

Cognicions

...Technology enablers & facilitators

SUCCESSFULLY DELIVERING USER ACCEPTANCE TESTING

FOR YOUR PROJECT

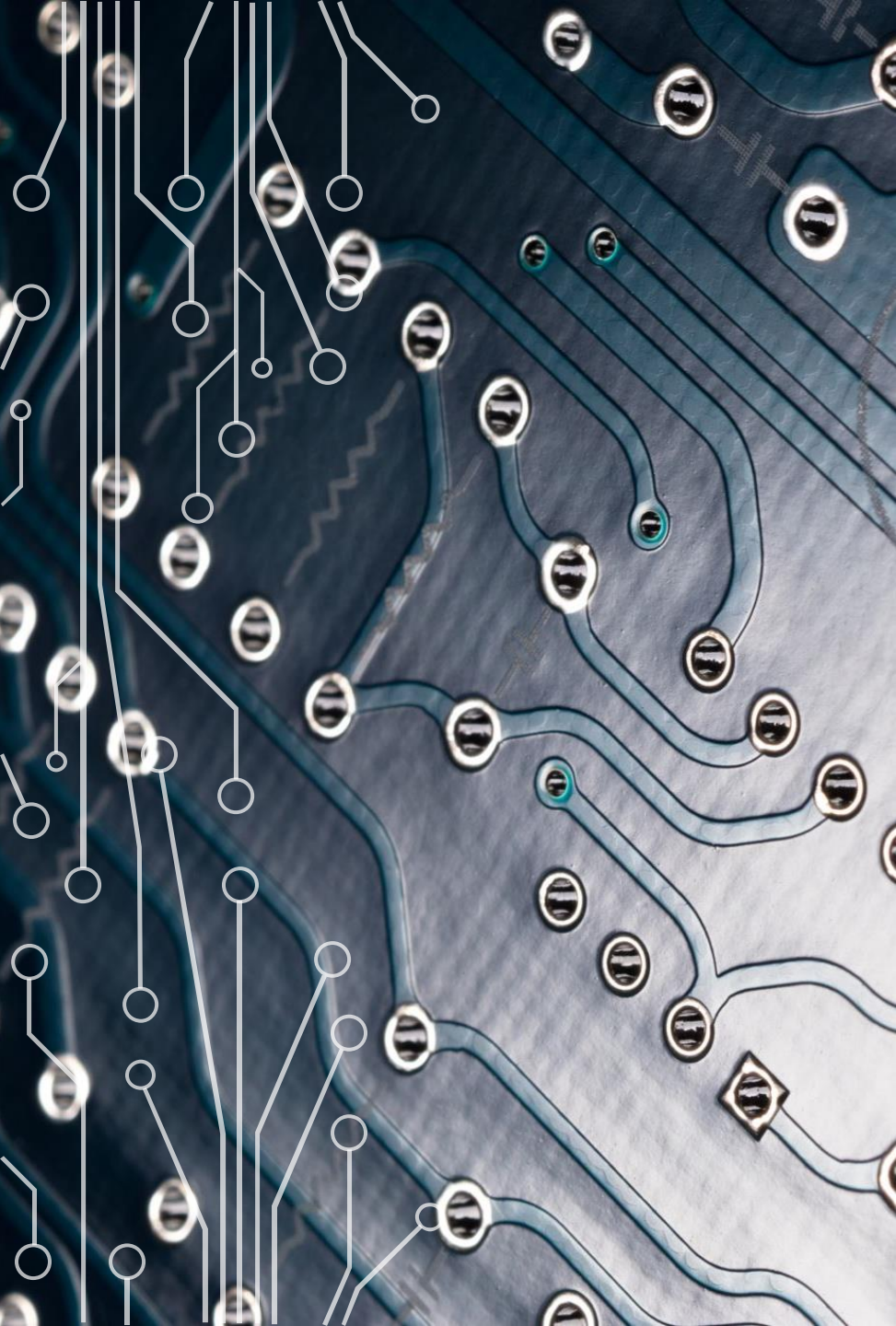
RONALD KOHLMAN

USER ACCEPTANCE TESTING – APPROACH

- ❖ To set up for success with User Acceptance Testing you will need a framework for defining the approach, effort, and capability for User Acceptance Testing your software delivery project.
- ❖ Following a framework ensures that testing delivers its outcomes, helping to make software better through verifying the correctness, completeness, and quality for the developed software.
- ❖ UAT is no different in its approach than any other technology testing phases, it needs structure.

