# Issue #11: 🧪 User Testing Protocol

**Repository:** `CherrelleTucker/codesign-toolkit` **URL:** https://github.com/CherrelleTucker/codesign-toolkit/issues/11 **Author:** @CherrelleTucker

**State:** `open` **Labels:** ⚙️ technical-codev, ⚙️ phase-development, 📈 difficulty-intermediate **Assignees:** None

**Created:** 2025-11-14T06:40:20Z **Last Updated in GitHub:** 2025-11-17T05:56:56Z **Worksheet Version:** 2025-11-17T05:57:05.585Z

---

# 🧪 User Testing Protocol

## *Structured Approach to Testing Earth Observation Solutions with Real Live Users*

**Tool Category**: Technical Co-Development | **Phase**: Development | **Difficulty**: 📈 Intermediate

> *Systematically test prototypes and solutions with actual users to validate usability, functionality, and alignment with real-world workflows.*

---

### 📋 Tool Summary Card

| Attribute | Value |
|---|---|
| 🎯 **Purpose** | Validate solution usability and functionality through structured user testing sessions |
| ⏱️ **Time Required** | 4-6 hours prep + 2-3 hours per testing session + 2 hours analysis |
| 👥 **Participants** | 3-5 representative users per round + UX facilitator + observer(s) |
| 📊 **Outputs** | Usability findings, task completion data, prioritized improvement recommendations |
| 🔄 **Frequency** | 2-3 rounds during development, plus pre-launch validation |
| 💼 **Materials** | Test protocol, working prototype, observation templates, recording setup |

---

### 🎯 When to Use This Tool

✅ **Essential For:**

- Testing interactive solutions (dashboards, tools, interfaces)
- Validating that solutions work in real user workflows
- Identifying usability problems before full deployment
- Demonstrating user-centered development to stakeholders

⚠️ **Consider Alternatives When:**

- Solution is purely backend/API with no user interface
- Only one user type with very simple interaction patterns
- Extremely limited time/resources for testing
- Solution is still in very early conceptual phase

---

## 🧪 Testing Protocol Framework

### Pre-Testing Setup (1-2 weeks before sessions)

```
## User Testing Plan: [Solution Name] - Round [X]
**Test Date(s):** [Planned dates] | **Facilitator:** [Name] | **Observers:** [Names]

## 🎯 Testing Objectives
**Primary Questions:**
- Can users successfully complete their core tasks with this solution?
- Where do users encounter confusion or friction in the workflow?
- How well does the solution integrate with users' existing processes?
- What aspects of the interface are intuitive vs. require training?

**Specific Hypotheses to Test:**
- [Hypothesis 1: e.g., "Users can create a wildfire alert in under 3 minutes"]
- [Hypothesis 2: e.g., "Data export format matches user workflow needs"]
- [Hypothesis 3: e.g., "Map interface is intuitive for non-GIS experts"]

## 👥 Participant Selection
**Target Participants per Session:** 3-5 users
**Total Sessions Planned:** [Number] over [timeframe]

**Participant Criteria:**
- **Primary Users:** [Specific roles/organizations who use solution daily]
- **Secondary Users:** [Occasional users or different workflow contexts]
- **Experience Levels:** Mix of novice, intermediate, and expert users
- **Technical Comfort:** Range from tech-savvy to minimal technical experience

**Recruitment Status:**
| Participant | Role | Organization | Experience Level | Confirmed |
|-------------|------|--------------|------------------|-----------|
| [Name] | [Role] | [Org] | [Novice/Intermediate/Expert] | ✅/❌ |
| [Name] | [Role] | [Org] | [Novice/Intermediate/Expert] | ✅/❌ |

## 📋 Test Scenarios & Tasks
### Scenario 1: [Real-world situation]
**Context:** "[Realistic scenario description that matches actual user work]"
**Tasks:**
1. [Specific task 1 - should be realistic and measurable]
2. [Specific task 2 - builds on previous task]
3. [Specific task 3 - represents complete workflow]

**Success Criteria:**
- Task completion: User completes without major assistance
- Time: Completes core workflow in [X] minutes or less
- Errors: Fewer than [X] significant errors or false starts
- Satisfaction: User rates experience 4/5 or higher

### Scenario 2: [Different use case or user type]
[Similar structure for each major use case to test]

## 🛠️ Technical Setup Requirements
**Environment:** [Production-like test environment with realistic data]
**Data:** [Representative but non-sensitive test datasets]
**Access:** [How users will access the solution during testing]
**Backup Plans:** [Alternative access methods if tech issues arise]
```

