

Professors IDI

IDI – Usability Testing

Outline

- **Concepts**
- Usability testing
- Formal usability tests
- Use cases

Concepts

- Usability:
 - **Ease of use** and **acceptability** of a system or product for a particular class of users carrying out specific tasks in a specific environment.
 - Where “**Ease of use**” affects user performance (efficacy, efficiency), satisfaction (comfort).
 - And “**Acceptability**” affects whether or not the product is used.

Concepts

- Usability:
 - 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.
 - To be useful, usability has to be specific. It must refer to particular tasks, particular environments and particular users.
 - **So has to be its testing!**

Concepts

- How to test?
 - Ease of use is inversely proportional to the number and severity of difficulties people have in using software.
 - Let's examine the difficulties!!!

Outline

- *Concepts*
- **Usability testing**
- Formal usability tests
- Use cases

Usability testing

- Two major families by goals:
 - **Determine usability problems** (*i.e. text editor*):
 - Discovery, prioritization, and resolution of usability problems
 - Iterative testing
 - **Measure task performance** (*i.e. 3D selection*). Include two fundamental tasks:
 - The development of the usability objectives
 - Iterative testing to determine if the product under test has met the objectives

Usability testing

- Great variety of usability tests:
 - Can be very informal or very formal
 - Observer might sit next to the participant, watch through a one-way glass, or watch the on-screen behaviour of a participant who is performing specified tasks.
 - Often use think-aloud (TA)
 - Observers might watch one or two participants at a time
 - Evaluated software can be varied:
 - Prototypes, under development, competitive products...

Usability testing

■ Think-Aloud:

- Participants must talk about what they are doing as they do it
 - Prompt participants to resume if they stop talking
 - What users say during tasks is more reliable than posterior interviews
 - In interviews users are inclined to answer what they think you would like them to
 - When people verbalize after the experiment, they only note what they remember
 - People tries to rationalize their behaviour (giving reasons why they did not see a button...)

Usability testing

- Think Aloud:
 - Can be apply to almost any usability testing method
 - Seem to work better with pairs of participants
 - Seem to be best suited than silent participation in problem discovery
 - Better for problem discovery than measurement

Usability testing

- Testing techniques:
 - Remote testing
 - Extra work
 - Heuristic/expert evaluation
 - Extra work
 - “Formal” usability tests

Outline

- *Concepts*
- *Usability testing*
- **Formal usability tests**
 - Environment
 - Usability tests tasks and roles
 - Development
 - Reporting
 - Cheap tests
- Use cases

Formal usability tests. Environment

- “Formal” usability tests require a controlled environment
 - Inside a room, outside...
 - Illumination conditions (useful for perception studies)
 - Devices used (e.g. computer with Internet connection and a browser, or a mobile...)
 - Other conditions (e.g. connection quality...)

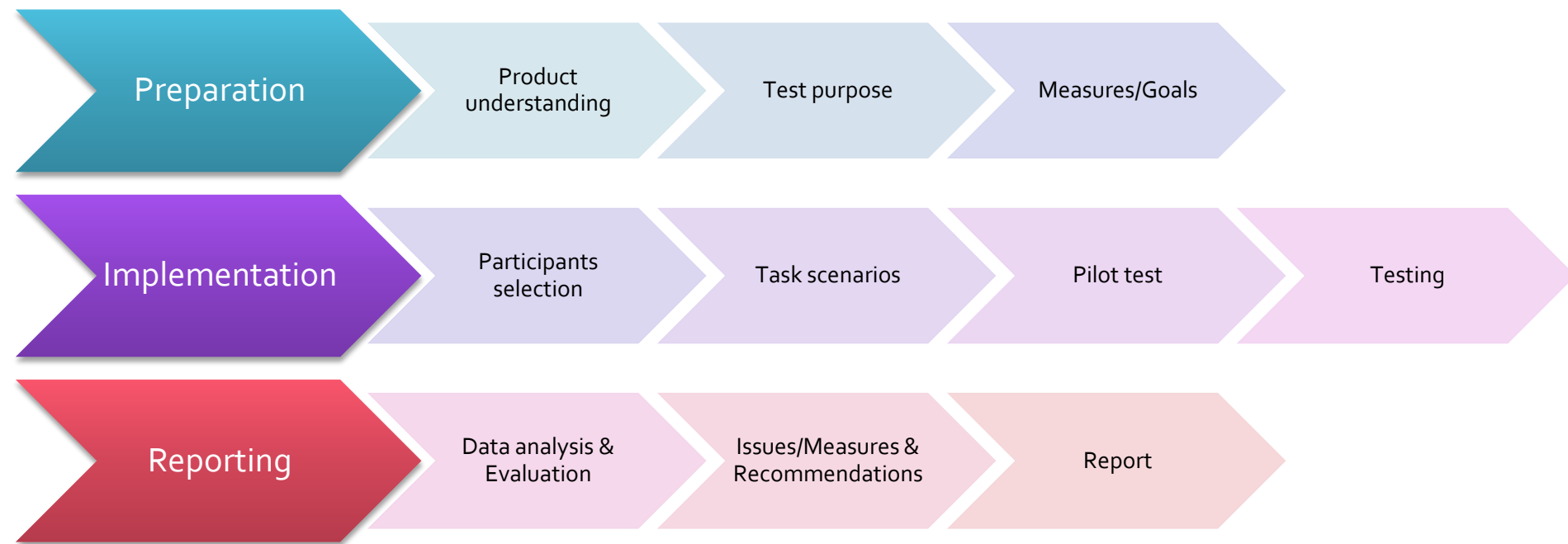
Usability lab 😊

Formal usability tests. Environment

- Set of soundproofed rooms
 - Proper recording and avoiding distractions to participants
- Different areas and equipment
 - Participant area (where the experiment is carried out)
 - Observer area with one-way glass
 - Executive viewing area behind the primary observer area
 - Video cameras and microphones, telephone...

Formal usability tests. Tasks and roles

■ Usability test workflow:



Formal usability tests. Tasks and roles

- Usability test roles:
 - **A:** Test administrator
 - **B:** Briefer
 - **CO:** Camera Operator
 - **DR:** Data Recorder
 - **HD:** Help Desk Operator
 - **PE:** Product Expert
 - **S:** Statistician

Formal usability tests. Tasks and roles

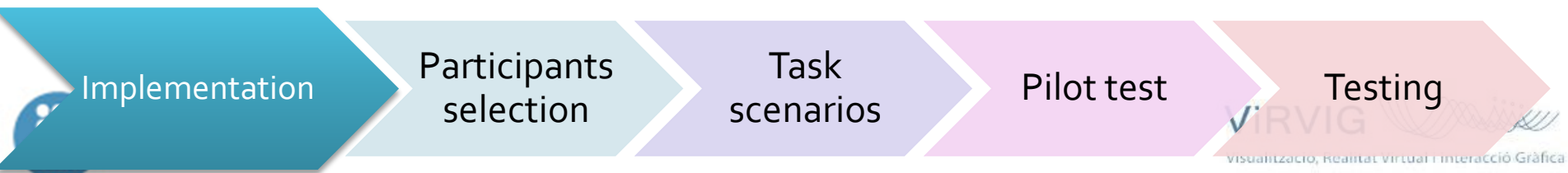
- Preparation:
 - Product understanding: Purpose of the product, parts ready to test, type of users...: **A, PE**
 - Test purpose: Product comparison, within/between subjects...: **A, S**
 - Measures/Goals: Number of iterations, counting mistakes/errors, timings...: **A, S**

