

# RRI-T

## REVERSE REQUIREMENTS INTERVIEW TESTING

---

Phương pháp Kiểm thử Toàn diện dựa trên Phỏng vấn Ngược

5 Personas × 7 Dimensions × 8 Stress Axes = Complete Quality

Version 1.0 | 2025

Evolved from RRI Methodology

Designed for Vietnamese Enterprise Software

# 1. NGUỒN GỐC & PHẢN BIỆN RRI

## 1.1 Những gì RRI làm tốt

RRI (Reverse Requirements Interview) giải quyết được vấn đề cốt lõi: phát hiện requirements ẩn trước khi code. Ba trụ cột mạnh nhất:

- Empathy-first thinking — “Nếu TÔI là user, tôi cần gì” thay vì hỏi user cần gì
- 3 Personas coverage — End User, BA, QA/Tester phủ 3 góc nhìn bổ trợ
- Stress Test Framework — 5 chiều (Time, Data, Error, Collab, Emergency) rất thực chiến

## 1.2 Phản biện — Những gì RRI chưa giải quyết

#	Điểm thiếu	Hệ quả	RRI-T bổ sung
1	Chỉ phát hiện, chưa kiểm chứng	Requirements không được verify	Test Case Generation (Q → A → R → P → T)
2	Thiếu chiều Infrastructure	Deploy, network, device bị bỏ qua	Testing Dimension #6
3	Thiếu chiều Data Integrity	Migration, backup bị bỏ qua	Testing Dimension #5
4	Không address Vietnamese UX	Dấu TV, VND, timezone	Vietnamese-Specific Testing
5	Thiếu Regression awareness	Fix gap A break gap B	Regression Testing Matrix
6	3 Personas chưa đủ	Thiếu DevOps + Security	Mở rộng 5 Personas
7	Stress Test 5 chiều	Thiếu Security, Infra, Locale	Mở rộng 8 chiều
8	Không đo coverage	Không biết test đủ chưa	Coverage Matrix + Scoring

## 1.3 Từ RRI đến RRI-T

RRI (Requirements Discovery)	→	RRI-T (Quality Verification)
"Tìm ra những gì CẦN BUILD"	→	"Chứng minh BUILD là ĐÚNG"
		+ cover những gì CHƯA NGHĨ TỚI

- RRI output → RRI-T input (requirements → test cases)
- RRI-T cũng phát hiện thêm requirements mới (testing reveals missing specs)
- Vòng lặp feedback: RRI-T gaps → quay lại RRI bổ sung

## 2. TRIẾT LÝ RRI-T

Đừng chỉ test xem phần mềm có CHẠY không. Hãy test xem nó có SỐNG ĐƯỢC trong tay người dùng thực tế không.

### 2.1 Sự khác biệt tư duy

Traditional Testing	RRI-T
Feature X hoạt động đúng spec không?	Người dùng THỰC SỰ dùng feature X như thế nào?
Test case từ spec → verify	Test case từ CUỘC SỐNG → discover + verify
Pass/Fail binary	Pass / Fail / Usable but Painful / Works but Missing
Test sau khi code	Test THINKING trước code, test EXECUTION sau code

### 2.2 Bốn kết quả kiểm thử






Thay vì chỉ Pass/Fail, RRI-T dùng 4 levels để capture đầy đủ chất lượng:

Kết quả	Ký hiệu	Ý nghĩa	Ví dụ
PASS	✓	Đúng spec VÀ tốt cho user	Save → thành công, có feedback
FAIL	✗	Sai spec hoặc không hoạt động	Save → lỗi 500
PAINFUL	⚠	Hoạt động nhưng UX kém	Save → mất 8 giây, không feedback
MISSING	□	Thiếu feature user cần	Không có auto-save

## 3. MÔ HÌNH 5 TESTING PERSONAS

### 3.1 Tổng quan

RRI dùng 3 personas để phát hiện requirements. RRI-T mở rộng lên 5 personas để kiểm thử toàn diện — vì testing cần cover cả góc nhìn vận hành hạ tầng và an ninh.

