

Skalierung von automatisierten Accessibility Regressions-Tests

Björn Scherer
CosmosDirekt (Generali Deutschland)

COSMOS DIREKT

Ein Unternehmen der



Motivation



Barrierefreiheit muss gewährleistet bleiben!

Ansatz

Automatisierung als
Gatekeeper für Regressionen

Dazu muss sie Skalieren! Aber WIE?

Björn Scherer

- » 15+ Jahre im QA-Umfeld
Schwerpunkt Automatisierung
- » Lead Test-Engineer
für Cosmos Direkt
- » Leitung Community of Practice
„Test & -Automatisierung“
- » HTW-Saar Dozent für
„Software Quality Engineering“

COSMOS DIREKT

- » hoher Grad an
Eigen-Entwicklungen
- » Dev: Web- & Hybrid-Apps
- » Fokus: Inhouse & Endkunden
- » Methodik: agile & DevOps
- » Test-Engineering als Plattform



Disclaimer

Automatisierung als Gatekeeper für Regressionen

**Die Automatisierung soll und kann
manuelle Prüfungen NICHT ersetzen.
Sie sind und bleiben wichtig!**



Non-text Content	Audio-only and video-only	Captions (prerecorded)	Audio Description or media alternative	Captions (live)	Audio description (prerecorded)	Sign-language (prerecorded)	Extended Audio-description (prerecorded)	Media alternative (prerecorded)	Audio-only (live)
Info and Relationships	Meaningful sequence	Sensory characteristics	Orientation	Identify Input Purpose	Identify Purpose	Use of Color	Audio Control	Contrast (Minimum)	Resize Text
Images of Text	Contrast (Enhanced)	Low or no background audio	Visual Presentation	Images of Text (no exception)	Reflow	No-text Contrast	Text spacing	Content on hover or focus	Keyboard
No Keyboard Trap	Keyboard (no exception)	Character Key Shortcuts	Timing Adjustable	Pause, Stop, Hide	No Timing	Interruptions	Re-Authenticating	Timeouts	Three Flashes or Below Threshold
Three Flashes	Animation from Interactions	Bypass Blocks	Page Titled	Focus Order	Link Purpose (in context)	Multiple Ways	Headings and Labels	Focus Visible	Locations
Link Purpose (Link only)	Section Headings	Focus not obscured (minimum)	Focus not obscured (enhanced)	Focus Appearance	Pointer Gestures	Pointer Cancellation	Label in Name	Motion Actuation	Target Size (enhanced)
Concurrent Input Mechanisms	Dragging Movements	Target Size (minimum)	Language of Page	Language of Parts	Unusual Words	Abbreviations	Reading Level	Pronunciation	On Focus
On Input	Consistent Navigation	Consistent Identification	Change on Request	Consistent Help	Error Identification	Labels or Instructions	Error Suggestion	Error Prevention (Legal, Financial, Data)	Help
Error Prevention (all)	Redundant Entry	Accessible Authentication (minimum)	Accessible Authentication (enhanced)	Parsing (obsolete and removed)	Name, Role, Value	Status Messages	* grün:	Es existieren ACT-konforme Prüfungen, die Aspekte des SuccessCriteria automatisiert evaluieren.	

Non-text Content	Audio-only and video-only	Captions (prerecorded)	Audio Description or media alternative	Captions (live)	Audio description (prerecorded)	Sign-language (prerecorded)	Extended Audio-description (prerecorded)	Media alternative (prerecorded)	Audio-only (live)
Info and Relationships	Meaningful sequence	Sensory characteristics	Orientation	Identify Input Purpose	Identify Purpose	Use of Color	Audio Control	Contrast (Minimum)	Resize Text
Images of Text	Contrast (Enhanced)	Low or no background audio	Visual Presentation	Images of Text (no exception)	Reflow	No-text Contrast	Text spacing	Content on hover or focus	Keyboard
No Keyboard Trap	Keyboard (no exception)	Character Key Shortcuts	Timing Adjustable	Pause, Stop, Hide	No Timing	Interruptions	Re-Authenticating	Timeouts	Three Flashes or Below Threshold
Three Flashes	Animation from Interactions	Bypass Blocks	Page Titled	Focus Order	Link Purpose (in context)	Multiple Ways	Headings and Labels	Focus Visible	Locations
Link Purpose (Link only)	Section Headings	Focus not obscured (minimum)	Focus not obscured (enhanced)	Focus Appearance	Pointer Gestures	Pointer Cancellation	Label in Name	Motion Actuation	Target Size (enhanced)
Concurrent Input Mechanisms	Dragging Movements	Target Size (minimum)	Language of Page	Language of Parts	Unusual Words	Abbreviations	Reading Level	Pronunciation	On Focus
On Input	Consistent Navigation	Consistent Identification	Change on Request	Consistent Help	Error Identification	Labels or Instructions	Error Suggestion	Error Prevention (Legal, Financial, Data)	Help
Error Prevention (all)	Redundant Entry	Accessible Authentication (minimum)	Accessible Authentication (enhanced)	Parsing (obsolete and removed)	Name, Role, Value	Status Messages	* grün:	Es existieren ACT-konforme Prüfungen, die Aspekte des SuccessCriteria automatisiert evaluieren.	

