



Qualitätssicherung von Software

Prof. Dr. Holger Schlingloff

Humboldt-Universität zu Berlin
und
Fraunhofer FIRST

Hinweis zur Exkursion

- Wir treffen uns am Freitag **pünktlich um 11:00 Uhr** beim Empfang (3. Stock) der Daimler-Chrysler Forschung (Abt. REI/SM), Alt-Moabit 96A, 10559 Berlin
- Wer zu spät kommt, den bestraft das Leben!
- Anmeldeliste in dieser Vorlesung!

Kapitel 2. Testverfahren

2.1 Testen im SW-Lebenszyklus

2.2 funktionsorientierter Test

- Modul- oder Komponententest
- Integrations- und Systemtests

2.3 strukturelle Tests, Überdeckungsmaße

2.4 Test spezieller Systemklassen

- Test objektorientierter Software
 - 2.5 automatische Testfallgenerierung
- Test graphischer Oberflächen
- Test eingebetteter Realzeitsysteme

2.6 Testmanagement und –administration

GUI-Testing

- *Capture-Replay-Technik*
- bekannteste Vertreter: Mercury WinRunner bzw. Astra, Rational Robot bzw. XDE Functional Tester
- HU-Tool „ATOS“ (Prof. Bothe)
- **Idee**
 - Aufzeichnung von Benutzerinteraktionen
 - Abspielen mit Vergleich auf Änderungen
- **Erweiterungen**
 - GUI-Map, Scripting Wizard
 - Alternativen, Wiederholungen
 - Checkpoints, Komparatoren



Vorgehensweisen GUI-Test

- **pragmatisch**
 - Aufzeichnung von Experimenten, Replay bei neuen Produktversionen
- **systematisch**
 - Festlegung von Testzielen (Qualitätskriterien, Requirements)
 - Definition von Testfällen und Testsuiten
 - Umsetzung (Implementierung) der Testfälle
 - automatische Durchführung, Ergebnisabgleich und Korrektur
- **Vor- und Nachteile** Testautomatisierung von Oberflächen
 - minimaler Aufwand für Regressionstests
 - systematische Überdeckung der Anforderungen
 - erhöhter Anfangsaufwand, Toolkosten

Aufzeichnung GUI-Test

- Umgebung aufsetzen, Aufzeichnungsoptionen setzen
- Aufzeichnung starten
- Wiederholung (experimentell oder nach Plan):
 - Testfälle durchführen
 - Beobachtungspunkte einfügen, Testverdikte festlegen
 - Kontrollpunkte setzen
- Aufzeichnung beenden
- entstandene Skripte editieren
 - Fallunterscheidungen, Verzweigungen
 - Wiederholungen, Datenbereiche

Wiedergabe GUI-Test

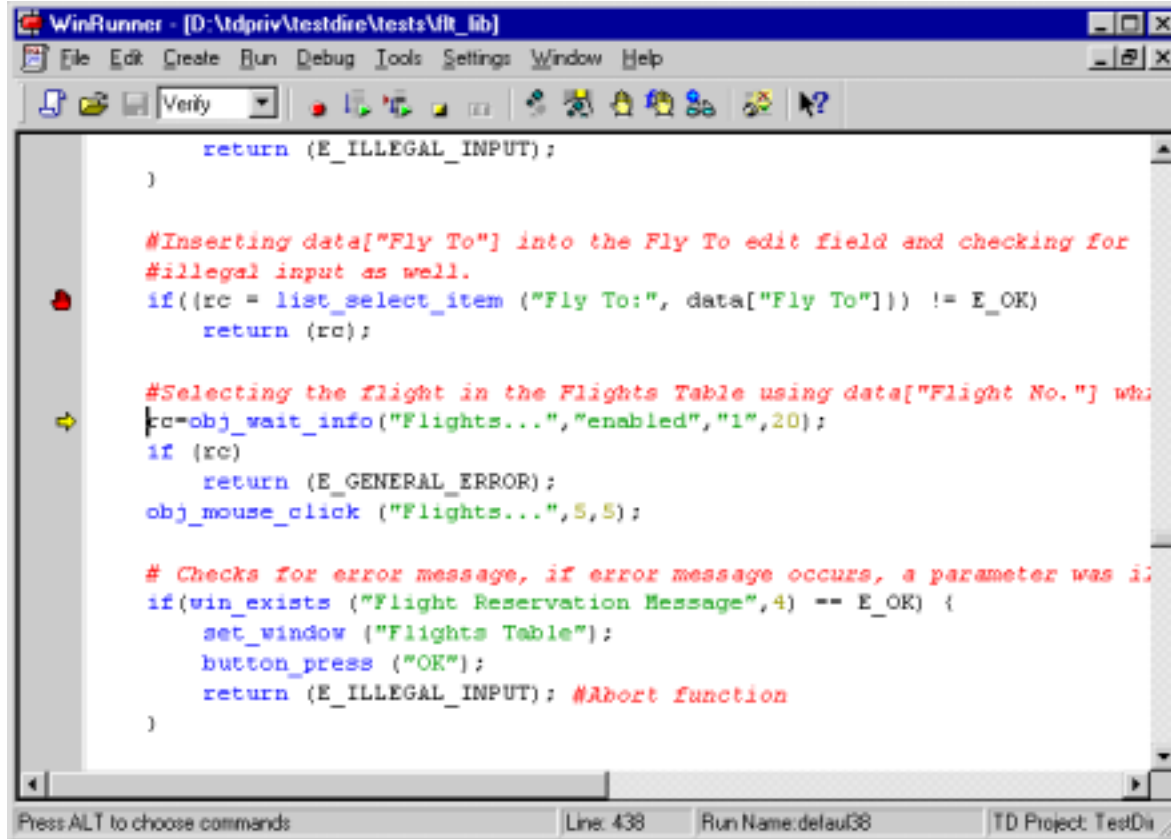
- Testskripte zu Testsuite zusammenstellen
- Umgebung aufsetzen, Wiedergabeoptionen setzen
- Wiedergabe mit Aufzeichnung der Ergebnisse
 - ggf. sofortige Korrektur von Skriptfehlern
- Auswertung der Ergebnisse
 - Für jeden nicht durchgelaufenen Testfall
 - manuelle Anpassung der Testskripte und erneute Durchführung oder
 - Eintrag von Items in Fehlerverfolgungssystem
 - Erzeugung eines Testreports (manuell oder automatisch)