Persona	Tư duy	Testing Focus	Nguồn gốc
 End User	Tôi dùng hàng ngày	Workflow, UX, efficiency, offline	RRI gốc
 Business Analyst	Rules phải đúng	Business rules, RBAC, audit, compliance	RRI gốc
 QA Destroyer	Tôi sẽ phá	Edge cases, errors, concurrent, undo	RRI gốc
 DevOps Tester	Hệ thống chịu được?	Deploy, scaling, monitoring, recovery	MỚI - RRI-T
 Security Auditor	Ai có thể lạm dụng?	AuthN/Z, injection, encryption, abuse	MỚI - RRI-T

### 3.2 Chi tiết: End User Tester

*Tư duy: “Tôi là nhân viên vận hành. Mỗi sáng mở app, tôi làm gì? Tôi cần gì để không phải suy nghĩ nhiều?”*

- Daily workflow hoàn chỉnh (happy path end-to-end)
- First-time user experience (onboarding)
- Repeated task efficiency (bulk operations, shortcuts)
- Context switching (đang làm A, chuyển sang B, quay lại A)
- Offline/poor connectivity behavior
- Mobile vs Desktop experience

### 3.3 Chi tiết: Business Analyst Tester

*Tư duy: “Mọi business rule phải được implement chính xác. Mọi con số phải khớp.”*

- Business rules accuracy (công thức tính, validation rules)
- Role-Based Access Control (đúng từng permission)
- Data consistency across modules
- Reporting accuracy (số liệu khớp giữa các báo cáo)
- Audit trail completeness

### 3.4 Chi tiết: QA Destroyer

*Tư duy: “Mọi thứ có thể sai SẼ sai. Tôi tìm mọi cách phá hệ thống.”*

- Edge cases (giá trị biên, null, empty, overflow)
- Error paths & recovery
- Concurrent operations
- Rapid sequential actions (double-click, mash submit)

### 3.5 Chi tiết: DevOps Tester (MỚI)

Tư duy: “App chạy đúng trên máy dev nhưng chết trên production. Tôi test ở điều kiện thực tế.”

- Deployment reliability (zero-downtime deploy)
- Scaling behavior (10 → 100 → 1000 users)
- Resource consumption (CPU, memory, disk)
- Backup/restore procedure
- Database migration safety

### 3.6 Chi tiết: Security Auditor (MỚI)

Tư duy: “Mọi input là hostile. Mọi user có thể là attacker.”

- Authentication & authorization bypass
- Input sanitization (XSS, SQL injection, CSRF)
- Data exposure (API responses leak sensitive data?)
- Rate limiting & abuse prevention
- Sensitive data handling (encryption, masking)









### 3.7 Ma trận Persona × Testing Aspect

Aspect	 User	 BA	 QA	 DevOps	 Security
UI/UX	•••	•	••	—	•
Business Logic	•	•••	••	—	•
Edge Cases	•	•	•••	•	••
Performance	••	—	•	•••	•
Security	—	•	•	••	•••
Data Integrity	•	•••	••	••	•
Infrastructure	—	—	•	•••	••

## 4. STRESS TEST FRAMEWORK 8 CHIỀU

### 4.1 Mở rộng từ 5 → 8 chiều

RRI gốc có 5 chiều Stress Test. RRI-T bổ sung 3 chiều quan trọng: Security Stress, Infrastructure Stress, Localization Stress.

#	Chiều	Câu hỏi gốc	Test Scenarios
1	 TIME	Deadline 5 phút, cần làm gì?	Bulk ops, shortcuts, auto-complete
2	 DATA	1000 dòng data, tìm thể nào?	Search < 500ms, filter < 1s, export 10K < 10s
3	 ERROR	Save xong phát hiện sai?	Undo/redo, version history, auto-save 30s
4	 COLLAB	5 người cùng edit 1 file?	Conflict warning < 2s, no data loss
5	 EMERGENCY	Supplier đột xuất hủy order?	Emergency workflow 2 clicks, notify < 30s
6	 SECURITY (MỚI)	NV nghỉ việc, revoke access?	Revoke ngay, sessions terminated, audit log
7	 INFRA (MỚI)	Server chết 2AM, sáng đến?	RTO < 15m, RPO < 5m, zero data loss
8	 LOCALE (MỚI)	Gõ tiếng Việt có dấu?	Nguyễn tìm được nguyên, VND đúng

### 4.2 Stress Combination Matrix

Sức mạnh thực sự nằm ở việc KẾT HỢP nhiều chiều cùng lúc:

Kết hợp	Scenario	Test Case
TIME × DATA	Deadline gấp + data lớn	Bulk approve 500 records trong 3 phút
COLLAB × ERROR	Nhiều người + conflict	3 users edit, 1 save fail, others?
COLLAB × SECURITY	Nhiều người + phân quyền	Admin và Viewer cùng mở record
INFRA × DATA	Server stress + data lớn	Migration 1M records + concurrent users
SECURITY × LOCALE	Auth + tiếng Việt	Username “Nguyễn Văn A” trong JWT
INFRA × EMERGENCY	Server sự cố + deadline	Prod down lúc close budget

## 5. BẢY TESTING DIMENSIONS

Mỗi feature/module phải được test qua 7 dimensions. Đây là các lens khác nhau nhìn vào cùng một functionality.

