



IBM Software Group

My Encounter and Interaction with Usability

Vivek S. Buzruk

Rational. software



@business on demand.

Basics

- What is most important aspect of successful software?

- Usability originates from integrity of Software with internal and external world.

- Usability testing is exploring/exploiting the software to judge whether the external world gets improved performance experience while complementing varied external world's feelings, opinion, mood, interest, knowledge, limitations, understanding and multipurpose increasing usage of the software.
 - ▶ In other terms – **Usability testing is testing the application from the perspective of convenience to end-user.**



1'st Encounter - As a Developer and User of IDE

- IBM Rational Apex Ada Developer environment (Native and Cross)
 - ▶ Validated compilation system and Integrated development environment for ADA language
 - ▶ A real great IDE from all angles. Matured usability and extensive consideration for usage.
- Few Points and examples of Usability – After and along with correctness, reliability, performance etc.

<i>1'st time IDE user</i> , most probably Developer or Project manager	Develop and Debug simple ADA program. Finds the Help system in place,	Import Ada Code from other environment, Browse the code and then build executable.	Start the maintenance, integration and enhancement of code. Navigation capabilities
<i><u>intermediate level (skill-set wise) IDE user</u></i>	Architect his system. Easy Build. Manage Configurations	Browse, maintain, Enhance the code (ie. Edit-compile-debug cycle)	Finds Training material and user manual to further improve his skill-set.
<i><u>Advanced and Regular IDE user</u></i>	Customization. Application and handling new release of IDE. Build and release of customer application. Controlled Iterative Development...		Discovers and integrates the IDE with the external world/environment.



1'st Encounter - As a Developer and User of IDE

- What about internal Integrity – This is where Architect plays important role.
 - ▶ Apex Ada IDE - well Architected and Designed for usability
 - ▶ Try Eclipse (www.eclipse.org) and Rational Software Architect
- Usability examples -
 - Consistency in user interface (Complying with X/Motif)
 - Menu grouping, division and items availability
 - Context menu availability
 - Error indications and reporting
 - Aesthetics
 - Color combination and window usage
 - Human factors considerations
 - Customization of user interface – compliance with X/Motif standard
 - Availability of command line interface and batch command language
 - Help System – in multiple formats and easily available
 - Context sensitive,
 - With every dialog
 - Language Reference manuals



2'nd Encounter – Testing of Core Banking application

- Not very popular, but definitely quite-known and used application
- Applet based interface –
 - ▶ Very bad but consistent user interface
 - ▶ Bad in Aesthetics. Appeared to be designed for functionality based on the terminal based interfaced.
- **I would have outright rejected but, found that the user community was using the system though with few complaints. Points considered important in terms of usability and functionality -**

<p><i><u>1'st time user of computerised application, Bankers</u></i></p>	<p>Create accounts, customers etc. and enter related information.- Does not make difference if it takes 5 pages</p>	<p>Verification of accounts and customers. This is where performance improvement was seen. Kind of Workflow but really primitive.</p>	<p>List based approach and availability of all commands at the outset.</p>
<p><i><u>intermediate level (skill-set wise) computer users</u></i> Mainly IT team and very few Bankers</p>	<p>Hot-Key based interface</p>	<p>Maintenance of users of software.</p>	<p>Easy generation of reports</p>
<p><i><u>Senior Management users</u></i></p>	<p>More concentration on report generation facility. Expected correctness and 1'st level information maintenance.</p>		<p>Considered the provider and his credibility.</p>

2'nd Encounter - What if such application is not used in controlled environment?

- Very difficult to succeed.
 - ▶ Not suited for the users/customers having better computerization exposure.