WinRunner von Mercury Interactive



Screenshots thanks to Mario Friske, FIRST

WinRunner Test Script Language (TSL)



```
WinRunner - [D:\tdpriv\testdire\tests\flt_lib]
File Edit Create Run Debug Tools Settings Window Help
Verify
return (E_ILLEGAL_INPUT);
)

#Inserting data["Fly To"] into the Fly To edit field and checking for
#illegal input as well.
if((rc = list_select_item ("Fly To:", data["Fly To"])) != E_OK)
    return (rc);

#Selecting the flight in the Flights Table using data["Flight No."] wh
rc=obj_wait_info("Flights...", "enabled", "1", 20);
if (rc)
    return (E_GENERAL_ERROR);
obj_mouse_click ("Flights...", 5, 5);

# Checks for error message, if error message occurs, a parameter was in
if(win_exists ("Flight Reservation Message", 4) == E_OK) {
    set_window ("Flights Table");
    button_press ("OK");
    return (E_ILLEGAL_INPUT); #Abort function
}
```

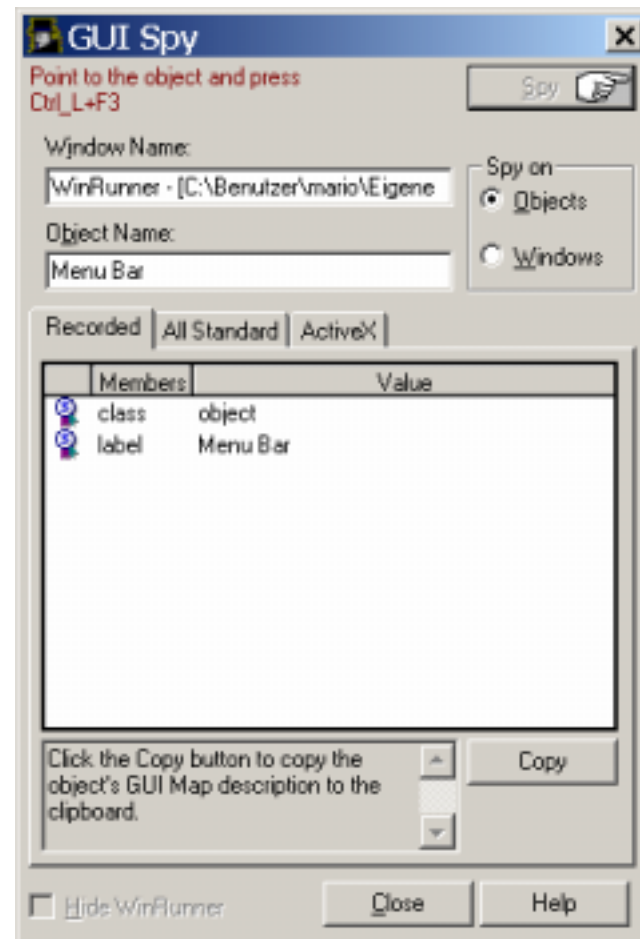
Press ALT to choose commands | Line: 438 | Run Name: default38 | TD Project: TestDir

- Verzweigungen
- Wiederholungen
- Makroaufrufe
- Dateneingaben
- Parallelität
- ...

