Computer Games 2011
Playtesting

Dr. Mathias Lux

Klagenfurt University

Agenda

- Motivation
- Usability Testing Basics
- Usability Tests
- Playtesting Specifics

(cc) by Warm ,n Fuzzy, http://www.flickr.com/photos/warmnfuzzy/466382462/

Motivation

- Playtesting forces to develop
  - for gamers
  - not for developers
- Software testers are not the target group
  - testers are necessary, skilled people
  - but playtests with “real people” are essential
- Playtests are a moderator
  - for controversy in game design

src. http://www.gamasutra.com/view/feature/3963/the_silent_revolution_of_.php

Motivation

- Partnership playtesting & design is constructive
  - instructive for designers to observe gameplay during playtesting
  - determine whether particular aspects of their design work out
- Playtests on mock-ups: early problem identification
  - the faster a problem is corrected, the less expensive it is
  - game development can therefore become truly "player-centric"
- Playtests allow examination of specific aspects
  - game balance, navigation, understanding of the game objectives, etc.

src. http://www.gamasutra.com/view/feature/3963/the_silent_revolution_of_.php

What kind of problems can be identified with playtesting?

- Accessibility and ease of use
  - interface, navigation within the game etc.
- Identification of sure-fire-wins
  - strategies that always work
- Fine-tuning of the game system
  - weapons, equipment, actions etc.
  - time spent on familiarizing with the game
  - ensure that the game maintains its balance and relevance even after long hours of gaming

src. http://www.gamasutra.com/view/feature/3963/the_silent_revolution_of_.php

What kind of problems can be identified with playtesting?

- Early reactions of different players
  - highlight first impressions and initial frustrations
  - e.g. demos giving wrong impression of a game
- Robustness of game & potential of maps
  - for multiplayer games

src. http://www.gamasutra.com/view/feature/3963/the_silent_revolution_of_.php

Agenda

- Motivation
- Usability Testing Basics
- Usability Tests
- Playtesting Specifics

(cc) by Warm ,n Fuzzy, http://www.flickr.com/photos/warmnfuzzy/466382462/

Usability Testing Basics

Ease of use affects the users’ performance and their satisfaction, while acceptability affects whether the product is used.

src. Bevan, N. Measuring usability as quality of use. Softw. Quality J. 4 (1995), 115–130.

Usability Testing Basics

- Usability must be considered before prototyping takes place
  - – late tests lead to high costs and frustration
- Usability problems often ignored by developers
  - – “We don’t have usability problems.”

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Usability: Basic Concepts

- Learnability,
  - so the user can rapidly begin working with the system
- Efficiency
  - enabling a user who has learned the system to attain a high level of productivity
- Memorability
  - allowing the casual user to return to the system after a period of non-use without having to relearn everything
- Low error rate
  - so users make fewer and easily rectifiable errors while using the system, and no catastrophic errors occur
- Satisfaction
  - making the system pleasant to use

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Agenda

- Motivation
- Usability Testing Basics
- Usability Tests
- Playtesting Sepcifics

(cc) by Warm ,n Fuzzy, http://www.flickr.com/photos/warmnfuzzy/466382462/

Usability Tests

- Inspection methods
  - – no users involved
  - – heuristic evaluation, cognitive walkthroughs, and action analysis
- Test methods
  - – with users
  - – thinking aloud, field observation, and questionaires

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Heuristic Evaluation

- Usability experts judge interactive elements
  - based on usability principles
- Aggregation of findings after evaluations

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Nielsen's Usability Heuristics

- **Visibility of system status**
  - The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **Match between system and the real world**
  - The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- **User control and freedom**
  - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- **Consistency and standards**
  - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- **Error prevention**
  - Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

src. http://www.useit.com/papers/heuristic/heuristic_list.html

Nielsen’s Usability Heuristics

- **Recognition rather than recall**
  - Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- **Flexibility and efficiency of use**
  - Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- **Aesthetic and minimalist design**
  - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- **Help users recognize, diagnose, and recover from errors**
  - Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- **Help and documentation**
  - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

src. http://www.useit.com/papers/heuristic/heuristic_list.html

Cognitive Walkthrough

- Task-oriented system exploration
  - by usability expert
- Simulating user behavior step by step
- Careful selection of tasks
  - in context of user goals

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Action Analysis

- Analysis on key stroke level
- Analysing actions of the users
  - by making them explicit (recording)
  - and discussing them

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Thinking Aloud

- User tells what s/he thinks
  - while using the system
- Typically users have tasks or problems to solve
  - e.g. “Create a new gnome wizard character”
- Interaction is recorded
  - Time 2 solve, problems, etc.

