Maximizing your success in user acceptance testing

Holly Robley, Product Manager
May 15, 2007
GTC West Education Days

What is user acceptance testing?

- General Concepts
  - What: UAT reviews how a system or software is handled by end users (proves things are "good to go")
  - Where: UAT is done on staging or production environments *outside* of the development (or engineering) process
  - When: UAT is completed near a project's release date
  - Who: UAT testers are not part of the technical team that developed the system or software
  - WHY: UAT can show a project team how effectively the system or software meets user requirements

How do we know how to do UAT?

- The Manual of Certified Standards in User Acceptance Testing
  - Where is the UAT handbook?
  - What steps are required to complete UAT?
  - Is there a standard that should be met?
  - How do other organizations do UAT?
- The "Real World" of UAT
  - Who is in charge of UAT?
  - Is UAT worth the effort?

iQurious
Discover new ways to simplify.

Background Information

- Experience is a great teacher
  - The "Power User" & the IBM mainframe
  - Data Coordinator, Tech Support, QA Engineer
    - Root Cause Analysis (PITs, Audits, etc.)
    - Beta Coordinator
    - CAB Administrator (Remedy lead, process enforcer)
- Attitude + Job Requirements = Involvement in UAT

iQurious
Discover new ways to simplify.

So: what are best practices?

- The real world
  - Should UAT be done only by the "Power User"?
    - What kind of feedback should UAT give?
  - Does it work to have large groups involved in a UAT?
    - What is needed to handle large groups?
  - What kinds of tests have we (collectively) done?
  - What kind of participation rates have we seen?

iQurious

What are the goals for your organization when you do user acceptance tests?

iQurious

What are your goals?

- Class is conducted in small-group format
  - Arranged in groups so that each of you gets a chance to discuss the class materials
  - You are each encouraged to share knowledge and ideas
  - We'd like to hear your ideas and build on them during today's session
- Identify your group's note taker, team lead, and speaker
- Take a few minutes to discuss what you want from this class
- Share your conclusions with the larger audience

iQurious
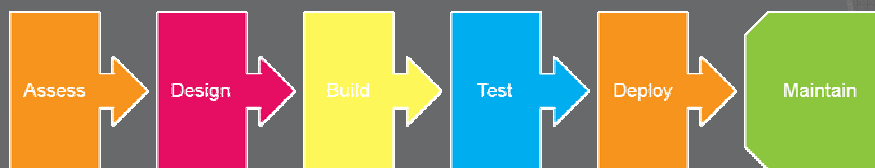Discover new ways to simplify.

Agenda

- Defining UAT in your environment
- Creating a UAT plan
- Techniques for maximizing success
- Presenting the data
- Hands-on workshop

- Each participant will receive templates, sample questions, and other materials on a CD at the end of today's class
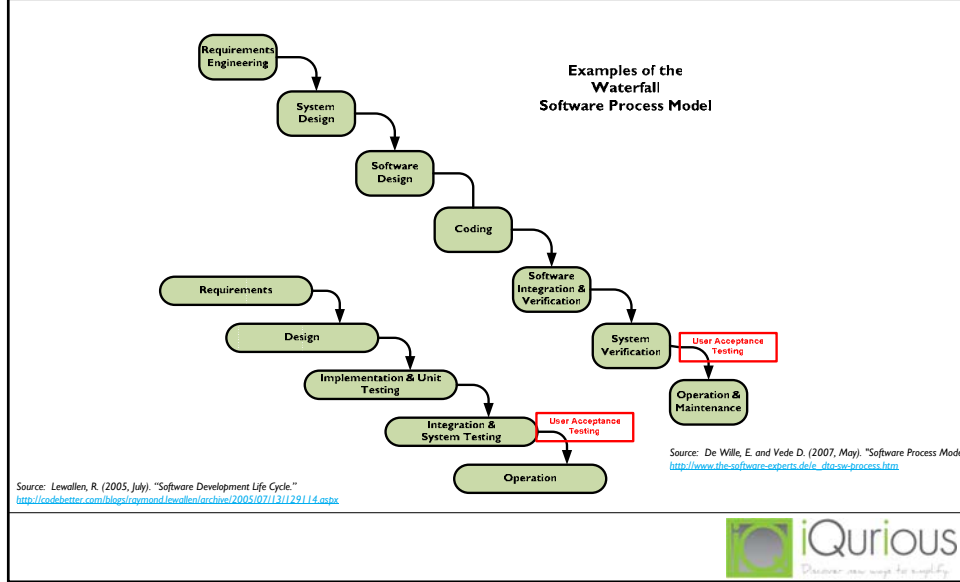
iQurious
Discover new ways to simplify.

