

fit@hcmus

Software Testing

CSC13003

Usability Testing

Content

- **GUI testing**
- Usability testing

What is GUI testing?

- Abbreviation of Graphic User Interface.
- Definition: Examine various elements of the user interface, such as buttons, menus, input fields, and visual design, to verify they work as intended

What is GUI testing?

- Purpose: The primary purpose of GUI Testing is to identify and rectify any issues or defects related to the user interface. It ensures that the application is visually appealing, user-friendly, and responsive.

What is GUI testing?

- **Functional Validation:** While GUI Testing focuses on the visual and interactive aspects of the software, it also checks whether the functionality tied to these elements is working correctly. It ensures that user interactions lead to the expected outcomes.

What is GUI testing?

- Common Elements: GUI Testing assesses common GUI elements, including buttons, checkboxes, radio buttons, dropdown menus, text fields, and error messages, among others.

Why is GUI testing necessary?

- **User-Centric Focus:** The user interface is the point of interaction between the software and the end user. Ensuring its reliability and usability is essential for a positive user experience.
- **Visual Consistency:** GUI Testing verifies that the software maintains visual consistency across different devices and platforms. It helps prevent issues like text overflow, misalignment, and distorted graphics.

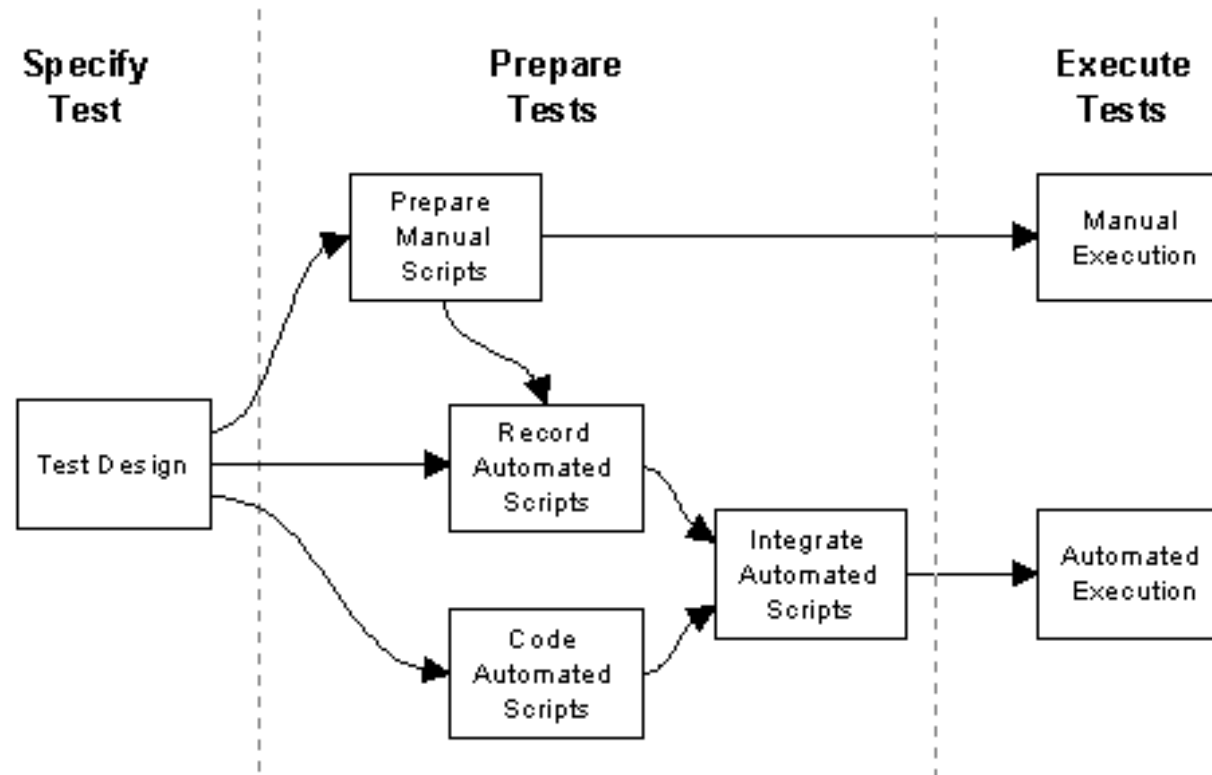
Why is GUI testing necessary?