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Field Observation

- Observe user in “natural environment”
- Be unobstrusive
  - – Noise and disturbance leads to errors & bias
- Possibility to take a video
  - – Note analyzing takes time (x 10)

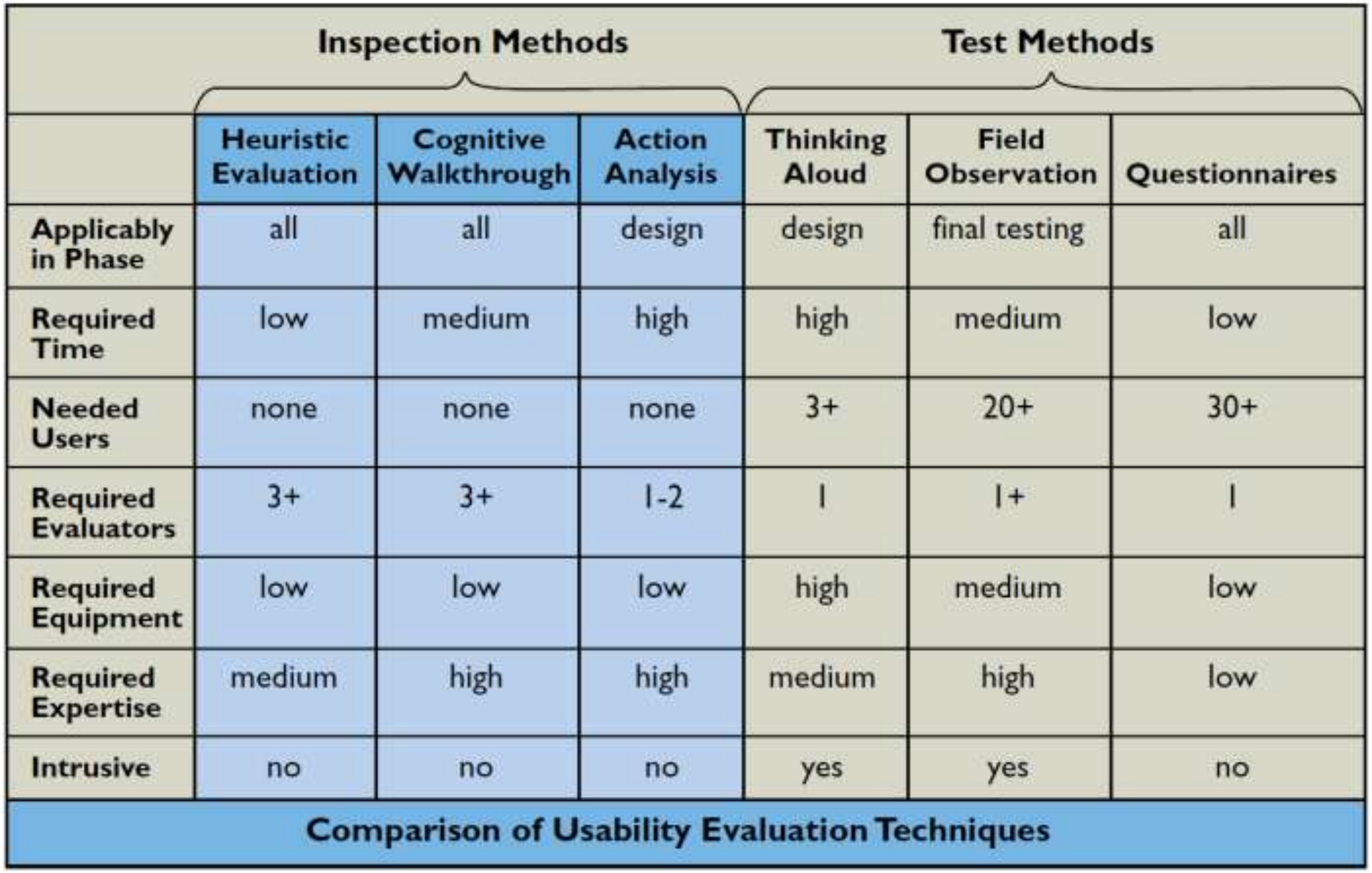
src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Questionnaires

- Collection of opinions
  - not necessarily general statement
- Needs a skilled questionnaire designer
  - Likert scales, statistical analysis etc.
- Alternative: semi structured interviews
  - needs skilled interviewers too

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

Comparison



	Inspection Methods			Test Methods		
	Heuristic Evaluation	Cognitive Walkthrough	Action Analysis	Thinking Aloud	Field Observation	Questionnaires
Applicably in Phase	all	all	design	design	final testing	all
Required Time	low	medium	high	high	medium	low
Needed Users	none	none	none	3+	20+	30+
Required Evaluators	3+	3+	1-2	1	1+	1
Required Equipment	low	low	low	high	medium	low
Required Expertise	medium	high	high	medium	high	low
Intrusive	no	no	no	yes	yes	no