Problem: Zuordnung von GUI-Elementen zu Skriptfunktionen („GUI-Mapping“)

WinRunner-Beispiel

Schrittweises
Erstellen der GUI-
Map:
Identifizieren
einzelner GUI-Objekte
mit dem GUI-Spy

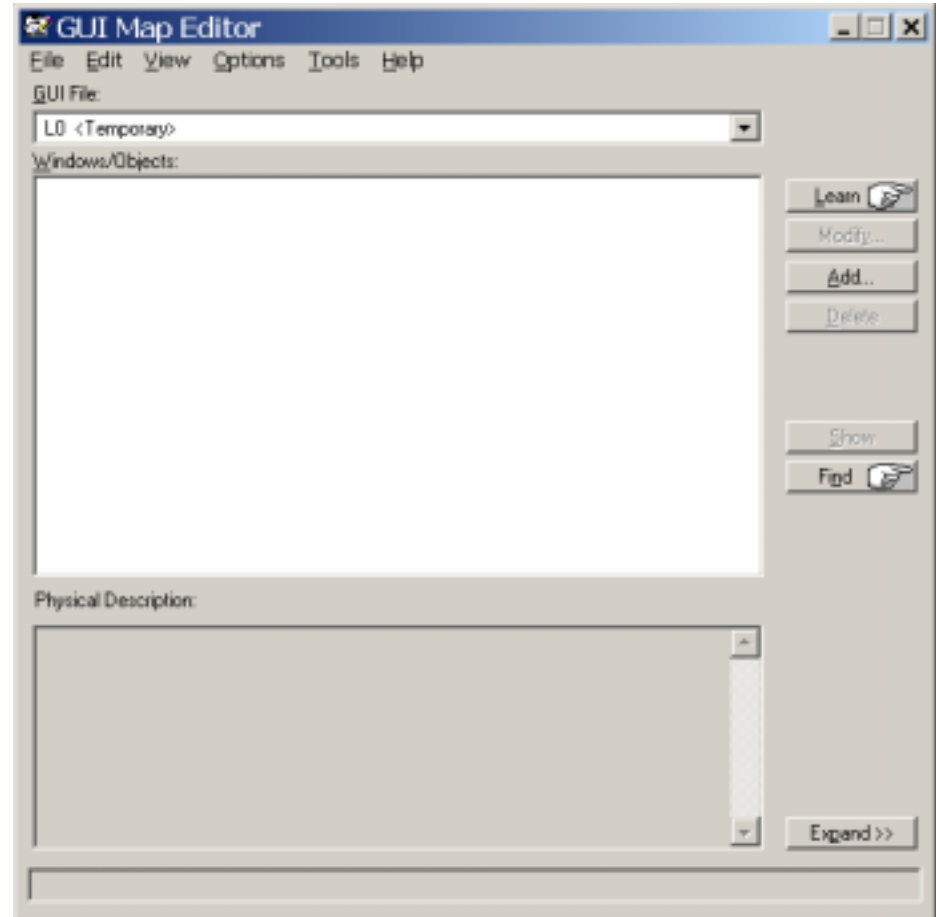


Verwaltung von GUI-Maps

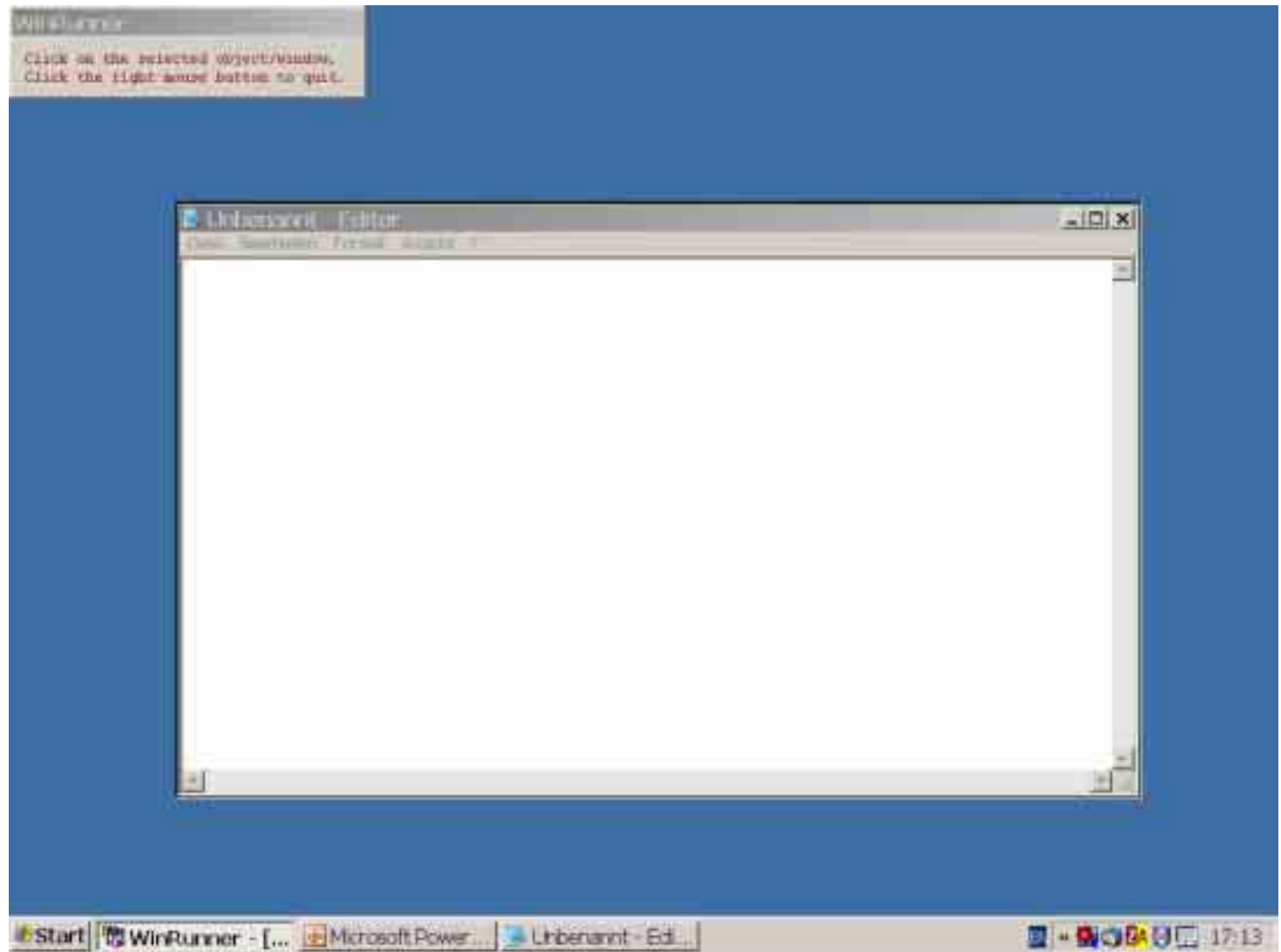
Erstellung GUI-Map:

Initial sind keine
Infos über GUI-
Elemente
vorhanden.

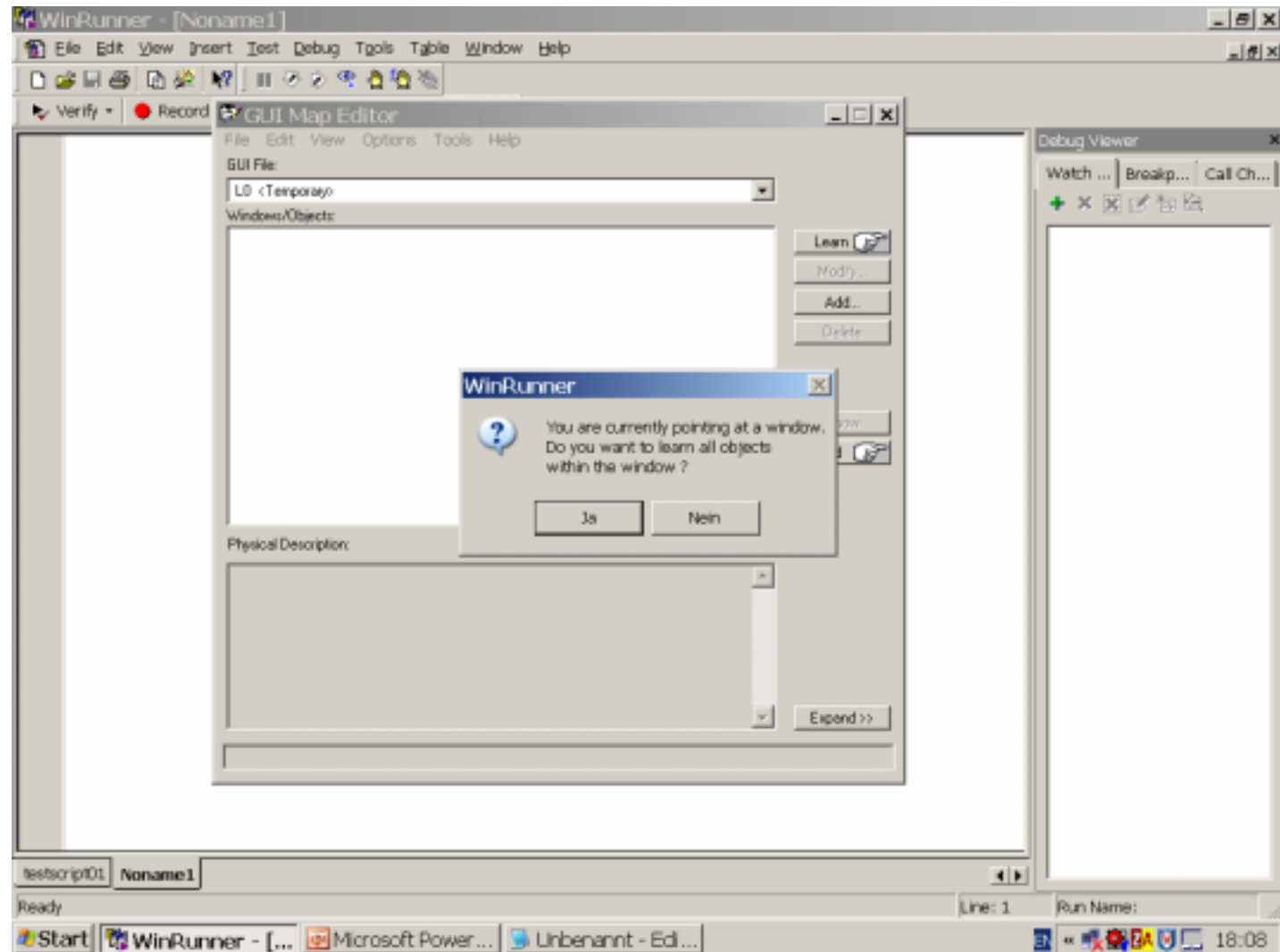
Große Applikationen
können hunderte
GUI-Elemente
enthalten.