- **Functional Verification:** GUI Testing ensures that the functionality tied to the GUI elements works as expected. It validates that buttons perform the intended actions, forms accept input correctly, and menus lead to the right options.
- **Usability and Accessibility:** It assesses the software's usability, making sure it's intuitive and easy to navigate. GUI Testing also checks for accessibility features to accommodate users with disabilities, meeting regulatory requirements.

Why is GUI testing necessary?

- **Brand Reputation:** A polished and bug-free user interface reflects positively on your brand's reputation. It can attract and retain users, fostering trust and loyalty.
- **Bug Detection:** Many critical bugs and issues manifest in the GUI layer. GUI Testing helps identify these issues early in the development process, reducing the cost of fixing them later.
- **Compliance:** In industries like healthcare and finance, compliance with regulatory standards is essential. GUI Testing ensures that the software complies with industry-specific guidelines and requirements.

Process



Common GUI bugs

- Data validation
- Incorrect field default
- Mishandling of server process failures
- Mandatory fields, not mandatory
- Wrong fields retrieved by queries

Common GUI bugs

- Incorrect search criteria
- Field order
- Multiple database rows returned, single row expected
- Currency of data on screens
- Window object/DB field correspondence

Common GUI bugs

- Correct window modality?
- Window system commands not available/don't work
- Control state alignment with state of data in window?
- Focus on objects needing it?

Common GUI bugs

- Menu options align with state of data or application mode?
- Action of menu commands aligns with state of data in window
- Synchronization of window object content
- State of controls aligns with state of data in window?

Types in GUI testing

<i>Stage</i>	<i>Test Types</i>
<i>Low Level</i>	Checklist testing Navigation
<i>Application</i>	Equivalence Partitioning Boundary Values Decision Tables State Transition Testing
<i>Integration</i>	Desktop Integration C/S Communications Synchronisation
<i>Non-Functional</i>	Soak testing Compatibility testing Platform/environment

Low level – Checklist testing

- GUI standards
- Application standards
- Color scheme
- Typography
- Layout and alignment
- Labels
- Error messages

Low level – Navigation testing

- Main Menu Navigation: Test the main navigation menu to ensure users can access all major sections of the application.
- Breadcrumb Navigation: Verify that breadcrumb trails accurately reflect the user's path and allow users to backtrack.

Low level – Navigation testing

- Links and Buttons: Check links, buttons, and call-to-action elements to confirm they lead to the expected screens or actions.
- Form Navigation: Ensure that users can move through forms easily, with proper focus order and validation messages.

Application testing

Technique	Elements to test
Equivalence Partitions and Boundary Value Analysis	<ul style="list-style-type: none">-Input validation-Simple rule-based processing
Decision Tables	Complex logic or rule-based processing
State-transition testing	<ul style="list-style-type: none">- Applications with modes or states where processing behavior is affected- Windows where there are dependencies between objects in the window.

Integration Level

- Desktop integration testing
- Client/Server communication testing
- Synchronization testing

Non-functional Level

- Soak testing
- Compatibility testing
- Platform/Environment testing

Challenges in GUI testing

- **Diverse Platforms and Devices:**
 - Challenge: Ensuring consistent GUI performance and appearance across various operating systems, browsers, and devices.
 - Solution: Comprehensive cross-browser and cross-device testing, along with responsive design practices.

Challenges in GUI testing

- Frequent UI Changes:
 - Challenge: Rapid iterations and updates in the UI can lead to test script maintenance challenges.
 - Solution: Implement a robust test automation framework, and use Page Object Model to decouple UI changes from test scripts.

Challenges in GUI testing

- Test Coverage:
 - Challenge: Ensuring complete coverage of all GUI elements and user workflows.
 - Solution: Develop a comprehensive test strategy that covers critical paths and edge cases.

Challenges in GUI testing

- Test Data and Environment Setup:
 - Challenge: Creating and managing test data and test environments can be time-consuming.
 - Solution: Use data provisioning tools and containerization for efficient environment setup.

Challenges in GUI testing

- Localization and Internationalization:
 - Challenge: Testing GUI for different languages, cultures, and regions.
 - Solution: Utilize localization testing tools and collaborate with native speakers.

Challenges in GUI testing

- Integration with Backend:
 - Challenge: Coordinating GUI tests with backend systems and APIs.
 - Solution: Implement end-to-end testing strategies and use mocking for backend components.