Herausforderungen:

Lange UseCases:

Masse von
Einzel-Checks
+ unerreichbare
Stellen

nur bedingt
CI-tauglich

schnelles
Feedback
VS.
breite
Abdeckung

Stateful Flows

» Zustände nur über Flows erreichbar

» Duzende States je Flow

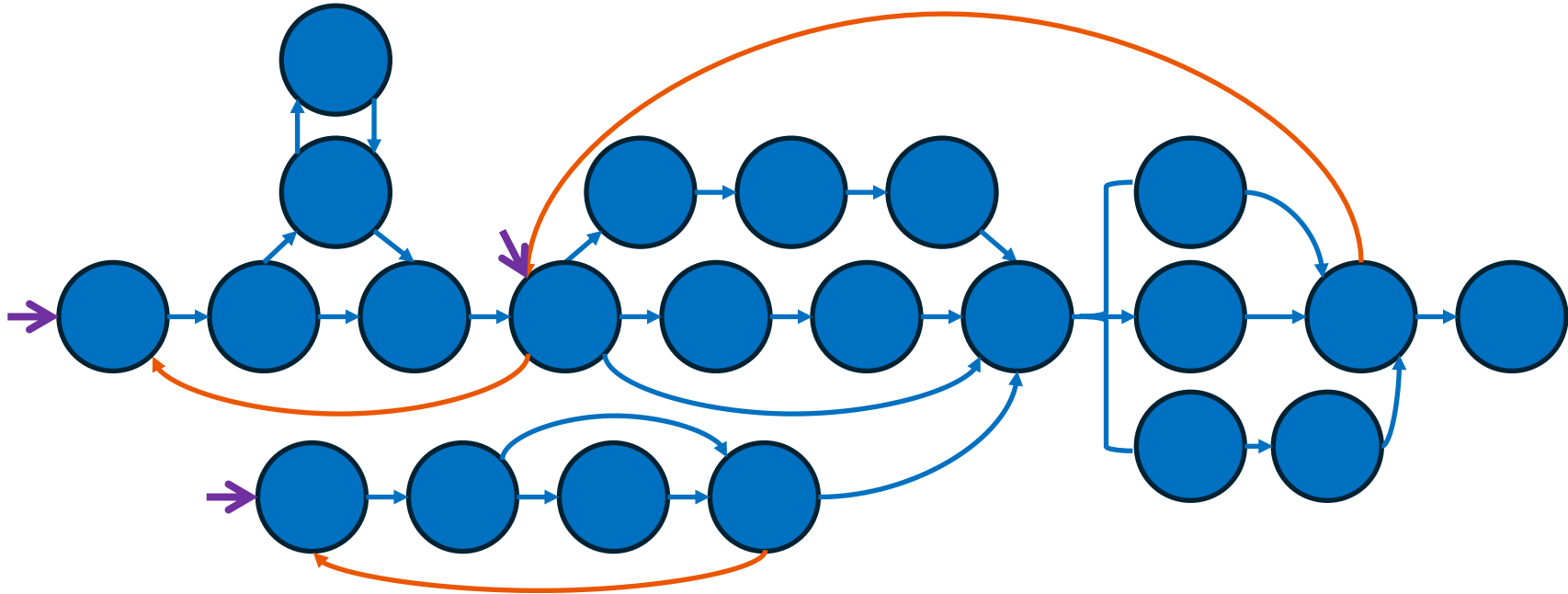
» **zusätzliche States**

(durch dynamische Masken,
Fehlerzustände, Overlays, etc.)

