HOCHSCHULE
DER MEDIEN

Testverfahren und -werkzeuge für barrierefreie Apps auf Android

Webinar
Computer Science and Media
Pirmin Gersbacher
20.05.2021

Hinweis zu den Grafiken in dieser Präsentation

Die Screenshots und Grafiken, die in dieser Präsentation enthalten sind, werden jeweils in einem YouTube-Video beschrieben. Das zugehörige YouTube-Video ist mit dem Screenshot verlinkt, so dass es mit einem Klick auf den Screenshot gestartet werden kann.

Agenda

1. Europäische Norm EN 301 549
2. Barrierefreiheitsprüfung
3. Analyse Tools
4. Automatisierte Testverfahren
5. Empirischer Vergleich
6. Zusammenfassung und Fazit

Europäischer Standard EN 301 549

Europäischer Standard EN 301 549

- Harmonisierte Norm im Rahmen der EU-Richtlinie für barrierefreie Inhalten im Web und bei Apps
- Erstellt von Europäischen Institut für Telekommunikationsnormen (ETSI)
- Hilft EU-Mitgliedstaaten elektronische Services barrierefrei zugänglich zu machen
- Spezifiziert die funktionalen Zugänglichkeitsanforderungen an ICT-Produkte und -Dienstleistungen

Europäischer Standard EN 301 549

- Beschreibung von Testverfahren und Evaluations-Methodiken
- Freiwilliges Referenzdokument zur Erfüllung der grundlegenden Anforderungen der EU-Richtlinie 2016/2102
- 162 Zugänglichkeitsanforderungen für mobile Anwendungen in verschiedenen Prioritätsstufen
- Beinhaltet u.a. Alle Erfolgskriterien aus WCAG 2.1 (Komformitätslevel AA)

Europäischer Standard EN 301 549

Auf EN 301 549 wird verwiesen in

- EU Richtlinie 2016 / 2102 (Zugänglichkeit mobile Anwendung öffentlicher Stellen)
- BITV 2.0 (Barrierefreie Informationstechnik Verordnung)

Zukünftig (2025):

- European Accessibility Act (Zugänglichkeit mobiler Anwendungen aller Wirtschaftsakteure)

Barrierefreiheitsprüfung von Apps

Barrierefreiheitsprüfung von Apps

Unterscheidung nach:

- User Tests
- Manuelle Testverfahren
- **Testen mit Analyse Tools**
- **Automatisierte Testverfahren**

User Tests

- Echte Nutzer mit Behinderungen für Evaluation eingesetzt
- Spezifischer Einblick & gründliche Evaluierung
- Schwierigkeiten Tester zu finden
- Ergebnisse können variieren

Manuelle Testverfahren

- Tester nimmt Rolle des Nutzers ein
- Tester muss mit allen Zugänglichkeitsanforderungen vertraut sein
- Gründliche Validierung
- Zeitaufwändig
- Ergebnisse können variieren
- Hilfsmittel (z.B. BIK-App-Test)

Analyse Tools

- Meist in Form von Apps
- Unterstützen Tester indem sie Testvorgang automatisieren
- Designer- / endnutzerbasiert
- Geben schnelles Feedback

Automatisierte Testverfahren

- Keine manuelle Erkundung notwendig
- Entwicklerbasiert
- Anzeige von Barrierefreiheitsverletzungen direkt im Code

  → Benötigt Zugriff auf Source Code
- Integration in Entwicklungsprozess
- Zusätzliche Unterscheidung in Statische Code Analyzer