Formal usability tests. Tasks and roles

- Product understanding (**A**):
 1. Understand the purpose of the product
 2. Parts of the product are ready for testing
 3. Types of people who will use the product
 4. Determine the use given to the product
 5. Conditions of usage of the product

Formal usability tests. Tasks and roles

- Implementation:
 - Participants' selection: **A**
 - Task scenarios: initial conditions, steps: **A**
 - Pilot test: Members of the team: **A, B, CO, DR, HD**
 - Testing, **A, B, CO, DR, HD**



Formal usability tests. Tasks and roles

- **Participants' selection**
 - Extra work

Formal usability tests. Tasks and roles

- **Test task scenarios:**
 - Must be representative
 - Core tasks: Features that everybody uses (write a text)
 - Peripheral tasks: Features used less often (table insertion)
 - Once the tasks are defined, scenarios of use must be created
 - Define initial conditions
 - Description of the scenario: what to do and why
 - Some action must be taken on finish
 - Should not provide step-by-step instructions but should include details
 - Not all users must be provided with the same scenarios (may depend on the user profile)

Formal usability tests. Tasks and roles

- Testing (**A, B, CO, DR, [HD]**):
 - Brief participants: **B**
 - Initial questionnaire: **B**
 - Develop tasks: **B, CO, DR, [HD], A**
 - Debrief: **B**
 - Final questionnaires: **B**

Formal usability tests. Tasks and roles

- Reporting (whole team):
 - Data Analysis & Evaluation: **A, S**
 - Issues/Measures & Recommendations: **A, S, team**
 - Report: **A, S, team**



Formal usability tests. Reporting

- **Reporting Results:**
 - Describe & prioritize the usability problems
 - Present quantitative measurements

Should lead to a recommendation

Formal usability tests. Reporting

Problem evaluation:

- Frequency: Number of users that find a problem divided by the number of users testing the app or web
 - Easy (objective) to evaluate
- Severity: Importance of the problem
 - Might be completely catastrophic or simply cosmetic
 - Difficult (more subjective) to evaluate

Formal usability tests. Reporting

- Reporting. *Usability problems*:
 - Should indicate the importance: severity
 - Can be classified:
 - Mistakes: Errors due to incorrect intention
 - Slips: Errors due to appropriate intention but incorrect action
 - Expertise does not affect on the number of errors
 - But affects how fast they are handled

Formal usability tests. Reporting

- Rating the severity of usability problems:
 - Some thoughts on severity and frequency
 - Local evaluation: Jeff Rubin, Jakob Nielsen...
 - Global evaluation: Dumas and Redish

Formal usability tests. Reporting

- Problem evaluation. **Dumas and Redish:**
 - **Level 1:** Prevents Task Completion
 - **Level 2:** Creates significant delay and frustration
 - **Level 3:** Problems have a minor effect on usability
 - **Level 4:** Subtle and possible enhancements/suggestions

Formal usability tests. Reporting

Problem evaluation. **Jeff Rubin:**

- **4: Unusable:** The user is not able to or will not want to use a particular part of the product because of the way that the product has been designed and implemented.
- **3: Severe:** The user will probably use or attempt to use the product here, but will be severely limited in his or her ability to do so.
- **2: Moderate:** The user will be able to use the product in most cases, but will have to undertake some moderate effort in getting around the problem.
- **1: Irritant:** The problem occurs only intermittently, can be circumvented easily, or is dependent on a standard that is outside the product's boundaries. Could also be a cosmetic problem.

Formal usability tests. Reporting

- Reporting. *Recommendations*:
 - Create a problem grid: frequency/impact
 - Global changes (prevent task completion) first
 - Must be checked:
 - *A missing help* may be a global problem or something related with a concrete UI
 - Try to give at least one recommendation for each problem
 - Present the different trade-offs clearly

Formal usability tests. Reporting

- Problem evaluation. Conclusions
 - Do not use a large number of categories
 - Do not get obsessed by the number of categories either
 - Different evaluators may disagree on some problems' severity
 - Treat frequency separately from severity
 - Do not forget to point out positive findings

Formal usability tests. Cheap tests

- Testing just a single person early is much better than 50 near the end
- The point of testing is to inform your judgment

Formal usability tests. Cheap tests

- Testing on the cheap
 - Guerrilla usability testing
 - Steve Krug's "usability testing on 10 cents a day"

Formal usability tests. Cheap tests

- Guerrilla usability testing
 - Take someone in a coffee or public space and ask her to use a website for a couple of minutes
 - Observe users
 - Ask open-ended questions such as “What would you do here?”
 - Get to know them a bit
 - Offer coffee or bagels
 - Analyse captured data
 - Considering your audience

Formal usability tests. Cheap tests

- “Usability testing on 10 cents a day”
 - Prepare some tasks to evaluate
 - Grab somebody from the company as user
 - Gather stakeholders in an observing room
 - Let the user do a set of tasks
 - Capture gestures, mouse, record...
 - Discuss over lunch (order pizza for everybody)
 - Report

Outline

- *Concepts*
- *Usability testing*
- *Formal usability tests*
- **Use cases**
 - Guerrilla testing: WhatsApp web app
 - Measure test: Depth perception in VR

Use case. WhatsApp web app

- Web application

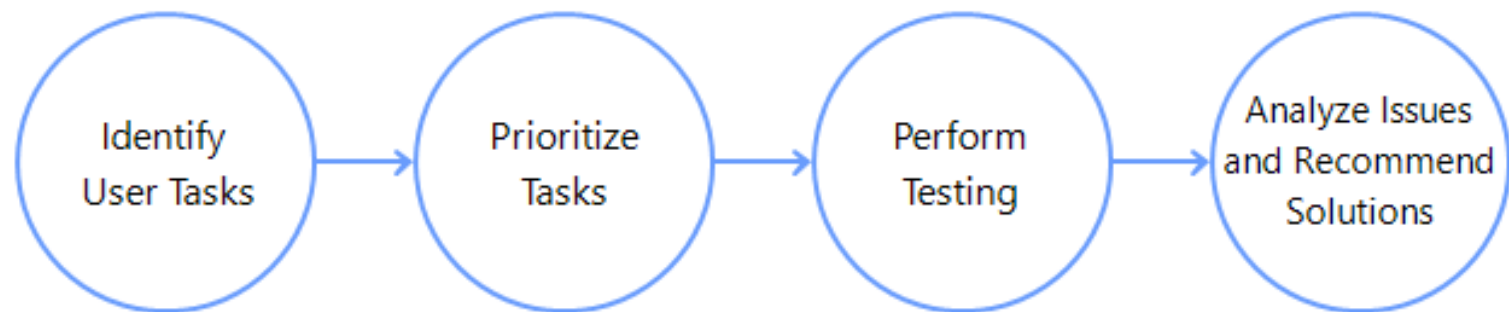


Use case. WhatsApp web app

- Type of usability test: Guerrilla
- Objective
 - Identify common problems on WhatsApp web
- Testing parameters
 - What is tested: Just two common tasks
 - Participants: 3 users, 2 never used it previously
 - Test procedure: Observation + interview