Automation in GUI testing

Loại kiểm thử	Manual vs Automated
Checklist testing	Manual: application conventions Automated: object states, menus, standard features
Navigation	Manual
Equivalence Partitioning, Boundary Values, Decision Tables, State Transition Testing	Manual: complicated cases Automated: simple cases
Desktop Integration, C/S Communications	Manual: complicated cases Automated: simple cases
Synchronization	Manual
Soak testing, Compatibility testing, Platform/environment	Automated

Content

- GUI testing
- **Usability testing**

Usability Testing

- Process that **employs representative target population participants** to evaluate product usability using specific usability criteria
- Usability testing is **not a guarantee for product success**, but it should identify at least the key problems

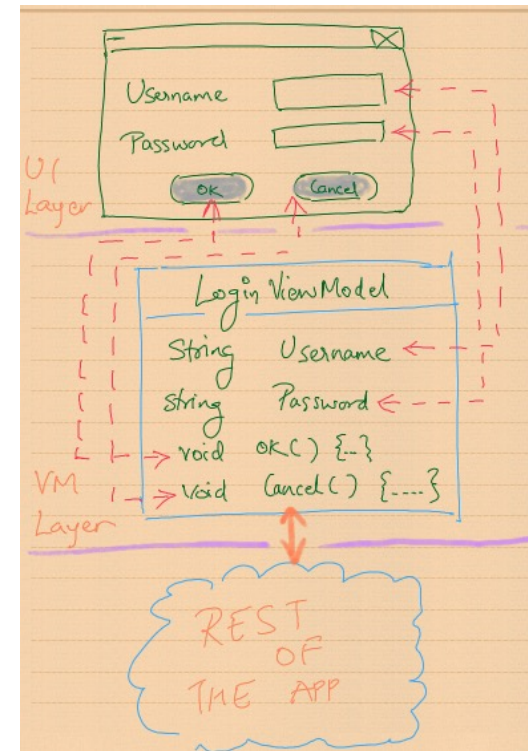


Basic components

1. Development of specific problems statements and test plans and objectives
2. Use of **representative sample** of end users
3. Representation of the **actual work environment**
4. **Observation of end users during** product use or review
5. Collection of **quantitative and qualitative** measurements
6. **Analysis of results and recommendations**

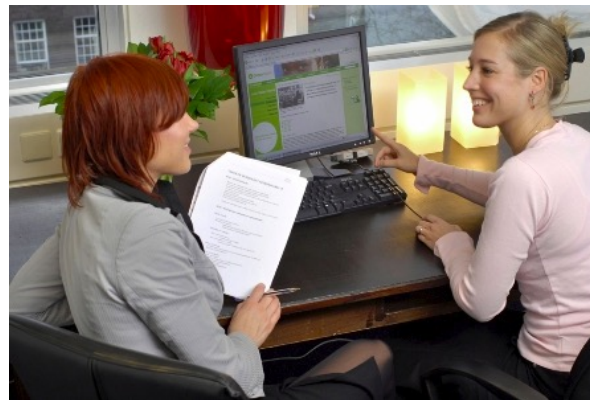
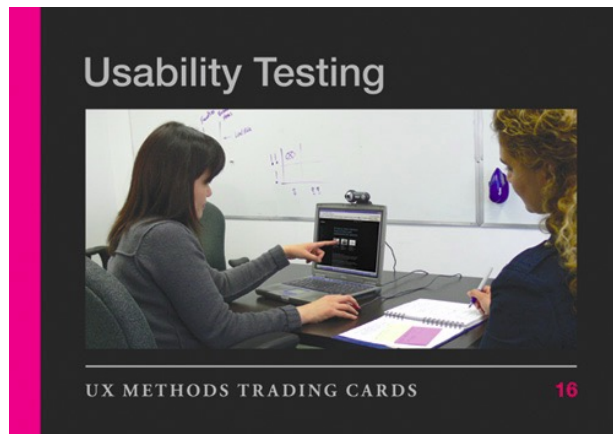
Types of usability tests – Exploratory

- Early in the process
- Can be based on any form of the GUI (sketch, wire diagrams etc.)
- Evaluate preliminary, basic design concept
- Perform representative tasks in a “shallow” mode
- Informal test methodology, a lot of interaction
- Discuss high level concepts



Types of usability tests – Assessment

- Done After fundamental concepts are done
- Evaluates usability of lower level operations
- The users actually perform set of well defined tasks
- Less interaction with test monitor
- Quantitative measurements are collected



