Statistical Analysis of results of quantitative evaluation

The following hypotheses were tested:

1. System scores at least 80 in the SUS (Bangor et al. 2008)
2. Task completion rate of 90% for ordering food (Sauro, J. 2010) important for restaurant that tasks are carried out successfully
3. the average number of errors per task is not higher than 0,66 (Sauro, J. 2010, S. 67 f.)
4. Users remember at least 80% of the total features of the prototype after completing the tasks
5. The average number of clicks must not deviate from the best case by more than two clicks
6. The user is able to order a previously determined dish within 15 seconds, starting on the home screen until the order is placed.

For all hypotheses, a mean value comparison had to be conducted. For determining which test was best fitted for said comparison, a Kolmogorow-Smirnow test was performed. For hypothesis 2, a binomial test was conducted since the variable is binary (successful or not successful). Only for task 3, the number of errors, normal distribution could be assumed. Subsequently, for testing hypothesis 3, a one sample one sided t-test was conducted, for the rest of the hypotheses a one sample one sided Wilcoxon signed rank test. An overview of the results can be found in tables 1 and 2.

Table 1 *One sample Wilcoxon signed rank tests*

	N	Mean	Std. Deviation	Test value	z	p	r
SUS Score	10	82.25	10.17	80	-1.33	0.092*	-0.42
Remembered Features	10	11.5	4.03	16.8	2.76	0.997	0.87
Number of Clicks	10	8.53	7.06	6.36	-0.37	0.356	-0.11
Time to complete	10	50.6	22.63	15	2.85	0,998	0.90

* Significant p<0.1

Table 2 *One sample t-Test*

	N	Mean	Std. Deviation	Test value	t(7)	p	d
Number of Errors	10	0.49	0.38	0.66	-1.298	0.117	0.459

Hypothesis 1 was the only hypothesis that was significant (z = -1.33, p = 0.092, r = -0.42). The effect size, according to Cohen (1988), can be interpreted as medium to high. The boxplot of the SUS scores can be found in figure 1. According to Bangor (2008), this indicated that the

overall app's usability can be considered at least "good". We can conclude that the overall concept is not bad, but there is some room for improvement to achieve "excellent" scores.

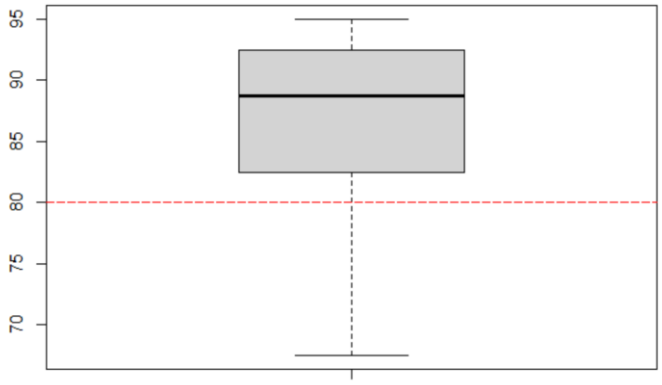

Figure 1: Boxplot SUS Scores

The null hypothesis of hypothesis 3, that the subject's values don't differ from 0.66 errors, could be rejected ($t(7)= -1.298$, $p = 0.117$, $d = 0.459$), though it was rather close to being significant on a value of $p>0.1$. Descriptively, the boxplot in figure 2 shows a clear tendency of the errors being lower than the desired value. Perhaps the results would be more clear with a higher sample size.

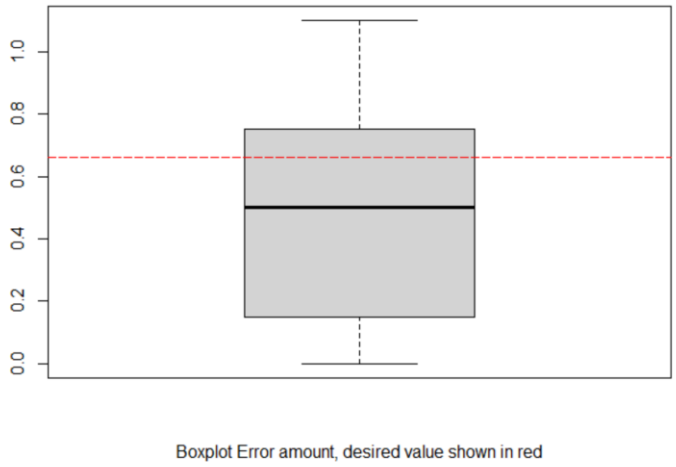

Figure 2: Boxplot of errors

For all other hypotheses, no significant effect could be found. Although all participants were able to successfully finish the task of ordering food for hypothesis 2, the result was not significant (exact binomial test, one sided, $p = .3487$, $n = 10$). This can probably be ascribed to the small sample size. Since all participants were able to successfully finish the task, we still see the hypothesis as fulfilled, at least descriptively. Still, usability heuristics like that of "transparency system status" should be reviewed to problems arising in critical situations.

