

STAREAST 2026

The QA Revolution

AI-Enhanced Testing for Modern
Quality Organizations

45-Minute Session | Live Demos | Real Case Studies | Action Plan Included

Session Roadmap

45 minutes of content • 15 minutes of Q&A

 0–8 min

The Shifting QA Landscape

 8–18 min

AI Meets the Atlassian
Ecosystem

 18–30 min

Metrics, Reports & Dashboards

 30–38 min

Edge Cases, Prediction & Case
Study

 38–45 min

Your AI-QA Action Plan

 45–60 min

Q&A

The QA World Has Changed

68%

of organizations are actively using
or have a Gen AI roadmap for QE

Source: World Quality Report 2024

72%

report faster automation processes
after Gen AI integration

Source: World Quality Report 2024

80%

of organizations using AI in testing
improved defect detection

Source: Capgemini Survey

50%

reduction in test case creation time
— e.g. Capital One with AI

Source: ISG / Capgemini, 2024



Fear It

- ✗ AI will replace QA roles
- ✗ Skills become obsolete
- ✗ Reactive, defensive posture
- ✗ Organizations leave you behind



Lead It

- ✓ AI amplifies QA expertise
- ✓ Skills become more valuable
- ✓ Proactive, strategic posture
- ✓ You become indispensable

The AI Tools You Need to Know

Each brings unique strengths — savvy QA teams leverage all three



Claude

by Anthropic

- ▶ Long-form reasoning
- ▶ Test plan drafting
- ▶ Nuanced analysis
- ▶ Document generation

Best for: Complex test strategies & documentation



ChatGPT

by OpenAI

- ▶ Conversational iteration
- ▶ Code generation
- ▶ Multi-turn refinement
- ▶ Wide plugin ecosystem

Best for: Iterative test case creation & scripting



Copilot

by Microsoft

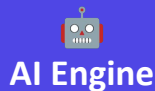
- ▶ IDE integration
- ▶ Inline code suggestions
- ▶ M365 ecosystem
- ▶ Real-time assistance

Best for: In-context automation & script writing

AI + Atlassian: A Powerful Partnership

Requirements & User Stories

Input: Confluence pages,
Jira epics, design docs



Bug Reports & Defects

Output: Jira issues with
full context & steps

Test Plans & Test Cases

Output: Jira test tickets,
Xray/Zephyr integration

Confluence Docs & Reports

Output: Auto-updated pages,
test metrics, dashboards

PART 1

From Requirements to Test Plans — Automatically

Stop spending hours writing what AI can draft in seconds

3 Steps from Requirements to Test Plan

01



Feed In Requirements

Point AI at your Confluence page, Jira epic, or paste requirements directly. Include acceptance criteria, user personas, and design specs for richer output.



02



AI Analyzes & Drafts

The AI identifies testable scenarios, infers edge cases, maps risk areas, and structures a comprehensive plan following your team's format.



03



Review, Refine & Publish

[TRY THIS PROMPT](#)

"Review this [user story + ACs]. Generate a test plan: happy path, negative tests, edge cases, performance, and accessibility. Format for Jira/Xray."



PROMPT

"You are a senior QA engineer. Review the following user story and acceptance criteria from Confluence, then generate a comprehensive test plan including: functional test scenarios, edge cases, negative tests, performance considerations, and accessibility checks. Format for direct import into Jira/Xray."

AI OUTPUT

TEST PLAN: User Authentication Flow v2.3

- ✓ TC-001 Valid login with correct credentials → expect dashboard redirect
- ✓ TC-002 Login with incorrect password → expect error message + lockout after 5 attempts
- ✓ TC-003 Session timeout after 30 min inactivity → expect re-auth prompt
- ⚠ TC-004 [EDGE] SQL injection attempt in username field → expect sanitization
- ⚠ TC-005 [EDGE] Login from unsupported browser → expect graceful degradation
- ✗ TC-006 [NEGATIVE] Empty credentials form submit → expect inline validation

Beyond Test Plans: Cases & Bug Reports



AI Test Case Generation

- ▶ Generates test cases directly from AC
- ▶ Includes happy path + negative paths
- ▶ Automatically assigns priority levels
- ▶ Maps to requirements for traceability
- ▶ Exports directly to Xray or Zephyr

TRY THIS PROMPT

"Generate test cases for [feature]. Include 3+ negative scenarios and 2 boundary conditions. For each: preconditions, steps, expected result, and priority."