→ keine Einzel-Checks
via URL möglich

→ Einzelchecks decken nicht ab,
was Nutzer wirklich erleben

Stateful Flows (Zustände nur über Flows erreichbar)



* vereinfachte, schematische Darstellung

Optionen?

~~Einzel-Check- Tests~~

~~1 A11Y Check
je Testfall~~

» zu kleinteilig

~~Test-Dopplung~~

~~bestehende Testfälle
kopieren und mit A11Y
Checks ergänzen~~

» Wartung

Integration in bestehende Tests

decken gesamte Logik
schon ab

» keine / wenig
neue Testfälle

```
it('check vertraege #bfsg', () => {  
  fixture.loginWithTestUser(fixture.td.account);  
  a11y.analyzeCurrentState('startseite');  
  
  fixture.startseite.gotoVertragsuebersicht();  
  a11y.analyzeCurrentState('vertragsuebersicht');  
  
  fixture.vertragsuebersicht.goToVertragsdetails(0);  
  a11y.analyzeCurrentState('vertragsdetails');  
  
  fixture.logout();  
  a11y.analyzeCurrentState('ausgeloggt');  
});
```

Erkenntnisse aus den Pilot-Projekten

80-90% der Anpassungen sind immer gleich

- » UseCase-Start
- » Seiten-/Masken-Wechsel / Navigation
- » Öffnen von Overlays / Modal-Dialogen



Automatische
Checks an den
„üblichen“ Stellen

Auto-Checks

- » decken 80-90% der Standard-Triggerpunkte ab
- » integriert in zentrale Mechanik (Ergänzung des Test-Framework)
- » steuerbar global im Projekt und lokal je Testfall

```
testcase.start();  
// incl. waiting for app  
  
buttonA.navigateTo(pageB);  
// incl. spinner handling  
  
layerX.openWith(buttonY);  
// incl. waiting for modal
```

Beispiel-Code

Auto-Checks + explizite Checks

**... ergänzt durch Einzel-Prüfungen
mit Fokus auf dynamische Inhalte**

- » nachgeladene Inhalte
- » aufgeklappte Bereiche
- » Meldungen (Validierung, Toasts, ...)

Auto-Checks + explizite Checks

Beispiel für nachgeladenen Inhalt

Erhalten Sie einen Rabatt auf Ihren Beitrag



Durch unseren Kundenbonus können Sie 10 % Rabatt auf Ihren Beitrag erhalten, sofern Sie bereits ein passendes Produkt bei uns abgeschlossen haben.

Haben Sie bereits eine Versicherung bei CosmosDirekt? ⓘ

Ja

Nein

Erhalten Sie einen Rabatt auf Ihren Beitrag



Durch unseren Kundenbonus können Sie 10 % Rabatt auf Ihren Beitrag erhalten, sofern Sie bereits ein passendes Produkt bei uns abgeschlossen haben.

Haben Sie bereits eine Versicherung bei CosmosDirekt? ⓘ

Ja

Nein

Sparen Sie Beiträge mit dem Kundenbonus!

Schließen Sie im Anschluss eine passende Hausratversicherung ab. Dadurch senken Sie **beide Beiträge um 10 %**. Die Hausratversicherung erhalten Sie zum Beispiel für 127 € im Monat*.

Auto-Checks + explizite Checks

... ergänzt durch Einzel-Prüfungen
mit Fokus auf dynamische Inhalte

- » nachgeladene Inhalte
- » aufgeklappte Bereiche
- » Meldungen (Validierung, Toasts, ...)

```
// trigger UI change
beitragPage
  .vorversicherung
  .select('nein');

// analyze changed state
a11y.analyzeCurrentState();
```

Beispiel-Code

Erkenntnisse aus den Pilot-Projekten

Manche Findings werden mehrfach gefunden

- » immer komplette Seite geprüft
- » Findings an zentralen Stellen „zählen“ öfter



Scoping der
State-Checks

Scoping der State-Checks

Erhalten Sie einen Rabatt auf Ihren Beitrag



Durch unseren Kundenbonus können Sie 10 % Rabatt auf Ihren Beitrag erhalten, sofern Sie bereits ein passendes Produkt bei uns abgeschlossen haben.

Haben Sie bereits eine Versicherung bei CosmosDirekt? ⓘ

Ja

Nein

Erhalten Sie einen Rabatt auf Ihren Beitrag



Durch unseren Kundenbonus können Sie 10 % Rabatt auf Ihren Beitrag erhalten, sofern Sie bereits ein passendes Produkt bei uns abgeschlossen haben.

