

# Master Thesis Presentation Structure & Alternatives

## "Designing an Accessibility Learning Toolkit: Bridging the Gap Between Guidelines and Implementation"

**Duration:** 20 minutes | **Total Slides:** 14-16 | **Format:** Academic presentation for CS commission

---

### RECOMMENDED CORE STRUCTURE

#### Slide 1: Title Slide

##### Content:

- **Title:** Designing an Accessibility Learning Toolkit: Bridging the Gap Between Guidelines and Implementation
  - **Student:** Gabriel Rovesti (ID: 2103389) | **Supervisor:** Prof. Ombretta Gaggi
  - **University of Padua, Department of Mathematics "Tullio Levi-Civita" | July 2025**
- 

#### Slide 2: Research Context & Motivation

**Time:** 2 minutes

##### OPTION A: Problem-Focused (Recommended)

- **Global Scale:** 1+ billion people with disabilities, 7 billion mobile users
- **Implementation Gap:** WCAG guidelines exist but practical implementation lacks guidance
- **Evidence:** 22/57 public apps fail accessibility (EU studies), 30% of Android apps have accessibility issues
- **Core Problem:** Abstract guidelines → Concrete implementation challenge

##### OPTION B: Technology-Focused

- **Mobile Revolution:** Explosive growth in mobile development frameworks
- **Accessibility Challenge:** Cross-platform frameworks need accessibility guidance
- **Current State:** Limited systematic comparison of accessibility implementation approaches

**Visuals:** Statistics infographic + before/after mobile interface examples

---

#### Slide 3: Research Questions & Contributions

**Time:** 2 minutes

##### OPTION A: Question-Driven (Recommended)

- **Three Research Questions:**
  - RQ1: Are framework components accessible by default?
  - RQ2: Can non-accessible components be made accessible?
  - RQ3: What is the development overhead for accessibility implementation?
- **Novel Approach:** First systematic quantitative comparison with formal metrics

#### OPTION B: Contribution-Driven

- **Three Main Contributions:**
  1. Formal evaluation framework with 6 quantitative metrics
  2. AccessibleHub: Theory-informed educational toolkit
  3. Systematic framework comparison with empirical validation

**Visuals:** Research questions flowchart OR contributions overview diagram

---

### Slide 4: Methodological Innovation

**Time:** 3 minutes

#### OPTION A: Metrics-Focused (Recommended)

- **Novel Evaluation Framework:** 6 formal metrics with mathematical definitions
  - Component Accessibility Score (CAS): % components accessible by default
  - Implementation Overhead (IMO): Additional LOC required for accessibility
  - Complexity Impact Factor (CIF): Weighted complexity assessment
  - Screen Reader Support Score (SRSS): Empirical testing (1-5 scale)
  - WCAG Compliance Ratio (WCR): % criteria satisfied by conformance level
  - Developer Time Estimation (DTE): Time-to-implement accessibility features

#### OPTION B: Standards-Focused

- **WCAG2Mobile Integration:** First implementation of W3C 2025 mobile accessibility standard
- **Methodological Rigor:** Formal metrics + empirical testing + systematic comparison
- **Reproducible Framework:** Documented methodology for future framework evaluation

#### OPTION C: Theory-Focused

- **Educational Foundation:** Constructivist + experiential learning theory integration
- **Quantitative Framework:** Mathematical formalization of accessibility evaluation
- **Empirical Validation:** VoiceOver + TalkBack testing on specific device configurations

**Visuals:** Metrics definition table + methodology flowchart

---

## Slide 5: AccessibleHub: Research Vehicle & Educational Tool

**Time:** 4 minutes

### OPTION A: Architecture-Focused (Recommended)

- **Purpose:** Interactive toolkit bridging WCAG theory → mobile implementation
- **7 Educational Modules:**
  - Components (progressive complexity), Best Practices (domain-specific)
  - Framework Comparison (analytical tool), Tools (workflow integration)
  - Settings (adaptive interface), Community (social learning)
- **Pedagogical Design:** Theory-informed learning progression + hands-on examples

### OPTION B: Feature-Focused

- **Interactive Learning:** Live code examples + copy-paste implementation patterns
- **Quantitative Dashboard:** Real-time accessibility metrics + compliance tracking
- **Comparative Analysis:** Side-by-side React Native vs Flutter implementation
- **Community Integration:** Connection to broader accessibility development ecosystem

**Visuals:** AccessibleHub home screen + architecture diagram + sample component screen

---

## Slide 6: Implementation Examples & Patterns

**Time:** 3 minutes

### OPTION A: Code-Focused (Recommended)

- **Progressive Complexity:** Buttons → Forms → Dialogs → Advanced components
- **Implementation Patterns:**

javascript

*// React Native: Property-based*

```
<TouchableOpacity
  accessibilityRole="button"
  accessibilityLabel="Submit form"
  accessibilityHint="Activates form submission">
```

*// Flutter: Widget-based*

```
Semantics(
  label: 'Submit form',
  button: true,
  child: ElevatedButton(...))
```