Anwen- dung der Learn- Funktion auf ein Fenster...

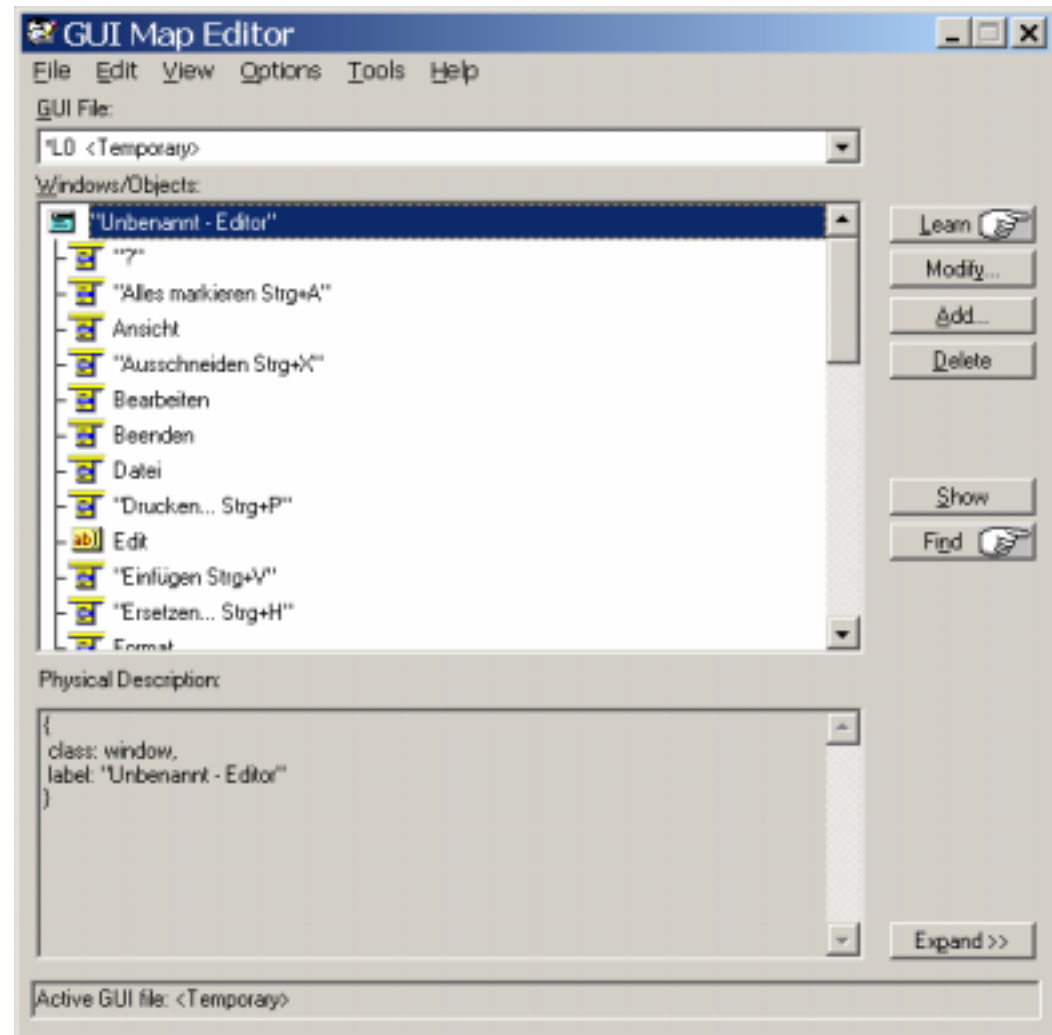


..führt zum
Scannen
der Gui...



Die erkannten
GUI-Elemente
sind dann in den
Testskripten
benutzbar.

Eingriff in
Betriebssystem
oder Middleware!

