

# Testprotokoll

## Benutzerfreundlichkeit des Dbuggers

### Anweisung an die Testleitung:

- Sorge für eine ruhige Umgebung und ausreichend Zeit.
- Stelle sicher, dass dein Desktop die Datei „nutzertest.rdbf“ enthält.
- Stelle sicher, dass das Programm geöffnet ist, eine Möglichkeit zum Zeitstoppen existiert, der Computer genug Strom hat, etc. Die Versuchsperson soll das Programm allerdings noch nicht sehen.
- Lies den Testpersonen die Testanweisung vor, auch wenn es dir komisch vorkommt. Dies dient der Standardisierung.
- Frage vor Beginn ab, ob eine Audio-Aufnahme in Ordnung ist.

### Anweisung an die Testleitung:

- Folgendes Vorlesen:

Liebe Teilnehmerin, lieber Teilnehmer!

Vielen Dank, dass du an unserer Nutzerstudie teilnimmst. Im Rahmen unseres PSE-Projektes haben wir ein Programm entwickelt, dessen Nutzerfreundlichkeit und Erlernbarkeit wir nun erproben wollen. Deine Teilnahme hilft uns sehr!

Unser Programm erfüllt die normalen Funktionen eines Debuggers, wie das Setzen von Breakpoints. Zusätzlich können mehrere Programme gleichzeitig gedebuggt und verglichen werden.

Der Ablauf wird folgendermaßen sein:

Zunächst wirst du einige standardisierte Anwendungsfälle ausführen. Diese werden dir jeweils vorgelesen werden. Bitte versuche, dabei „laut zu denken“. Zum Beispiel: „Ich frage mich, wie ich X tun kann.“ Oder: „Ich suche gerade eine Funktion, um Y zu tun.“ Das ist am Anfang ungewohnt, aber deine Testleitung wird dich ggf. mit Fragen dabei unterstützen.

Das hilft uns, Schwachstellen und Schwierigkeiten des Programmes zu entdecken.

Versuche zunächst, die Aufgabe im Programm ohne Hilfe zu lösen.

Solltest du danach nicht weiterkommen, ziehe bitte das Nutzermanual zu Rate. Auch hier gilt, dass uns dein lautes Denken weiterbringt. Du könntest zum Beispiel sagen: „Ich suche einen Eintrag zum Thema Z.“

Solltest du auch hier nicht weiterkommen, kannst du deine Testleitung fragen.

Im Anschluss an den Praxisteil haben wir noch ein paar Fragen an dich.

Wenn du dich unwohl fühlen solltest, ist es jederzeit möglich, das Experiment zu unterbrechen.

Hast du noch Fragen?

### Anweisung an die Testleitung:

- Starte (bei Einverständnis) nun die Audioaufnahme.
- Wenn du Antworten auf Fragen gibst, sollten diese ebenfalls auf der Aufnahme zu hören sein.
- Falls die Testperson das „laute Denken“ vergisst, kannst du sie mit Nachfragen dazu anhalten, zum Beispiel:
  - „Was denkst du gerade?“
  - „Bist du auf eine Schwierigkeit gestoßen?“
  - „Worüber denkst du nach?“

Anweisung an die Testleitung:

- Lies jeweils die Aufgabenstellung vor und lasse sie von der Versuchsperson bearbeiten.
- Stoppe bei jeder Aufgabe die Zeit und dokumentiere sie in der angefügten Tabelle.
- Die Überschrift soll mit vorgelesen werden.