### Testing Session Structure (2-3 hours per session)

```
## Session Agenda Template

### Pre-Session (15 minutes before participant arrival)
- [ ] Technical setup verified and tested
- [ ] Recording equipment ready (with participant permission)
- [ ] Test scenarios and materials prepared
- [ ] Observer roles and note-taking assignments confirmed

### Introduction & Consent (10 minutes)
**Script:** "We're testing the solution, not testing you. There are no wrong
answers, and if something doesn't work, that's valuable feedback for us."

- Welcome and introductions
- Explain purpose: improving the solution based on real user experience
- Recording consent and confidentiality assurance
- Encourage thinking aloud throughout the session
- Questions about the participant's current work and experience

### Context Setting (10 minutes)
**Script:** "Tell me about how you currently handle [relevant task area]. What tools
do you use? What works well? What's frustrating?"

- Understand participant's current workflow
- Identify baseline expectations and pain points
- Confirm participant fits target user profile
- Set context for realistic test scenarios

### Task Testing (60-90 minutes)
**For Each Scenario:**

**Setup (2-3 minutes):**
- Present realistic scenario context
- Confirm participant understands the situation
- Remind about thinking aloud

**Task Execution (10-20 minutes per task):**
```

```
- Observer notes: What does the user try first?
- Observer notes: Where do they hesitate or look confused?
- Observer notes: What do they say about what they expect vs. what happens?
- Facilitator: Ask clarifying questions but avoid leading
- Facilitator: Provide hints only if user is completely stuck

**Task Debrief (5 minutes per task):**
- "How did that feel?"
- "Was anything surprising or confusing?"
- "How does this compare to how you do this now?"
- "What would make this easier or more efficient?"

### Solution Feedback (15 minutes)
- Overall impressions of the solution
- Most/least useful features
- Missing functionality or capabilities
- Integration concerns with existing workflow
- Training needs assessment

### Wrap-up (10 minutes)
- Thank participant for time and insights
- Confirm next steps and how feedback will be used
- Ask about willingness to participate in future testing
- Provide contact information for follow-up questions
```

## 📊 Observation & Data Collection

### Real-Time Observation Template

```
## User Testing Observation Sheet
**Participant:** [ID/Name] | **Date:** [Date] | **Observer:** [Name] | **Session:** [Round X]

### Participant Background
**Role:** [Job title/function]
**Experience:** [Relevant domain experience]
**Tech Comfort:** [Self-reported comfort with technology 1-5]
**Current Tools:** [What they use now for similar tasks]

### Task Performance Tracking
| Task | Start Time | Completion Time | Success Level | Error Count | Assistance Needed |
|------|------------|-----------------|---------------|-------------|-------------------|
| Task 1: [Description] | [Time] | [Time] | Complete/Partial/Failed | [Number] | None/Hints/Significant |
| Task 2: [Description] | [Time] | [Time] | Complete/Partial/Failed | [Number] | None/Hints/Significant |
| Task 3: [Description] | [Time] | [Time] | Complete/Partial/Failed | [Number] | None/Hints/Significant |
### Behavioral Observations
**Positive Indicators:**
- [ ] User immediately understood interface elements
- [ ] User followed expected workflow path
- [ ] User expressed satisfaction or approval
- [ ] User completed tasks faster than expected
- [ ] User discovered helpful features independently

**Concern Indicators:**
- [ ] User hesitated or seemed confused about next steps
- [ ] User clicked/tried multiple options before finding correct path
- [ ] User expressed frustration or confusion
- [ ] User asked for help or clarification multiple times
- [ ] User mentioned this being harder than current method

### Quotes & Specific Feedback
**Direct User Quotes:**
- "[Quote about what worked well]"
- "[Quote about confusion or problems]"
- "[Quote about comparison to current tools]"
- "[Quote about missing features or needs]"

### Observer Notes & Insights
**Workflow Integration:**
- [How well solution fits user's actual work process]

**Interface Usability:**
- [Specific UI elements that worked well or caused problems]

**Performance Issues:**
- [Any technical problems or slow response times]

**Training Implications:**
- [What users would need training on vs. what was intuitive]
```

