bei OSS-Veröffentlichungen

## Zielgerichteter Fragebogen vor Veröffentlichungsprozess

- Fremdcode? Wenn ja, welche Lizenzen?
- Patente?
- Möglicher Wettbewerb intern und extern?
- Kurzes Interview zu den Themen des Fragebogens

## Bürokratie nicht als Selbstzweck verstehen!

- Fördert gute Projekte
- Verhindert unnötige Arbeit und Probleme



# Risikomanagement: Immer am Ball bleiben!

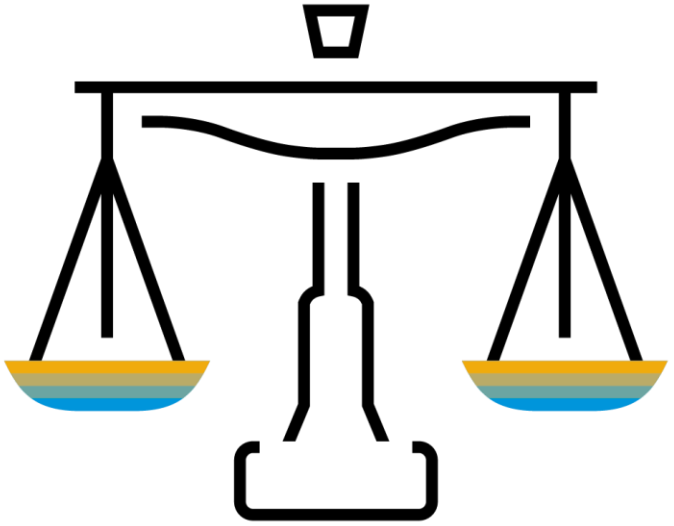
Einmal prüfen ist keinmal prüfen!

- Inkonsistente Kontrollen:
  – Strikte Prüfung bei/vor Veröffentlichung
  – Wenige/keine Prüfungen im weiteren Projektverlauf
  – Fluktuation wird nicht berücksichtigt
- Folge:
  – Zu Beginn Policy-konforme Projekte verletzen über die Zeit Regeln
  – Ne länger man Projekte laufen lässt, umso schwieriger wird Korrektur
  – Verantwortlichkeiten diffundieren
- Notwendige Konsequenz:
  – Regeln über den kompletten Lebenszyklus des Projekts überprüfen
  – Maintainer immer aktuell halten
  – Wenn Regeln länger nicht eingehalten werden: Sunset des Projekts

# Bei Einführung von Guidelines: Immer auf gute Balance achten!

## Überregulierung unbedingt vermeiden!

- Guidelines und Regeln müssen überschaubar und verstehbar sein
- Zu viele/intransparente Regeln führen dazu, dass …
  – Sie bewusst oder unbewusst ignoriert werden
  – Innovative Projekte gar nicht erst angegangen werden
  – Sich Projekte unnötig verlangsamen oder zu teuer werden



## Kommunikation oft wichtiger als Regeltext

- Inhalte und Gründe für Regeln erklären (Trainings, Q&A usw.)

## Verantwortung wahrnehmen!

- Vieles kann und muss an Entwicklung delegiert werden
- Aber: Vieles kann zentral erledigt und automatisiert werden

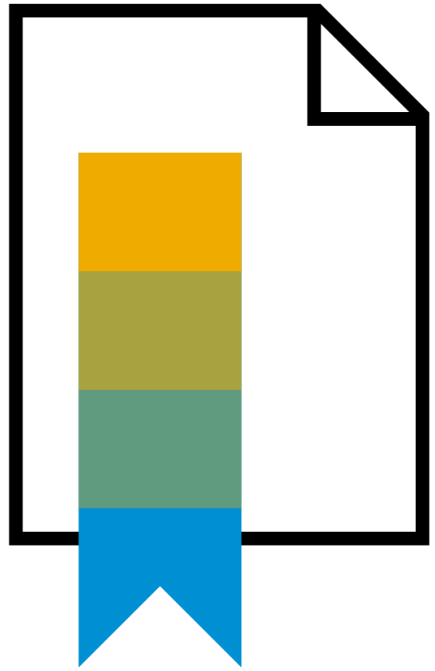
# Formalisierung über eine eigene Open Source-Policy

## Open Source ist wichtig fürs Business bei SAP

- ~100% aller Lösungen beinhalten OSS oder hängen von OSS ab
- OSS erleichtert/beschleunigt Produktentwicklung
- Beiträge zu OSS liegen im strategischen Interesse der Firma