WHAT IS USER ACCEPTANCE TESTING (UAT)?

- ❖ User Acceptance Testing (UAT) is a critical phase in the software development lifecycle where the end-users, or their representatives, evaluate a software application to ensure it meets their requirements and is fit for its intended purpose.
 - ✓ Validates end-to-end business processes
 - ✓ Validates system transactions
 - ✓ Validates user access
 - ✓ Confirms non-system requirements have been delivered to support the solution
 - ✓ Acceptance Testing is the process of validating that a completed system complies with its documented requirements
 - Whilst, for example, System Testing is modular and process specific, Acceptance Testing focuses on the interaction between completed modules



WHY IS UAT ESSENTIAL?

- ❖ UAT is vital because it helps identify issues and defects (bugs) that may have been overlooked in earlier testing phases.
- ❖ It also ensures that the software product aligns with user expectations, improving user satisfaction and reducing post-release defects.

IMPORTANCE OF TESTING

- ❖ Important reasons why testing should be ranked as a top concern, especially at the start of a project:
 1. A poor testing program can:
 - a. cause mission failure
 - b. significantly impact operational performance and reliability, and
 - c. double or triple field support and maintenance costs
 2. A good testing program is a major project cost
 3. Complex programs can spend more than half their total program effort on testing activities
 4. To make testing effective, you must take the time up front to plan and organise it properly
 5. A good testing program will help significantly as you define your early requirements and design work. This help is critical, and it can have a major influence on overall project success
 6. A good testing program forces you to face and deal with problems as the work is done, and when the cost of rework and fixes is much lower
 7. A good testing program can't totally make up for poor software quality, but it does help prevent many ills and will at least let you know you are in trouble early
 8. Testing is insurance, many have learned the hard way

UNDERSTANDING UAT

- ❖ User Acceptance Testing (UAT) is a critical phase in the software development lifecycle, focusing on ensuring that the software meets the needs and expectations of its intended users.
- ❖ It is a key delivery from any project.
- ❖ The business stakeholders are generally responsible for delivery of UAT execution, using their end users to validate the new solution meets their needs.

THE UAT PROCESS

- ❖ The UAT process typically consists of several well-defined phases, each with its own objectives and activities. Understanding this process is essential for a successful UAT implementation.



- ❖ **Planning:** produces test strategies, approach, objectives, and plans
- ❖ **Preparation:** produces detailed test plans and test cases / scripts based on the test objectives and conditions; and establishes the test environment
- ❖ **Execution:** the execution of the cases / scripts against the application(s) and the recording of test results
- ❖ **Reporting:** the reporting of test progress and results, defects found, metrics, and test assessment reports

KEY COMPONENTS OF THE UAT PROCESS

**Test
Planning**

**Test Case
Design**

**Test
Execution**

**Defect
Reporting**

**Regression
Testing**

**User
Feedback
Collection**

**Approval
and Sign-Off**

UAT TEST TYPES

Type	Definition
Requirements Testing	Testing compliance with the Business Requirements.
Usability testing	Testing the usability and consistency of the application.
Business Process Testing	Testing of new or re-engineered business processes and ensuring that any systems processes integrate with the manual processes. The user wants to determine if the new software/release can support the business process
Documentation Testing	Assess whether User Guides, Procedure Manuals, and Online Help are fit for purpose.
Regression Testing	Regression testing of any existing and unchanged components that the business deems necessary to test.
Business Scenario Testing	Testing of existing or new business scenarios, ensuring that the transactional flow of these scenarios is valid. The user may want to assess the critical business scenarios with one or two critical variations.
High Risk Transactional Testing	<p>The thorough testing of any high-risk transactions that is, high risk for the business. Most businesses have one or two critical transactions that MUST be processed correctly.</p> <p>High risks are for example any of the following: Complex; extensive; likely to be buggy; high \$ value and are based on a risk analysis by the business.</p>



PLANNING FOR UAT

Setting Clear UAT Objectives

Identifying Test Scenarios

Business Scenarios

Acceptance Criteria

Environment Preparation

- ❖ Effective planning for UAT sets the stage for a successful testing phase.
- ❖ It ensures that testing goals are well-defined, resources are allocated appropriately, test scenarios are identified, testers and test environments are selected, and criteria for success are established.