Dimension	Mục tiêu	Key Metrics
D1: UI/UX	Giao diện trực quan, responsive, accessible	0 visual deviation, WCAG 2.1 AA
D2: API	Mọi endpoint đúng contract, error graceful	100% match OpenAPI spec
D3: Performance	Nhanh và ổn định dưới tải thực tế	p95 < 500ms, 0 errors under load
D4: Security	Dữ liệu an toàn, access đúng quyền	0 unauthorized access, TLS 1.2+
D5: Data Integrity	Dữ liệu đúng, đầy đủ, nhất quán	100% roundtrip accuracy
D6: Infrastructure	Hệ thống ổn định, recoverable	RTO < 15m, 0 downtime deploy
D7: Edge Cases	Xử lý mọi tình huống bất thường	Graceful handling, no crash

### 5.1 D1: UI/UX Testing

Area	Kiểm tra	Tiêu chí Pass
Visual Consistency	Font, color, spacing theo design system	0 deviation từ design tokens
Responsive	Mobile 375px, Tablet 768px, Desktop 1440px, 4K	Layout không vỡ, content không mất
Interactive States	Hover, focus, active, disabled, loading, error	Mỗi state có visual feedback rõ
Form Behavior	Validation timing, error messages, tab order	Real-time validation, lỗi inline
Navigation	Breadcrumb, back/forward, deep linking	Mọi page reachable, back hoạt động
Loading States	Skeleton, spinner, progress bar	Feedback < 100ms, spinner sau 300ms
Empty States	No data, first use, search no results	Có message hướng dẫn
VN Text Overflow	Tiếng Việt dài hơn EN ~30%	UI không bị vỡ layout

### 5.2 D2: API Testing

Area	Kiểm tra	Tiêu chí Pass
Contract	Request/Response schema validation	100% match OpenAPI spec
Status Codes	200, 201, 400, 401, 403, 404, 409, 422, 500	Đúng code cho đúng scenario
Validation	Required fields, types, ranges, formats	400 với message rõ cho mỗi vi phạm
Pagination	Cursor-based, offset, limit, sorting	Consistent, no duplicates
Error Responses	Format consistency, error codes	JSON chuẩn: {code, message, details}
Idempotency	POST/PUT retry safety	Duplicate request không tạo duplicate data

### 5.3 D3: Performance Testing

Performance Budget:

Metric	Target	Degraded	Failed
First Contentful Paint	< 1.5s	< 3s	> 3s
Time to Interactive	< 3s	< 5s	> 5s
API Response (p50)	< 200ms	< 500ms	> 500ms
API Response (p95)	< 500ms	< 1s	> 1s
Bundle Size	< 500KB	< 1MB	> 1MB
Database Query	< 50ms	< 200ms	> 200ms

Load Test Scenarios:

Scenario	Users	Duration	Success Criteria
Normal Load	50 concurrent	30 min	p95 < 500ms, 0 errors
Peak Load	200 concurrent	15 min	p95 < 1s, error < 0.1%
Stress Test	500 concurrent	10 min	Graceful degradation, no crash
Endurance	100 concurrent	4 hours	No memory leak, consistent p95
Spike Test	0 → 300 → 0 in 1m	5 min	Recovery < 30s, no data loss

### 5.4 D4: Security Testing

Area	Kiểm tra	Tiêu chí Pass
AuthN	Login, logout, token refresh, session timeout	Token secure, timeout đúng
AuthZ	RBAC per endpoint, horizontal access	User A KHÔNG access data User B
Input Sanitization	XSS, SQL Injection, Command Injection	Mọi input được escape/sanitize
Data Exposure	API response chỉ allowed fields	Không leak password, PII
File Upload	Type check, size limit	Reject executable, limit 10MB
Headers	CSP, HSTS, X-Frame-Options	Tất cả security headers present
Encryption	HTTPS only, data at rest	TLS 1.2+, AES-256 sensitive data