Use case. WhatsApp web app

- Test process



Use case. WhatsApp web app

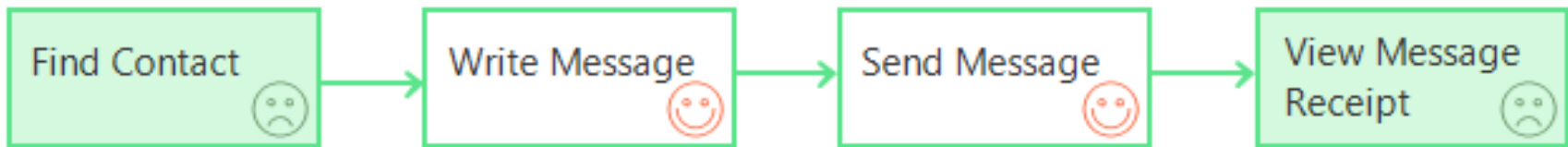
- User tasks
 - Send a message to a friend
 - Share photos with a friend

Use case. WhatsApp web app

- Development
 - Give the instructions to the users
 - Users are observed with performing actions
 - Asked about the experience on certain subtasks

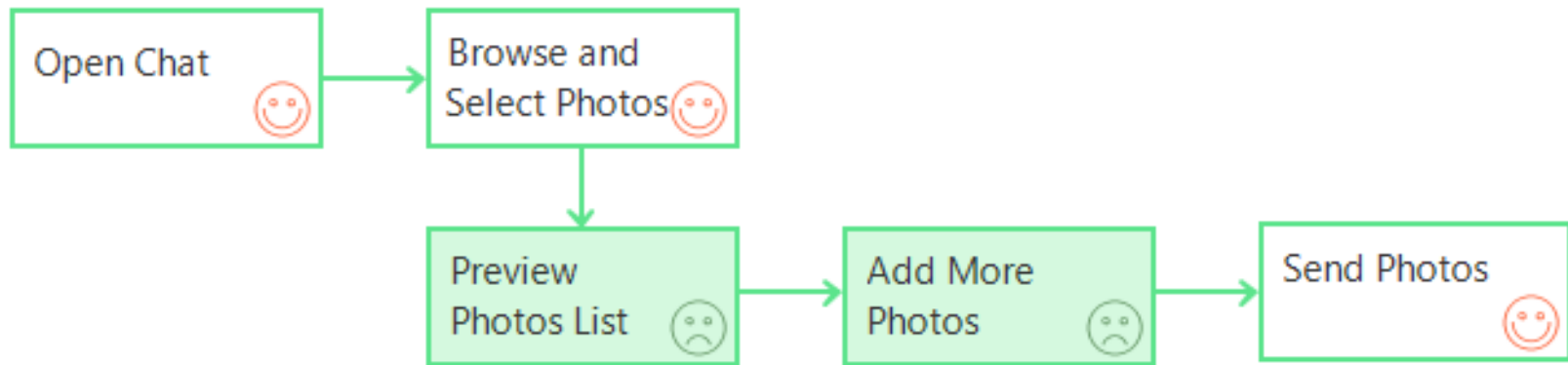
Use case. WhatsApp web app

- Development of task 1



Use case. WhatsApp web app

- Development of task 2



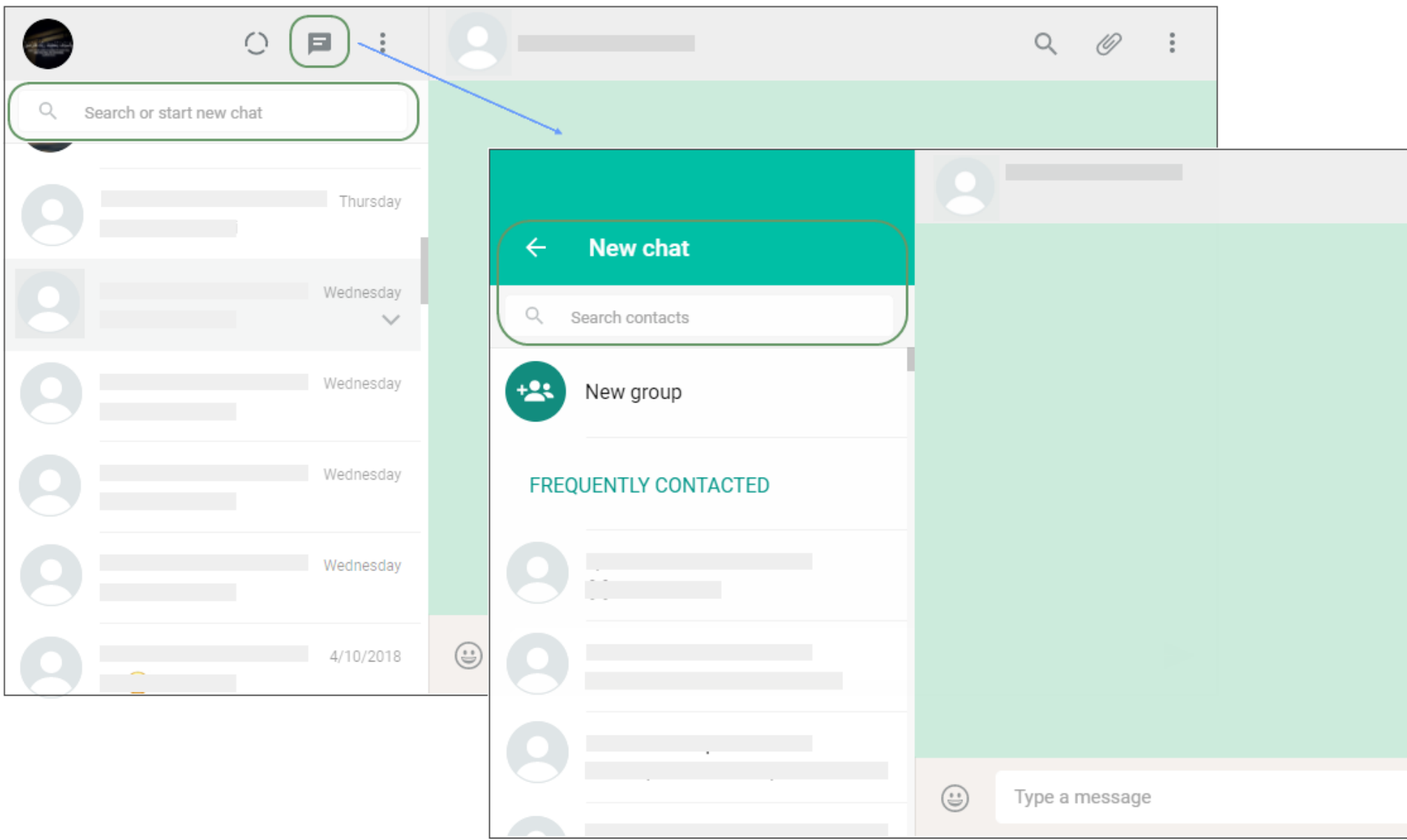
Use case. WhatsApp web app

- Analyse problems. Finding a contact:
 - There are two ways to start a new chat:
 - (i) Search within Chats list, (ii) Go to New Chat icon on top and search contact
 - The user was not clear to differentiate between these two options