### Post-Session Analysis Framework

```
## Testing Round [X] Analysis Report
**Test Dates:** [Range] | **Participants:** [Number] | **Analyst:** [Name] |
**Report Date:** [Date]

## 📊 Quantitative Results Summary
### Task Completion Rates
| Task | Attempted | Completed Successfully | Completion Rate | Avg Time |
|------|-----------|----------------------|-----------------|----------|
| [Task 1] | [N] participants | [N] completed | [X]% | [X] minutes |
| [Task 2] | [N] participants | [N] completed | [X]% | [X] minutes |
| [Task 3] | [N] participants | [N] completed | [X]% | [X] minutes |

### User Satisfaction Scores (1-5 scale)
```

```
- **Overall Experience:** [Average score] (Range: [lowest] - [highest])
- **Ease of Use:** [Average score]
- **Usefulness:** [Average score]
- **Integration with Workflow:** [Average score]
- **Likelihood to Adopt:** [Average score]

## 🎯 Key Findings by Priority

### 🔴 Critical Issues (Must Fix Before Launch)
**Issue 1: [Description]**
- **Impact:** [How this affects users and success metrics]
- **Evidence:** [Specific observations, participant count affected]
- **Recommended Action:** [Specific design/development change needed]
- **Effort Estimate:** [Development time/complexity]

**Issue 2: [Description]**
- **Impact:** [User impact and business impact]
- **Evidence:** [Supporting data from testing]
- **Recommended Action:** [Specific solution approach]
- **Effort Estimate:** [Resource requirements]

### 🟡 Important Improvements (Should Address Soon)
**Improvement 1: [Description]**
- **Benefit:** [How this would improve user experience]
- **Evidence:** [User feedback and observations]
- **Recommended Action:** [Implementation approach]
- **Effort Estimate:** [Development requirements]

### 🟢 Enhancement Opportunities (Future Versions)
**Enhancement 1: [Description]**
- **Potential Value:** [Long-term user benefit]
- **Evidence:** [User suggestions and workflow analysis]
- **Recommended Action:** [Future development consideration]

## 💬 Representative User Feedback
**What Users Liked:**
- "[Positive quote about specific feature]"
- "[Comment about workflow improvement]"
- "[Praise for intuitive design element]"

**What Users Found Challenging:**
- "[Quote about confusion or difficulty]"
- "[Feedback about missing functionality]"
- "[Concern about integration with existing tools]"

**Suggestions for Improvement:**
- "[User-suggested enhancement]"
- "[Workflow integration recommendation]"
- "[Feature request based on current work methods]"

## 📋 Recommendations & Next Steps
### Immediate Actions (This Sprint)
```

```
1. [Specific action item with owner and deadline]
2. [Another immediate fix needed]

### Short-Term Improvements (Next 2-4 weeks)
1. [Enhancement with implementation plan]
2. [Another important improvement]

### Validation Testing Needed
- [ ] Re-test critical issue fixes with 2-3 users
- [ ] Validate new features with different user types
- [ ] Test integration scenarios more thoroughly

### Next Testing Round Planning
**Focus Areas:** [What to emphasize in next round]
**Participant Needs:** [Different user types or scenarios to include]
**Timeline:** [When to conduct next testing round]
```

