

# Usability/Accessibility testing

Software Testing: IN3240 / IN4240

# Summary

## Usability testing

HCI definition, framework, and guidelines

User-centric design processes

## Accessibility testing

Context of accessibility

Accessibility personas and accessible design

Web-content accessibility guidelines

Assistive technologies and tools



# Part I: Close-ended questions

# Question 1

Which of the following is a **purpose** of **HCI testing**?

- a. It tests that the software testing is approved by users
- b. It tests that the software is precise in its calculations
- c. It tests that the software is understandable
- d. It tests that the software has all related documentation in place



# Question 1: Clues

Which of the following is a **purpose** of **HCI testing**?

**We** (humans) **interact** with **computers** in various ways

Using desktop applications

Web browsers

Mobile devices

**Interface** between **us** and **computers** is crucial

*How* do we **interact** with technology?

How can we *best* **enable** this **interaction**?

Do some user **groups** have **advantages** / **disadvantages** over

others?



# Question 1: Clues

Which of the following is a **purpose** of **HCI testing**?

HCI: **Human-Computer Interaction**

*“The **extent** to which a product can be **used** by specified users to achieve specified **goals** with **effectiveness**, **efficiency**, and **satisfaction** in a specified **context** of use.” (ISO 9241-11)*

**Purpose** of HCI

- Understandable**
- Easy to **learn**
- Effective to **use**
- Easy to **remember**
- Satisfactory to **use**

# Question 2

Which **components** constitute the **HCI framework**?

- a. Maintainability, Portability, Security
- b. Performance, Load, Stress
- c. Laws, Industry-specific standards, Rules and Regulations
- d. Interface standards, Usability, Interface dynamics, Aesthetics



# Question 2: Clues

Which **components** constitute the **HCI framework**?

## HCI Framework

An **abstraction** to enable **contextualisation**

Considerations for HCI → Present a **global view**

Aesthetics

Interface Dynamics

Usability

Interface Standards





# Question 2: Clues

Which **components** constitute the **HCI framework**?

## Interface Standards

The underlying standards for designing user interfaces

## Criteria

Adhering to **best practices**

**Consistent** behaviour and design

## Objectives

Decrease the work load

Faster development



# Question 2: Clues

Which **components** constitute the **HCI framework**?

## Usability

Concerned with how *easy* something is to use

Consists of ...

Effectiveness

Efficiency

Satisfaction

Key considerations

**Understanding** the target **users** and their **needs** → Create **user-centric**  
designs



# Question 2: Clues

Which **components** constitute the **HCI framework**?

## Interface Dynamics

Interfaces (whether visual or API) must adhere to specific criteria

## Criteria

Responsive and fast

Adaptable to the users needs and context

Empowering the user

Captivating

Dynamic



# Question 2: Clues

Which **components** constitute the **HCI framework**?

## Aesthetics

How the system / IT artefact appears

Responsible for the first impression

## Criteria

Design should be modern, fresh, and appealing


Recognition of a company's applications

A company's graphical profile



# Question 3

Which of the following represent **interface dynamics principles**?

- a. Software has to be responsive, fast and adaptable to user needs and the given context
  - b. Software have to have the same response time for all devices that run on it
  - c. Software has to respond quickly to fast-changing needs
  - d. Systems have to be tested for load and stress, to verify their dynamic metrics
- 

# Question 3: Clues

Which of the following represent **interface dynamics principles**?

Interface dynamic principles

Principles related to the dynamic aspects / nature of interfaces

Characterised by constant change, activity, or progress

**Interfaces** must be **designed** in such a way that they **are**

**Responsive** and **fast**

**Adaptable** to the users' **needs** and context

**Empowering** to the user

**Captivating**



# Question 4

Which of the following is a **good practice** when using **system alerts**?

- a. Never write a short message – People need as many details as possible about the alert
- b. Use capital letters or exclamation marks - Users see it better
- c. Never use error codes, jargon or technical terms – Speak the users language
- d. Place the system alert on the top-left of the page – Users see it first

# Question 4: Clues

Which of the following is a **good practice** when using **system alerts**?