Use case. WhatsApp web app

- Analyse problems. Finding a contact:
 - Searching in Chats list gives an assumption that search will run through the Chats list only, and in fact it works for both Chats and Contacts
 - On the other side, New Chat option also provides a list of Chats as well as Contacts. This requires some kind of clarity

Use case. WhatsApp web app



Use case. WhatsApp web app

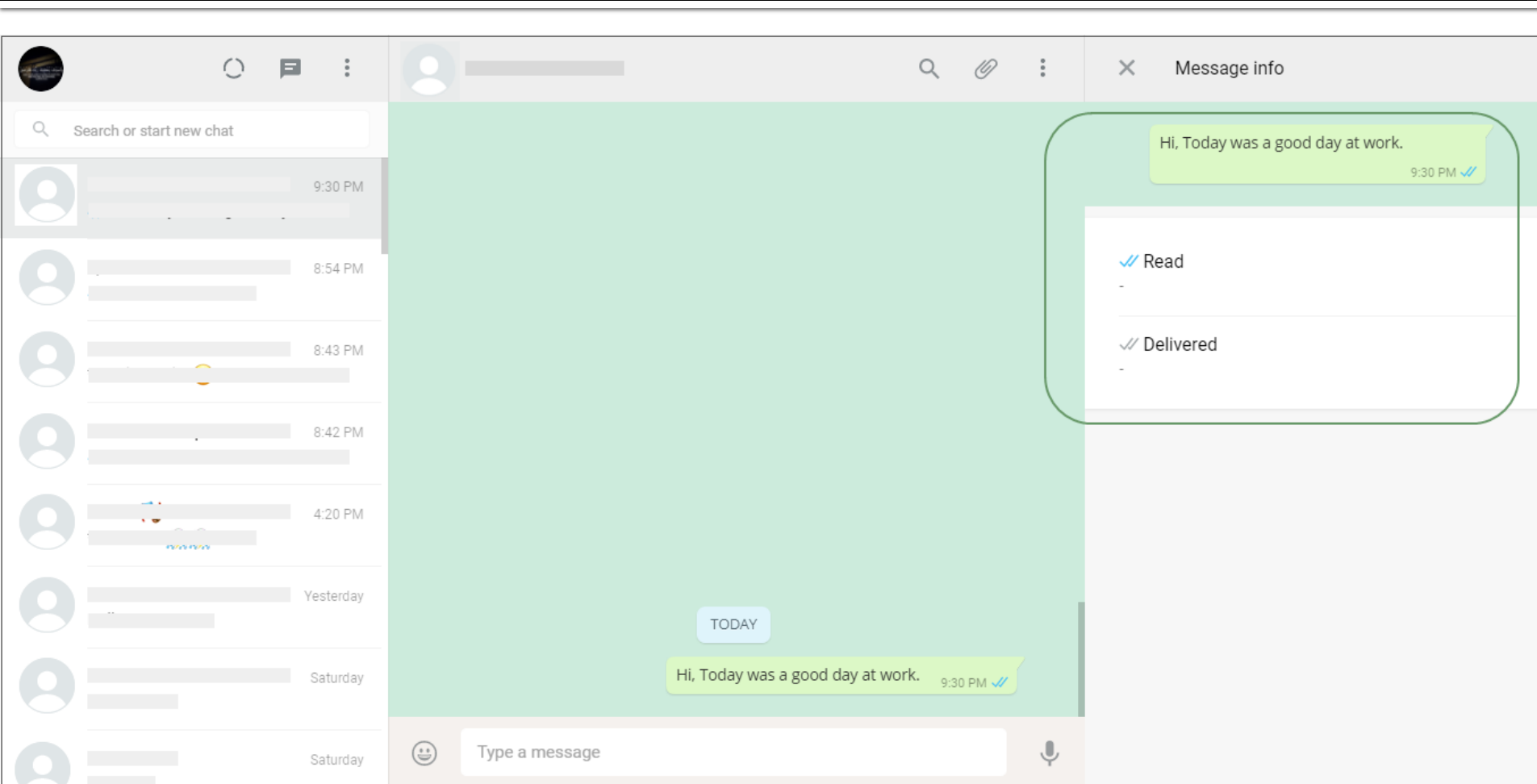
■ Recommendation:

- A clear separation between Chats and Contacts is needed
 - Can be done by giving a filter option in Contact list, or a single list can be sorted based on Recent Chats or Contact names.

Use case. WhatsApp web app

- Analyse problems. Viewing Message Receipt:
 - In Message Info pane, the area showing message status is merged with the Message pane
 - Also, it is not clear that user is viewing status of which message
 - Also, it took time for the user to find Close icon on top on Message Info pane