TEST CASE DESIGN

- ❖ Designing UAT test cases involves the creation of detailed testing scenarios based on the test objectives and user requirements. Well-designed test cases ensure thorough test coverage.
 - **Start with Requirements:**
 - Begin by referencing the user requirements, user stories, and acceptance criteria to identify what needs to be tested. These are used to define your test cases or test scripts.
 - **Use Realistic Business based Scenarios:**
 - Create test cases that mimic real-world usage. These should represent the most common and critical user actions.
 - Keep in mind that you are not replicating the testing that has already been performed by the delivery team, e.g. systems functional testing.
 - **Test Data:**
 - Define the data and inputs required for each test case. Ensure that the test data reflects different usage scenarios and user profiles.
 - **Positive and Negative Testing:**
 - Include both positive test cases (valid inputs leading to expected outcomes) and negative test cases (invalid inputs or actions leading to error conditions).
 - **End-to-End Testing:**
 - For complex systems, consider end-to-end testing that covers the entire business scenario transactional journey.

- ❖ You need to have an environment to perform your UAT within. These are some of the questions you should ask:
 - Why do you need it?
 - What do you need?
 - How will you get it ready?
 - When will you need it?
 - What should it look like?
- ❖ UAT environments must be established prior to the commencement of the "Execute Testing" test activity.
- ❖ Inadequate or not ready test environments will result in delays starting execution, significant cost overruns and ineffective defect management
- ❖ Establishing a test environment includes shaking it out, prepare test cases to achieve this
- ❖ Test environments should be established in time to allow any test data creation and software component migration to occur

UAT ENVIRONMENT



TEST EXECUTION

- ❖ This is where the planned test cases and scripts are put into action.
- ❖ It covers the test execution process, recording defects and issues, collaboration with testers, and the critical aspect of managing the test environment.
- ❖ The software is evaluated against the acceptance criteria.
- ❖ Ensure that you are ready to start
 - Satisfy your Entry Criteria
- ❖ Test Execution cycle
 - Confirm test readiness for each “Drop” using predefined entry criteria
 - Execute relevant Regression test cases
 - Execute test cases for relevant Drop’s features
 - Coordinate with external parties as necessary (e.g. Vendor)
 - Document execution results in test management tool
 - Capture and categorise defects in test management tool
 - Execute defect resolution process
 - Report on testing results, status, and issues

DEFECT REPORTING



Recording and managing defects and issues is a critical part of UAT.



A defect is:

An incorrect step, process, or data definition in a computer program
Discrepancies could be an error, a Defect, an environment problem or a problem with the test or data



Causes of Software Defects:

Human error, mistakes do happen, which produces a Defect (fault or bug) in the code, in software or a system, or in a document

If a Defect in code is executed, the system will fail to do what it should do (or something it shouldn't), thus causing a failure

Defects may result in failures, but not all Defects do so

Failures also can be caused by environmental conditions, data, infrastructure...

- ❖ Your UAT testers from the business may not necessarily have skills or training in the defect management tool that you propose to use for tracking your defects within.
- ❖ Experience has shown that involving the business testers in using the prescribed tool to a) write test cases, b) execute test cases, and c) write up defects has ongoing benefits to the tester's skills, the management of the project's (and future) UAT efforts, and the organisation's retention of knowledge.

UAT PROCESS

Test Planning:

In this phase, the UAT team outlines the test strategy, defines objectives, and establishes criteria for success. It's crucial to plan thoroughly to ensure that UAT aligns with the project's goals and user expectations. This plan outlines the scope, resources, schedule, and acceptance criteria for the testing process.

Test Case Design:

Test cases are designed and written based on user requirements and expected workflows. Create business scenarios that mimic real-world usage, focusing on critical paths and user interactions. Business scenarios and key end-to-end business transactions are the source of your test case design, ensuring that testing aligns with the end-users' perspective.