Haben Sie bereits eine Versicherung bei CosmosDirekt? ⓘ

Ja

Nein

Sparen Sie Beiträge mit dem Kundenbonus!

Schließen Sie im Anschluss eine passende Hausratversicherung ab. Dadurch senken Sie **beide Beiträge um 10 %**. Die Hausratversicherung erhalten Sie zum Beispiel für 1 27 € im Monat*.

Scoping der State-Checks

```
// ganze Seite
```

```
a11y.analyzeCurrentState(  
  'BeitragSeiteMitHinweis'  
);
```

Beispiel-Code

```
// nur relevanter Teil
```

```
a11y.analyzeComponent(  
  beitragePage.bonusHinweis  
);
```

Beispiel-Code

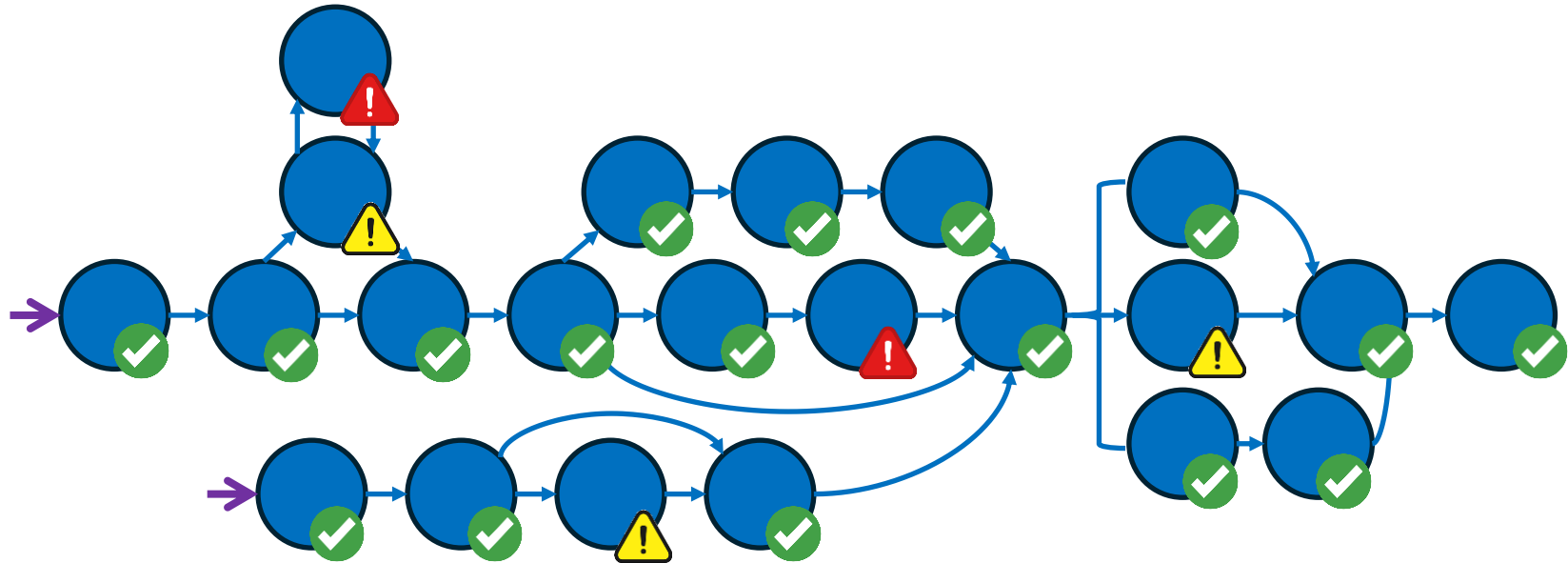
Erkenntnisse aus den Pilot-Projekten

Kein Gesamt-Ergebnis

- » viele State-Checks (mit Einzel-Ergebnissen)
- » decken nicht ab:
 - was Nutzer wirklich erleben
 - was das Business interessiert