Use case. WhatsApp web app



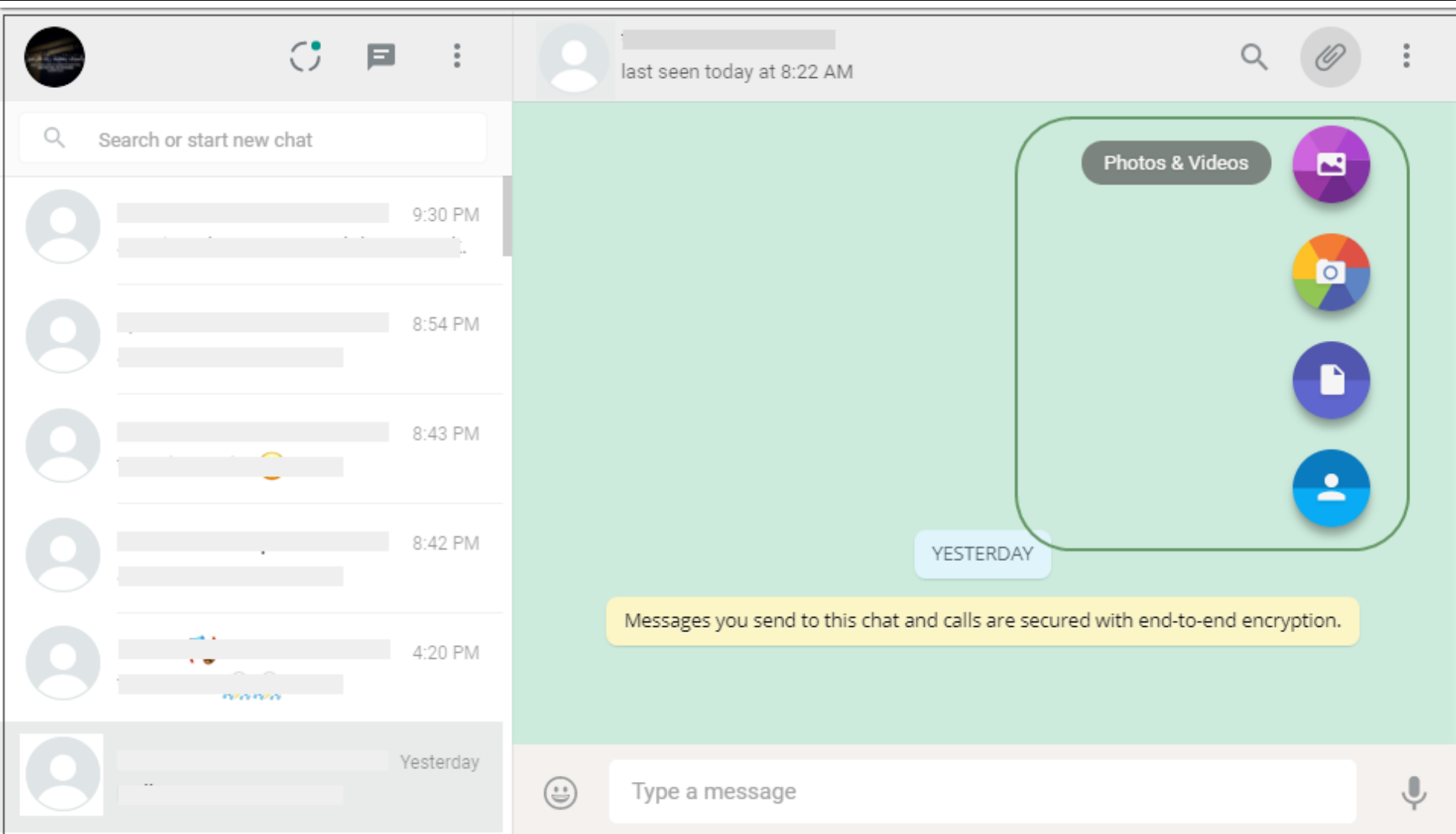
Use case. WhatsApp web app

- **Recommendation:** The area of Message info pane and Message pane needs to differentiate clearly
 - Since this is desktop version and Message area is still visible when Info pane is opened, the link between message and its info could be made more prominent

Use case. WhatsApp web app

- Analyse problems. Using attach
 - The Attach menu and tooltips do not match with the UI
 - Shows a totally different theme and experience in current screen

Use case. WhatsApp web app



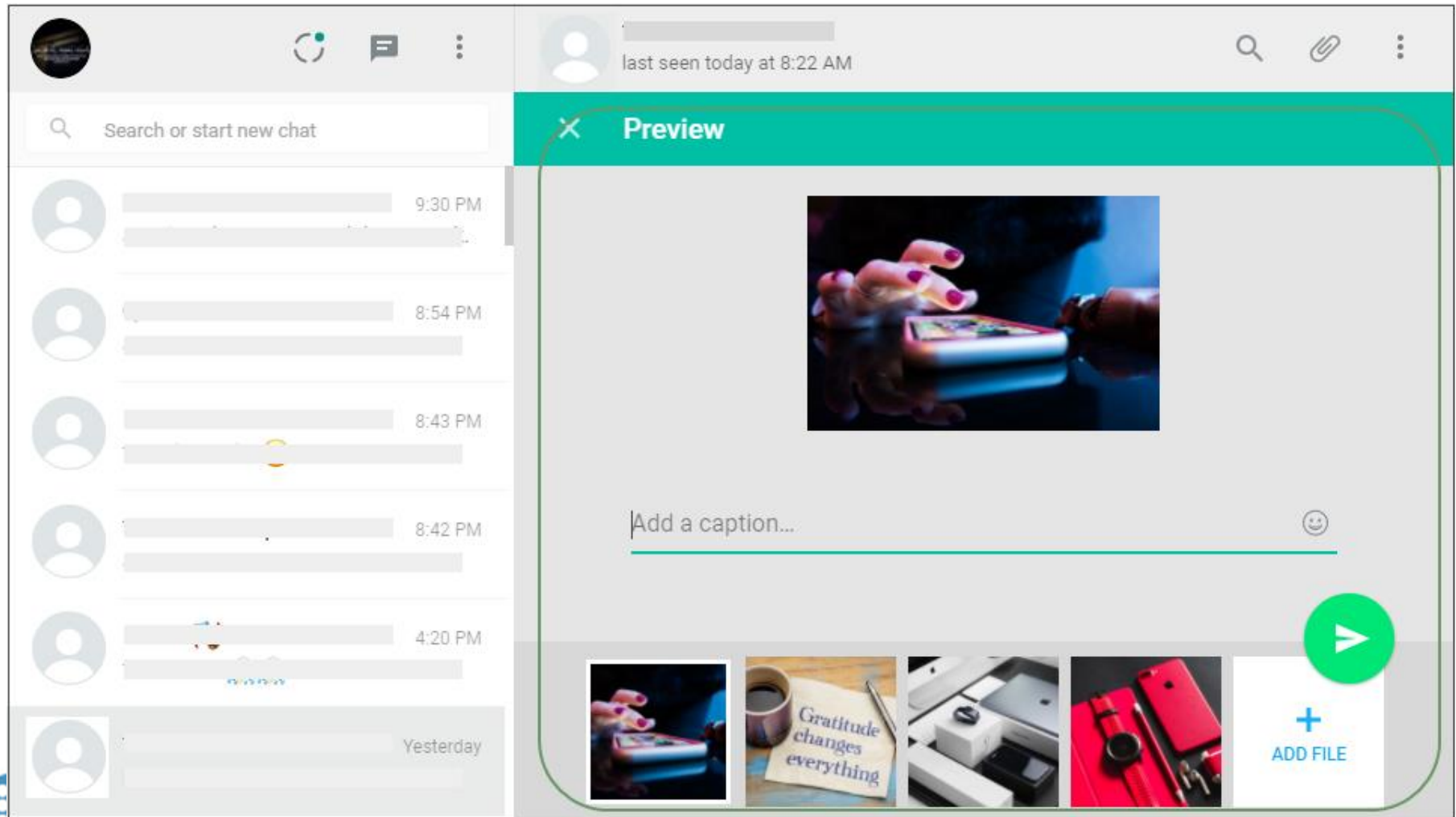
Use case. WhatsApp web app

- Recommendation: Make menu placement and theme consistent with UI.
 - Instead of Tooltips, the option names along with icons seems more helpful.