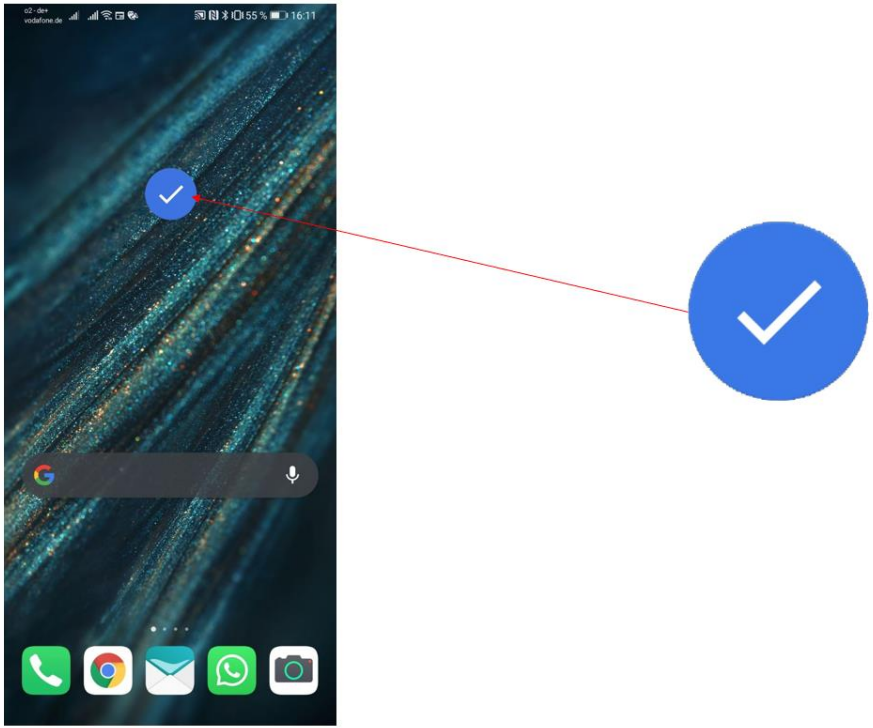
  & automatisierte UI-Tests

Analyse Tools

Accessibility Scanner

- Herausgeber: Google
- App

Accessibility Scanner - Funktionsweise



Accessibility Scanner - Weitere Funktionen

- Share Mechanismus (Textfile und Screenshot)
- Aufnahme Session

Accessibility Scanner - Abdeckung

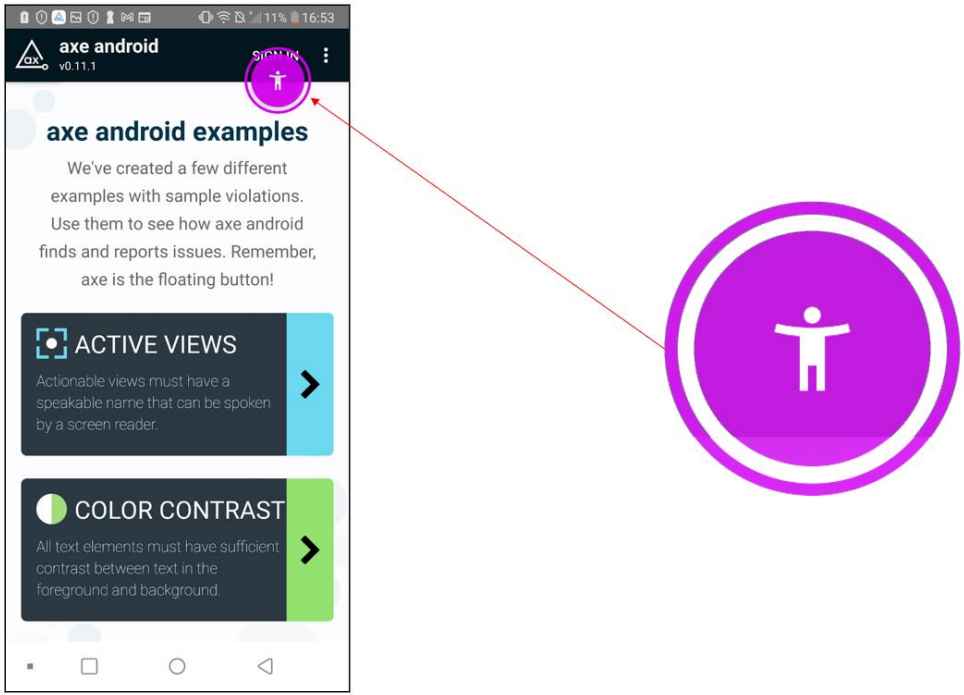
Überprüfung von:

- Inhaltslabels
- Label Duplikate
- Zielelementgröße
- Textkontrast
- Bildkontrast

axe for Android

- Deque Systems
- App

axe for Android - Funktionsweise



axe for Android - Abdeckung

Überprüft:

- Inhaltslabels
- Zielelementgröße
- Textkontrast
- Bildkontrast
- Fokusmanagement

Accessibility Insights for Android

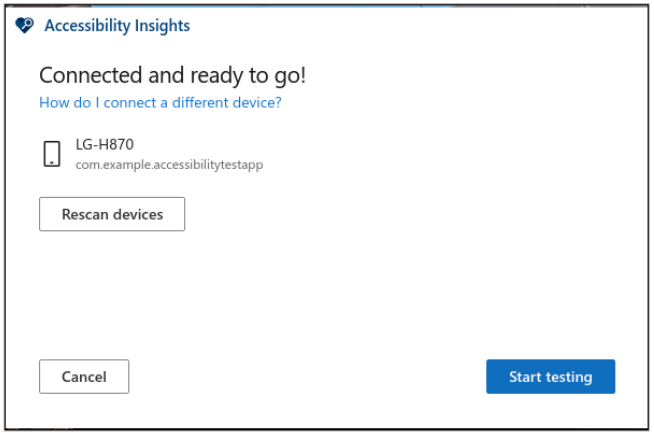
- Herausgeber: Microsoft
- Desktop Anwendung

  https://accessibilityinsights.io/en/downloads/

Accessibility Insights - Funktionsweise

Testgerät (Smartphone):

Computer:



Accessibility Insights - Abdeckung

- Gleiche Accessibility Bibliothek wie axe Android

Inspector Ally

- Im Rahmen der Masterarbeit erstellt
- Inspector App - Hilfstool
- Vorbild: Seiteninspektoren bei Browser
- Ziel: Einsatz von Screenreader beim Testen reduzieren

Inspector A11y - Funktionsweise

Automatisierte Testverfahren

Android Lint

- Herausgeber: Google
- Statische Code Analyse auf strukturelle Probleme
- Überprüft Quellcode auf potenzielle Fehler und macht Vorschläge hinsichtlich Sicherheit, Performance, Usability und Barrierefreiheit

Android Lint

- In Android Studio bereits integriert
- Alternativ: Grade Wrapper
- *File (Rechtsklick)* > *Analyze* > *Inspect Code*
- Code Analysiert und Ergebnisse in verschiedene Kategorien aufgelistet (Inspection Results)

Android Lint -Funktionsweise

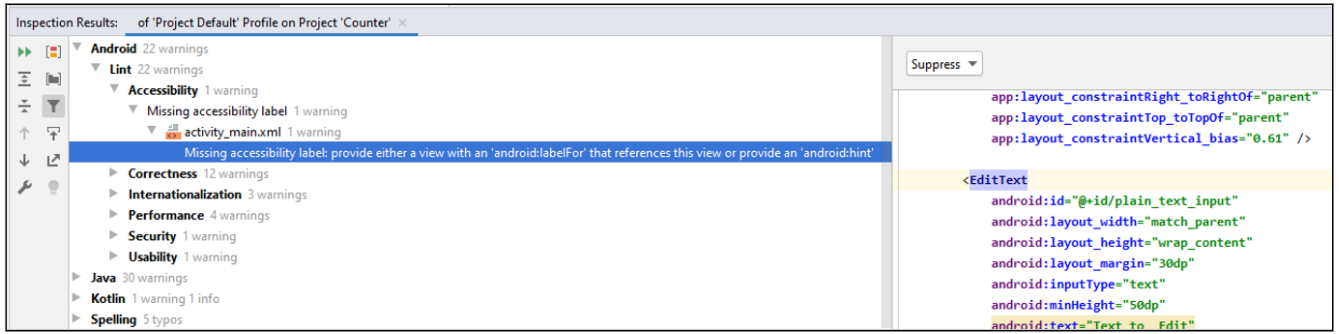


Image without contentDescription inspection

Image without `contentDescription`

Non-textual widgets like ImageViews and ImageButtons should use the `contentDescription` attribute to specify a textual description of the widget screen readers and other accessibility tools can adequately describe the user int

