Understanding Quality through the Consumer's Eyes

Reality Unveiled During PAS UAT

Michelle Dunn
Calgary, Alberta
mandjdunn@shaw.ca

Introduction

- Software development teams typically never use the application they are responsible for creating.
- Join me in the journey of discovering what happens after the software vendor has released their product.

Ready … Set … Ship!

- Your company is now ready to deliver a new software product to the customer.
- You have met the schedule, the scope and the quality requirements of the contract.

… More Testing???

- You have completed various forms of testing throughout the SDLC
- Your daily automated suite contains
  - 20,000+ JUNIT tests
  - 1,000 GUI (front end tests)
  - 350 System tests
  - 100 Report tests
- So why does your customer still feel the need to do rigorous User Acceptance Testing? What is UAT?

Why the Secrecy Surrounding UAT?

- Although most large companies perform UAT prior to deploying an application, there is limited articles/books published on the subject by leading Test Experts.
  - I could not find one reference to UAT in my favorite book "Lessons Learned in Software Testing" by James Bach.
  - The bestselling test book of all time "Testing Computer Software" by Cem Kaner has only one paragraph on the subject.
- If it's not talked about, then it cannot be important … right?

UAT Redefined

- A Common Definition (coined by me!)
  - "UAT is performed to determine if the application will support the needs of the user in the real world."
- What does this mean?
- Let's probe further and dissect the term UAT

UAT: Who is the "User"

- Who will be affected by the project/application?
  - Project Team
  - Support and Operation Team
  - End User
  - End User's customer
  - Legal or regulatory authority
  - Monetary authority
  - Anyone else?

UAT: What is "Acceptance"?

- "Acceptance Testing is any testing done by one party for the purpose of accepting another party's work"
  - Michael Bolton, 2001
- What testing needs to be performed to determine if the software application is acceptable?

UAT: What is "Testing"

- "Testing is questioning a product in order to evaluate it"
  - James Bach, "Rapid Software Testing", 2007
- Test Techniques:
  - Functionality
  - Regression
  - Scenario
  - Reliability
  - End to End
  - Usability
  - Security
  - Scalability
  - Performance
  - And more "ility" Tests

What is Your Company's UAT Goal?

UAT Goal: Beta Release

- User base tests the application
  - Example: Book4Golf
  - Rigorously tested by the software development team
  - Mass software release where users were asked to test
  - Beta users had no motivation to test, thus only usability bugs were communicated to team
  - UAT:
    - User: Public Tested (no test experience)
    - Acceptable: Acceptable to the Shareholders
    - Testing: No Testing Performed

UAT Goal: Rubber Stamp

- Demonstration of product to users
  - Example: Neteller
  - Rigorously tested by software development team
  - UAT Performed the day before release
    - 30 Minute Scripted Test Performed by Customer Support
    - Demo to the Product Managers
  - Bugs found would not stop the release
  - UAT:
    - User: Customer Support, Product Managers
    - Acceptable: Yes, Acceptable to the Shareholders
    - Testing: Scripted and Demo only

UAT Goal : Thorough Form of Testing

- Test the application End to End
  - Example: PAS
  - Rigorously tested by Development Team
  - UAT performed 8+ months before Go Live
    - Performed by SME and trained Testers
    - Test to validate functionality and to find bugs
    - Bugs found could stop the release
  - UAT:
    - User: All users that would be touched by application
    - Acceptable: Yes, Acceptable to the Advisory Board
    - Testing: All forms of testing