Use case. WhatsApp web app

- Analyse problems. Attaching photos:
 - Close icon with Preview title is confusing.
 - The user clicked it just to close the preview of selected photos, but it discards all the selected photos.
 - Adding more files option is not clear.
 - The Attach icon still displays on top, but it is not functional. The user clicked on that icon first.
 - It is difficult to navigate large number of selected files.

Use case. WhatsApp web app



Use case. WhatsApp web app

- Recommendation:
 - Rename preview area to Attachments to avoid any confusion for the user.
 - Provide scrolling in thumbnails area
 - User should be able to add more files by clicking an Add icon with caption

Use case. WhatsApp web app

- More Observations
 - Using a scrollbar requires high accuracy to hold the bar and scroll it
 - Cursor is changed to resize when user tries to scroll Message pane
 - No keyboard scroll allowed in Contacts & Contact/Group Info
 - Little visibility of actions' visual feedback (bottom left)
 - Were skipped multiple times
 - Status cannot be updated on desktop version
 - Users cannot see others' status

Use case. Depth perception in VR

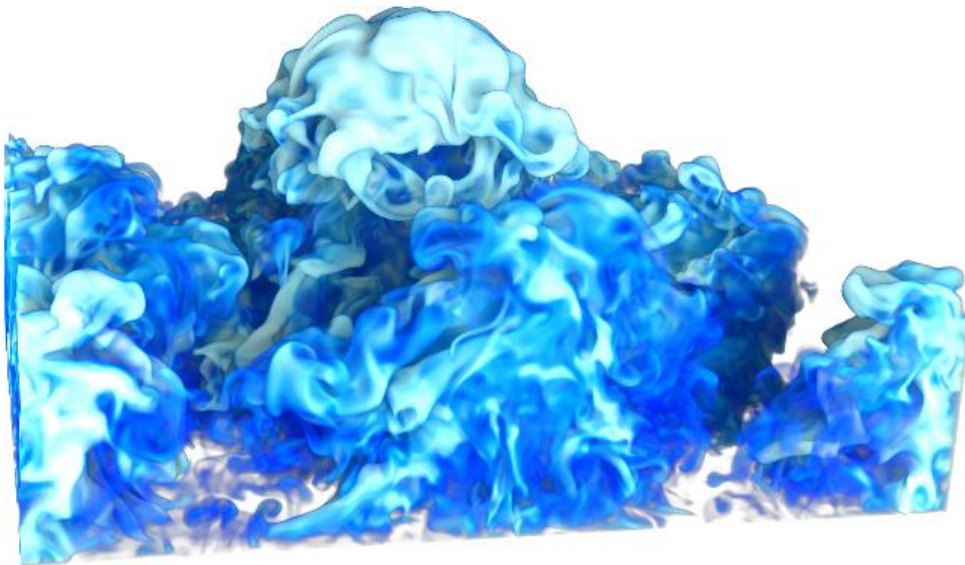
- Goal:
 - Evaluate performance of shading technique in VR environments
- Context:
 - Perception of complex, volume datasets is difficult in VR
 - Shading techniques may enhance shape and depth perception

Use case. Depth perception in VR

- Purpose of the test:
 - Analyze whether shading techniques influence the perception of shapes and depth in VR
- Methodology:
 - Provide images under different shading conditions
 - Ask the users to classify two points of the scene placed at different depths
 - Analyze the results obtained

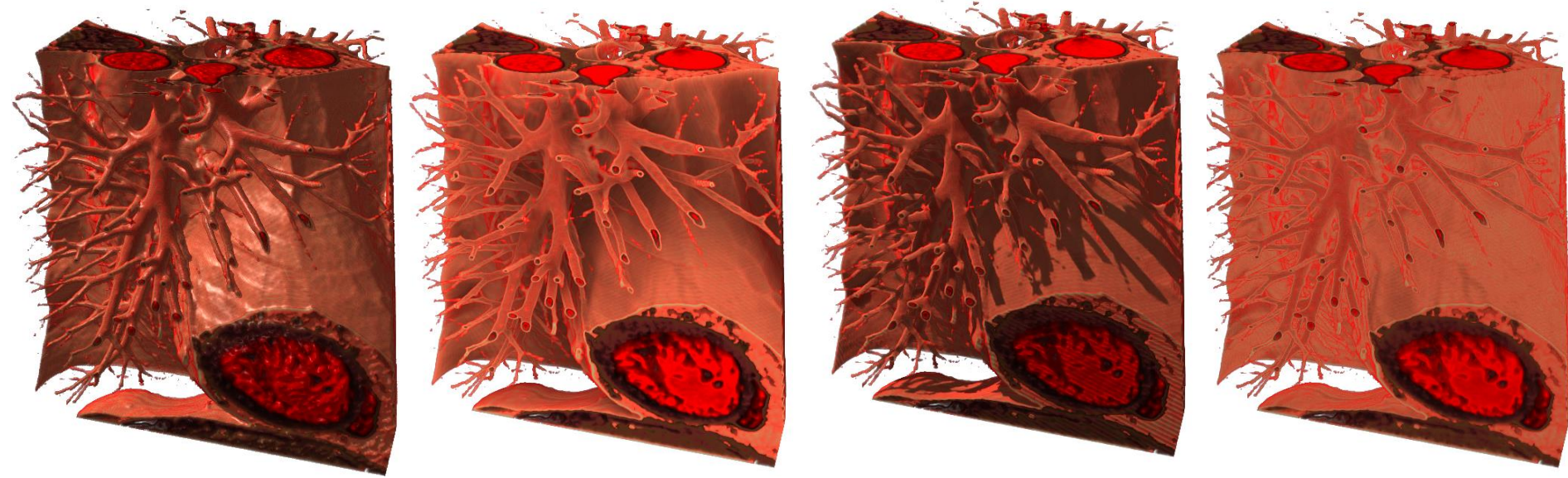
Use case. Depth perception in VR

- Sample images:



Use case. Depth perception in VR

- Shading techniques:



Use case. Depth perception in VR

- Test preparation:
 - Select shading models (4)
 - Select models (likely unknown to users)
 - Determine number of participants, iterations
 - Low level perception problem -> should be > 10
 - Latin squares balance results -> 16 per experiment
 - Two tasks

Use case. Depth perception in VR

- Images selection:
 - Select models likely unknown
 - Avoid previous knowledge
 - Random shading sorting
 - Avoid learning (shading)
 - Random model sorting
 - Avoid learning (model)
 - Latin squares
 - Avoid fatigue and learning (within users)

Use case. Depth perception in VR

- Measures (what we measure in the test):
 - Time to answer
 - Correctness

Use case. Depth perception in VR

- Variables to include in the analysis (to discard confounding or correlating variables)
 - Shading technique
 - Depth values
 - May analyze if absolute difference correlates with correctness
 - Previous VR background
 - Information of images for left and right eye
 - Luminance of the points' environment
 - Correlation between depth and shading maps