Note that elements in application screens that are purely decorative and do not any content or enable a user action should not have accessibility content descri this case, just suppress the lint warning with a tools:ignore="ContentDescription

Note that for text fields, you should not set both the `hint` and the `contentDescription` attributes since the hint will never be shown. Just set th See http://developer.android.com/guide/topics/ui/accessibility/checklist.html#spec

Issue id: ContentDescription

Android Lint - Abdeckung

- Statische Code Analyse

  Aber:

- Inhalte und Komponenten oft dynamisch erzeugt

  z.B. Dynamische Eigenschaften wie Größe und Kontrast

- Kann nur Eigenschaften während Entwicklung überprüfen
  - Fehlende Inhaltsbeschreibung
  - Fehlende Beschriftungen Formularfelder

Espresso

- Herausgeber: Google
- UI-Test Framework
  - UI-Test Automatisierung
  - Testen der UI um sicherzustellen, dass Spezifikationen erfüllt

Espresso

- Barrierefreiheitstests in Testsuite integrieren
- Verbesserung der Barrierefreiheit in der Entwicklungsphase
- Für Zugänglichkeitsprüfung benötigt zusätzlich Accessibility Bibliothek
- Einfache Integration des Accessibility Framework for Android (ATF)

ATF

- Java-basierte Bibliothek
- Verwendet “Check-based” System
- Jeder Check bewertet Aspekt der UI für Verbesserung der Barrierefreiheit
- Open Source

  https://github.com/google/Accessibility-Test-Framework-for-Android

Integration von ATF in Espresso

androidTestimplementation in build.gradle hinzufügen:

```
androidTestImplementation 'androidx.test.espresso:espresso-accessibility:3.3.0-alpha05'
```

Import der Klasse in Testfile:

```
import androidx.test.espresso.accessibility.AccessibilityChecks
```

Befehl in Testfile um Accessibility Checks zu aktivieren:

```
AccessibilityChecks.enable()
```

Bzw. zur Betrachtung der kompletten View Hierarchie:

```
AccessibilityChecks.enable().setRunChecksFromRootView(true)
```

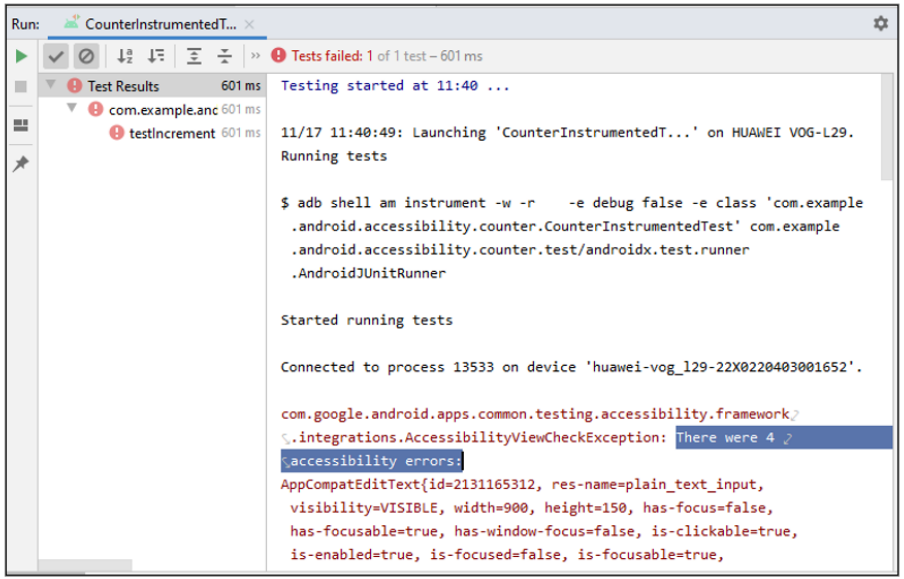
Espresso - Link zum Codelab

https://developer.android.com/codelabs/a11y-testing-espresso#0https://developer.android.com/codelabs/a11y-testing-espresso