## Policy zum unternehmenseinheitlichen Risikomanagement

- IP-Management
- Lizenz-Compliance
- Security
- Ermöglicht Zertifizierungen wie OpenChain



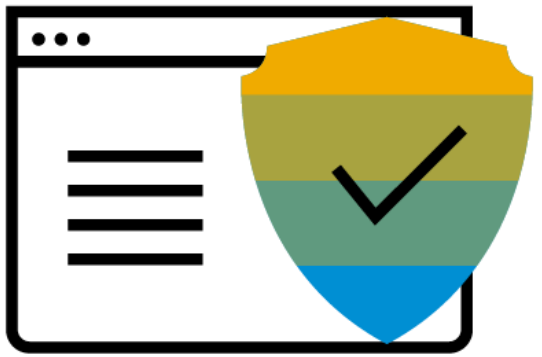
# Identifizierbarkeit von Beiträgen, Nutzung richtiger Lizenzen

## OSS-Beiträge von SAP-Beschäftigten

- User muss SAP-E-Mail-Adresse verwenden
- Öffentliche GitHub-User müssen mit SAP-ID verknüft werden
- Beiträge nur unter Bedigungen vorab geprüfter CLAs
- Trennung von privaten und beruflichen Beiträgen

## Berücksichtigung von **Lizenz-Risikoklassen**

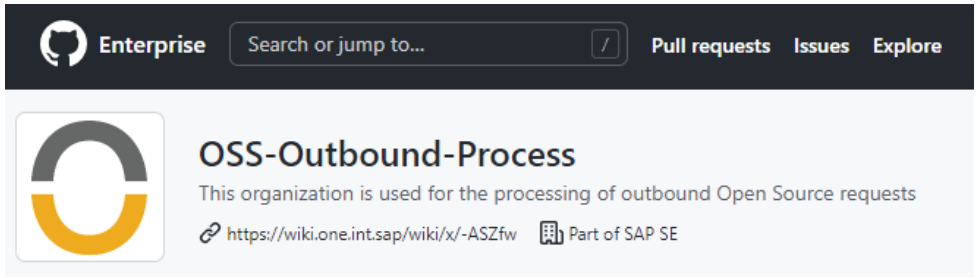
- **Niedriges** Risiko (z.B. BSD, Apache 2.0, MIT)
  – Nutzung möglich
  – Apache 2.0 ist Standard für SAP-eigene Projekte
- **Mittleres** Risiko (z.B. EPL)
  – Nutzung möglich, Vorsicht bei Modifikationen
- **Hohes** Risiko (z.B. [A/L]GPL, CC-BY-SA)
  – Nutzung nur in genehmigten Ausnahmefällen



# Prozessimplementierung über GitHub Enterprise

## Automatisierter Veröffentlichungsprozess

- Erstantrag über GitHub-Issue bei SAP OSPO
- Eigenes GitHub-Repository pro Antrag
- Einzelne Prozessschritte per GitHub-Issues abgebildet




## Bot-unterstützte Prozesschritte

- OSPO-Team-Mitglieder werden Anträgen dynamisch zugewiesen
- Automatisiertes Triggern von Code-Scans
- Erstellung der GitHub-Teams und –Repositories aus GitHub-Issue

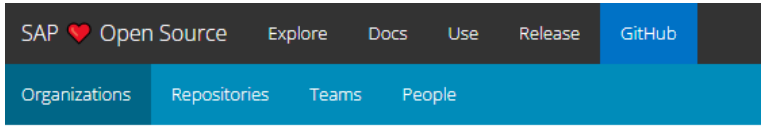
## Initiale Compliance sichergestellt

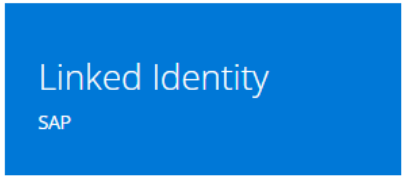
- Nutzung der Apache 2.0-Lizenz
- Copyright-Metadaten per Template im Repository

# Werkzeuge zur Unterstützung bei Projektbeteiligung und –gründung

## Open Source Management Portal

- Link zwischen GitHub-ID und Unternehmens-ID
- Verpflichtend für SAP-Entwickler auf GitHub
- Open Source-Projekt von Microsoft




## CLA-Assistant

- Managt Unterzeichnung des CLAs bzw. DCOs
- Standard für alle SAP-GitHub-Organisationen
- Open Source-Projekt von SAP