Test Execution:

Testers execute the test cases, either manually or with automation tools, to identify defects and assess the software's performance. Testers execute test cases and assess the software's performance against predetermined acceptance criteria. During this phase, it's essential to record detailed information about any issues encountered.

Defect Reporting:

Defects or issues identified during test execution phase by the testers are recorded, tracked, and prioritised. These reports are critical for developers to understand and fix any problems. The project team collaborates to resolve these issues promptly.

Regression Testing:

After defects are addressed, regression testing is conducted to ensure that the changes made do not introduce new issues or negatively impact other areas of the software.

User Feedback Collection:

Beyond defect identification, testers collect valuable feedback from users regarding their experience with the software. This information can lead to improvements in usability and overall satisfaction.

Approval and Sign-off:

Once the software meets the defined acceptance criteria a UAT report is prepared, summarising the testing process, results, and any outstanding issues. This report is often approved by the user community and used to determine whether the software is ready for production release.



COMMUNICATION AND COLLABORATION

- ❖ Communication and collaboration are key to a successful UAT phase:
 - **Stakeholder Engagement:**
Actively involve stakeholders, including end-users, business analysts, and developers, to ensure alignment with project goals and user expectations.
 - **Regular Meetings:**
Schedule regular meetings to discuss progress, challenges, and feedback. These meetings can help resolve issues promptly and improve the overall testing process and the quality of the product.
 - **Clear Documentation:**
Ensure that all test documentation, including test cases, scripts, and defect reports, is clear, well-structured, and easily accessible to all team members.
 - **Feedback Channels:**
Establish channels for collecting user feedback and encourage open and honest communication. User feedback can be invaluable for improving the product.
 - **Defect Resolution Collaboration:**
Foster effective collaboration between testers and developers to expedite defect resolution. Encourage clear and structured communication about defects.

A magnifying glass is positioned over a bar chart. The chart has two groups of bars labeled 'Q2' and 'Q3'. Each group contains two bars, one blue and one green. The magnifying glass is focused on the 'Q2' group. Overlaid on the left side of the image is a white circuit board pattern with lines and nodes.

CHALLENGES IN UAT

- ❖ Despite its importance, User Acceptance Testing (UAT) is susceptible to various challenges and pitfalls that must be recognised and addressed, including:
 - Inadequate Test Planning
 - Lack of User Involvement
 - Ambiguous Acceptance Criteria
 - Defect Mismanagement
 - Insufficient Test Data
 - Resistance to Change
 - Scope Creep
 - Resource Constraints

KEY TAKEAWAYS

UAT Essentials:

UAT is a critical phase in the software development lifecycle, focusing on user satisfaction and validating that the software meets business requirements.

Phases of UAT:

UAT consists of several phases, including planning, test design, execution, defect management, and reporting.

Best Practices:

Effective communication, collaboration, test data management, automation, and user-centred testing are essential best practices for successful UAT.

Challenges and Solutions:

UAT can face challenges, including scope creep, resource constraints, and resistance to change. However, these challenges can be addressed with proper planning and collaboration.

UAT Tools and Technologies:

UAT management software, test data management tools, and collaboration and communication tools play a crucial role in streamlining the UAT process.