### OPTION B: Educational-Focused

- **Learning Pathway:** Scaffolded component complexity with theoretical grounding
- **WCAG Mapping:** Each component explicitly connected to success criteria
- **Multi-Modal Learning:** Visual examples + code patterns + accessibility outcomes

**OPTION C: Pattern-Focused**

- **Architectural Differences:** Property-based vs Widget-based accessibility models
- **Cross-Platform Consistency:** How frameworks handle platform-specific requirements
- **Implementation Efficiency:** Code patterns that minimize development overhead

**Visuals:** Side-by-side code comparison + component screenshots + WCAG mapping

---

**Slide 7: Quantitative Results Overview**

**Time:** 2 minutes

**OPTION A: Metrics Dashboard** (Recommended)

- **AccessibleHub Results:**
  - CAS: 100% component implementation (20/20)
  - WCR: 88% WCAG 2.2 compliance (weighted across A/AA/AAA)
  - SRSS: 4.3/5.0 average (VoiceOver + TalkBack testing)
  - IMO: 23.3% average implementation overhead

**OPTION B: Framework Comparison**

- **Default Accessibility:** React Native 38% vs Flutter 32%
- **Implementation Efficiency:** React Native 45% less code required
- **Cross-Platform Consistency:** React Native more consistent screen reader behavior

**Visuals:** Metrics dashboard screenshot + key numbers highlighted

---

**Slide 8: Framework Comparison Results**

**Time:** 3 minutes

**OPTION A: Comprehensive Comparison** (Recommended)

Metric	React Native	Flutter	Difference
Default Accessible Components	38%	32%	+6%
Implementation Overhead (LOC)	Baseline	+45%	React Native advantage
Screen Reader Support	4.2/5.0	3.8/5.0	+10.5%
WCAG AA Compliance	92%	85%	+8.2%

**OPTION B: Architectural Focus**

- **Implementation Models:** Property-based vs Widget-based approaches
- **Code Efficiency:** React Native requires 27%-200% less code for equivalent functionality
- **Platform Integration:** Different approaches to iOS/Android accessibility API integration

**OPTION C: Developer Experience Focus**

- **Learning Curve:** React Native more accessible to web developers
- **Implementation Time:** DTE measurements show React Native 35% faster
- **Maintenance Overhead:** Flutter's explicit semantics aid long-term maintenance

**Visuals:** Comparison table + architectural diagrams + code overhead chart

---

**Slide 9: Implementation Overhead Analysis**

**Time:** 2 minutes

**OPTION A: Detailed Breakdown** (Recommended)

- **Overhead Range:** 8.7% (simple content) to 36.3% (complex navigation)
- **Complexity Drivers:** Interaction patterns > Visual complexity
- **Key Contributors:** Focus management, element hiding, state communication
- **Framework Difference:** React Native consistently requires less code

**OPTION B: Practical Implications**

- **Resource Planning:** 20-25% additional development time for comprehensive accessibility
- **ROI Analysis:** Implementation cost vs user base expansion benefits
- **Team Considerations:** Framework choice impacts development velocity

**Visuals:** Overhead breakdown chart + complexity factors visualization

---

**Slide 10: Screen Reader Testing & Validation**

**Time:** 2 minutes

**OPTION A: Empirical Results** (Recommended)

- **Testing Methodology:** VoiceOver (iOS 16) + TalkBack (Android 15) on specific devices
- **Results:** React Native 4.2/5.0 average, Flutter 3.8/5.0 average
- **Platform Variance:** React Native more consistent cross-platform behavior
- **Key Findings:** Property-based approach translates better to assistive technologies

**OPTION B: Validation Approach**

- **Multi-Platform Testing:** Systematic evaluation across iOS/Android
- **Real-World Usage:** Testing with actual screen reader users and scenarios
- **Compliance Verification:** Mapping implementation to WCAG2Mobile requirements

**Visuals:** Testing results table + platform comparison chart + sample screen reader behavior

---

## Slide 11: Extended Accessibility Principles

**Time:** 2 minutes

### OPTION A: Beyond WCAG (Recommended)

- **Novel Principles:** 10+ extended principles beyond standard WCAG requirements
- **Mobile-Specific:** Touch optimization, swipe efficiency, battery considerations
- **Educational:** Pedagogical accessibility principles for developer learning
- **Social:** Community-centered accessibility development approaches

### OPTION B: Theoretical Contributions

- **Quantitative Measurement:** First formal metrics for accessibility implementation
- **Educational Framework:** Theory-informed approach to accessibility learning
- **Implementation Patterns:** Reusable code patterns for common accessibility challenges

### OPTION C: Skip This Slide (If time is tight)

- Integrate key points into other slides

**Visuals:** Extended principles overview + examples of mobile-specific considerations

---

## Slide 12: Research Impact & Practical Implications

**Time:** 2 minutes

### OPTION A: Multi-Stakeholder Impact (Recommended)

- **For Developers:** Evidence-based framework selection + realistic resource planning
- **For Project Managers:** Quantified implementation costs + ROI justification
- **For Educators:** Proven interactive learning approach + curriculum integration
- **For Researchers:** Reproducible evaluation methodology + formal metrics