Comparison of Usability Evaluation Techniques

src. A. Holzinger, Usability Engineering Methods For Software Developers, Communications Of The ACM, January 2005/Vol. 48, No. 1

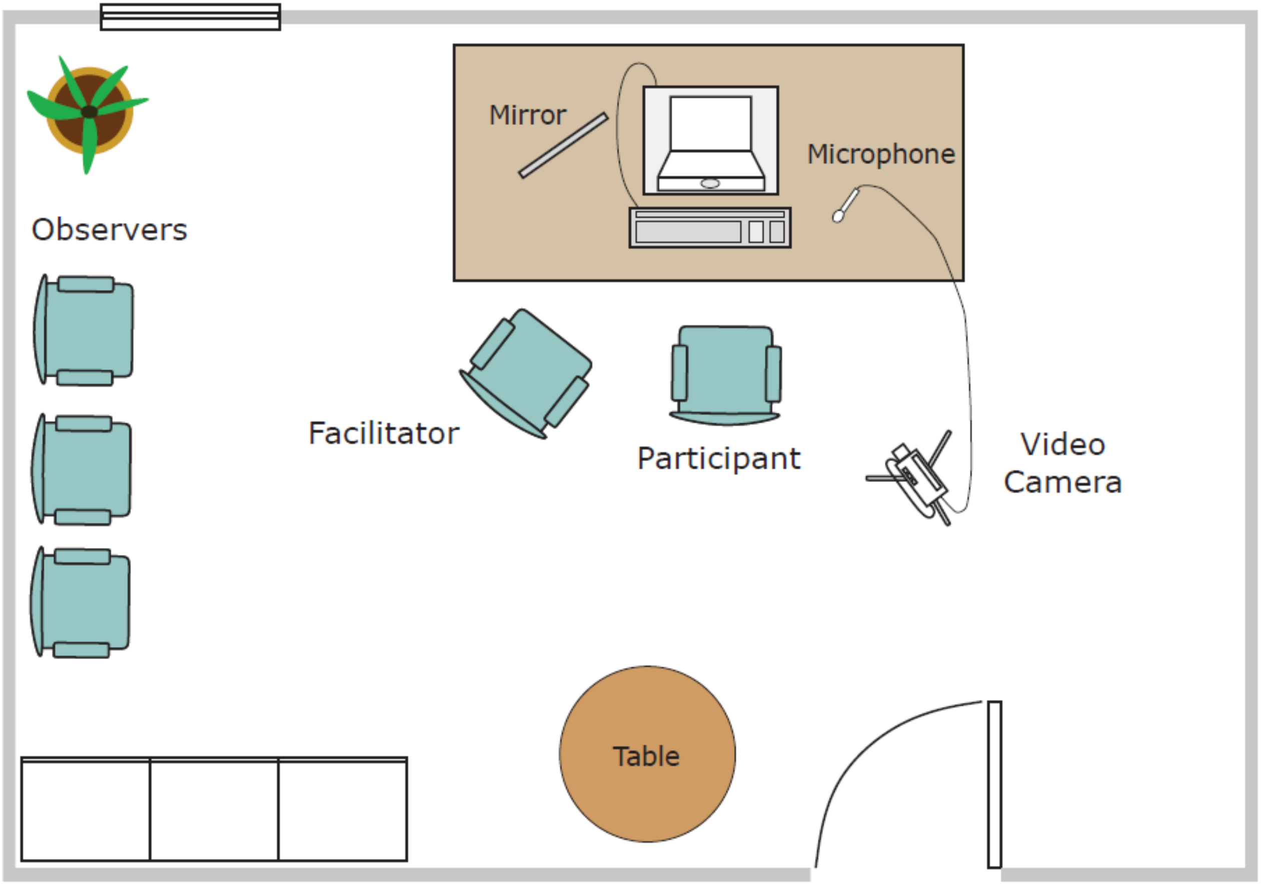
Six Stages

1. Develop the test plan
2. Select and acquire participants
3. Prepare test materials
4. Run a pilot test
5. Conduct the real test
6. Analysis and final report

Note: Always do a pilot test!

src. K. Andrews, Human-Computer Interaction Lecture Notes, 2011, IICM, TU Graz

Typical Test Setup



src. K. Andrews, Human-Computer Interaction Lecture Notes, 2011, IICM, TU Graz

Usability Test Environment

- Eliminate uncontrolled parameters
  - give subjects the “same” environment
  - do not interfere
- Capture the information you need
  - record videos of screen, controller, face, body etc.
  - log important information
  - provide a synced time stamp (e.g. software)