## HCI Guidelines

A collection of **interface standards**

Windows, OS X, and web guidelines such as W3C

## Purpose of the guidelines

Provide developers / testers with a **set of best practices**

Consistent **behaviour** and **design** of software

**Decrease** people's **workloads**

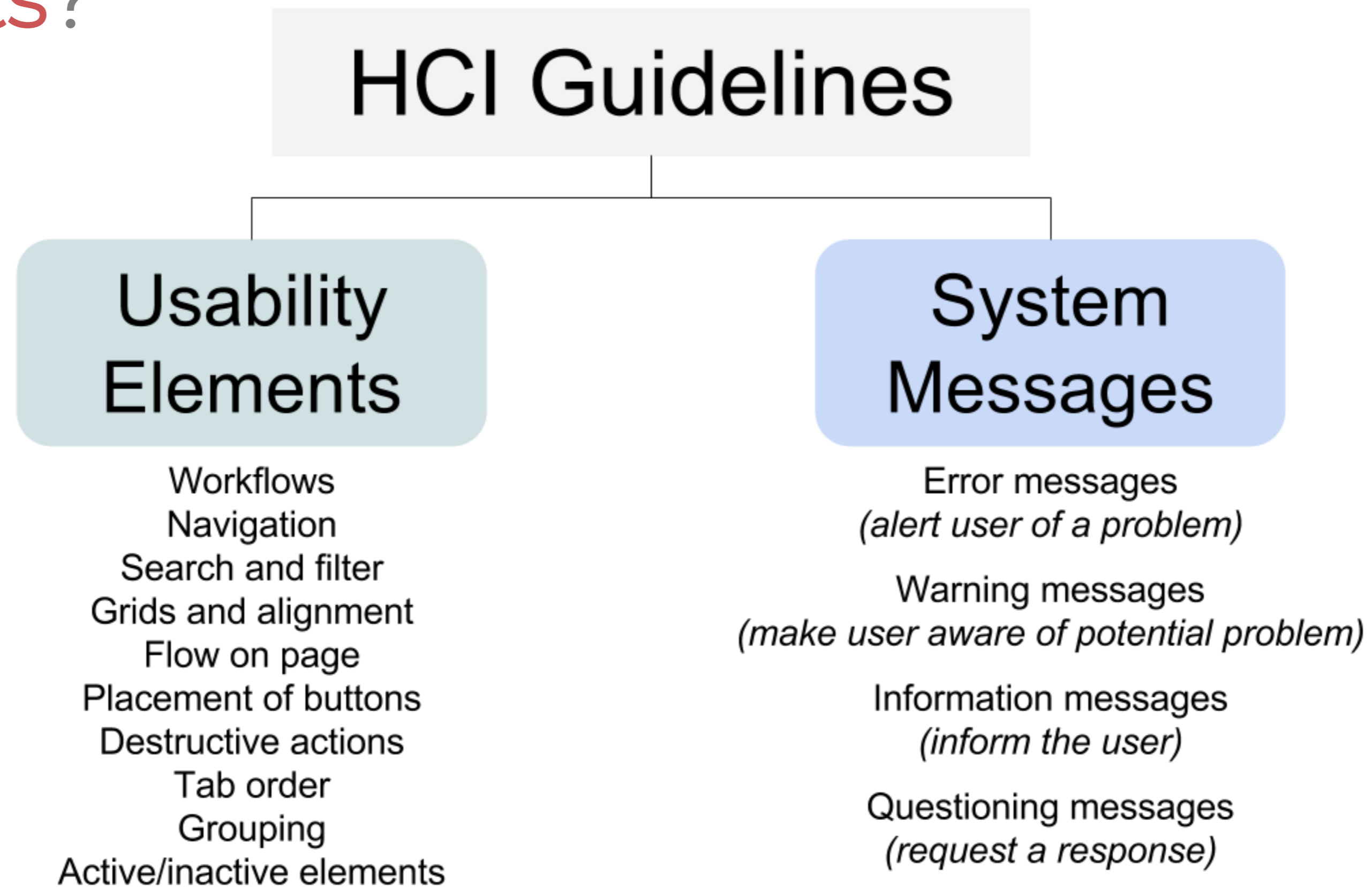
**Faster development** cycles





# Question 4: Clues

Which of the following is a **good practice** when using **system alerts**?