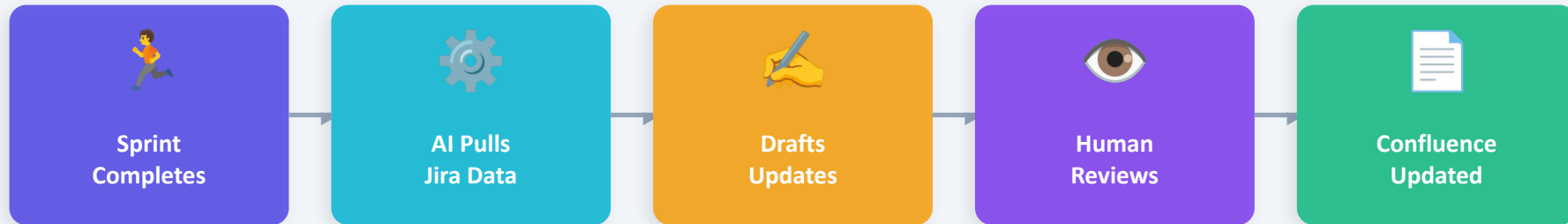
AI Bug Report Builder

- ▶ Describe bug in plain language
- ▶ AI structures into Jira-ready format
- ▶ Auto-fills: steps, severity, component
- ▶ Suggests labels & fix version
- ▶ Links to related failing test cases

TRY THIS PROMPT

"Bug found: [plain description]. Write a Jira-ready report: summary, steps to reproduce, expected vs actual, severity, environment, and suggested labels."

Confluence That Keeps Itself Fresh



What Gets Auto-Updated in Confluence:



Test coverage metrics & pass/fail rates



Sprint test summary reports



Traceability matrices (requirements ↔ tests)



Open defect summaries by component



Trend charts for quality metrics over time



Release readiness scorecards

TRY THIS PROMPT

"Summarize this sprint's test results for Confluence: total tests run, pass/fail rate, top defects by severity, coverage gaps identified, and a release readiness recommendation."

PART 2

Flipping Quality Metrics on Their Head

From lagging indicators to real-time quality intelligence

Auto-Spotting Coverage Gaps

BEFORE AI

- ✗ Manual gap analysis takes days
- ✗ Relies on individual team member knowledge
- ✗ Coverage blind spots persist undetected
- ✗ Requirements change → tests don't update
- ✗ Hard to quantify what's actually untested

AFTER AI

- ✓ Real-time gap analysis across all stories
- ✓ AI cross-references every requirement
- ✓ Flags untested code paths immediately
- ✓ Auto-proposes new tests when ACs change
- ✓ Coverage heatmaps by feature & risk

VS

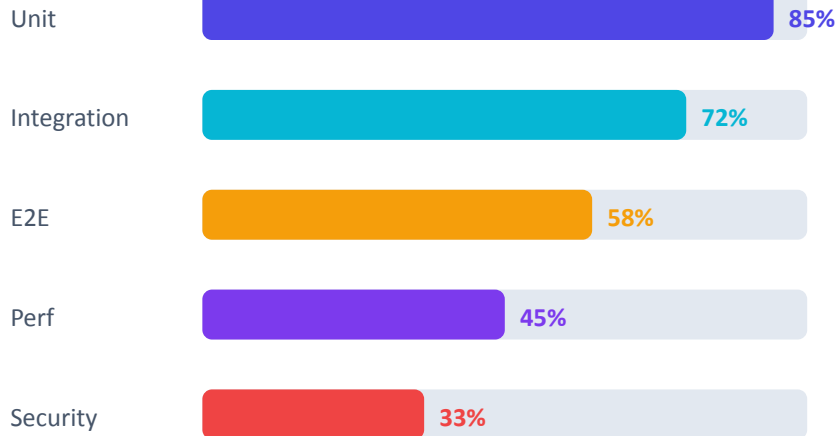
TRY THIS PROMPT

"Here are my ACs: [paste]. Here is my test suite: [paste]. Identify any requirements not covered by an existing test and suggest 3 new test cases to close the gaps."

Reports That Make Stakeholders Take Notice

Test Coverage by Layer

EXAMPLE



TRY THIS PROMPT

"Summarize this sprint's test results by layer. Output a coverage % per layer and flag any below 50% as at-risk."



Auto-Generated Sprint Reports

AI summarizes test results, defect trends, and risk areas after every sprint — no manual writing.



Executive Quality Dashboards

Stakeholder-friendly views: release readiness score, defect escape rate, coverage trends.



Predictive Quality Scores

AI rates release risk based on test results, open bugs, and historical patterns.

PART 3

Beyond Traditional Manual Testing

AI catches what human eyes miss — and predicts what hasn't happened yet

Catching What Traditional Testing Misses



Boundary Conditions

- Min/max field values at exact limits
- Zero and negative input handling
- String length edge cases
- Date/time boundary rollovers



Concurrency Scenarios

- Race conditions in async processes
- Simultaneous multi-user edits
- Session conflict handling
- Database lock scenarios



Environmental Variance

- Low bandwidth / network latency
- Memory pressure scenarios
- Different locale & timezone combos



Data Combinations

- Special characters in all inputs
- Unicode & right-to-left text
- Extremely long data strings

TRY THIS PROMPT

"What edge cases might a QA engineer miss when testing [feature]? Think: boundary values, concurrency, data integrity, failure/recovery, locale, and adversarial inputs."