Defining UAT in your environment

iQurious

Process Models

The bottom line of every life cycle (development method)

iQurious

Waterfall Process Model

Examples of the
Waterfall
Software Process Model

Source: Lewallen, R. (2005, July). "Software Development Life Cycle."
http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx

Source: De Wille, E. and Vede D. (2007, May). "Software Process Model."
http://www.the-software-experts.de/e_dta-sw-process.htm

The V-Shaped Process Model

Examples of the
V-Shaped
Software Process Model

Source: Lewallen, R. (2005, July). "Software Development Life Cycle."
http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx

Source: De Wille, E. and Vede D. (2007, May). "Software Process Model."
http://www.the-software-experts.de/e_dta-sw-process.htm
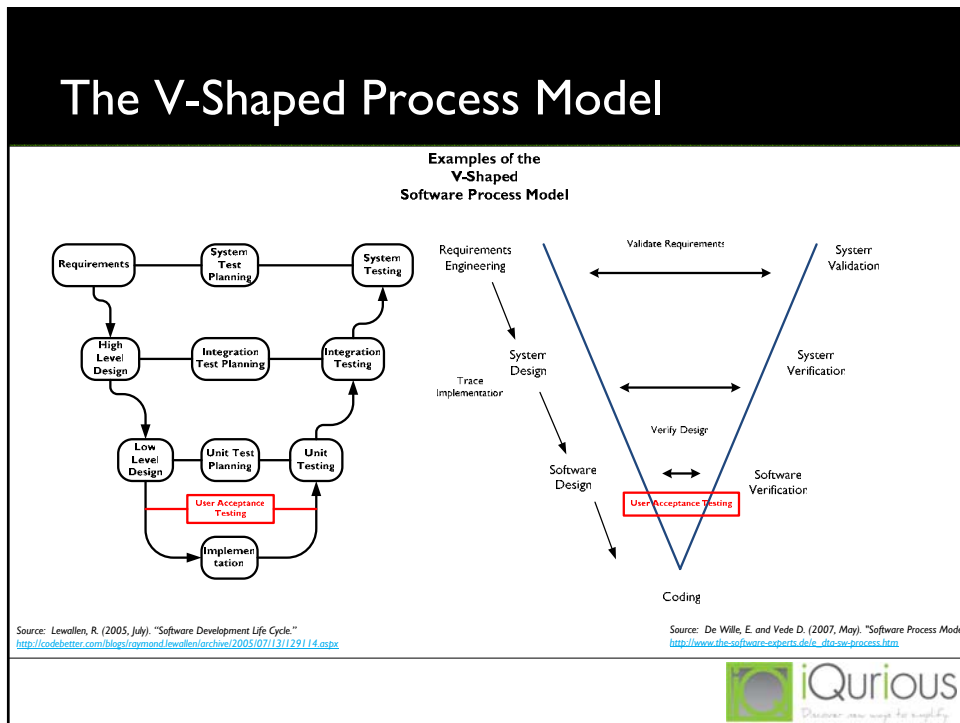
The Modified "V"

Modified "V" Software Process Model
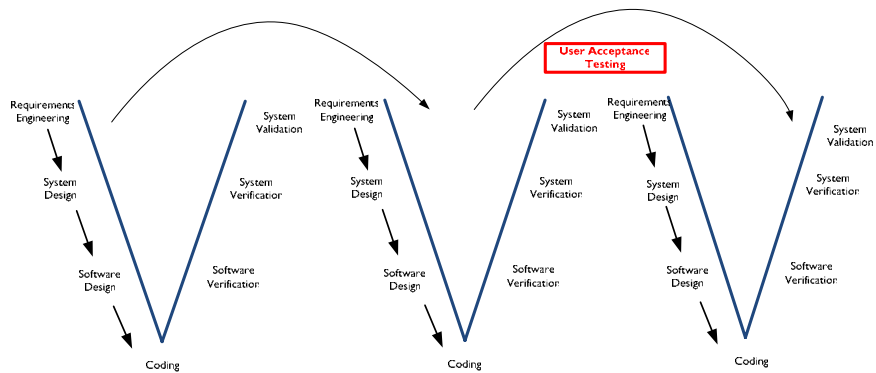
User Acceptance Testing