Espresso

- Integriert in Entwicklungsprozess und bestehende UI-Tests
- Benötigt Virtuelles Device oder Smartphone (dynamische Inhalte)
- Probleme erkennen bevor sie in eigentliche App gelangen
- Gut skalierbar

Espresso - Funktionsweise



Espresso und ATF - Abdeckung

- Inhaltslabels
- Zielelementgröße
- Textkontrast
- Bildkontrast

Empirischer Vergleich

Empirischer Vergleich

- Bisher: Bewertung anhand Herstellerangaben
- Jetzt: Empirische Untersuchung anhand einer Testanwendung
- Ziel:
  - Überprüfen inwieweit Kriterien der EU Norm EN 301 549 erfüllt werden
  - An welchen Stellen können Tools unterstützend eingesetzt werden
  - Welche Kriterien manuell überprüft werden müssen

Empirischer Vergleich - Untersuchte Kriterien

- 11.1.1.1a Alternativtexte für Bedienelemente
- 11.1.1.1b Alternativtexte für Grafiken und Objekte
- 11.2.5.5 Berührungszielbereichsgröße
- 11.1.4.3 Kontraste von Texten ausreichend
- 11.1.4.6 Kontraste von Texten erweitert
- 11.1.4.11 Kontraste von Grafiken und Bedienelementen
- 11.3.3.2 Beschriftungen von Formularelementen vorhanden
- 11.1.3.1 Lineare Navigation