### 5.5 D5: Data Integrity Testing

Area	Kiểm tra	Tiêu chí Pass
CRUD Consistency	Create → Read → Update → Delete lifecycle	Data roundtrip 100% chính xác
Cross-Module	Data shared giữa modules	Thay đổi module A reflect ở B
Calculation	Công thức, aggregation, rounding	Khớp 100% với Excel reference
Migration	Schema change, data migration	Zero data loss, rollback safe
Backup/Restore	Backup procedure, restore verify	Restore khớp 100% backup point



Concurrent Write	Optimistic/pessimistic locking	No lost updates, conflict detected
Temporal	Timestamps, timezone, date ranges	UTC storage, GMT+7 display đúng

## 5.6 D6: Infrastructure Testing

Area	Kiểm tra	Tiêu chí Pass
Deploy	Zero-downtime deployment, rollback	0 downtime, rollback < 5 min
Env Parity	Dev $\approx$ Staging $\approx$ Prod	Config drift < 5%
Health Checks	Liveness, readiness, startup probes	Correct status within 10s
Monitoring	Metrics, dashboards, alert rules	Alert fires within 60s
Scaling	Horizontal scale up/down	Scale < 2 min, no request drops
Resource Limits	CPU/Memory limits, disk usage	Graceful at 90% capacity

## 5.7 D7: Edge Case & Error Recovery

Category	Test Cases	Expected
Boundary Values	0, 1, max-1, max, max+1, negative	Accept valid, reject with message
Empty/Null	Null, undefined, empty string, whitespace	Không crash, appropriate state
Special Characters	Unicode, emoji, HTML tags, SQL chars	Escape/render correctly
Large Input	10MB paste, 100K characters	Truncate hoặc reject with limit
Rapid Actions	Double-click, rapid submit, tab-mash	Debounce, no duplicate ops
State Transitions	Offline $\rightarrow$ Online, sleep $\rightarrow$ wake	State recovery, no data loss
Network	Slow 3G, packet loss, timeout	Retry logic, offline indicator
Time-based	Midnight, DST, leap year, timezone	Correct calculations

## 6. QUY TRÌNH 5 PHASES

Phase	Tên	Mô tả	Thời gian	Output
1	PREPARE	Gather context, setup framework	1-2 giờ	Test Plan
2	DISCOVER	5-persona interview sinh test cases	3-5 giờ	Raw Test Cases
3	STRUCTURE	Phân loại, ưu tiên, viết formal	2-4 giờ	Structured Test Suite
4	EXECUTE	Chạy tests, ghi kết quả 4 levels	2-8 giờ	Test Results
5	ANALYZE	Phân tích gaps, regression, feedback	1-2 giờ	Gap Report

### 6.1 Phase 2: DISCOVER — Phân bố



Round	Persona	Số lượng	Focus
1	 End User	25-35	Workflow, UX, daily tasks
2	 Business Analyst	20-30	Rules, data, compliance
3	 QA Destroyer	25-35	Edge cases, error paths
4	 DevOps	15-20	Infrastructure, performance
5	 Security Auditor	15-20	Security, access control
	TỔNG	100-140	Full coverage

### 6.2 Phase 3: STRUCTURE — Ưu tiên hóa

Ma trận Impact × Likelihood:

	Unlikely	Possible	Likely
Critical Impact	P1	P0	P0
Major Impact	P2	P1	P0
Minor Impact	P3	P2	P1

### 6.3 Phase 5: ANALYZE — Release Gate

Coverage	Status	Action
≥ 85%	 Green	Release approved
70-84%	 Yellow	Release with known issues documented
< 70%	 Red	Block release, fix critical first

**Release criteria:** Tất cả 7 dimensions ≥ 70%, ít nhất 5/7 ≥ 85%, 0 items P0 ở trạng thái FAIL.

## 7. TEST CASE GENERATION: Q → A → R → P → T

### 7.1 Mở rộng từ RRI format

RRI dùng Q → A → R → P (Question → Answer → Requirement → Priority). RRI-T thêm T = Test Case để khép kín vòng lặp.

ID: [MODULE] - [DIMENSION] - [NUMBER]

Persona: [👤 | 📋 | 🔍 | 🛠️ | 🔒]

Q: [Câu hỏi từ góc nhìn persona]

A: [Expected behavior]

R: [Requirement rút ra]

P: [P0|P1|P2|P3]

T: TEST CASE

Precondition: [Setup]

Steps: 1... 2... 3...