# Question 4: Clues

Which of the following is a **good practice** when using **system alerts**?

**Guidelines** for using **system alerts**

How they **behave** | The way they must be **designed** and **used**

**Error** and **warning** messages

Explain the problem + Provide a solution

**Good** practices

**Never** use **error** codes, **jargon**, or **technical** terms → Speak user's language

**Never** use **capital letters** or **exclamation** marks → Considered aggressive



# Question 4: Clues

Which of the following is a **good practice** when using **system alerts**?

Other **considerations**

Keep the **messages short** and **concise**

Should **explain** something of **value** to the user

Do not make them read entire paragraphs / incident descriptions

Use the **right** action **buttons**

**Errors** and **Warnings** are never OK → Use **“Close”**

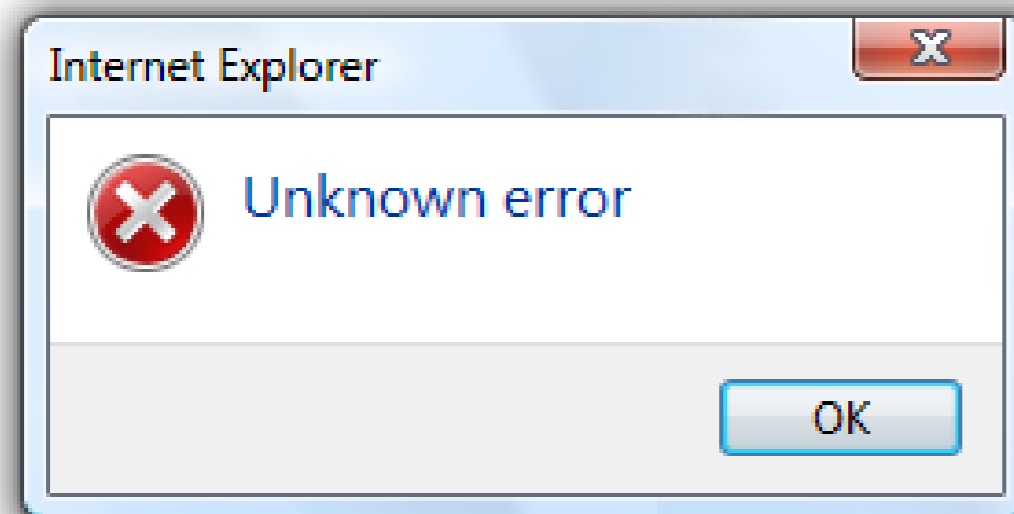
Having **“Yes”, “No”** and **“Cancel”** for a question is confusing



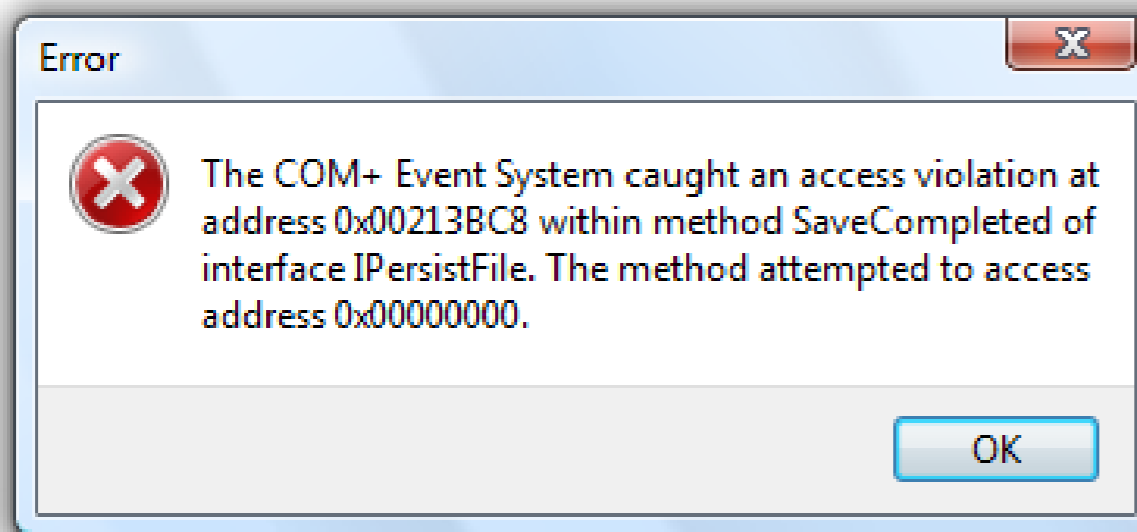
# Question 4: Clues

Which of the following is a **good practice** when using **system alerts**?