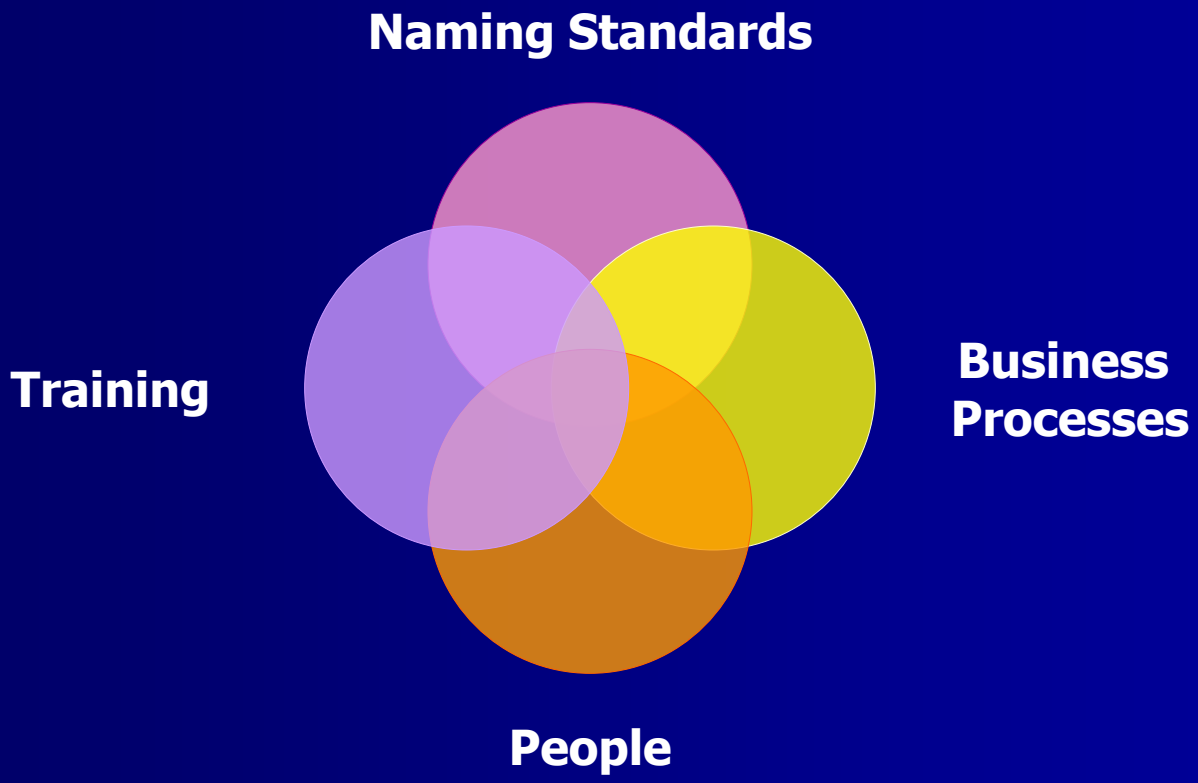
Why Are UAT Goals so Different?

- Mandate to use application?
- Mission Critical application?
- Size and Complexity of application?
- Risk to Organization?
- Risk to Life?
- Monetary Risk?