Requirements Engineering → System Design → Software Design → Coding → Software Verification → System Verification → System Validation (repeated three times)
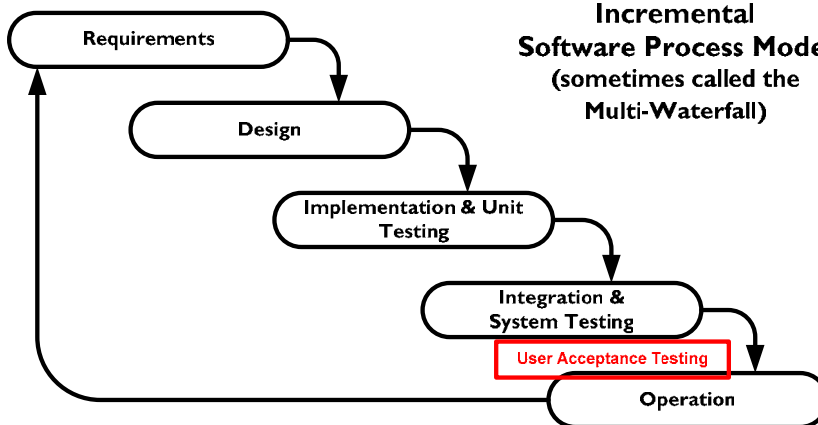
Source: De Wille, E. and Vede D. (2007, May). "Software Process Models"
http://www.the-software-experts.de/e_dta-sw-process.htm
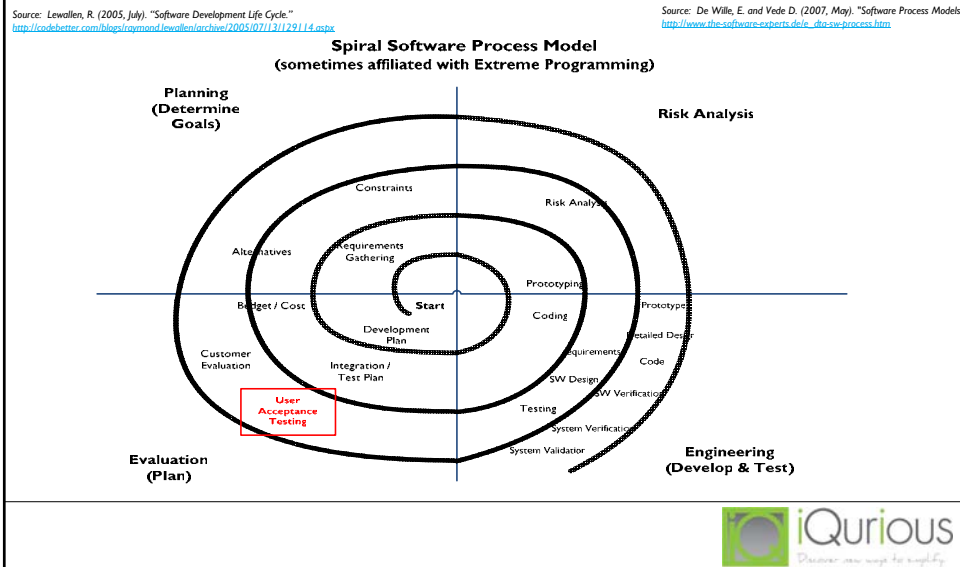
The Incremental Process Model

Requirements → Design → Implementation & Unit Testing → Integration & System Testing → User Acceptance Testing → Operation

Incremental
Software Process Model
(sometimes called the
Multi-Waterfall)

User Acceptance Testing

Source: Lewallen, R. (2005, July). "Software Development Life Cycle."
http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx

The Spiral Process Model

Source: Lewallen, R. (2005, July). "Software Development Life Cycle."
http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx

Source: De Wille, E. and Vede D. (2007, May). "Software Process Models"
http://www.the-software-experts.de/e_dta-sw-process.htm

Spiral Software Process Model
(sometimes affiliated with Extreme Programming)

Basic steps for UAT

- Plan: Identify what needs to work in the simplest terms
- Specify: Forget the complicated test scenarios and figure out what "easy" steps will confirm UAT success
- Execute: Run the UAT in as efficiently and quickly as possible
- Record: Make sure processes are in place to "measure" the success of UAT
- Complete: Set a FINAL date when UAT summary will be complete.

A checklist for success

- Thorough requirements document
- Complete development cycle
- Comprehensive System Tests
- Concise documentation
- Straightforward UAT Test Plan
- Appropriate resources (Human and System) available for tests

iQurious

The "sweet spot"

- How do we find the ideal place for UAT in the development/rollout cycle?
  - What are your concerns?
  - Who: in your organization: is affected?
  - What is helpful feedback?
  - How do you know that your objectives were met?

iQurious

Is "no-go" an option?