- One can not say Usability was given very little importance
 - ▶ Tester's angle is different from end-user's angle
 - ▶ But,
 - Definitely there were many points beyond Aesthetics

- Missing aspects
 - ▶ Release and enhancement management – Important to handle with ease by Administrator's and IT team's angle
 - ▶ Documentation – from intermediate and advanced user point of view
 - ▶ Use of standards and integration with Environment –From Administrator's and IT team perspective
 - ▶



Usability Testing – RUP and other definitions

- Usability tests focus on human factors, **aesthetics**, consistency in the user interface, **online and context-sensitive help**, wizards and agents, **user documentation**, and training materials.
- Usability testing looks at how easy the product is to learn and use.
- You might do this testing with end users but many usability tests (such as performance tests, or counts of the number of steps involved to complete a task) can be done by anyone.
- Cem Kaner and James Bach's Definition -
 - ▶ **Usability**. How easy is it for a real user to use the product?
 - **Learnability**: operation of the product can be rapidly mastered by the intended user.
 - **Operability**: product can be operated with minimum effort and fuss.
 - **Throughput**: how quickly the user can do the complete task.
 - **Accessibility**: product is usable in the face of the user's limitations or disabilities



Applying Usability Tests – IDE Example

User Category		Type of (Multipurpose) Usage based Usability Testing		Broader Usability points
<u>Type of User</u>	<u>Kind of User</u>	Work Environment Creation	Edit-Compile-Debug Cycle	Consistency, Aesthetics , help, ...
1'st Time	Developer	Points to consider: <i>Learnability:</i> Accessibility Operability Throughput		Overall considerations Work Environment Creation Edit-Compile-Debug Cycle
	Evaluator			
	Project Manager			
...				
Advanced and Regular User	Developer	Points to consider: Throughput Operability Accessibility Learnability		Environment Integration
	Architect			
	Release Manager			

Test Cases



Applying Usability Tests – Web or other interface

User Category		Type of (Multipurpose) Usage based Usability Testing		Broader Usability points
<u>Type of User</u>	<u>Kind of User</u>	Environment Creation and initial setup	Usage Cycle 1	Consistency, Aesthetics , help , ...
1'st Time	Internal level 1 Staff	Points to consider: <i>Learnability:</i> Accessibility Operability Throughput		Overall considerations Environment Creation Usage Cycle 1 Usage Cycle n Quick usage 1 Quick usage n
	Customer			
	Manager,....			
...				
Advanced and Regular User	Internal level x Staff	Points to consider: Throughput Operability Accessibility Learnability		Environment Integration
	Customer			
	Release Manager,....			

Test Cases



Usability Analysis and results

- The Usability analysis and results need to be verified by
 - ▶ Architect and
 - ▶ Product Manager

- Decide the steps to get further input from users of software.



Usability Testing and other Testing Techniques

- Usability testing does not really overlap with other testing techniques, but may get certain inputs from other test cases.
- Usability testing is not user-interface testing, but few Consistency and Aesthetics aspects might have been considered in user-interface testing.
- Usability testing need to be aligned with
 - ▶ User testing - systematically simulate a user.
 - User testing involves a variety of users and user roles, not just one.
 - ▶ Exploratory Testing – Many times the “Learn and Test” approach of Exploratory approach reveals many flaws in usability and its integration with user types and environment.
 - Adaptability related tests is another exploratory testing example which generates lot of input for usability testing.
 - ▶ Accessibility testing – Though it is a separate testing technique, but it discusses more from functionality angle for a particular user group.



Usability Testing and Automation

- Use Functional Tester to automate certain parts of Usability testing.
 - ▶ For example, number of steps for a certain tasks
 - ▶ Or availability of help/documentation in required places.

- Along with such above tests, property verification techniques can be used for better understanding of “consistency” and even human factors.

- Sometimes one may have to modify the scripts and having good scripting language can be really useful.

- **But** many times usability is manual testing and so, **tools like Manual Tester** can help to record the test and also to record the results of usability testing during (manual) execution.



Summary

- Usability testing brings out the experience and the impression felt by different types of users along with increased performance results.
- Increase the frequency of Usability Testing and if possible, involve real end-users for getting better feedback.





Analyst

WebSphere
Business
Modeler
& Monitor

Rational
Software
Modeler

Architect

Rational
Software
Architect

Developer

Rational
Web/App
Developer

WebSphere
Integration
Developer

Tester

Rational
Functional
& Manual
Tester

Rational
Performance
Tester

Deployment
Manager

Tivoli
Configuration
Manager

Tivoli
Monitoring

Customer
Extensions

ECLIPSE

3rd Party
ISV Tools



Rational Team Unifying Platform



Rational Portfolio Manager