Examples



What are the **problems** with this **error** message?



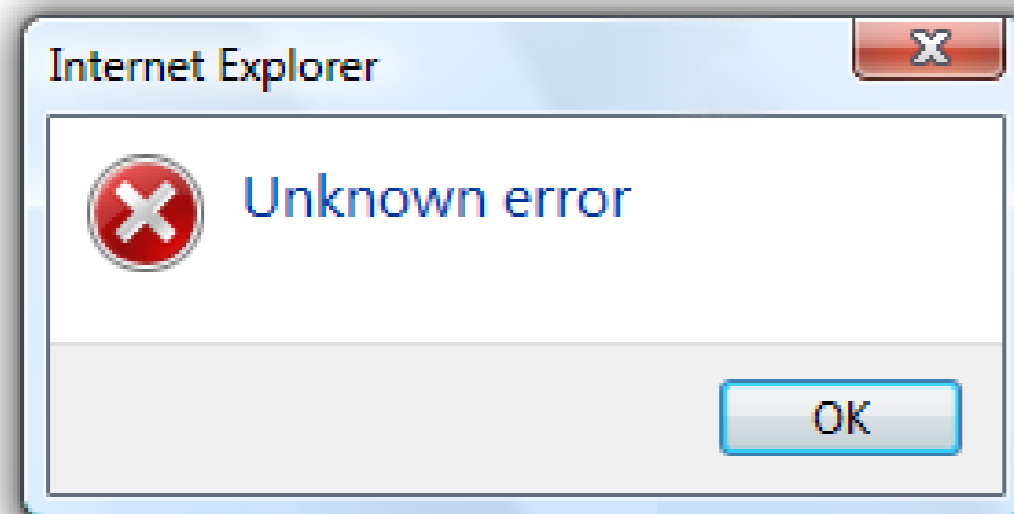
What are the **problems** with this **error** message?



# Question 4: Clues

Which of the following is a **good practice** when using **system alerts**?

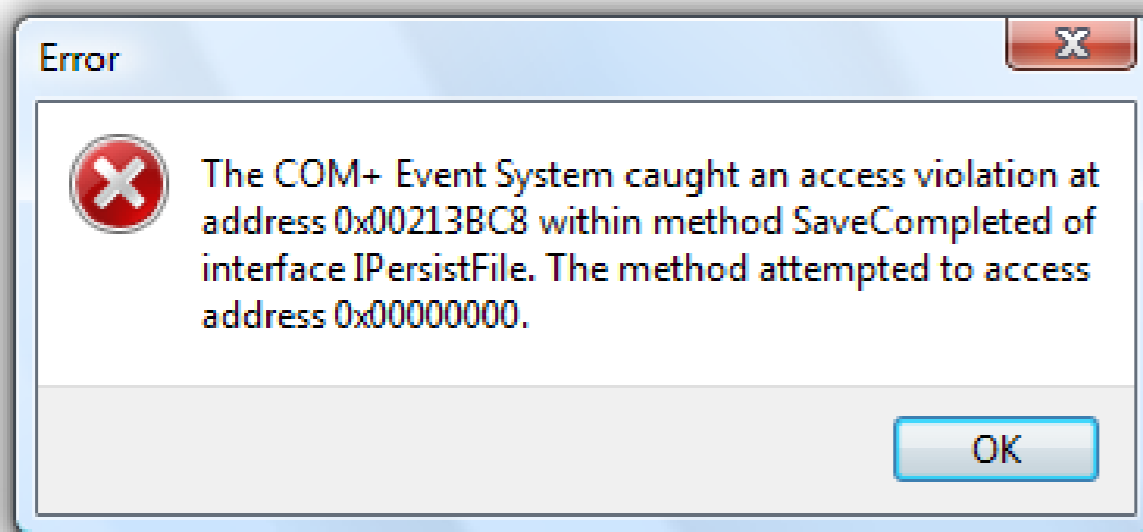
Examples



Does **not** give a specific **problem**

There is **nothing** the **user** can do

Instead: Design **proper error handling** into the system



Does **not help** the user **understand** the **problem**

Is **useless** to the **user**

Cause: **Programmers** make error messages for

**themselves**

# Question 5

The **purpose** of **HCI testing** is to make a software **system** **easy** to **learn** and easy to **remember**.