For hypothesis 4 specifically, the developers of the app might have specified the features in too much detail. Furthermore, not all features were tested, and it would have been hard for the users to remember features that they never used. We could assume this value to be a measure of the UI's memorability in the sense of Nielsen's five quality components of usability, and would therefore have to attest a low degree of memorability to the app. As the measurement seems to not be useful, drawing a conclusion upon the results of this conclusion doesn't seem useful.

The following rejected hypothesis that the overall number of clicks might not deviate by two from the ideal case ($z = -0{,}37$, $p = .0356$, $r = -0.11$) is a self-set goal to understand if users understand the app well enough to use optimal ways of usage. Shortcuts could thereby be seen as a part of Nielsen's heuristic of "user control and freedom". More experienced users might be more successful in completing the tasks with fewer clicks, a test with users who are familiar with the app would be necessary to test that.

Literature

Bangor, Aaron; Kortum, Philip T.; Miller, James T. (2008): An Empirical Evaluation of the System Usability Scale. In: *International Journal of Human-Computer Interaction* 24 (6), S. 574–594. DOI: 10.1080/10447310802205776.

Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences. Hoboken: Taylor and Francis.

Sauro, J. (2010). A practical guide to measuring usability. https://measuringu.com/wp-content/uploads/2017/05/quantitativeusabilitytestonline.pdf

Appendix

Guideline Quantitative Evaluation

Agenda

1. Set Up and what to measure
2. Introduction
3. Demographic Information
4. Scenario
5. Test Cases
6. SUS
7. Follow Up Questions

Setup and what to measure

The quantitative evaluation is held in **Zoom** (because of Covid and also because of the recording). During the whole session the test person has to **share their screen**. The prototype is made available to the test person via the following link: https://interactionprototyping.github.io/WS2022_WeDeserveArrays/finalcode/

The subject must open the link in **Chrome** and also **open the console** so that the test supervisor can see and read it. The subject should adjust the screen to the size of the **Nexus 6** phone.

Before starting the test, the test administrator must ensure that they can start a **screen recording**. In doing so, the subject's **image and sound** must be recorded! If only the screen can be recorded, the sound must be recorded separately.

Make sure you have the **SUS**-questionnaire at hand for later. (It is in the Drive Folder Iteration 4)

The Measurements:

- Time user takes to complete the task
- Number of clicks (total) the user takes to complete the task
- Number of errors for the task (please note down the errors)
- Number of users completing task successfully (without assistance)
- Number of system features the user can remember after the test (will be measured in the end)

Introduction

Briefly explain the process to the subject: The test person is recorded from now until the end of the evaluation. First they are asked some questions about themselves, then they have to solve some tasks with the prototype and afterwards they have to fill in a questionnaire.

Try not to give too much information on the app or the project yet, provided they have not had any contact with the project so far.

Demographic Information

1. Age
2. Gender
3. Occupation
4. Have you been involved earlier on in this project? (tested a previous prototype etc.)

Scenario

Imagine you are in a restaurant with your friends that uses a new app called Me'n'U. You have been on a special diet for a while. The app has some new features that you want to try out with your friends today. **You go through the Onboarding-Screens and select that you are allergic to soy and then that you are a pescetarian.** When you sit down at your table, there is a QR-Code on the table top.

Test Cases

The test leader reads the tasks **aloud** to the respondent, several times if necessary, so that the respondent has understood all the points **before starting the task**.

Try not to help with the tasks if possible. The test person should try to solve everything themselves without assistance and as fast as they can, but they should not feel rushed. When the task has been solved, the test person may be interrupted if they are unsure.

1. **Order** - Add a vegan pizza to your order and place your order. Do not pay right away. (Start: Home; Finish: Home)
    1. criteria to determine if the task was successfully solved:
        1. Pizza Regina Vegana is added to order
        2. order is placed
        3. Pay later is chosen

Favourites - Add one of the dishes in Antipasti to your favourites. Then, go to your favourites list. (Start: Home; Finish: Your Favourites)

a. criteria to determine if the task was successfully solved:
    1. User shows all the dishes
    2. One of the antipasti is added to favourites list
    3. Favourites screen is opened

Call a Waiter - Call a waiter to your table using the app. (Start: Your Favourites; Finish: Home)

a. criteria to determine if the task was successfully solved:
    1. Call a waiter button is pressed

Preferences - Edit your dietary needs you have set previously. Change your allergy from soy to peanuts. Then, go back to the Home screen. (Start: Home; Finish: Home)

a. criteria to determine if the task was successfully solved:
    1. Allergy is changed from “soy” to “peanut”

Past Orders - (There should be something that was already ordered) Look at the dishes you have already ordered, but not paid yet. (Start: Home; Finish: Already ordered drop-down is opened)

a. criteria to determine if the task was successfully solved:
    1. “Already ordered” drop-down is opened