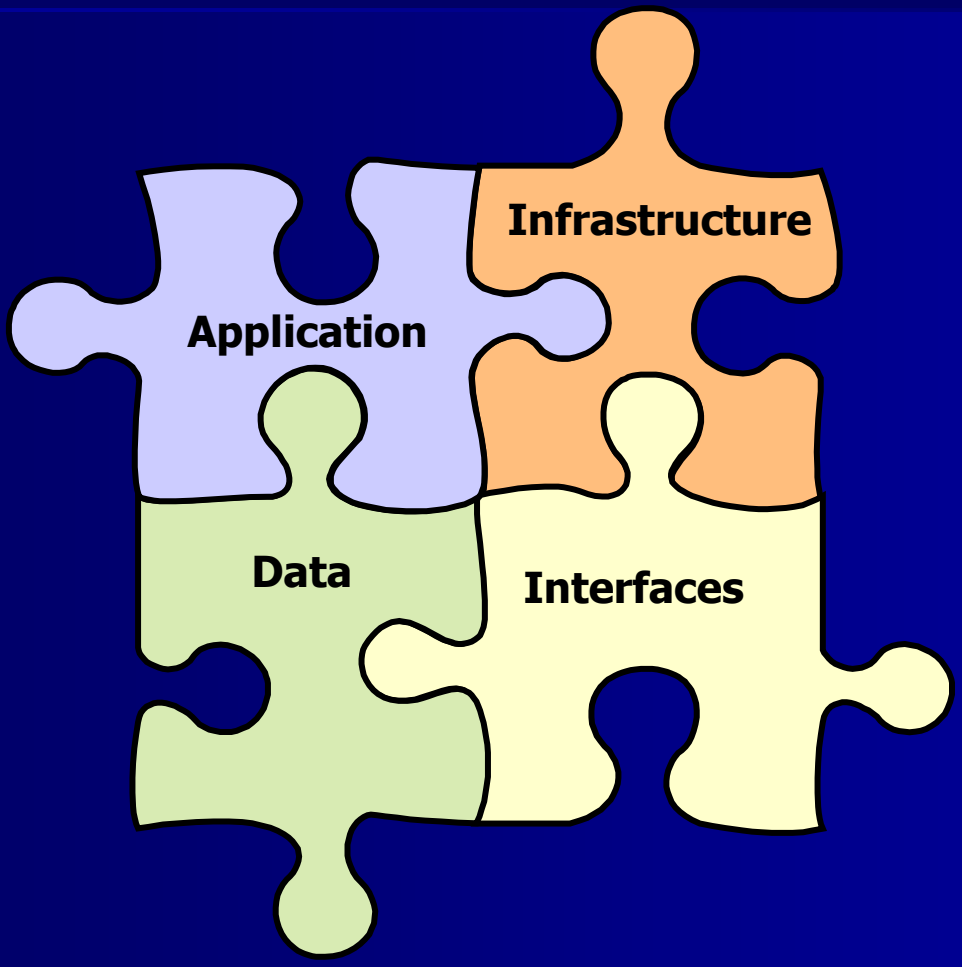
Concerns a Customer Has

1. Will the client deliver the application on time, in scope and with quality?
2. Will the application work with my infrastructure, with my data load and with my user load?
3. Will the application integrate with interfaces and secondary applications easily?
4. Will I meet my business timeframes and regulatory requirements?
5. Will the Service Level Agreements be met?
6. Will the Application Upgrade without data corruption? Will functionality regress?

Application Usage



The Application is One Piece of the Puzzle



What is UAT Test Coverage?

- "Test Coverage is the proportion of the product that has been tested"
- Additionally Test Coverage for UAT not only includes testing the application but also:
  - Testing using real environments, with real data, with real business scenarios , with real users
  - End to End testing from "cradle to grave"

Determining your UAT Test Coverage

ASK: What is the complete picture?

1. Application
2. Interfaces
3. Environment
4. Data
5. Performance
6. Security

Determining your UAT Test Coverage Cont'd

- Application
  - Business Functionality
    - Business Processes
    - Application Functionality (80% rule)
    - Complex Functionality
- Interfaces
  - Push to or Pull from (Applications, Databases, Secondary Applications, Reports, Etc.)
- Environment
  - Platforms /Configuration / Infrastructure

Determining your UAT Test Coverage Cont'd

- Data
  - Data Characteristics
  - Naming Standards
- Performance
  - Data Load
  - User Load
  - Timeframes
- Security

What is UAT Risk Based Testing?

Testing the real and potential risks that threaten the quality of the application

Defining Test Risk

- Using the Test Coverage Matrix created, ask the Business
  - What Must Work?
  - What Do You Value?
  - What Threatens the Value of the Product?
  - Brainstorm functional, technical, performance, security risks that the test coverage matrix may have missed

Categorize the Test Risk

- High, Medium, Low
- How bad would it be if this problem occurred?
- What are the chances that this problem can occur?
- Has it been tested by the Development Team?
- New or Legacy Application?

Create Tests to Exploit Risk

- What tests must be run to exploit these risks?
- What business processes, data, infrastructure, and interfaces must be used in the test?
- Continually Ask the Test Team

  "Are we Testing the Right Stuff the Right Way?"

Finding UAT Defects

- All High Severity bugs found will not be fixed
- Bugs that are a nuisance in testing become issues in PROD where:
  - Workarounds need to be developed
  - Training needs to be incorporated
  - Users must live with the bug until the bug is fixed and that release is in PROD

UAT Regression Testing

- Test Risk evolves once the application is Live and in Production
- Review Test Risks from before and question the test's current value
  - Why repeat the test?
  - What is it's return on investment (ROI)?
  - Can the test be run the same way?
- Add new tests to exploit new risks
  - Data Migration and Data Integrity
  - New Functionality and Bug Fixes

Questions

?

Acknowledgements

- Sabina Fabbian