- a. True
- b. False



# Question 5: Clues

The **purpose** of **HCI testing** is to make a software **system** **easy** to **learn** and easy to **remember**.

Purpose of HCI testing

To ensure optimal user experience

Users should experience some *value* from using the system

<b>Understandable</b>
Easy to <b>learn</b>
Effective to <b>use</b>
Easy to <b>remember</b>
Satisfactory to <b>use</b>



# Question 6

HCI testing's **primary concern** are the **aesthetics** of a **software** program.

- a. True
- b. False





# Question 7

Which of the following **elements** are **components** of the **HCI testing framework**? (Draw arrows)

Components of the HCI testing framework	Laws and regulations
	Interface standards
	Interface dynamics
	Accuracy
	Usability
	Compliance
	Aesthetics

# Question 8

When **designing** \_\_\_\_\_ software **systems**, one **has** to:

- **Understand** how the users **think** and **behave**
- Gather **fact** and **data** instead of relying on opinion and speculation
- Perform **studies**, **design** and **test** on users before implementation
- **Iterate**



# Question 9

When **specifying demands** (creating requirements) for **user-centric** software systems, we can **use personas**.

- a. True
- b. False



# Question 9: Clues

When **specifying demands** (creating requirements) for **user-centric** software systems, we can **use personas**.

Specifying **demands**

A **crucial** step in the development and testing processes

We want to **know** exactly **what** the system should do / **how** it should behave

Serve to ...

**Document** the problems

**Summarise, analyse, and compare** results

Create the **basis** for problem **conceptualisation**

Ensure **quality control** mechanisms



# Question 9: Clues

When **specifying demands** (creating requirements) for **user-centric** software systems, we can **use personas**.

## Personas

Design and specification tool

Description of a **representative** user

Provide **information** about

**Who** the users are

**Goals, motivations,** and **activities** of usage

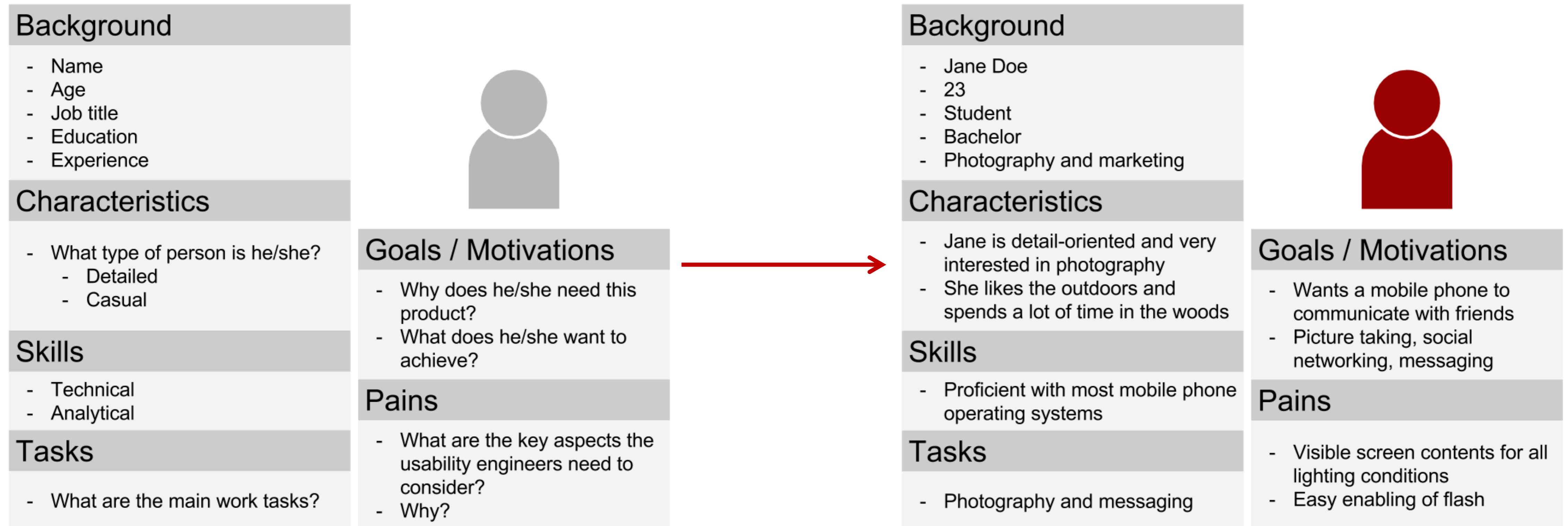
Informed based on **research** and **checked** to **validate**