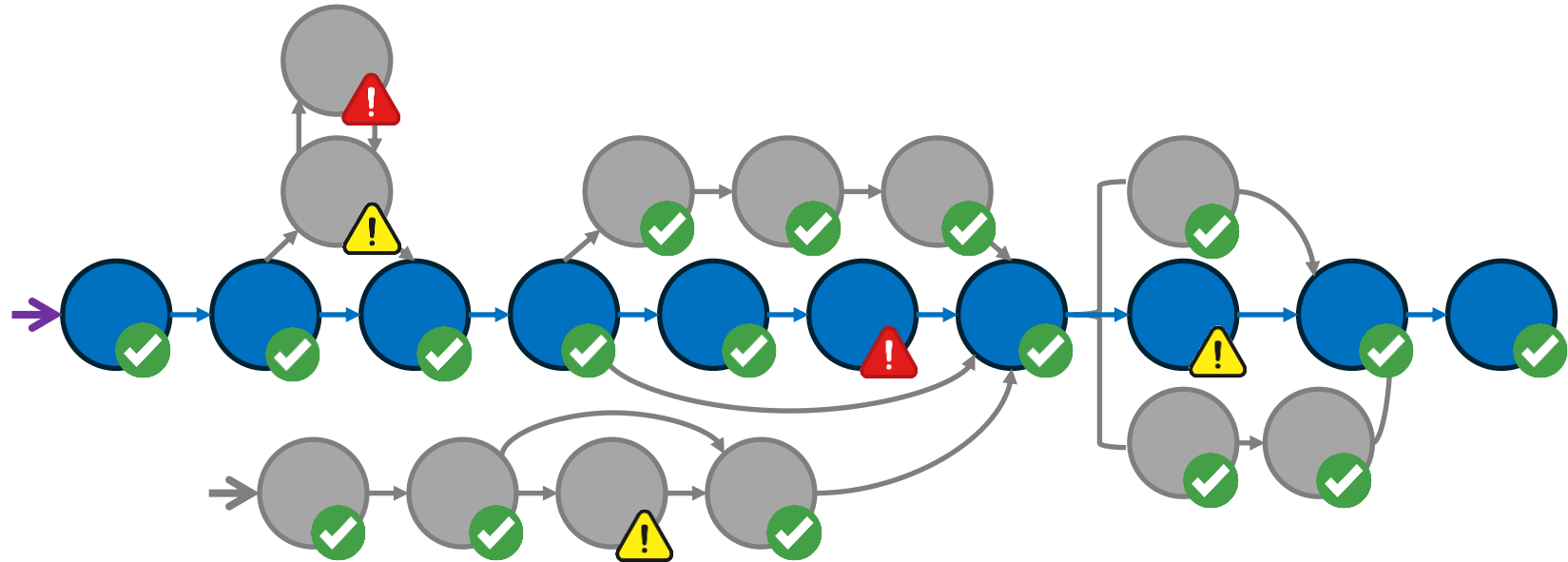
Aggregation von
Einzel-Results



Gesamt-Ergebnis aus vielen Einzel-Ergebnissen?

17x ✓ 3x ! 2x !

» UseCase-abhängig !



UseCase 1 » Gesamt-Ergebnis 



UseCase 2 » Gesamt-Ergebnis

Ergebnis » Aggregation

- » Findings sammeln
(über alle Triggerpunkte)
- » je UseCase verdichten
(1 Gesamt-Ergebnis)
- » Details bleiben drill-down-fähig

Ergebnis » Aggregation

analyzed 46 states

```
// checks
// ... by status
passes: 851,
violations: 1,
inapplicable: 1916
```

```
// violations
// ... by impact
critical: 0,
serious: 1,
moderate: 0,
minor: 0
```

```
// ... by tag
'wcag2a': 1,
'cat.keyboard': 1,

// ... by rule
'scrollable-region
-focusable': 1,
```

Beispiel-Ergebnis aus 1 UseCase

Ergebnis » Einzel-Check » Report

Axe-core® Accessibility Results for step 'HinweisLayer' (idx 41)

Page URL: </cosmos-xy-antrag-app/#/antrag/13-hinweise>

axe-core found **1** violation

#	Description	Axe rule ID	WCAG	Impact	Count
1	Scrollable region must have keyboard access	scrollable-region-focusable	WCAG 2 Level A, WCAG 2.1.1, WCAG 2.1.3	serious	1

Failed

1. Scrollable region must have keyboard access

[Learn more](#)

scrollable-region-focusable

WCAG 2 Level A, WCAG 2.1.1, WCAG 2.1.3

Ensure elements that have scrollable content are accessible by keyboard

serious

Issue Tags: `cat.keyboard` `wcag2a` `wcag211` `wcag213` `TTv5` `TT4.a` `EN-301-549` `EN-9.2.1.1` `EN-9.2.1.3`