- What is "Done"?
  - Feature Freeze: Major design elements are done
  - Code Freeze: The system/software has successfully completed a full internal test run
  - Release Candidate: Everything is "ready" for release
- Is the Rollout date set in stone?
- What are your options if the feedback is negative?

iQurious

Who will you target?

- Is there an "ideal user"?
- Do you foresee "pain points" with specific users?
- What are the "risks" of integrating outside users into your test cycle?

iQurious

Group activity

Discuss the "Sweet Spot", when you're "done", your "ideal user", the "pain points", and UAT risks

iQurious

UAT is not...

- Smoke testing
- Functionality testing
- Regression testing
- Platform or Compatibility testing
- Alpha or Beta testing (we'll discuss why)
- Subject-matter Expert testing
- Boundary testing

iQurious

Requirements

- Why is this project being created?
- What are the changes that will occur to complete the project?
- What organizations are involved in completing the project?
- How long will it take to get the changes done?
- Who is responsible for ensuring the project is successful?
- What is "successful"?

iQurious

What kind of output do you want?

- Define what will be helpful or unhelpful
- How can you guide users to give you the feedback you want?
- No handholding! What must be done so that UAT testers can complete their task with a minimal impact on the developers and testers?

iQurious

Group activity

Define the "Ultimate UAT"

iQurious

Creating a UAT plan

iQurious

Time frame for testing

- Perils of a long time frame
  - Many users will only test for last 3 days
  - Users need sense of immediacy
- Ideal timeframe is under 2 weeks
  - Keeps the ball rolling
  - Still allows users to get vital work done around testing

iQurious
Discover new ways to simplify.

Communication plan

- No surprises: get level-heads involved
- Create your communication model
  - Identify:
    - Who will gather results
    - What output will be collected
    - How results will be communicated to project team
- Plan ways to overcome "honesty obstacles"
- Protect developers and UAT testers from one another

iQurious
Discover new ways to simplify.

Making good questions

- The key to valuable feedback: A plan without a "Plan".
- Sample questions
  - Does the system work?
  - Do you like the UI?
  - Did you have any issues?
- Components of a good question
  - Simple language: keep it brief
  - Multiple choice / Satisfaction Ratings
  - Limit the number of questions asked!

iQurious
Discover new ways to simplify.

Sample Requirements documents & UAT plans

iQurious
Discover new ways to simplify.

Techniques for maximizing success

iQurious

Find a power user

- Power users are often used in beta testing
- Where do they fit in UAT?
  - Used as liaison between users and IT/developers
  - Higher level of honesty
  - Can be used as first-level support

iQurious

How to keep power users

- Rewards: Integrating expert users into UAT cycles can be a win-win for the overall project.
  - Improved Requirements documents
  - Better development test cycles
  - Early buy-in: Experts can help support your organization's deployment of a major change
  - Influence: Power users have egos too! The more you use them, the better they feel.

iQurious
Discover new ways to simplify.

Motivating users to participate

- Common target is 30% detailed feedback
- Getting there:
  - Raffle: Games are good!
  - Nagging e-mails: Guilt works too…
  - Buy-in from management
  - Make it mandatory
  - Be the good guy
  - Unintended benefits: UAT can strengthen relationships between organizations

iQurious
Discover new ways to simplify.

Group activity

- Brainstorm on ways to motivate testers
- Create a nagging email

iQurious

Presenting the final data

- Summary report
  - A successful UAT program should have an unnecessary summary report
  - Create good communication plan before UAT begins
  - Nothing in summary report should be a surprise
- Why I don't like doing UAT summary reports

iQurious

Graphs & charts

- Who looks at graphs and charts?
  - Everybody (if they're pretty enough)
- Information can be used by different groups
  - Bring your results to the Project team and the users
  - Make sure positive comments are broadcast
  - Reinforce connections UAT might offer
- Stay practical: Keep It Simple…

iQurious

Needs of different departments

- Who needs to see your results?
  - Engineering / Development
  - Quality (QA, QC, System Test)
  - IT Support
  - Management (Executive level)
  - Others?

iQurious

Now it's your turn

UAT: Internal Access Portal

- Define UAT for your environment
- Create requirements document outline
- Identify internal tests
- Create UAT plan outline
  - Line of support for testers
  - Information needed from UAT (questions for testers)
  - List ideas to motivate users
- Decide on output elements for UAT summary

Before we get started

Q & A

iQurious

Group activity

iQurious

Takeaways

- ◉ CD contents
  - Presentation
  - Sample documents
  - Templates for you to create UAT plan and summary docs
- ◉ Leave your e-mail to get more stuff
  - HollyR@iQurious.com

iQurious