Use case. Depth perception in VR

- Experiment setup:
 - 3D TV
 - Users placed at fixed distance
 - Chair to reduce movements
 - Avoid parallax as confounding variable
 - Dark room
 - External light (for virtual light source consistency analysis)

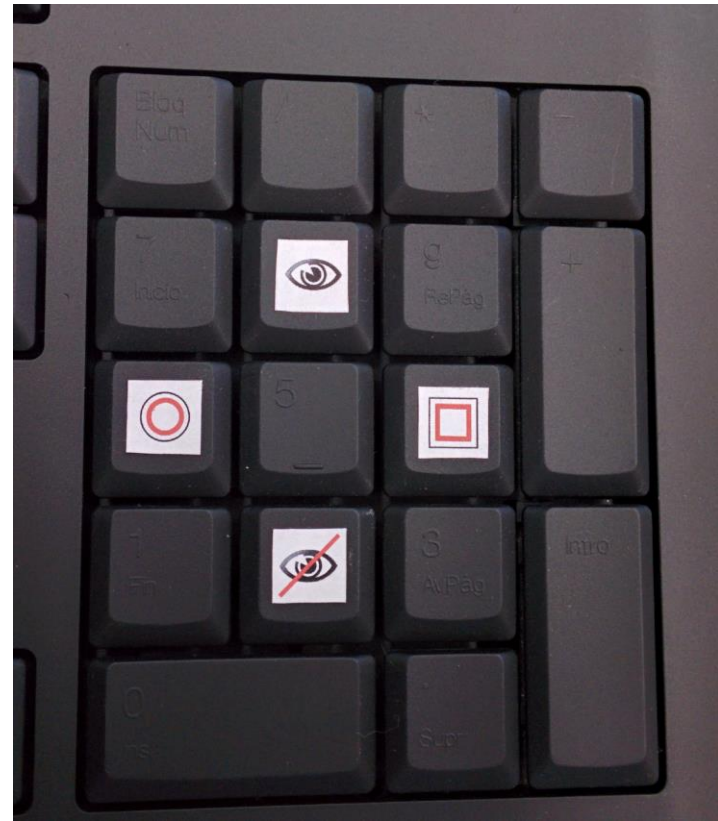
Use case. Depth perception in VR

- Experiment setup:



Use case. Depth perception in VR

- Experiment setup:
 - Modified keyboard to facilitate entry
 - Will compute timings

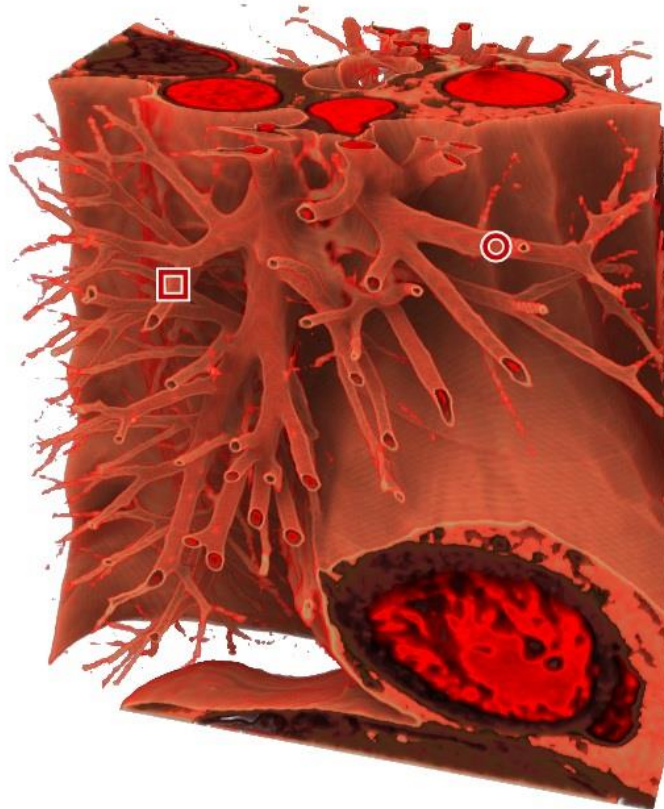


Use case. Depth perception in VR

- Experiment setup:
 - Initial questionnaire (background, VR exposition...)
 - Initial training
 - Tasks
 - May rest between tasks
 - Post questionnaires