#	Issue Description	To solve this violation, you need to...
1	Element location <code>.modal-body</code> Element source <code><div _ngcontent-ng-c1221586233="" class="modal-body"></code>	Fix any of the following: • Element should have focusable content • Element should be focusable

Herausforderungen:

Lange UseCases:

Masse von
Einzel-Checks

+ unerreichbare
Stellen

nur bedingt
CI/CD-tauglich

schnelles
Feedback
VS.
breite
Abdeckung

Warum sind A11Y-Tests nicht CI/CD-tauglich?

zu „lax“

- Check-Ergebnisse nur als Reports
- Kein Einfluss aufs Ergebnis
- Manuelles Review notwendig

zu „streng“

- Erwartungshaltung: 0 Findings
- TestRuns: immer rot
-> Ergebnisse werden ignoriert
- CI/CD: jedes kleinste Finding bricht die (Release-) Pipeline



» kein Vertrauen, keine Akzeptanz

Erkenntnis

Bedarf zur Nutzung der A11Y-Tests in Pipelines

- » Akzeptanz im Team schaffen
- » Mittelweg zwischen „zu lax“ und „zu streng“



anpassbare
Fail-Policy

Fail-Policy

```
// by impact  
critical  
serious  
moderate  
minor
```

```
// by tag  
wcag2a  
wcag2aa  
  
// by rule  
color-contrast  
heading-order  
meta-viewport  
...
```

- Erwartungshaltung
anpassbar an Bedarf und
Reifegrad
- mehrere „Dimensionen“

~~zu „lax“ / zu „streng“~~
» **genau richtig**

Fail-Policy » mehrstufig

global

- » App-übergreifend
- » zentral gepflegt

```
critical: 0,  
serious: 1,  
moderate: 5,  
minor: 10,
```

einfache Beispiel Policy

je App

- » Anpassung an App-Spezifika

je Testfall

- » Vergleich mit Ist-Zustand
(nicht schlechter werden)

je Check

- » Einschränkung einzelner Regeln
(„won't fix“)

Bonus

Herausforderungen:

Lange UseCases:

Masse von
Einzel-Checks

+ unerreichbare
Stellen

nur bedingt
CI/CD-tauglich

schnelles
Feedback
vs.
breite
Abdeckung

Erkenntnis

Widersprüchliche Ziele

» breite Abdeckung » trotzdem schnelles Feedback

Test-Vorgehen muss angepasst werden

Hybride Strategie
zur Ausführung
der A11Y-Tests

Hybride Strategie » 2 Modi

„always on“

- automatisch Teil von jedem funktionalen Test-Run
- wenige, kritische UseCases
- A11Y-Modus „*lokal*“
(für einzelnen Testfall)

» Fast Feedback

A11Y-Fokus

- extra gestarteter Test-Run mit Fokus auf A11Y
- mit vollständigem A11Y-Testset
- A11Y-Modus „*global*“
(für alle Testfälle im Test-Run)

» breite Abdeckung

Hybride Strategie » Anwendung

Vorgehen	„always on“ » Fast Feedback	A11Y-Fokus » breite Abdeckung
Scrum (2-Wochen-Sprints)	zu jedem Test-Stage-Deployment (z.B. 1-2x täglich oder als Nightly-Run)	wöchentlich (z.B. Mid-Sprint und Pre-Release)
DevOps (continuous)	„on push“ (Teil der CI/CD-Pipeline) Fast Feedback, alles Wichtige abdecken	zyklisch (z.B. Nightly-Run) nicht produktions-verhindernd, aber trotzdem Fehler finden

**Empfehlung**

Zyklische Ausführung » Automatisierung

(statt „dran denken müssen“)

Cron-Triggered Jobs

- A11Y-Job je App mit fester Ausführungszeit
- Dev-Team: Verantwortung über Umsetzung, Review, etc.

Zentraler Service

- Zentral organisierter A11Y-Trigger
- Dev-Team: A11Y-Config je App
- Automatische Info der Ergebnisse ans Team

FAZIT: A11Y-Automation ist skalierbar

~~Lange UseCases:
Masse von Einzel-Checks
+ un erreichbare Stellen~~

**UseCase-
Ergebnisse
+ Integration in
funktionale Tests**

~~nur bedingt
CI/CD-tauglich~~

**aussagekräftige
Nutzung in CI/CD
durch
Fail-Policies**

~~schnell **VS.** breit~~

**schnell UND breit
durch hybride
Strategie**