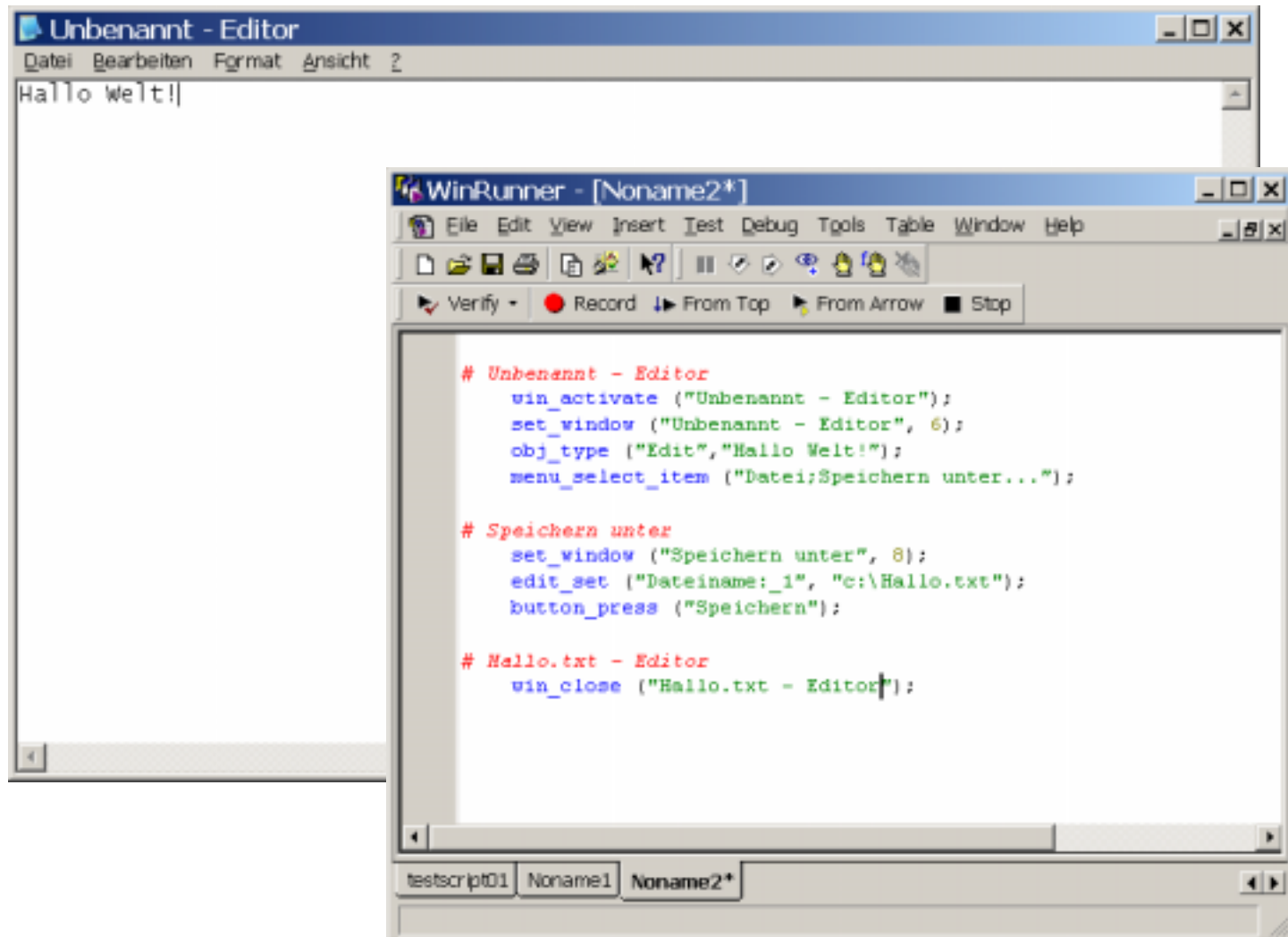


Aufzeichnung der Testskripte

- Es gibt zwei Aufnahmemodi:
 - analog (Pixelgenaues Ansteuerung)
 - kontextsensitiv (Arbeiten mit identifizierten Objekten, geht leider nicht immer)
- Beide Modi lassen sich kombinieren
- Bei widerspenstigen Applikationen simuliert man die Tastaturbedienung
(sollte jede ordentliche Applikation können, ist jedoch leider nicht immer der Fall)

Ein typischer einfacher Testfall ...

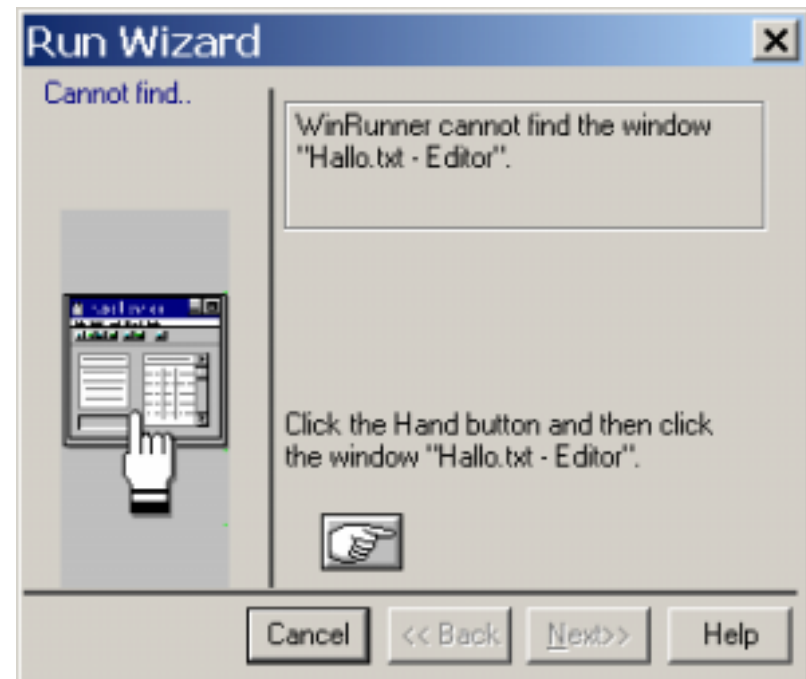
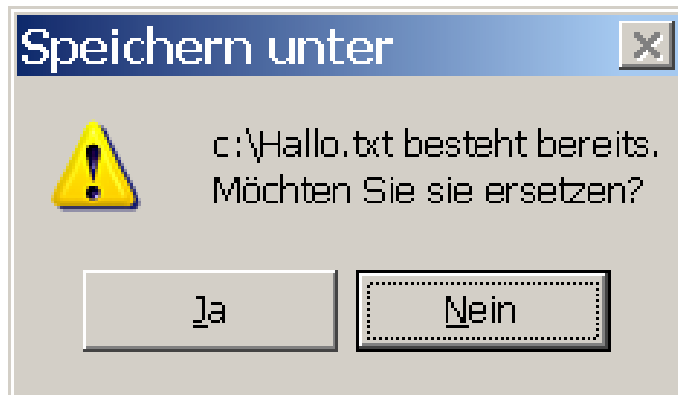
(„Hallo Welt!“ im MS Notepad schreiben und speichern)