## REUSE-Framework

- Korrekte Copyright- und Lizenz-Annotation aller Projektdateien
- Open Source-Projekt der FSFE
- Verpflichtend für alle SAP-Open Source-Projekte
- Automatisierte Metadaten-Generierung mit eigenem OSS-Tool

# Werkzeuge für Day-2-Operations

## FOSSTARS – Open Source „Rules of Play“

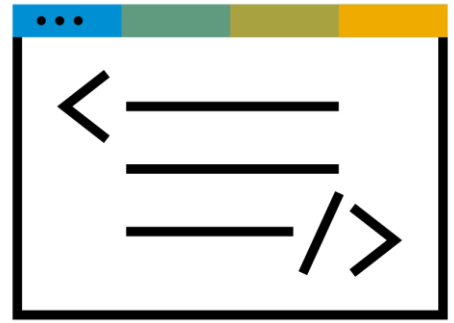
- Wöchentliche Prüfung aller Projekte
- Getrackte GitHub Issues bei Policy-Verletzungen
- Besonderes Augenmerk auf REUSE-Framework-Probleme
- Open Source-Projekt von SAP

## BlackDuck – Komponenten-Scan

- Monatlicher Scan aller Projekte
- Prüfung auf inkompatible Lizenzen
- Nutzung des BlackDuck Detect-Scripts von Synopys

## Weitere interne automatisierte Werkzeuge

- Erkennen und Archivieren inaktiver Projekte
- Aktivierung des Issue-Trackers für alle Projekte



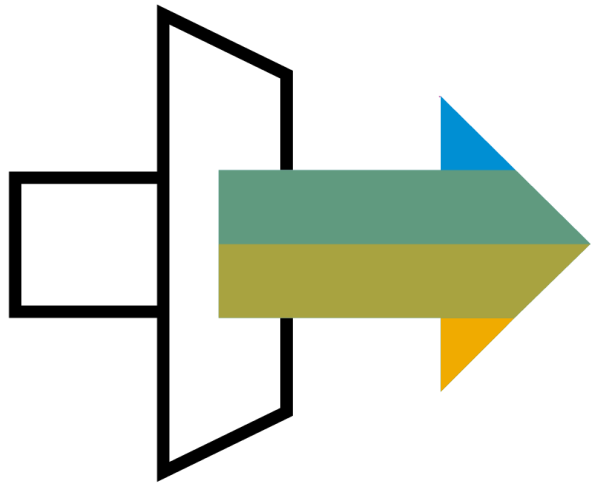
# Zusammenfassung und Ausblick

## Copyright- und Lizenz-Compliance geht alle an

- Open Source Program Office arbeitet Leitlinien aus
- Klare Guidance, welche Lizenzen genutzt werden dürfen
- Aufklärung über mögliche Copyright-Verletzungen
- Leitlinien idealerweise als Unternehmenspolicy umsetzen
- Policy möglichst einfach und verständlich halten
- Umsetzung der Kontrollen so weit wie möglich automatisieren

## Weitere Verbesserungen in der Pipeline

- Projekt-Releaseprozess wird Probot-basiert weiter vereinfacht
- Weitere Investitionen in Dokumentation und Training
- Große Herausforderungen:
  – False Positives bei BlackDuck-Komponenten-Scan
  – Neue Technologien wie GitHub CoPilot



# Danke für Ihre Aufmerksamkeit.

Kontaktinformation:

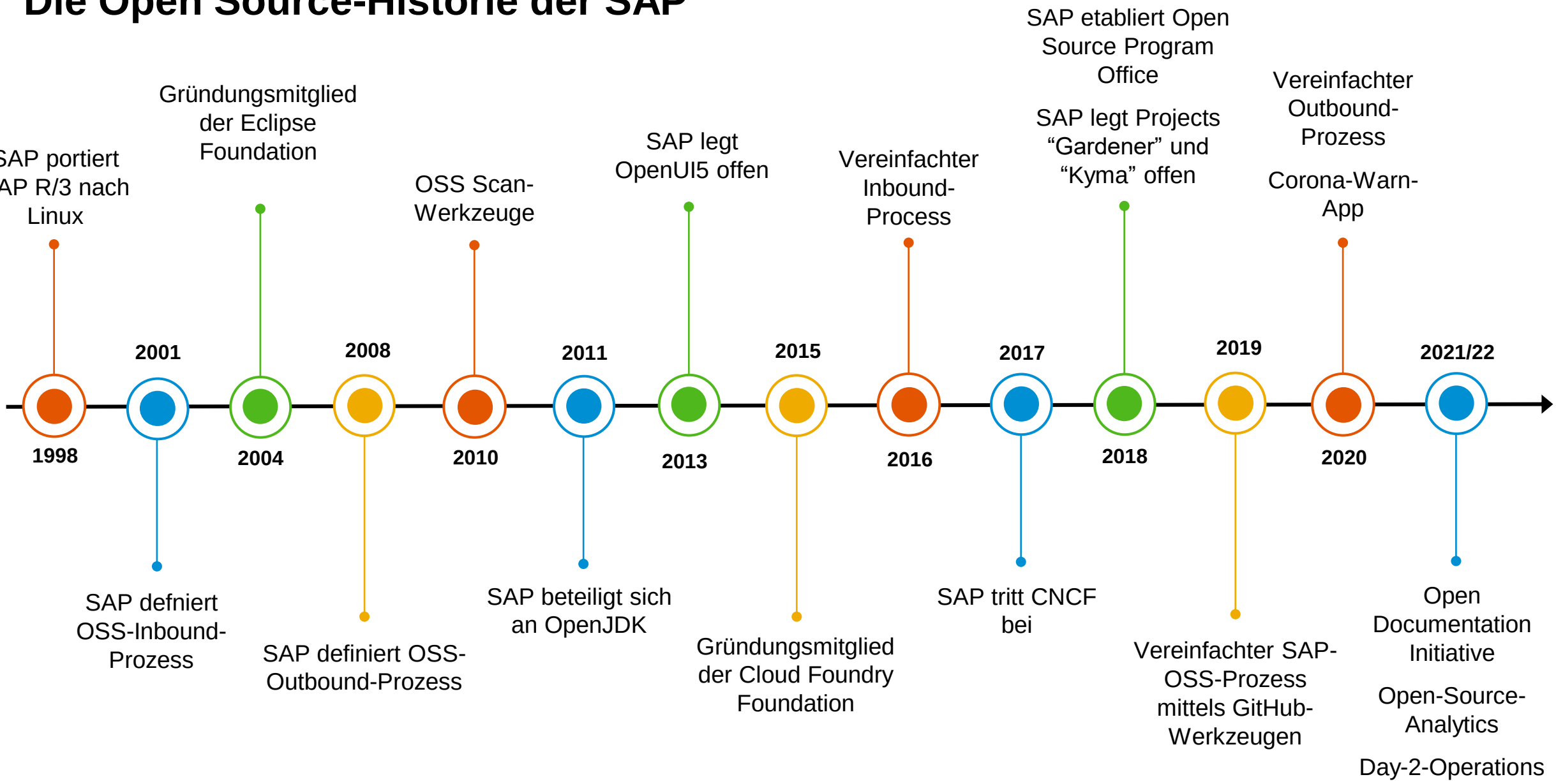
**Sebastian Wolf**
sebastian.wolf@sap.com

**SAP Open Source Program Office**
opensource.sap.com
community.sap.com/topics/open-source
ospo@sap.com
twitter.com/sapopensource

THE BEST RUN SAP

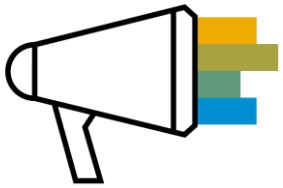
# Anhang

# Die Open Source-Historie der SAP

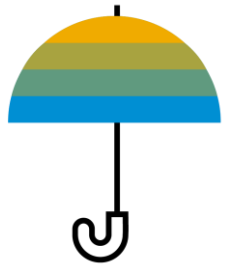


# SAP Open Source Program Office – Unsere Mission

- Zentrales Team für alle Open Source-Themen bei SAP
- Unterstützung aller SAP-Teams bei der Verwendung und Publikation von Open Source
- Risikomanagement nach den SAP-Compliance-Regeln



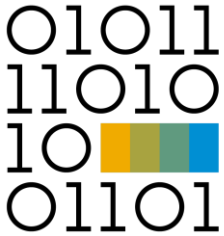
**Strategie und Policy**
Wir definieren, entwickeln und fördern die SAP Open Source-Strategie und Policy



**Compliance und Risikomanagement**
Wir minimieren und handhaben die Risiken, Verpflichtungen und Herausforderungen bei Open Source-Software, sowohl bei Verwendungen als auch eigenen Beiträgen

**Unterstützung und Training**
Wir unterstützen und trainieren die SAP-Teams darin, von Open Source-Software zu profitieren.

**Prozesse und Werkzeuge**
Wir verbessern regelmäßig unsere Open Source-Management-Prozesse und -Werkzeuge



**InnerSource**
Wir fördern und vereinfachen InnerSource bei SAP