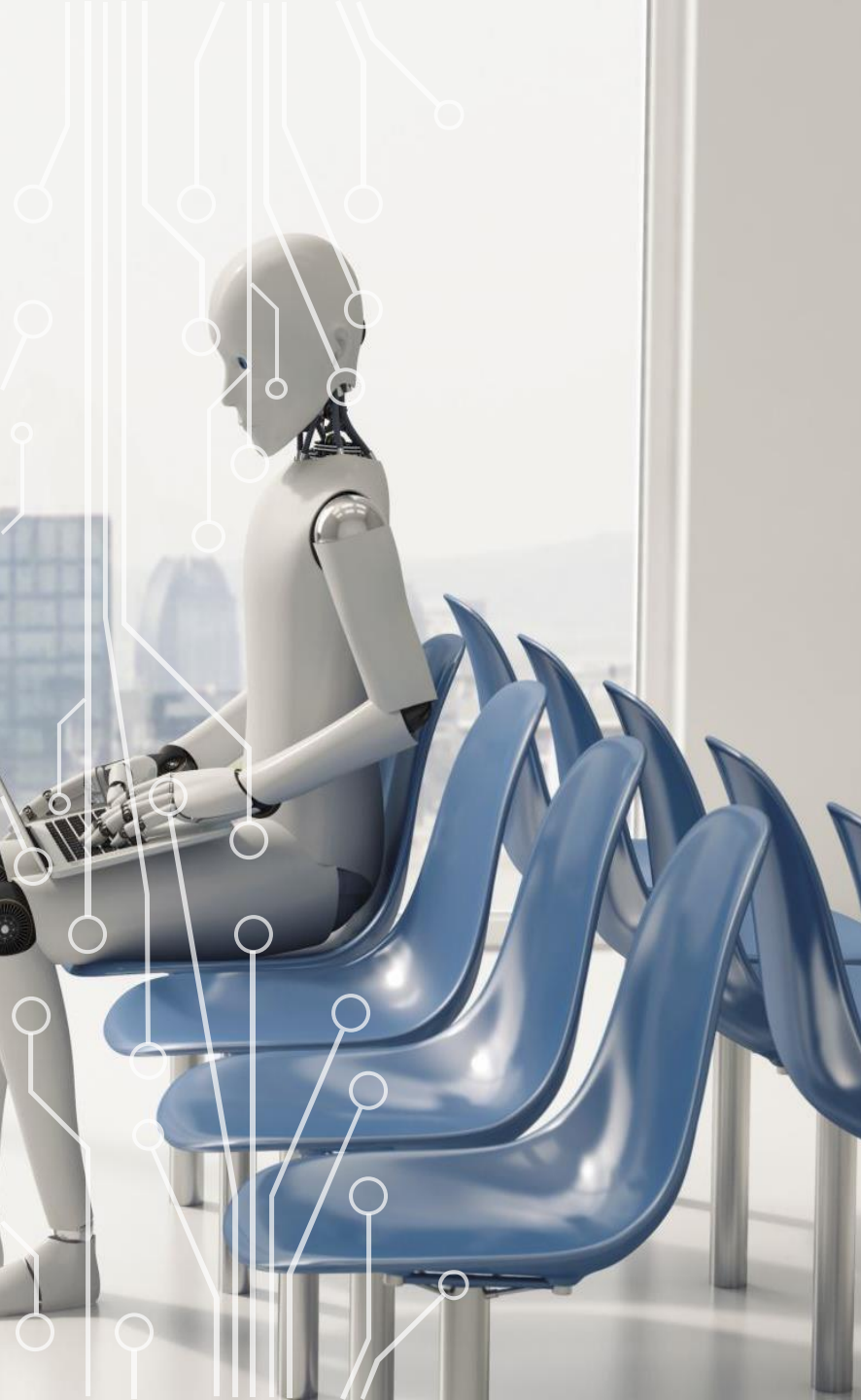
Collaboration and Communication Tools:

Industry-Specific UAT:

UAT in regulated industries, such as healthcare, pharmaceuticals, and financial services, must adhere to specific regulatory requirements and prioritise data privacy and security.

Future Trends:

The future of UAT involves the integration of AI and machine learning, UAT for VR and AR, and UAT in the IoT.



ABOUT RONALD

- ❖ Ronald is a highly experienced and knowledgeable IT professional in the field of program and test management.
- ❖ He has had many roles working across transformational initiatives and complex enterprise technology solutions.
 - Leadership in Transformational Programs
 - Global Experience and Cross-Continental Team Leadership
 - Governance Frameworks and Tools
 - Delivery of Complex Technology Solutions
 - Executive-Level Engagement and Consulting
- ❖ He has been writing and publishing technology industry specific documents for several years. Imparting his practical working experience within these documents.
- ❖ You can purchase his book:
“**Successfully Delivering User Acceptance Testing for your Project**” on Amazon at: <https://amzn.asia/d/bCleko7>
- ❖ He also has another book on UAT available:
“UAT Planning Guide” on Amazon at: <https://amzn.asia/d/aEY33P>

Cognicions

...Technology enablers & facilitators



COGNICIONS.COM



COGNICIONS PTY LTD

ABN 83 611 219 642



MELBOURNE

PO BOX 125, OLINDA,
VICTORIA 3788



+61 (0) 402 448 050



INFO@COGNICIONS.COM