Designing an Accessibility Learning Toolkit

Bridging the Gap Between Guidelines and Implementation

Gabriel Rovesti (ID: 2103389)
July 2025

Table of Contents

Outline

Research Context & Motivation

The Scale

- 1+ billion people with disabilities globally
- 7 billion mobile device users
- Explosive growth in mobile applications

The Problem

- Abstract WCAG guidelines exist
- Limited practical implementation guidance
- Gap between theory and mobile development

Evidence

- 22/57 EU public apps fail accessibility
- 30% of Android apps have accessibility issues
- Developers struggle with implementation

Core Challenge

How do we bridge the gap between abstract accessibility guidelines and concrete mobile implementation?

Outline

Università degli Studi di Padova

Research Questions & Contributions

Primary Research Questions

1. How can we systematically evaluate accessibility implementation across frameworks?
2. What design patterns optimize both accessibility and developer experience?
3. How effective are theory-informed educational approaches?

Research Contributions

- Novel evaluation framework with 6 formal metrics
- First systematic quantitative comparison of React Native vs Flutter
- AccessibleHub: Open-source learning toolkit

Innovation

Theory-Practice Bridge

- WCAG2Mobile mapping
- Quantitative metrics
- Educational design theory

Outline

Methodological Innovation

Six Formal Accessibility Metrics

1. **CAS**: Component Accessibility Score
2. **WCR**: WCAG Compliance Rate
3. **SRSS**: Screen Reader Support Score
4. **IMO**: Implementation Overhead
5. **API**: Accessibility API Coverage
6. **DTE**: Development Time Efficiency

WCAG2Mobile Integration

- Direct mapping: WCAG 2.1 AA success criteria $\rightarrow$ mobile components
- Weighted scoring based on impact and frequency
- Cross-platform consistency validation

Formula

$$CAS = \frac{\sum_{i=1}^{n} w_i \cdot c_i}{\sum_{i=1}^{n} w_i}$$

$$WCR = \frac{|S_p|}{|S_t|} \times 100\%$$

where S_p = passed criteria, S_t = total criteria

Outline

AccessibleHub: Research Vehicle

Educational Design Principles

- **Scaffolded Learning**: Progressive complexity
- **Theory-Practice Integration**: WCAG ↔ Code
- **Multi-Modal**: Visual + Audio + Hands-on
- **Community-Centered**: Social learning

Technical Architecture

- React Native foundation
- Expo development workflow
- Cross-platform deployment
- Open-source MIT license

Core Features

- 20+ Interactive components
- WCAG 2.1 AA compliance
- Screen reader optimized
- Real-time testing
- Community contributions

Availability

- GitHub: gabrielrovesti/AccessibleHub
- Documentation & guides
- Vid

Outline

Implementation Examples & Patterns

React Native Button

```
<TouchableOpacity
  accessibilityRole="button"
  accessibilityLabel="Submit form"
  accessibilityHint="Validates and submits the current form
        "
  onPress={handleSubmit}>
  <Text>Submit</Text>
</TouchableOpacity>
```

Key Patterns

- Explicit semantic roles
- Descriptive labels
- Action hints
- Focus management

Flutter Button

```
Semantics(
  label: 'Submit form',
  button: true,
  onTap: () => handleSubmit(),
  child: ElevatedButton(
    onPressed: handleSubmit,
    child: Text('Submit'),
  ),
)
```

Architecture Differences

- Property-based vs Widget-based
- Implicit vs Explicit semantics
- Platform integration approaches

Outline

Università degli Studi di Padova

Quantitative Results

AccessibleHub Performance

- **CAS**: 100% component implementation (20/20)
- **WCR**: 88% WCAG 2.1 AA compliance
- **SRSS**: 4.3/5.0 average (VoiceOver + TalkBack)
- **IMO**: 23.3% average implementation overhead

Framework Comparison Results

- React Native: 38% default accessible components
- Flutter: 32% default accessible components
- React Native: 45% less implementation code
- Screen reader consistency: React Native advantage

Key Metrics

Metric	Score
CAS	100%
WCR	88%
SRSS	4.3/5.0
IMO	23.3%

Outline

Università degli Studi di Padova

Conclusions

Research Contributions

1. **Methodological**: Novel evaluation framework with 6 formal metrics
2. **Practical**: AccessibleHub toolkit with theory-informed design
3. **Empirical**: First systematic quantitative framework comparison

Future Research Directions

- Framework expansion: SwiftUI, Jetpack Compose
- User studies: Developer effectiveness measurement
- Automation: CI/CD pipeline integration
- Community: Open source ecosystem development

Impact

Theory-Practice Bridge

- Quantitative accessibility metrics
- Open-source educational toolkit
- Evidence-based framework decisions

Thank You

Questions & Discussion

Gabriel Rovesti
gabriel.rovesti@studenti.unipd.it
GitHub: gabrielrovesti/AccessibleHub