Expected: [Kết quả chi tiết]

Dimension: [D1-D7]

Stress: [Chiều stress]

Result: [✅ | ❌ | ⚠️ | 📄] [Notes]

### 7.2 Ví dụ: End User Persona

Field	Content
ID	OTB-D1-007
Persona	👤 End User
Q	Đang edit budget, sắp gọi 15 phút. Quay lại, data có còn?
A	Auto-save mỗi 30 giây, data phải còn nguyên
R	REQ-045: Implement auto-save every 30s during editing
P	P0
Steps	1. Edit 3 fields 2. Đợi 35s 3. Close tab 4. Re-open
Expected	3 thay đổi giữ lại + “Draft auto-saved at [time]”
Dimension	D1 (UI/UX) + D5 (Data Integrity)
Stress	ERROR (recovery from interruption)

### 7.3 Ví dụ: Security Auditor Persona

Field	Content
ID	OTB-D4-003
Persona	🔒 Security Auditor
Q	User A có thể sửa API request để xem budget department khác?
A	KHÔNG — API phải check department-level authorization
R	REQ-180: Enforce department-level access control

<b>P</b>	P0
<b>Steps</b>	1. User A GET budget Marketing 2. User A PUT 3. User A list
<b>Expected</b>	Step 1-2: 403 Forbidden. Step 3: empty. Audit log ghi nhận
<b>Dimension</b>	D4 (Security)
<b>Stress</b>	SECURITY (authorization bypass)

## 8. COVERAGE MATRIX






### 8.1 Module × Dimension Scoring

Mỗi module được evaluate coverage trên cả 7 dimensions:

Coverage Score = (PASS count) / (Total test cases) × 100%

Target: ≥ 85% PASS rate per dimension → ● Release  
 Warning: 70-84% → ● Conditional  
 Block: < 70% → ● No release

### 8.2 Persona Coverage Check

Persona	Min Test Cases	Primary Dimensions
 End User	25	D1: UI/UX, D7: Edge Cases
 Business Analyst	20	D2: API, D5: Data Integrity
 QA Destroyer	25	D7: Edge Cases, D2: API
 DevOps	15	D3: Performance, D6: Infrastructure
 Security Auditor	15	D4: Security

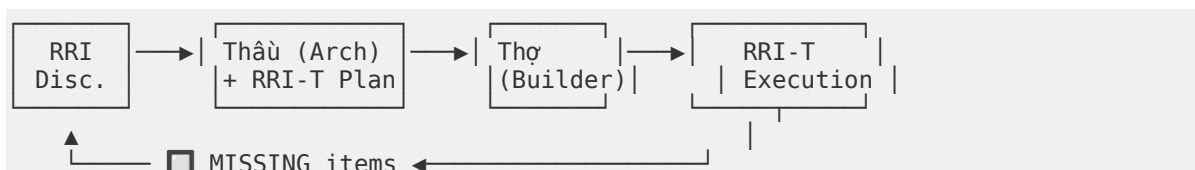
## 9. VIETNAMESE-SPECIFIC TESTING

Hầu hết testing frameworks thiết kế cho English-first. Khi build cho VN, có edge cases đặc thù:

#	Area	Test Case	Expected
1	Dấu tiếng Việt	Search “nguyen” tìm “Nguyễn”?	Yes — diacritic-insensitive
2	Unicode sorting	Sort: Ân, Bình, Cường, Đức	Đúng thứ tự TV (Đ sau D)
3	VND Currency	Hiển thị 1234567 VND	1.234.567 đ (chấm phân cách ngàn)
4	Phone format	+84 912 345 678 hoặc 0912345678	Accept cả hai, normalize
5	Date format	Hiển thị date	23/02/2025 (DD/MM/YYYY)
6	Timezone	Server UTC, display local	GMT+7 consistently
7	Địa chỉ VN	Vietnamese address structure	Số nhà/Đường/Phường/Quận/TP
8	CCCD/CMND	12 digits (CCCD), 9 digits (CMND)	Accept cả hai formats
9	Mã số thuế	MST 10 or 13 digits	Validate đúng format
10	Text overflow	VN text ~30% longer than EN	UI không vỡ layout
11	Font rendering	Diacritics ở small font 10px	Dấu rõ ràng, không cắt
12	Input methods	Telex, VNI, VIQR	Tất cả smooth
13	PDF export	Export với Vietnamese content	Dấu đúng trong PDF