## 🔧 Troubleshooting Common Testing Challenges

### 👥 "Users are too polite and don't give critical feedback"

**Strategies to Encourage Honest Feedback:**

**Reframe the Context:**

- "We're looking for problems - finding them helps us make this better"
- "The best feedback you can give us is pointing out what doesn't work"
- "We'd rather hear about issues now than have users struggle later"

**Ask Specific Probing Questions:**

- "What would make this faster for you?"
- "How does this compare to what you use now?"
- "What would you change about this screen?"
- "What information is missing that you'd need?"

**Use Comparison Techniques:**

- "If you were training someone new, what would you warn them about?"
- "What would your colleagues say about this approach?"
- "Where do you think people might get confused?"

### 🖥️ "Technical issues disrupt testing sessions"

**Prevention Strategies:**

- **Pre-Session Testing:** Test all technology 30 minutes before each session
- **Backup Plans:** Have alternative access methods ready (different browsers, devices, etc.)
- **Simple Scenarios:** Start with basic tasks to confirm system stability
- **Technical Support:** Have someone available to troubleshoot during sessions

**Recovery Techniques:**

- **Acknowledge and Move On:** "Let's try a different approach while we fix that"
- **Paper Prototyping:** Use screenshots or sketches if system is down
- **Focus on Workflow:** Discuss user needs and current processes instead
- **Reschedule Gracefully:** Respect participant time and offer alternative session times

---

### 📊 "Results are inconsistent across different users"

**Analysis Approaches:**

**Segment by User Characteristics:**

- **Experience Level:** Do novice vs. expert users have different experiences?
- **Role Type:** Are different job functions using the tool differently?
- **Technical Comfort:** Does tech-savviness affect success patterns?
- **Organizational Context:** Do different agencies have different workflow needs?

**Look for Underlying Patterns:**

- **Task Complexity:** Are some tasks inherently more difficult for everyone?
- **Interface Elements:** Do specific UI components cause consistent problems?
- **Workflow Assumptions:** Are we assuming workflows that don't match reality?
- **Training Needs:** Are some things learnable vs. fundamentally confusing?

**Design for Flexibility:**

- **Multiple Pathways:** Provide different approaches for different user types
- **Progressive Disclosure:** Simple default with advanced options available
- **Customization Options:** Let users adapt interface to their preferences
- **Contextual Help:** Provide assistance where needed without cluttering interface

---

## 📈 Success Metrics & Benchmarks

### Usability Benchmarks for Earth Observation Solutions

```
## Testing Success Criteria

### Task Completion Metrics
**Acceptable Performance:**
- **Core Task Completion:** >80% of users complete primary workflows successfully
- **Time to Competency:** Users complete familiar tasks within 150% of expert time by session 2
- **Error Recovery:** Users recover from mistakes without significant assistance >90% of the time
- **Task Efficiency:** Core workflows take ≤20% longer than users' current methods

### User Experience Metrics
**Target Satisfaction Scores (1-5 scale):**
- **Overall Experience:** >4.0 average across all participants
- **Ease of Learning:** >3.8 (recognizing some complexity is expected)
- **Workflow Integration:** >4.2 (critical for adoption success)
- **Usefulness:** >4.5 (must provide clear value over current methods)
```

```
### Adoption Readiness Indicators
**Pre-Launch Validation:**
- [ ] 90% of test participants can complete core tasks independently
- [ ] Users identify clear benefits over current tools/methods
- [ ] No critical usability issues remain unresolved
- [ ] Users express confidence in recommending solution to colleagues
- [ ] Integration concerns are addressed or have clear mitigation plans

### Training Requirement Assessment
**Minimal Training Needed (>80% intuitive):**
- Basic navigation and core functionality
- Data input and standard output generation
- Common workflow patterns

**Focused Training Needed (40-80% intuitive):**
- Advanced features and customization options
- Integration with existing systems
- Troubleshooting and error handling

**Intensive Training Required (<40% intuitive):**
- Complex analytical functions
- Administrative or configuration tasks
- Specialized use cases or edge scenarios
```

