# Introducing: the Souza Method.

# H.U.M.N.
# Human-centric, User-focused, Multidisciplinary, Nuanced

## A Practical QA Framework for the Age of AI

**Author: Peter Michael Souza Junior. August 8, 2025**

# The AI Production Paradox

## The Human Bottleneck is Now the Most Critical Point of Opportunity

- AI writes code FAST. It builds plans and graphics FAST.
- Planning, design, and implementation are merging into one hyper-fast cycle.
- All of this, but humans must still approve everything. The human review process is still the same bottleneck.
- With all this AI momentum upstream, QA cannot be stuck downstream at the end of the production cycle.

Human-centric, User-focused, Multidisciplinary, Nuanced

# The Solution: A Major Paradigm Shift

## Shifting From QA as a Final Gate to QA as a Continuous Partner

- The only way to keep pace and add value, is for QA to be present from project inception, as a fully invested, and fully informed risk partner.
- This requires a new mindset: Defined by The **HUMN** Method.
- **Human-centric**
- **User-focused**
- **Multidisciplinary**
- **Nuanced**

# 1. Human-centric

## Trust Your Instincts

- **Principle**: AI can follow specs, but it can't feel when a product is wrong. QA professionals must use their innate human intuition to question design and functionality on a fundamental level.
- **Action**: Conduct exploratory "gut-check" sessions on designs before a single line of code is written. Challenge counter-intuitive flows before the project plan is set in stone.
- **Value**: Catches costly architectural flaws, usability obstacles, and product inconsistency, all at the cheapest possible stage: the beginning.

# 2. User-focused

## Champion the End-User

- **Principle**: A product that is functional, but also frustrates users, is a failure. Every testing activity must be anchored to the experience of the person who will actually be using the product.
- **Action**: Test for usability, acting from a place of empathy, and build test scenarios from real user stories and user impact, not just technical specs.
- **Value**: Directly connects QA activities to user satisfaction, retention, and business KPIs.

# 3. Multidisciplinary

## Quality is a Conversation, Not a Monologue

- **Principle**: In a fast, AI-driven cycle, quality cannot be a silo. QA must be an integrated partner with Design, PM, Engineering, Legal, and Compliance.
- **Action**: Embed QA in design reviews and backlog grooming. Create shared regulatory checklists with Legal. Build a culture where QA is a collaborative resource, not a final roadblock.
- **Value**: Prevents silos, reduces rework, and mitigates compliance and legal risks before they become crises.

# 4. Nuanced

## See the Forest and the Trees

- **Principle**: A feature can meet every spec and still harm the overall product. QA must understand the real-world context of the entire system.
- **Action**: Conduct end-to-end workflow testing for every release. Deliver a high-level "Readiness Report" that assesses real-world viability beyond individual ticket acceptance criteria.
- **Value**: Protects the brand by preventing the release of features that work in isolation but fail in a practical, integrated workflow.

# Principles into Practice

## The Souza Format Templates

- A methodology is only as powerful as its practice. **Principles** are good, **guidance** is better, and **habits** are ***Make Or Break***.
- The Souza Format templates embed the **HUMN** mindset directly into your team's daily workflow in tools like Jira, Shortcut, or Monday.com.

# HUMN: PM Team Epic Template

## Ensuring a Holistic View from Day One

- Purpose: Cross-team engagement, Nuanced overview, Minimal changes through development cycle.
- Key Fields:
  - **Title**: Clear, concise name of the epic.
  - **Description**: Detailed overview of the epic's purpose and its connection to larger business goals.
  - **Feature Elements & Design Requirements**: A list of the major feature objectives to be accounted for and links to design requirements.
  - **Associated Tickets**: A checklist of all child tickets required to complete each feature element and the epic.
  - **Technical Considerations & Dependencies**: Notes on technical risks or dependencies on other teams/epics.
  - **Stakeholders**: Key contacts from PM, Design, Engineering, Legal, etc.
  - **Overall Acceptance Criteria**: The high-level conditions that define the success of the entire epic.
- Example: E-commerce checkout epic with stakeholder roles
- Why It Matters: Aligns teams, ensures QA involvement early