## 10. TÍCH HỢP VIBECODE KIT

### 10.1 Workflow: RRI → Code → RRI-T



### 10.2 Khi nào chạy RRI-T?

Timing	Scope	Trigger
Pre-code (Phase 1-3)	Test plan + test cases	Sau khi Thầu design xong
During-code	Unit + integration tests	Thợ viết code kèm tests
Post-code (Phase 4-5)	Full 7-dimension execution	Trước merge/deploy
Post-deploy	Smoke + regression	Sau mỗi deployment
Monthly	Full regression suite	Scheduled maintenance

### 10.3 Prompt Template cho Claude Code

ROLE: RRI-T Test Engineer

MODULE: [module name]

PERSONA: [chọn 1 trong 5 personas]

DIMENSION: [chọn 1 trong 7 dimensions]

Hãy đặt ra 20 câu hỏi từ góc nhìn [persona] về [module], tập trung vào dimension [dimension].

Format output: Q→A→R→P→T cho mỗi câu hỏi.

Include Vietnamese-specific test cases nếu relevant.

## 11. CASE STUDY MINH HỌA

### 11.1 Áp dụng RRI-T cho DAFC OTB Budget Module

Sau RRI phát hiện 212 requirements, RRI-T chạy để tạo test suite toàn diện.

Dimension	Tests	✓	✗	⚠	☐	Coverage
D1: UI/UX	35	28	3	4	0	80% 🟡
D2: API	30	27	2	1	0	90% 🟢
D3: Performance	20	18	1	1	0	90% 🟢
D4: Security	18	15	2	0	1	83% 🟡
D5: Data	25	22	1	0	2	88% 🟢
D6: Infrastructure	15	13	1	1	0	87% 🟢
D7: Edge Cases	30	24	3	2	1	80% 🟡
TỔNG	173	147	13	9	4	85% 🟢

### 11.2 ROI của RRI-T

Metric	Không RRI-T	Có RRI-T
Bugs found pre-release	~10	22 (13 FAIL + 9 PAINFUL)
Bugs found post-release	~15-20	~2-3 (estimated)
Cost per pre-release bug	\$100	\$100
Cost per post-release bug	\$1,000-5,000	\$1,000-5,000
Estimated savings	—	\$12,000-85,000 per module
Time invested	0	~15 giờ
Confidence at release	Low	High



## 12. CHECKLIST TỔNG HỢP

### Phase 1 — PREPARE

- ☐ Thu thập RRI output / requirements / specs
- ☐ Setup test environment + test data
- ☐ Define scope boundaries
- ☐ List all user roles
- ☐ Identify critical paths

### Phase 2 — DISCOVER

- ☐ End User interview: 25-35 test ideas
- ☐ BA interview: 20-30 test ideas
- ☐ QA Destroyer interview: 25-35 test ideas
- ☐ DevOps interview: 15-20 test ideas
- ☐ Security interview: 15-20 test ideas

### Phase 3 — STRUCTURE

- ☐ Classify test cases vào 7 Dimensions
- ☐ Prioritize Impact × Likelihood matrix
- ☐ Viết  $Q \rightarrow A \rightarrow R \rightarrow P \rightarrow T$  cho P0 và P1
- ☐ Review Vietnamese-specific checklist

### Phase 4 — EXECUTE

- ☐ Chạy tất cả P0 test cases trước
- ☐ Ghi kết quả 4-level (✓✗⚠️🔴)
- ☐ Log FAIL làm bug tickets
- ☐ Log PAINFUL làm improvement tickets
- ☐ Feedback MISSING về requirements

### Phase 5 — ANALYZE

- ☐ Tính coverage per dimension
- ☐ Check release gate criteria
- ☐ Regression trên tất cả fixes
- ☐ Generate Coverage Matrix report

## CÔNG THỨC THÀNH CÔNG RRI-T

EMPATHY × STRUCTURE × EXHAUSTIVENESS × MEASUREMENT

(5 Personas) (7 Dimensions) (100+ Test Cases) (Coverage Matrix)

=

## COMPLETE QUALITY ASSURANCE

---

*RRI tìm ra requirements TRƯỚC KHI code. RRI-T chứng minh code ĐÁP  
ỨNG requirements — và phát hiện những gì requirements CHƯA  
COVER.*

RRI-T Methodology v1.0

Built on the shoulders of RRI  
Designed for Vietnamese Enterprise Software  
Optimized for Vibecode Kit + Claude Code