Types of usability tests – Validation

- Done late in development cycle, close to release
- Goal is to certify product usability, “disaster insurance” against launching a poor product
- Often the first time when the whole product is tested (including help and docs)
- Evaluate wrt. some predetermined usability standard or benchmark
- Standards come from **previous testing, competitive information, marketing** etc.
- Very specific quantitative tests
- Can establish standards for future products
- Can be also done by beta customers

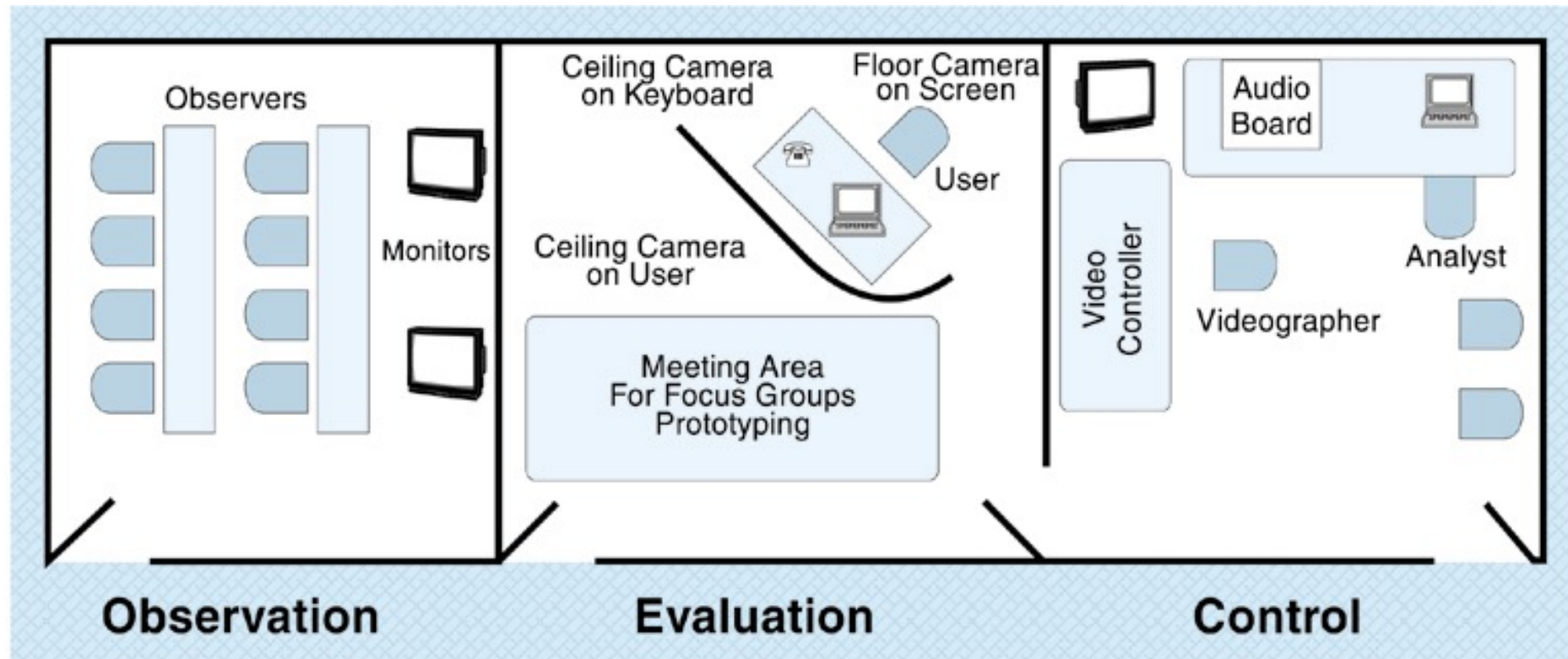
Types of usability tests – Comparison

- Can be done at any point in the development cycle
- Compare alternatives using objective measures
- Can be informal or formal, depending when it is done
- Often, best of alternative designs is combined

Test environments

- Simple single room setups
 - Observer/monitor close to evacuator
 - Observer removed from evaluator
- Electronic observation room
- Classic elaborate usability lab
- Mobile lab

Example



Typical test plan format

- Purpose: what is the main purpose of the test
- Problem statement: specific questions you want resolved
- Test plan and objectives: tasks the user will do
- User profile: who will be the users
- Method and test design: how will you observe it, how will you collect the data etc.
- Test environment and equipment
- Test monitor role
- Evaluation measures and data to be collected: how will you collect the feedback and how will you evaluate it
- Report: what will final report contain