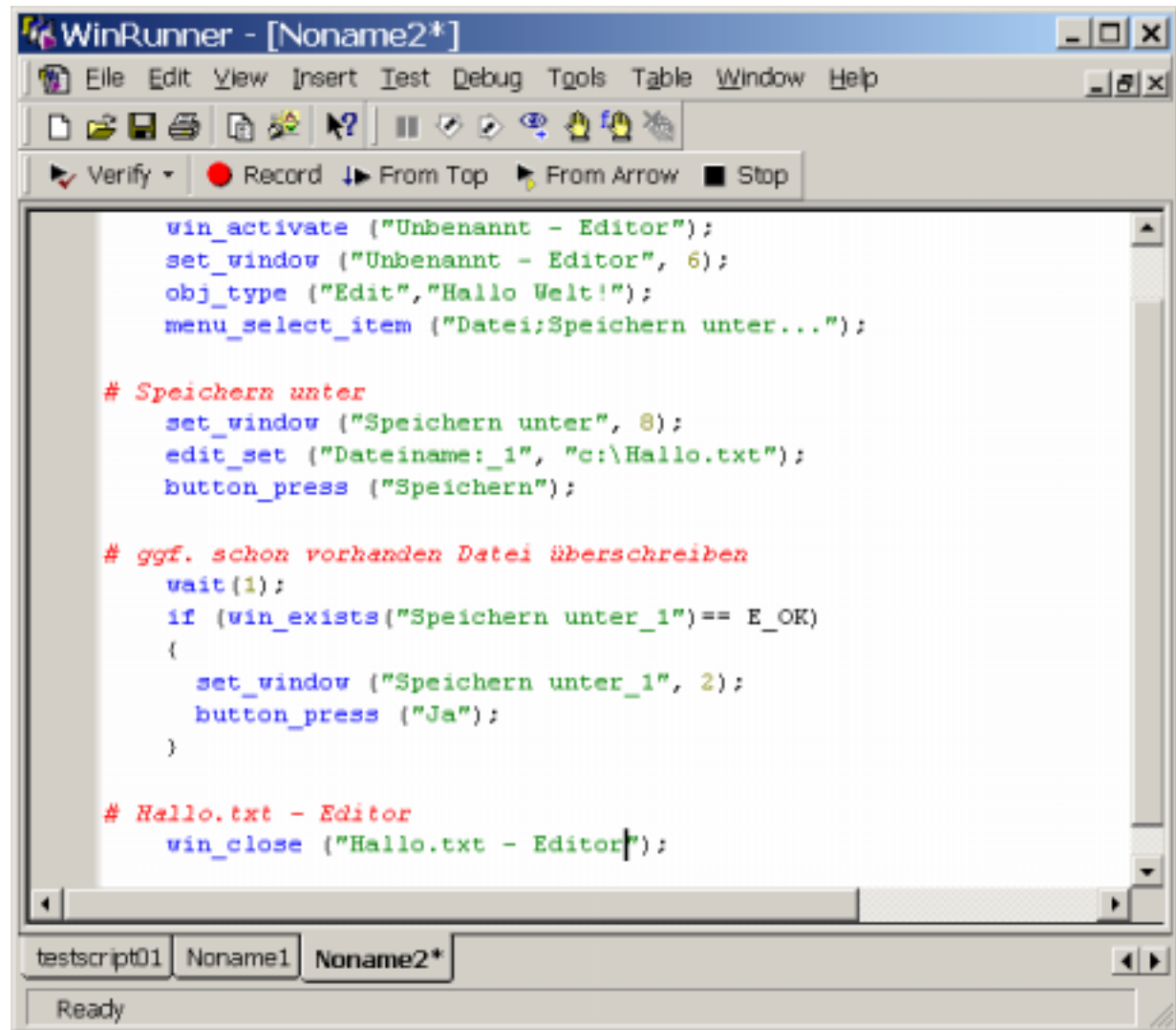
Die automatische Durchführung...