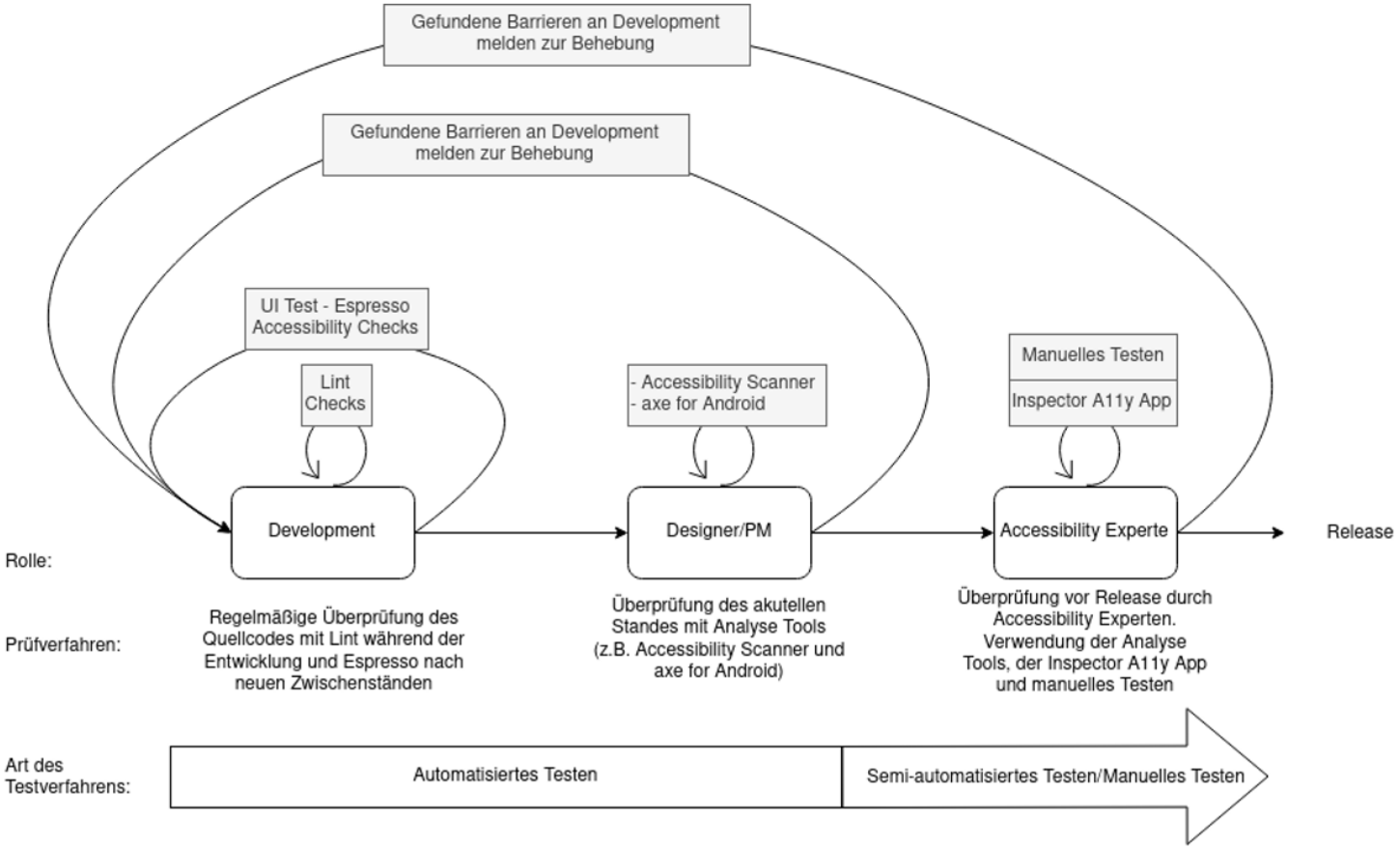
EN 301 549	Evaluations-Kriterium	Accessibility Scanner	Accessibility Engine (axe) for Android	Accessibility Insights for Android	Lint	Espresso und ATF
11.1.1.1a	Alternativtexte für Bedienelemente	Teilweise bestanden	Teilweise bestanden	Teilweise bestanden	Teilweise bestanden	Teilweise bestanden
11.1.1.1b	Alternativtexte für Grafiken und Objekte	Teilweise bestanden	Teilweise bestanden	Teilweise bestanden	Teilweise bestanden	Teilweise bestanden
11.2.5.5	Target Size	Bestanden	Bestanden	Bestanden	Nicht geprüft	Teilweise bestanden
11.1.4.3	Kontraste von Texten ausreichend	Bestanden	Teilweise bestanden	Nicht bestanden	Nicht geprüft	Teilweise bestanden
11.1.4.6	Kontraste von Texten ausreichend (erweitert)	Bestanden	Nicht geprüft	Nicht geprüft	Nicht geprüft	Nicht geprüft
11.1.4.11	Kontraste von Grafiken und Bedienelementen ausreichend	Bestanden	Nicht geprüft	Nicht geprüft	Nicht geprüft	Nicht geprüft
11.1.3.2	Sinnvolle Reihenfolge	Nicht geprüft	Nicht geprüft	Nicht geprüft	Nicht geprüft	Nicht geprüft
11.1.3.1e	Beschriftungen von Formularelementen programmatisch ermittelbar	Nicht bestanden	Bestanden	Nicht bestanden	Teilweise bestanden	Nicht bestanden

Schluss

Zusammenfassung Vorteile der Testverfahren

Accessibility Scanner	axe for Android	Accessibility Insights	Espresso	Android Lint
• Größte Abdeckung • Einfache Bedienung	• Einfache Bedienung	• Anbindung an GitHub-Repository	• Integration in bestehende UI-Tests • Gut skalierbar	• Integriert in Android Studio • Schnelle Überprüfung auf statische Probleme