assumptions



# Question 9: Clues

When **specifying demands** (creating requirements) for **user-centric** software systems, we can **use personas**.



# Question 10

Is it **allowed** to use **low-fidelity prototyping** when **designing** a **user-centric software system**?

- a. Yes
- b. No





# Question 10: Clues

Is it allowed to use low-fidelity prototyping when designing a user-centric software system?

Development of concept

We have a problem to solve

Creative process → How can we solve this problem?

Prototyping

Exploring various ways of representing and solving the problem

Low-fidelity prototyping

High-fidelity prototyping





# Question 10: Clues

Is it **allowed** to use **low-fidelity prototyping** when **designing** a **user-centric software system**?

**Low-fidelity** prototyping

**Mock-ups** and wireframes

Start by **sketching** on **paper** → Paper prototyping

Focus on **structure** and **function** → As opposed to details

Do **not** apply any **design**

Create **multiple** concepts

**Evaluate**, **refine**, and **narrow** down → Based on **feedback**

Note: It is **not** supposed to be **pretty**!



# Question 10: Clues

Is it allowed to use low-fidelity prototyping when designing a user-centric software system?

Characteristics of low-fidelity prototyping

**Fast** → Quick progress on requirements specification and testing

**Easy** → Anyone can do it

No specialised skills required

**Simple** → No unnecessary details included

Non-technical individuals can easily understand the concept

**Cheap** → Takes pen and paper

**Relevant** → Communicates the essentials



# Part II: Exercises and Open-ended questions

# Exercise 1: Design of Things

Watch the video on “Thoughtful design”

[www.youtube.com/watch?v=E\\_rwwEo5YhY](http://www.youtube.com/watch?v=E_rwwEo5YhY)

Follow-up questions

- Can you give examples of everyday things that seemed confusing to you?
- Have you experienced confusing software?



# Exercise 2: SiO Case Study

The **Student Organisation** (SiO) has updated their **website** ([www.sio.no](http://www.sio.no)) to make it more **effective, efficient,** and **user-friendly**.

Now they are **conducting “user observation”** for **usability testing**.

A **group of students** are **invited to perform some actions** on the **site**.



# Exercise 2: SiO Case Study

Which of the following **points** must an **observer** keep in **mind** while conducting “**user observation**”?

1. Try to cover a huge number of tasks and to make a lot of observations
2. Give the students time to perform the task instead of interrupting or showing them how to perform said task
3. The focus of the observation should be to check the expertise level of the users
4. The focus of the observation is to discover the problems in the software with the help of the user

# Exercise 2: Clues

Which of the following **points** must an **observer** keep in **mind** while conducting “**user observation**”?