Aufgaben:

1. Lass dir das Nutzermanual anzeigen.
2. Ändere die Sprache zu einer Sprache deiner Wahl.
3. Füge ein neues Programmfenster hinzu.
4. Lösche ein Programmfenster.
5. Einfache Addition
  - a. Schreibe in einem der Textfenster ein Programm zur Addition zweier Zahlen. Das Programm soll zwei Zahlen als Input nehmen und die Summe wieder ausgeben.
  - b. Lass dir vom Programm die Summe berechnen.
  - c. Lass dir in einer beliebigen Zeile den Wert aller Variablen deines Programmes anzeigen.
6. Addition mit zwei Funktionen
  - a. Verändere dein Programm nun so, dass eine Funktion die zwei zu addierenden Zahlen empfängt, diese aber an eine andere Funktion zur Addition weitergibt.
  - b. Speichere deine Programme.
7. Vergleich von zwei vorgelieferten Programmen.

Hinweis: Es gibt verschiedene Möglichkeiten, die folgenden Aufgabenteile zu lösen.

  - a. Lade vom Desktop die Datei „nutzertest.rdbf“
  - b. Finde die Stelle, an der die Variablen b aus Programm A und d aus Programm B denselben Wert haben. ((Alternativ: An der Variable b aus A den Wert 16 hat.))
  - c. Lass dir zu jedem Zeitpunkt die Differenz von Variable c aus Programm A und Variable i aus Programm B anzeigen.
  - d. Sorge dafür, dass das Programm anhält, sobald Variable b aus Programm A und Variable d aus Programm B denselben Wert haben.
8. Lösche eine Watch-Expression.

# Allgemeine Informationen

1. Alter: \_\_\_\_\_

2. Geschlecht:     ☐ weiblich                      ☐ männlich

3. Beruf: \_\_\_\_\_

4. Was ist Dein Bezug zur Informatik?

---

5. Was ist Deine Erfahrung im Umgang mit Programmen zum Debuggen von Software?

---

6. Wie intuitiv war die Nutzung des Dbuggers für dich?

(1 = nicht intuitiv, 5 = intuitiv)

(nicht intuitiv)	1	2	3	4	5 (intuitiv)
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

7. Wie nutzerfreundlich hast du den Dbugger empfunden?

(1 = nicht nutzerfreundlich, 5 = nutzerfreundlich)

(nicht nutzerfreundl.)	1	2	3	4	5 (nutzerfreundl.)
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. Wie hilfreich fanden Sie das Nutzermanual?

(1 = nicht hilfreich, 5 = hilfreich)

(nicht hilfreich)	1	2	3	4	5 (hilfreich)
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9. Wie übersichtlich fanden Sie die Oberfläche des Programms?

(1 = nicht hilfreich, 5 = hilfreich)

(nicht hilfreich)	1	2	3	4	5 (hilfreich)
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

10. An welcher Stelle hattest du bei der Nutzung des Programms besondere Schwierigkeiten?

---

---

**11. Was hätte dir den Einstieg in das Programm erleichtert?**

---

---

**12. An welcher Stelle hattest du bei der Nutzung des Manuals besondere Schwierigkeiten?**

---

---

**13. Was hätte dir die Nutzung des Manuals erleichtert?**

---

---

**14. Was hätte die Oberfläche des Programmes in deinen Augen übersichtlicher gemacht?**

---

---

**15. Gibt es eine Funktion oder einen Hinweis, den du vermisst hast?**

---

---

**16. Hier ist Platz für weitere Anmerkungen und Verbesserungsvorschläge:**

---

---

Vielen Dank für Deine Teilnahme! Du hast uns damit sehr geholfen!

**Auswertung:**

Aufgabe	Zeit	Absturz-Fehler	Zeitverlust-Fehler	Ärgernisse

**In Programm A:**

```
int main (int n) {
    int a = 1;
    int b = 0;
    int c = 34723;
    while (a < 2*n) {
        b = b + a;
        a = a + 2;
        c = c*a-b;
    }
    return b;
}
```

**In Programm B:**

```
int foo(int k) {
    int i = 0;
    int d = 0;
    while(i < k) {
        i = i + 1;
        d = d + k;
    }
    return d;
}
int main (int k) {

    int a = 0;
    a = foo(k);
    return a;
}
```