Szenario


Gefundene Barrieren an Development melden zur Behebung
Gefundene Barrieren an Development melden zur Behebung
UI Test - Espresso Accessibility Checks
Lint Checks
- Accessibility Scanner
- axe for Android
Manuelles Testen
Inspector A11y App
Development
Designer/PM
Accessibility Experte
Release
Rolle:
Prüfverfahren:
Regelmäßige Überprüfung des Quellcodes mit Lint während der Entwicklung und Espresso nach neuen Zwischenständen
Überprüfung des akutellen Standes mit Analyse Tools (z.B. Accessibility Scanner und axe for Android)
Überprüfung vor Release durch Accessibility Experten. Verwendung der Analyse Tools, der Inspector A11y App und manuelles Testen
Art des Testverfahrens:
Automatisiertes Testen
Semi-automatisiertes Testen/Manuelles Testen

Zusammenfassung

Vorteile

- Schneller Überblick
- Größere Datenmenge in kürzester Zeit
- Konsistent
- Unterstützung von Personen mit weniger Expertise
- Decken einige der häufigsten Verletzungen auf

Nachteile

- Decken nur kleine Teilmenge der Kriterien von EN 301 549 ab
- Teilweise unvollständige Prüfung
- Anwender könnten sich in falscher Sicherheit wiegen

Fazit

- Ohne manuelles Testen ist keine vollumfängliche Überprüfung einer App auf Barrierefreiheit möglich
- Auch wenn viele Kriterien nicht untersucht werden so sind die untersuchten Elemente (Text, Bilder, Buttons, Formularfelder,...) die am häufigsten vorkommenden Elemente

Fazit

- Anwendungen werden ständig weiterentwickelt
- Zukünftig durch KI weitere Tests und automatische Fehlerbehebung möglich
- Aber auch jetzt schon regelmäßig Funktionalitäten ergänzt
- Dem Tester wird Arbeit abgenommen und Barrieren aufgezeigt
- Können Grundlage für umfassendere Teststrategien bilden

Referenzen (1)

- Amtsblatt der Europäischen Union. (2012). *VERORDNUNG (EU) Nr. 1025/2012 DES EUROPÄISCHEN PARLAMENTS UND DES RATES*. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2012:316:0012:0033:DE:PDF
- Android Developers. (2020a). *Automated Accessibility Testing using Espresso*. https://developer.android.com/codelabs/a11y-testing-espresso?index=.%2F.index#2
- Android Developers. (20. April 2020b). *Test your app's accessibility*. https://developer.android.com/guide/topics/ui/accessibility/testing
- Android Developers. (28. April 2020c). *Build more accessible apps*. https://developer.android.com/guide/topics/ui/accessibility
- Android Developers. (18. Februar 2021). *AccessibilityService*. https://developer.android.com/reference/android/accessibilityservice/AccessibilityService
- Bundesamt für Justiz. (19. September 2020b). *§ 1 BITV 2.0 - Einzelnorm: Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (Barrie-refreie-Informationstechnik-Verordnung - BITV 2.0) § 1 Ziele*. https://www.gesetze-im-internet.de/bitv_2_0/__1.html

Referenzen (2)

- Deque. (29. Oktober 2020). *Android Accessibility Tools & Audits | Deque*. https://www.deque.com/android-accessibility/
- Deque Systems, I. (9. März 2021). *Accessibility Engine (axe) for Android*. https://play.google.com/store/apps/details?id=com.deque.axe.android
- Dohmke, T. (10. März 2020). Accessibility Insights for Android is here. *Open Source Blog*. https://cloudblogs.microsoft.com/opensource/2020/03/10/accessibility-insights-android-now-available/
- ETSI (2020). *EN 301 549 V3.2.1: Accessibility requirements for ICTproducts and services* (Harmonised European Standard). https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_20/en_301549v030201a.pdf
- Richtlinie (EU) 2016/2102 des Europäischen Parlaments und des Rates (2016). https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX%3A32016L2102
- Directive (EU) 2019/882 of the European Parliament and of the Council, 2019 (2019). https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32019L0882&from=DE