**Good practices** for user **observation**

Start looking for **potential users** in **advance** of the study

Be **realistic** about how **many observations** you will have **time** for during the study

**Give** the user **time** to **perform** the **task**

**Refrain** from **interrupting** the users or **showing** them how to do something

Make the study a **positive experience** for the user

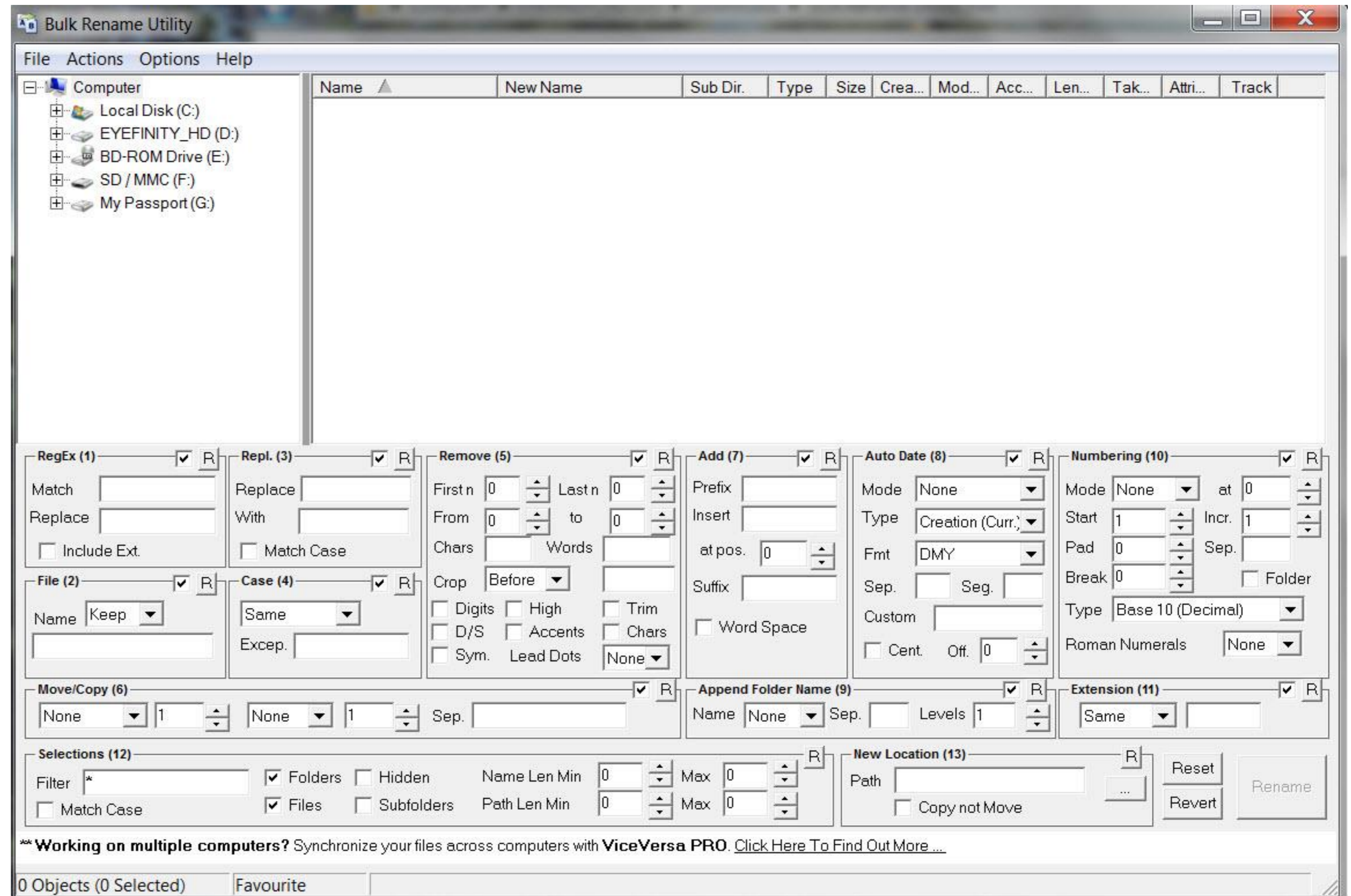
**Recall** that you are **not testing** the **user**



# Exercise 3: Usability Issues

What are some **usability issues** with the following application?

How would you **improve** the **HCI** aspects of this application?





# Exercise 4: Usability Issues

Can you give **examples** of **poorly designed websites**?

Follow-up questions

- What are the usability issues with the site?
- Why is this poor practice?
- What should be done to improve the HCI aspects?



**The seminar slides are made by**

Yulai Fjeld, revised by Eva Hadler Vihovde