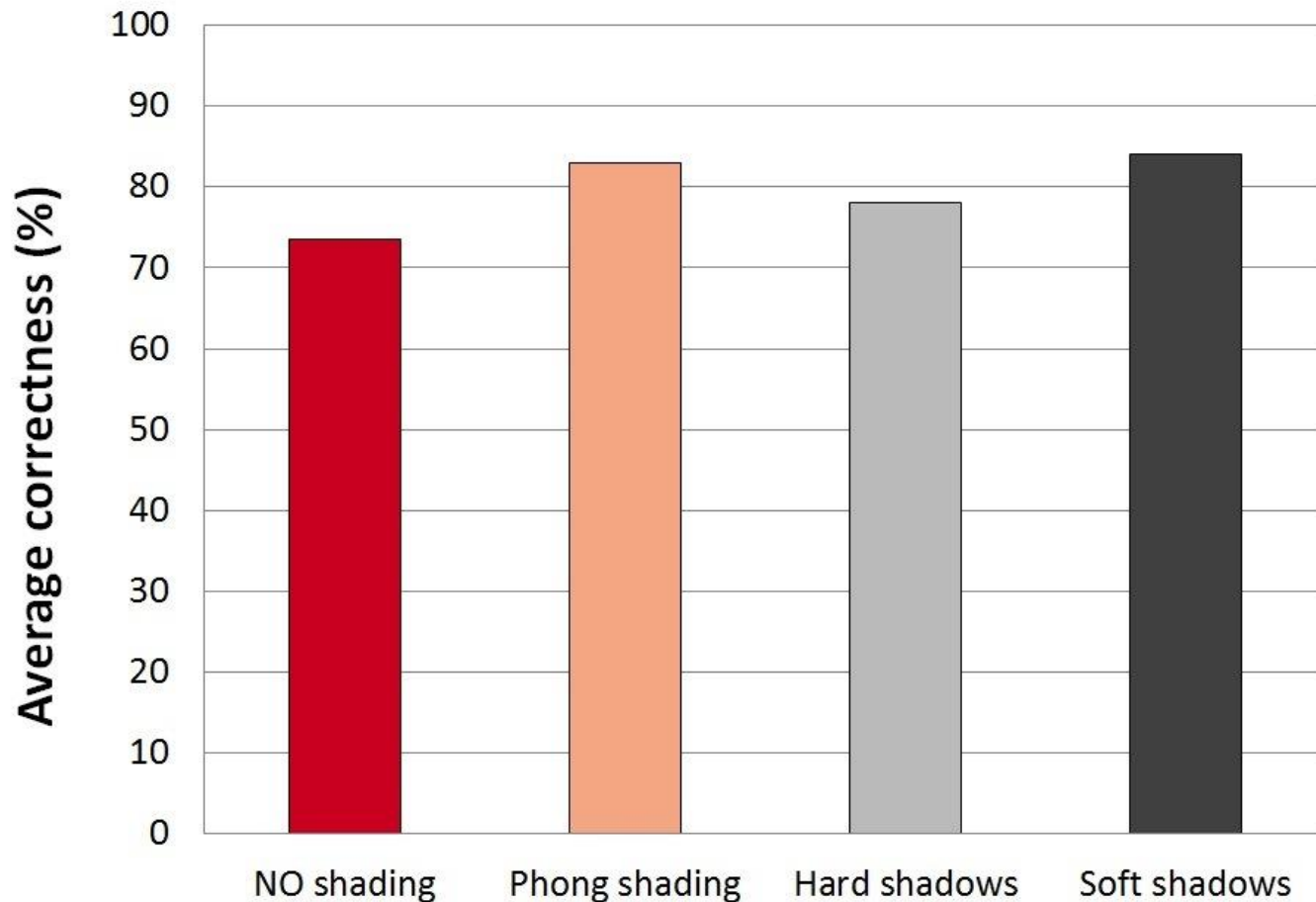
Use case. Depth perception in VR

- Task: Select the closer point. 2-alternative forced choice (2AFC)



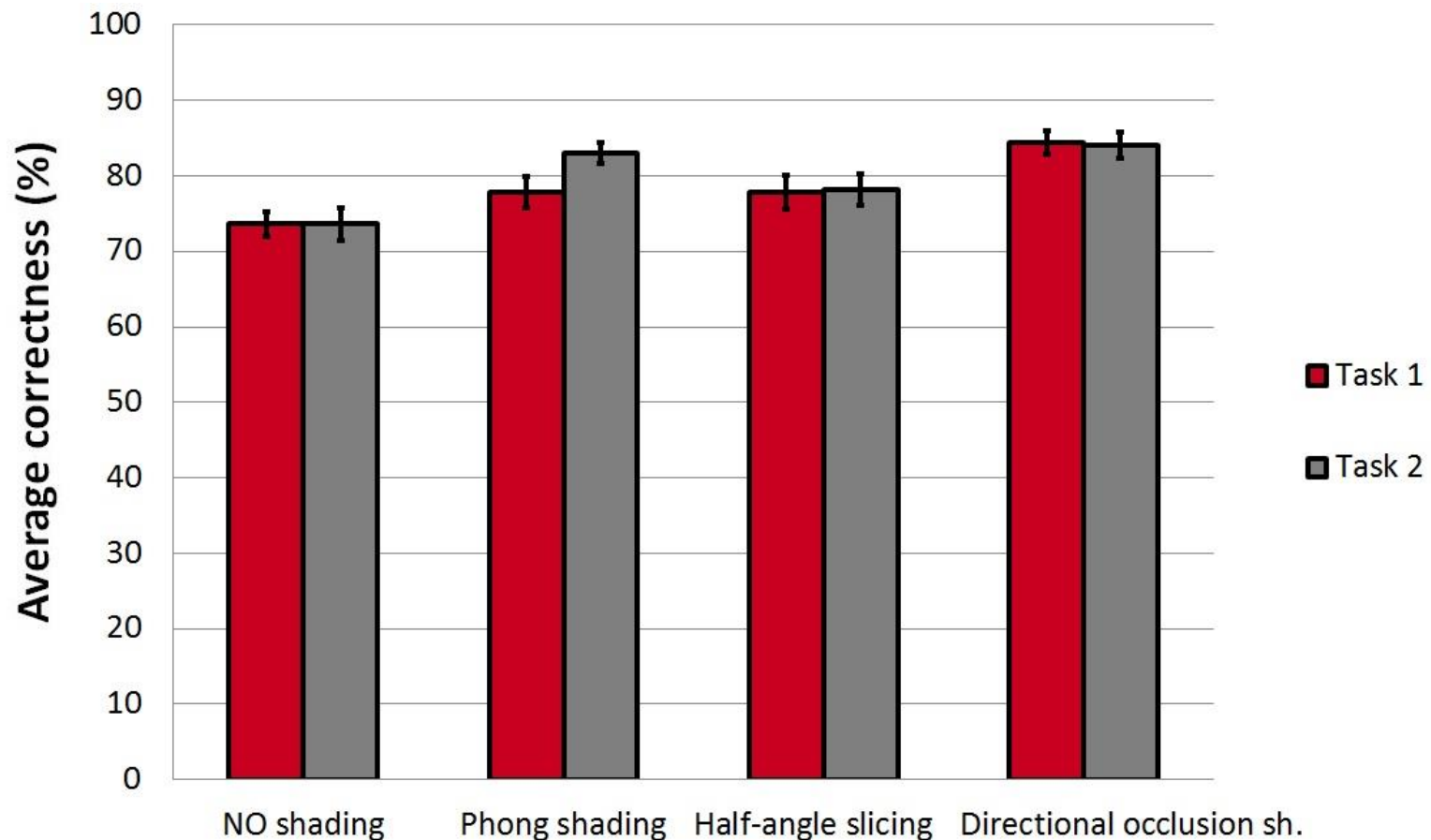
Use case. Depth perception in VR

- Analysis of results



Use case. Depth perception in VR

■ Analysis of results



Use case. Depth perception in VR

- Statistical analysis:
 - **ANOVA test:** One-way analysis of variance to reject the null hypothesis that all correctness means are equal between shading techniques.
 - For a significance level of $\alpha = 0.05$, a Bonferroni post-hoc test with the same acceptance level to reveal differences between the individual shading techniques
 - **Result:** reject the null hypothesis when $p < 0.05$

Use case. Depth perception in VR

- Statistical analysis.
 - Chi-square test of association for the categorical variables relative depth and users' answers from tasks 1 and 2

Variables	χ^2	<i>p</i> value	Correct answers for each depth category
T1: relative depth vs. users' answers	5.991	<0.0001	<0.05: 66 % 0.05–0.1: 88 % >0.1: 86 %
T2: relative depth vs. users' answers	5.991	<0.0001	<0.05: 63 % 0.05–0.1: 86 % >0.1: 87 %

Use case. Depth perception in VR

- Guidelines and recommendations
 - Using advanced volumetric shading improves depth perception
 - Among the tested shading models the simulation of soft shadows by using directional occlusion shading for desktop-based VR seem to yield better results

Use case. Depth perception in VR

- Guidelines and recommendations
 - Real illumination does not affect depth perception when using advanced volume illumination techniques
 - External lighting may be carefully controlled to provide a pleasant environment
 - Specular highlights on the screen, reflections, or over-illuminated areas will certainly affect the correct perception of the data

Use case. Depth perception in VR

- Guidelines and recommendations
 - When trying to judge depth in volume models, the X/Y relative position of the markers or the luminance of the points to classify seems to have no importance

Professors IDI

IDI – Usability Testing