## 🔄 Integration with Development Cycles

### Agile Development Integration

```
## User Testing in Development Sprints

### Sprint Planning Considerations
**Testing-Ready Definition:**
- Core functionality works without major bugs
- User interface is complete enough for realistic task testing
- Test data and scenarios are prepared
- Users are available and scheduled

**Testing Sprint Timeline:**
- **Week 1:** Feature development completion, testing preparation
- **Week 2:** User testing sessions, immediate feedback integration
- **Week 3:** Fix implementation, validation testing with subset of users
- **Week 4:** Sprint review with testing outcomes, next sprint planning

### Continuous Feedback Loop
**Daily Standups:** Include user testing insights in development discussions
**Sprint Reviews:** Demonstrate improvements made based on user feedback
**Backlog Prioritization:** Use testing results to prioritize future development
**Stakeholder Updates:** Regular communication about user validation progress
```

### Testing Round Planning

**Round 1: Early Prototype (Week 6-8 of development)**

- **Focus:** Core functionality and workflow validation
- **Participants:** 3-4 primary users, high tolerance for rough edges
- **Success Criteria:** Users understand concept and can complete basic tasks

**Round 2: Beta Version (Week 10-12 of development)**

- **Focus:** Interface refinement and integration testing
- **Participants:** 5-6 users including secondary user types
- **Success Criteria:** Smooth task completion, positive user satisfaction

**Round 3: Pre-Launch Validation (Week 14-15 of development)**

- **Focus:** Final validation and adoption readiness
- **Participants:** 4-5 users representing full user diversity
- **Success Criteria:** Users ready to recommend to colleagues, minimal support needed

---

## 🔗 Related Tools & Integration

**This Protocol Works With:**

- **Requirements Definition Canvas** - Test against co-defined requirements
- **Stakeholder Mapping Workshop** - Recruit representative test participants
- **Progress Communication Dashboard** - Share testing results with stakeholders

**This Protocol Informs:**

- **Decision Documentation Template** - Document design decisions based on testing
- **Training Material Development Kit** - Design training based on usability findings
- **Adoption Monitoring Framework** - Establish baseline expectations for adoption

**External Tool Integration:**

- **Screen Recording Software:** Capture user sessions for detailed analysis
- **Survey Tools:** Collect quantitative satisfaction and preference data
- **Project Management:** Track testing findings as development backlog items
- **Analytics Platforms:** Measure actual usage patterns post-deployment

---

## 📚 Source Attribution

**Primary Sources:**

- **NSITE Solution Project Requirements and Expectations** - User testing requirements during solution development
- **Solution Co-Development Toolkit Narrative** - Iterative user validation and feedback integration
- **MSFC Coordination on Solutions Co-Development Toolkit** - User engagement during technical development phases

**Supporting Sources:**

- **SERVIR Service Design Tool 2021** - User validation approaches and stakeholder feedback collection
- **Meeting Notes - Technical Development CoDesign Toolkit Working Group** - Testing integration with development processes

**Methodology Foundation:**

- User experience research and usability testing best practices
- Human-computer interaction evaluation frameworks
- Agile development user validation approaches adapted for Earth observation contexts

---

## 💬 Community Discussion

**Share your testing experience:**

- What testing approaches work best for your user types and solution complexity?
- How do you balance thorough testing with development timeline constraints?
- What unexpected insights have you gained from user testing sessions?
- How do you adapt testing methods for remote or distributed users?

**Protocol improvements:**

- What testing scenarios would you add for specific Earth observation applications?
- How do you handle testing with users who have security or data access constraints?
- What tools and technologies work best for remote user testing?

---

🔧 **Tool Maintainer:** @your-username | 📅 **Last Updated:** [Today's Date] | 📌 **Version:** 1.0