Task selection for evaluation

- Tasks to be evaluated are functions users want to do with the product. **Focus is on user view of the tasks and NOT at the components and details** that you used to implement it. Examples:
 - Create and file document
 - Import several images
 - Find the right document
- Objective is to indirectly expose usability flaws by asking the user to perform typical tasks and **NOT telling them exactly how to do it.**
- Choose **key and most frequently** done tasks
- The **task has to be specific and measurable** (quantitatively or qualitatively)

Task components

TASK	DESCRIPTION
Task	Load paper into copier
Machine state	Paper tray empty
Successful completion criteria	Paper properly loaded
Benchmark	Completed in 1 minute

Selection of evaluators and test groups

- Evaluators should **be representative of the targeted users**
- Independent groups or within-subject design (but be careful to avoid exposing users to same tests since this will bias the results)
- Adequate numbers of testers
- Offer motivation and rewards

Measurements and Questionnaires

- Performance data: measures of user behavior such as **error rates, number of accesses to help, time** to perform the task etc.
 - Usually can and should be objectively and automatically measured
- Preference data: measures of **user opinion, thought process such as rankings, answers to questions, comments etc.**
 - Use questionnaires.

Some performance measures (measure what can be measured)

- Time to complete each task
- Number and percentage of tasks completed successfully/unsuccessfully
- Time required to access information
- Count of incorrect selections
- Count errors
- Time for system to respond
-

Data should be collected automatically or manually in an objective way.

Questionnaires (for preference data)

Likert scale

- I found GUI easy to use (check one)
 - ___ Strongly disagree ___ Disagree
 - ___ Neither agree or disagree
 - ___ Agree ___ Strongly agree
 - (can also assign numbers from -2 to 2)
- *Semantic differentials*
- I found File Open menu (circle one)
 - Simple 3 2 1 0 1 2 3 Complex

Questionnaires

Fill in questions

- I found the following aspects of GUI particularly easy to use (list 0-4 aspects)

Questionnaires

Check-box

- Please check the statement that best describes your usage of spell check
 - I always use spell check
 - I use spell check only when I have to
 - I never use spell check

Questionnaires

Branching questions

- Would you rather use advanced search
 - NO (skip to question 19)
 - YES (continue)
- What kind of advanced search would you like? (check one)
 - Boolean
 - Relevance

Summarizing performance results

- Performance data
 - Mean time to complete
 - Median time to complete
 - Range (high and low)
 - Standard deviation (độ lệch) of completion times
 - System response time statistics
- Task accuracy
 - % of users completing the task within specified time
 - % of users completing the task regardless (không quan tâm) of time
 - Same as above, with assistance
 - Average error rate

Summarizing preference results

- For limited choice questions
 - Count how many participants selected each choice (number and %)
 - For Likert scale or semantic differentials provide average scores if there are enough evaluators
- For free form questions
 - List questions and group answers into categories, also into positive and negative answers
- For free comments
 - List and group them at the end of the report

Analyzing Data

- Identify and focus on tasks that **did not pass the test** or showed significant problems
- Identify **user errors and difficulties**
- Identify **sources of errors**
- Prioritize problems by criticality = severity AND probability of occurrence
- Analyze differences between groups (if applicable)
- Provide recommendations at the end

Problems statements and performance data to collect

Problem Statement	Performance Data Collected
How effective is the tutorial	Compare error rates of users who used and not used this
How easy is it to perform task X	Error rate OR Number of steps needed

Note: this is Performance data measurement only. You also need to asses user Preference data (see next slide)

Problems statements and preference data to collect

Problem Statement	Preference Data Collected
How effective is the tutorial	Ask user to rate it from very ineffective to very effective (Lickert scale or semantic differentials) + free comments
How easy is it to perform task X	Ask user to rate it from very easy to very difficult (Lickert scale or semantic differentials) + free comments

Relate problem statements with tasks

Problem Statement	Task
How effective is the tutorial	GroupA: Import image w/o using tutorial GroupB: Same but use tutorial first
How easy is it to Create Virtual Machine	Create Virtual machine with "this" properties using New VM Wizard

Task components

TASK	DESCRIPTION
Task	Create VM using New VM Wizard
Machine state	VMware WS SW just loaded
Successful completion criteria	Working VM created
Benchmark	Completed in 30 sec.