Test Plan

- Purpose
- Problem Statement
- User Profile
- Method (Test Design)
- Task List
- Test Environment
- Data to be Collected
- Content of Report

src. K. Andrews, Human-Computer Interaction Lecture Notes, 2011, IICM, TU Graz

Test Plan: Task List

- Prioritize tasks by frequency and criticality
- Choose most frequent and critical to test
- Make a task list for test team (internal use)
- For each task:
  - Define any prerequisites
  - Define successful completion criteria
  - Specify maximum completion time
  - Define what constitutes an error

src. K. Andrews, Human-Computer Interaction Lecture Notes, 2011, IICM, TU Graz

Example tasks

- Task 2: Create an Avatar
  - *prerequisite:* login with registered user
  - *completed:* planet selection view reached
  - *max. time:* 5 min.
- Task 3: Select a home planet
  - *prerequisite:* logged in and created an avatar
  - *completed:* game start tutorial
  - *max. time:* 1.5 min.

Acquire Participants

- Split users into different user groups
  - based on their characteristics and needs
- Test each user group separately
  - test at least 5 test users per user group
- Acquire test users
  - via agency, students, customers, internal personnel
- Maintain a database of potential test users
- Screening questionnaire
  - ensure users fit profile

src. K. Andrews, Human-Computer Interaction Lecture Notes, 2011, IICM, TU Graz

Test Materials

- Orientation Script
- Background Questionnaire
- Nondisclosure and Consent Form
- Training Script (if any)
- Task Scenarios
- Data Collection Forms
- Debriefing Topic Guide
- Post Test Questionnaire
- Checklist

Conduct Test

- Test facilitator handles all interaction with participant
  - other team members and observers remain completely quiet
- Do not prompt or bias user during test
  - beware of non-verbal signals
- Only assist if user in severe difficulty
  - make note of when and what help given
- Conduct debriefing interview or questionnaire
- Save screen shots of any interesting problems and positive findings right after the test
  - they may not be reproducible again later

More information …

- See lecture notes of K. Andrews
  - http://courses.iicm.tugraz.at/hci/

Agenda

- Motivation
- Usability Testing Basics
- Usability Tests
- Playtesting Specifics

(cc) by Warm ,n Fuzzy, http://www.flickr.com/photos/warmnfuzzy/466382462/

Playtesting

- Respect work of design teams
  - Be prepared for arguments
  - No offense
- Adapt to needs of design teams
  - designers should require experimentation
  - test their hypotheses regarding design issues
- Deliver to concerned parties early
  - time for game development is always short

src. http://www.gamasutra.com/view/feature/3985/the_silent_revolution_of_.php

Playtesting Campaign Preparation

- One month of preparation per campaign
- Define its objective
  - to determine what types of playtesters we shall have to recruit
- Define scale of the sessions
  - 1, 2, 4, 8, 12 players
- Define duration
  - from half a day to a full week

src. http://www.gamasutra.com/view/feature/3985/the_silent_revolution_of_.php

Playtesting Campaign Preparation

- Logistics
- Legal framework
  - non-disclosure agreement
  - monetary compensation
- Prepare the design teams
  - to utilize the playtests

src. http://www.gamasutra.com/view/feature/3985/the_silent_revolution_of_.php

Playtesting Campaign Preparation

- Recruiting playtesters
  - as many candidates as possible to have a solid pool of playtesters
- Evaluate candidates
  - gaming competence
  - ability for analysis and self-expression
- Evaluation process
  - initial selection questionnaire
  - evaluation during the sessions themselves
  - observe the candidates at play
- Establish a protocol to obtain consistent results
  - no "all-purpose" evaluation protocol
  - adapt to specific circumstances as the situation mandates

src. http://www.gamasutra.com/view/feature/3985/the_silent_revolution_of_.php

Playtesting Campaign Preparation

- Good playtests require a stable build
  - the game *must* be sufficiently stable
  - maps must be rid of the most detrimental bugs
- Game delivery
  - dev team delivers a playtest-ready version
  - internal debug team reviews the game

src. http://www.gamasutra.com/view/feature/3985/the_silent_revolution_of_.php

Debriefing

- Designer ask most pressing questions
- Semi-structured interviews
  - bias from different interviewers
- Create a report
  - as soon as possible
  - statistical facts vs. subjective observations

Playtest Sessions

- Design team observes playtesters
  - Learn from their interaction
  - Rapidly create new approaches
  - Discuss with testers (afterwards)
- Collect
  - relevant statistical data
  - from a representative sample of users

Thanks …

… for your attention