Predict Quality Issues Before They Surface

FROM REACTIVE → TO PREDICTIVE → TO PREVENTIVE



Then

Reactive

Find bugs after users report them in production



Traditional QA

Proactive

Find bugs before release through thorough testing



AI-Enhanced QA

Predictive

AI flags risky code areas before testing even begins



Future State

Preventive

AI suggests design changes to eliminate bug classes

TRY THIS PROMPT

"Here are our open defects and coverage data: [paste]. Rate this release 1-5 for readiness. List the top 3 risk areas and note which should block release vs. be accepted as known issues."

PART 4

What the Research Actually Shows

The honest picture — promise, progress, and the gap still to close

What the Research Actually Shows

Testing ranked #1 for AI value

60% of DevOps practitioners say testing is the most valuable area for AI investment — above coding, deployment, and planning.

Tricentis / Techstrong Research, 500+ DevOps practitioners, 2024

Human oversight remains essential

71% of practitioners check AI outputs at least half the time. Nearly 1 in 5 review every single AI output before trusting it.

Tricentis / Techstrong Research, 500+ DevOps practitioners, 2024

AI boosts individuals, not yet systems

75% of developers report personal productivity gains from AI. Yet at the system level, AI adoption correlated with a 1.5% drop in throughput and 7.2% reduction in delivery stability.

DORA Accelerate State of DevOps Report, 39,000+ professionals, 2024



What this means for QA teams

The gap between individual gains and system-level results is exactly where QA professionals add irreplaceable value.

AI is a tool. You're the strategy.

PART 5

Your AI-Enhanced QA Action Plan

Walk out of this room with a clear path forward

Three Phases to AI-Enhanced QA

1

Foundation (Weeks 1–4)

- ▶ Pick ONE workflow: test plan generation from requirements
- ▶ Set up Claude or ChatGPT with your team's test template
- ▶ Run AI + human side-by-side for 2 sprints, compare quality
- ▶ Document your prompt patterns in Confluence
- ▶ Measure time saved and coverage improvement

2

Integration (Months 2–3)

- ▶ Connect AI to Jira: auto-create test tickets from stories
- ▶ Implement AI bug report builder into your workflow
- ▶ Set up Confluence auto-update for sprint test summaries
- ▶ Expand to 2–3 more workflows based on Phase 1 learnings
- ▶ Train the full team on effective AI prompting

3

Leadership (Months 4–6)

- ▶ Build AI-powered quality dashboards for leadership
- ▶ Implement predictive quality scoring for releases
- ▶ Establish team prompt library & AI usage standards
- ▶ Present ROI to leadership and secure ongoing buy-in
- ▶ Become the internal AI-QA champion for your org

What to Remember

01 AI doesn't replace QA expertise — it amplifies it. Your domain knowledge makes AI outputs valuable.

02 Start with test plan generation. It's the highest-ROI, lowest-risk entry point for AI in QA.

03 The Atlassian ecosystem is your force multiplier. Jira + Confluence + AI = end-to-end quality workflow.

04 Metrics matter. Track time saved, coverage gained, and defect reduction to build internal support.

05 Lead from the front. QA professionals who master AI today will define quality standards for the next decade.

TEST PLAN

"Review this [user story + ACs]. Generate a test plan: happy path, negative tests, edge cases, performance, and accessibility. Format for Jira/Xray."

BUG REPORT

"Bug found: [plain description]. Write a Jira-ready report: summary, steps to reproduce, expected vs actual, severity, environment, and suggested labels."

COVERAGE GAPS

"Here are my ACs: [paste]. Here is my test suite: [paste]. Identify uncovered requirements and suggest 3 new test cases to close the gaps."

RELEASE RISK

"Given these open defects and coverage data: [paste]. Rate this release 1-5 for readiness. List the top 3 risks and note which should block release vs. be accepted."

TEST CASES

"Generate test cases for [feature]. Include 3+ negative scenarios and 2 boundary conditions. For each: preconditions, steps, expected result, and priority."

CONFLUENCE

"Summarize this sprint's test results for Confluence: total tests run, pass/fail rate, top defects by severity, and a release readiness recommendation."

EDGE CASES

"What edge cases might a QA engineer miss testing [feature]? Focus on: boundary values, concurrency, data integrity, failure recovery, and adversarial inputs."



Questions?

15 minutes for Q&A

Don't have a question? Try this prompt when you get back to your desk:

"Write a test plan for [your next user story] including edge cases, negative tests, and Jira import format."

Thank You!

The QA Revolution is yours to lead.

StarEAST 2026

Session materials in conference app

Keep the conversation going

Ask your AI questions any time

Starter Prompt Library

bit.ly/qa-ai-prompts (see session handout)

Connect on LinkedIn

Search: The QA Revolution Talk

STAREAST 2026