### OPTION B: Industry Impact

- **Framework Selection:** Data-driven guidance for accessibility-focused projects
- **Resource Planning:** Realistic overhead estimates for project planning
- **Quality Assurance:** Formal metrics for measuring accessibility implementation quality

## Slide 13: Research Contributions Summary

**Time:** 1.5 minutes

### OPTION A: Three-Pillar Approach (Recommended)

1. **Methodological:** Novel evaluation framework with 6 formal metrics + WCAG2Mobile integration
2. **Practical:** AccessibleHub toolkit with theory-informed educational design
3. **Empirical:** First systematic quantitative comparison of accessibility implementation

### OPTION B: Innovation Focus

- **First of Its Kind:** Systematic accessibility comparison with formal methodology
- **Theory-Practice Bridge:** AccessibleHub connects abstract guidelines to concrete code
- **Open Source Impact:** Tools and methodology available for community adoption

**Visuals:** Three pillars diagram + contribution summary table

---

## Slide 14: Future Research Directions

**Time:** 1 minute

### OPTION A: Immediate Extensions (Recommended)

- **Framework Expansion:** SwiftUI, Jetpack Compose accessibility analysis
- **User Studies:** Developer effectiveness measurement with AccessibleHub
- **Automation:** Integration with CI/CD pipelines for accessibility testing
- **Community:** Open source contribution and ecosystem development

### OPTION B: Long-term Vision

- **Industry Adoption:** Standardization of quantitative accessibility metrics
- **Educational Integration:** Curriculum integration at universities and bootcamps
- **Policy Impact:** Evidence for accessibility compliance cost-benefit analysis

### OPTION C: Research Extensions

- **Longitudinal Studies:** Long-term accessibility maintenance costs
- **Cross-Cultural:** Accessibility implementation across different development cultures
- **Emerging Technologies:** AR/VR accessibility framework evaluation

**Visuals:** Research roadmap + community adoption potential

---

## Slide 15: Questions & Discussion

**Time:** Remaining time

### OPTION A: Simple Thank You

- **Summary:** AccessibleHub bridges theory-practice gap with quantitative evidence
- **Availability:** Open source toolkit + reproducible methodology
- **Contact:** [gabriel.rovesti@studenti.unipd.it](mailto:gabriel.rovesti@studenti.unipd.it) | GitHub: [gabrielrovesti/AccessibleHub](https://github.com/gabrielrovesti/AccessibleHub)

### OPTION B: Call to Action

- **For Academia:** Adopt formal metrics for accessibility research
- **For Industry:** Use AccessibleHub for evidence-based framework decisions
- **For Community:** Contribute to open source accessibility toolkit development

**Visuals:** QR codes for repositories + contact information + University branding

---

## ALTERNATIVE PRESENTATION STRATEGIES

### Strategy A: Research-Heavy (Recommended for Academic Audience)

- Emphasize methodological innovation (Slides 4, 8, 11)
- Dedicate more time to formal metrics and WCAG2Mobile integration
- Position AccessibleHub as research vehicle rather than primary contribution

### Strategy B: Tool-Focused (If Commission Prefers Practical Work)

- Lead with AccessibleHub demonstration and practical impact
- Minimize theoretical foundation in favor of implementation examples
- Emphasize immediate developer utility and adoption potential

### Strategy C: Balanced Academic-Practical (Safest Option)

- Current structure balances research rigor with practical contribution
  - Each section reinforces both theoretical foundation and practical application
  - Appeals to both research-oriented and industry-connected commission members
- 

## SLIDE CONTENT DEVELOPMENT NOTES

### Code Examples to Include:

- Button accessibility properties (React Native vs Flutter)
- Form validation with accessibility announcements
- Focus management in modal dialogs
- Custom accessibility actions for complex gestures



## **Screenshots to Prepare:**

- AccessibleHub home screen with metrics
- Component comparison screen
- Framework comparison analytical interface
- Settings screen showing accessibility customization

## **Data Visualizations Needed:**

- Implementation overhead by component type
- WCAG compliance across frameworks
- Screen reader testing results
- Development time estimation comparisons

## **Backup Content (If Questions Arise):**

- Detailed WCAG2Mobile mapping tables
  - Extended accessibility principles explanation
  - Educational theory foundation
  - Complete methodology with mathematical formulations
  - User study plans and community adoption strategies
- 

## **TIMING FLEXIBILITY NOTES**

### **If Running Long** (Remove/Shorten):

- Slide 11 (Extended Principles) - integrate into other slides
- Detailed code examples in Slide 6
- Future work details in Slide 14

### **If Running Short** (Expand):

- Live demonstration of AccessibleHub navigation
- Detailed methodology explanation
- Extended results discussion with Q&A

### **Emergency Backup Plan:**

- Core story: Problem (2 min) → AccessibleHub (5 min) → Results (8 min) → Impact (3 min) → Questions (2 min)