# HUMN: Engineering Work Template

## Translating User Needs into Technical Action

- Purpose: User-focused reasoning via technical communication
- Key Fields:
  - **Title**: Clear, concise summary of objective.
  - **Description**: Brief overview of the task, its purpose, and links to relevant designs.
  - **Implementation Details**: The proposed technical approach.
  - **Acceptance Criteria**: A list of requirements that must be met for the ticket to be considered complete.
- Example: Login feature ticket with details and acceptance criteria
- Why It Matters: Aligns developers and QA on user expectations

# HUMN: QA Bug Template

## Focusing on Context, Not Just Defects

- Purpose: Human-centric, User-focused bug reporting
- Key Fields:
  - **Title**: Clear, concise summary of error.
  - **Current Result**: Description of the observed issue or bug.
  - **Expected Result**: Explanation of the intended or expected behavior.
  - **Steps to Reproduce**: Detailed steps necessary to replicate the issue.
- Example: Broken checkout button with user impact (“Blocks purchase”)
- Why It Matters: Prioritizes user experience in bug fixes

# HUMN: QA Automation Template

## Ensuring Automation Delivers Real Value

- Purpose: Nuanced, targeted automation
- Key Fields:
  - **Title**: Clear, concise statement of objective.
  - **Automation Goal**: What is the primary purpose of this automated test? (e.g., "Verify the complete user checkout flow with a valid credit card.”)
  - **Initial Status**: The state of existing automaton for this task (e.g., None, Flaky, Failing).
  - **Linked Manual Case #**: The ID of the manual test case this script is automating. This maintains a clear link between manual and automated suites.
  - **Test Data Requirements**: Specific data needed for the test to run (e.g., test account credentials, specific product SKU).
  - **Dependencies**: Any preconditions required for the test to run (e.g., environment setup, dependent services must be active, feature flags enabled).
- Example: Shopping Cart Checkout flow automation with preconditions and assertions
- Why It Matters: Aligns automation with user-focused goals

# HUMN: Pull Request Template

## Merging Code with Clarity and Empathy

- Purpose: Communicates empathy by anticipating reviewer needs
- Key Fields:
  - **Title**: A clear, conventional title describing the change (e.g., [FEAT] Add user profile avatar upload or [FIX] Correct tax calculation in checkout).
  - **Description of Change**: A high-level summary of the purpose of the PR. What problem is being solved? What is the goal?
  - **Linked Ticket(s)**: Direct link(s) to the associated Engineering Ticket(s) or PM Epic(s). This immediately provides the "why" behind the PR.
  - **Implementation Details**: A brief overview of the technical approach. Were there any key decisions, trade-offs, or new libraries introduced? This section is primarily for fellow engineers.
  - **How to Test**: Clear, numbered steps that a QA professional or another engineer can follow to verify the changes. This is the most critical section for ensuring the change works as expected.
- Example: Updated component package used in the shopping cart
- Why It Matters: Communicates the Why and How, limiting PR review/revise cycle time

# HUMN: How-To Documents

## Defeating Knowledge Silos, One Process at a Time

- Purpose: A knowledge multiplier and source of truth.
- Key Fields:
  - **Title**: An action-oriented title starting with "How to..." (e.g., "How to Publish a New Component to the Design System").
  - **Objective**: A single sentence explaining what the reader will be able to accomplish after following this guide.
  - **Prerequisites**: A checklist of anything the user must have or know before they begin (e.g., “Two users with Admin permission”).
  - **Step-by-Step Instructions**: Unambiguous Actions, Single Steps, and Screenshots (Guide someone who lacks product knowledge).
  - **Verification**: A clear description of the expected outcome. How does the user know they have successfully completed the task? (e.g., "You will see a 'Task Successful' message in Slack.").
  - **Troubleshooting / Common Issues**: A section detailing potential errors and how to resolve them.
  - **Who to Contact**: Domain experts, Authors, and Responsible Parties.
- Example: A document that outlines how to mark a retail product with temporary price drop
- Why It Matters: Reduces institutional siloing in legacy features