Pay - (There should be something that was already ordered) Pay for your dishes. (Start: Order; Finish: Home)

a. criteria to determine if the task was successfully solved:
    1. The dishes are payed for

Check Out - You are done. Check out from your table. Do not give Feedback. (Start: Home; Finish: Scan QR)

a. criteria to determine if the task was successfully solved:

1. “No, thanks” is chosen

Going Back - You are back at the restaurant and you decide to order a glass of vine. After ordering that you go for one salad together with a pizza of your choice. After that they pay how they want to and can leave. (Start: Scan QR; Finish: Scan QR)

a. criteria to determine if the task was successfully solved:

1. Wine added to order
2. Wine order placed
3. Pizza and salad added to order
4. Pizza and salad order placed
5. Pay
6. Check Out

SUS

After all tasks have been solved, the respondent may now **stop sharing their screen**. The **screen recording is not stopped**. To answer the questionnaire, the test administrator now shares the screen so that the respondent can see the SUS questionnaire.

Explain the SUS: The respondent should not think too long about the questions and go by their initial gut feeling. However, there is no time limit, so read the tasks carefully and give an appropriate score. The test supervisor does not interfere. He only writes down the answers (or writes them out later from the recording).

→ **Now do the SUS: https://forms.gle/s3m3kbvq1dgWcfMx6**

Follow Up Question

1. Try to list the features of our app. (Maybe also including features that you did not test during the tasks)

That’s it.
Thank you for participating!

R Code

```
#install.packages("stargazer")
#install.packages("xlsx")


library(xlsx)
library(stargazer)

#### set workspace and read data
setwd("insert path to ‘Results_quantitativeEvaluation’ here")

demographics <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 1)
time <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 2)
errors <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 3)
clicks <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 4)
success <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 5)
sus <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 6)
features <- read.xlsx("Results_quantitativeEvaluation.xlsx", sheetIndex = 7)
```

```
age_mean <- demographics$Mean[1]
gender <- demographics[2, 2:11]

#### SUS analysis ####
sus_scores <- as.numeric(sus$NA..11[3:12])
SUS_mean <- mean(sus_scores)
SUS_sd <- sd(sus_scores)
ks.test(sus_scores, "pnorm", 1, 2)
wilcox.test(sus_scores, mu = 80, alternative = "greater", exact = FALSE)
boxplot(sus_scores, xlab="Boxplot SUS Scores, desired value shown in red",
show.names=TRUE)
abline(h = 80, col="Red", lt=5)
z <- qnorm(0.09171)
r <- z/sqrt(10)

#### TASK COMPLETION RATE ####
task1_success <- success[1, 2:11]
binom.test(10,10, 0.9, alternative="greater")


#### ERRORS ####
errorAmount <- as.numeric(errors$Mean[1:8])
errorAmount_mean <- mean(errorAmount)
errorAmount_sd <- sd(errorAmount)
ks.test(errorAmount, "pnorm", 1, 2)
t.test(errorAmount, mu = 0.66, alternative="less")
cohensD(errorAmount, mu = 0.66)
boxplot(errorAmount, xlab="Boxplot Error amount, desired value shown in red")
abline(h = 0.66, col="Red", lt=5)

#### REMEMBERED FEATURES ####
rememberedFeatures <- as.numeric(features[1, 2:11])
rememberedFeatures_mean <- mean(rememberedFeatures)
rememberedFeatures_sd <- sd(rememberedFeatures)
ks.test(rememberedFeatures, "pnorm", 1, 2)
wilcox.test(rememberedFeatures, mu = 16.8, alternative = "greater", exact = FALSE)
z <- qnorm(0.9971)
r <- z/sqrt(10)


#### NUMBER OF CLICKS ####
bestCase <- as.numeric(clicks$Best.Case[1:8])
bestCase_mean <- mean(bestCase)
clicksAmount <- as.numeric(clicks$Mean[1:8])
clicksAmount_mean <- mean(clicksAmount)
clicksAmount_sd <- sd(clicksAmount)
ks.test(clicksAmount, "pnorm", 1, 2)
wilcox.test(clicksAmount, bestCase, mu = 2, alternative="less", exact = FALSE)
z <- qnorm(0.3563)
r <- z/sqrt(10)
```

```
#### TIME TO COMPLETE ORDER ####
timeToComplete_Task1 <- as.numeric(time[1, 2:11])
timeToComplete_Task1_mean <- mean(timeToComplete_Task1)
timeToComplete_Task1_sd <- sd(timeToComplete_Task1)
ks.test(timeToComplete_Task1, "pnorm", 1, 2)
wilcox.test(timeToComplete_Task1, mu = 15, alternative = "less", exact = FALSE)
z <- qnorm(0.9978)
r <- z/sqrt(10)
```