# HUMN: Plain Language Documentation Template

## The Single Source of Truth for Feature Behavior, Given/When/Then

- Purpose: The GWT Testing Document is the ultimate tool that forces clear communication and agreement on feature behavior for non-technical team members
- Key Fields:
  - **Feature Name**: A high-level name for the feature being described (e.g., Feature: User Login).
  - **User Story**: A brief, one-sentence description of the goal from a user's perspective (e.g., "As a registered user, I want to log in to my account so that I can access my dashboard.").
  - **Scenario**: A descriptive title for specific test cases (e.g., Scenario: Successful login with valid credentials).
  - **Given:** [The context or precondition]. A state that must be true before the test begins. (e.g., Given I am on the login page).
  - **And:** [An additional precondition]. (e.g., And I have a valid, registered user account).
  - **When:** [The action performed by the user]. A single, specific action. (e.g., When I enter my correct email and password and click the 'Sign In' button).
  - **Then:** [The expected outcome]. The verifiable result of the action. (e.g., Then I should be redirected to my account dashboard).
  - **And:** [An additional outcome]. (e.g., And I should see a welcome message with my name).
- Example: Authentication flow for a subscription based checkout page
- Why It Matters: This document serves as the acceptance criteria for Legal team verification as well as Engineering Management.

# HUMN: README.MD

## Onboarding Your Team with Clarity

- Purpose: A well-crafted README is an act of empathy. It anticipates the questions a newcomer will have and answers them preemptively.
- Key Fields:
  - **Project Description**: A clear title and a one-sentence description of what this project is and does.
  - **Core Features**: A high-level bulleted list of the main functionalities.
  - **Technology Stack**: A list of the key frameworks, languages, and libraries used.
  - **Prerequisites**: A checklist of software or tools that must be installed on a local machine before proceeding (e.g., Node.js v18+, Docker).
  - **Getting Started**: Quickstart or initial state confirmation (Local Installation, Dependencies, Environment variables, Running)
  - **Running Tests**: The command(s) needed to execute the test suite (e.g., npm test).
  - **How to Contribute**: A link to a CONTRIBUTING.md file or brief instructions on the team's branching and PR strategy.
- Why It Matters: A strong README dramatically reduces dependency on senior team members, and establishes a professional standard for the project.

# The Old Way vs. The HUMN Way

## A Fundamental Change in the QA Mission

**The Old Way (The Gatekeeper):**

- QA is at the end of the cycle.
- Primary Role: Validate tickets.
- Focus: "Did we meet requirements?"
- Risk Scope: Functional bugs.
- Result: A bottleneck that causes lifecycle churn.

**The HUMN Way (The Partner):**

- QA is involved from project inception.
- Primary Role: Assess risk and champion the user.
- Focus: "Did we build the right thing, right?"
- Risk Scope: Functional, Usability, Legal, and Brand.
- Result: A catalyst that accelerates quality delivery.

# Implement HUMN Today!

- **Easy Wins** (This Week - Individual Actions):
  - Use the QA Bug Template for your next bug report.
  - Examine your project's README and identify one improvement.
- **Team Plays** (This Month - Collaborative Actions):
  - Run one "Gut-Check" session on a new feature before development starts.
  - Introduce and adopt the PR Template for all new code merges.
  - Create one How-To document for a critical team process.
- **Cultural Shifts** (This Quarter - Process Changes):
  - Invite QA Leads to all new project design reviews.
  - Adopt the QA Bug, QA Automation, and PR templates for one complete project.

# HUMN: Q&A

- How will you apply HUMN in your work?
- Contact: Petermsouzajr.com petermsouzajr@gmail.com