(lässt sich natürlich schwer mit einem Screenshot darstellen)

...klappt auch nicht, denn wir vergaßen...



Deswegen noch
schnell eine
Fallunter-
scheidung
eingebaut...



```

WinRunner - [Noname2*]
File Edit View Insert Test Debug Tools Table Window Help
[Icons]
Verify [Record] [From Top] [From Arrow] [Stop]

win_activate ("Unbenannt - Editor");
set_window ("Unbenannt - Editor", 6);
obj_type ("Edit", "Hallo Welt!");
menu_select_item ("Datei;Speichern unter...");

# Speichern unter
set_window ("Speichern unter", 8);
edit_set ("Dateiname: 1", "c:\\Hallo.txt");
button_press ("Speichern");

# ggf. schon vorhandene Datei überschreiben
wait(1);
if (win_exists("Speichern unter_1") == E_OK)
{
    set_window ("Speichern unter_1", 2);
    button_press ("Ja");
}

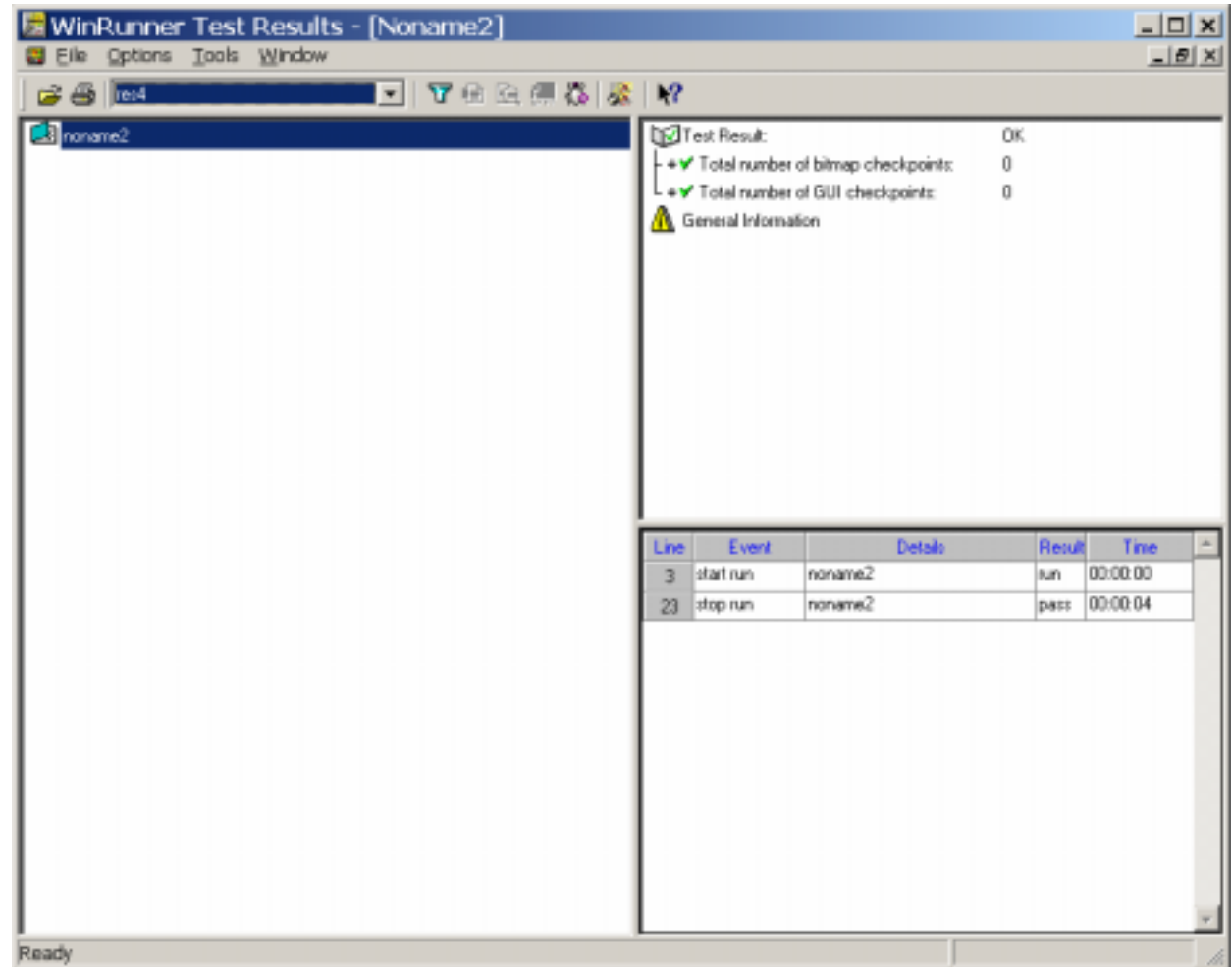
# Hallo.txt - Editor
win_close ("Hallo.txt - Editor");
  
```

testscript01 Noname1 Noname2* Ready

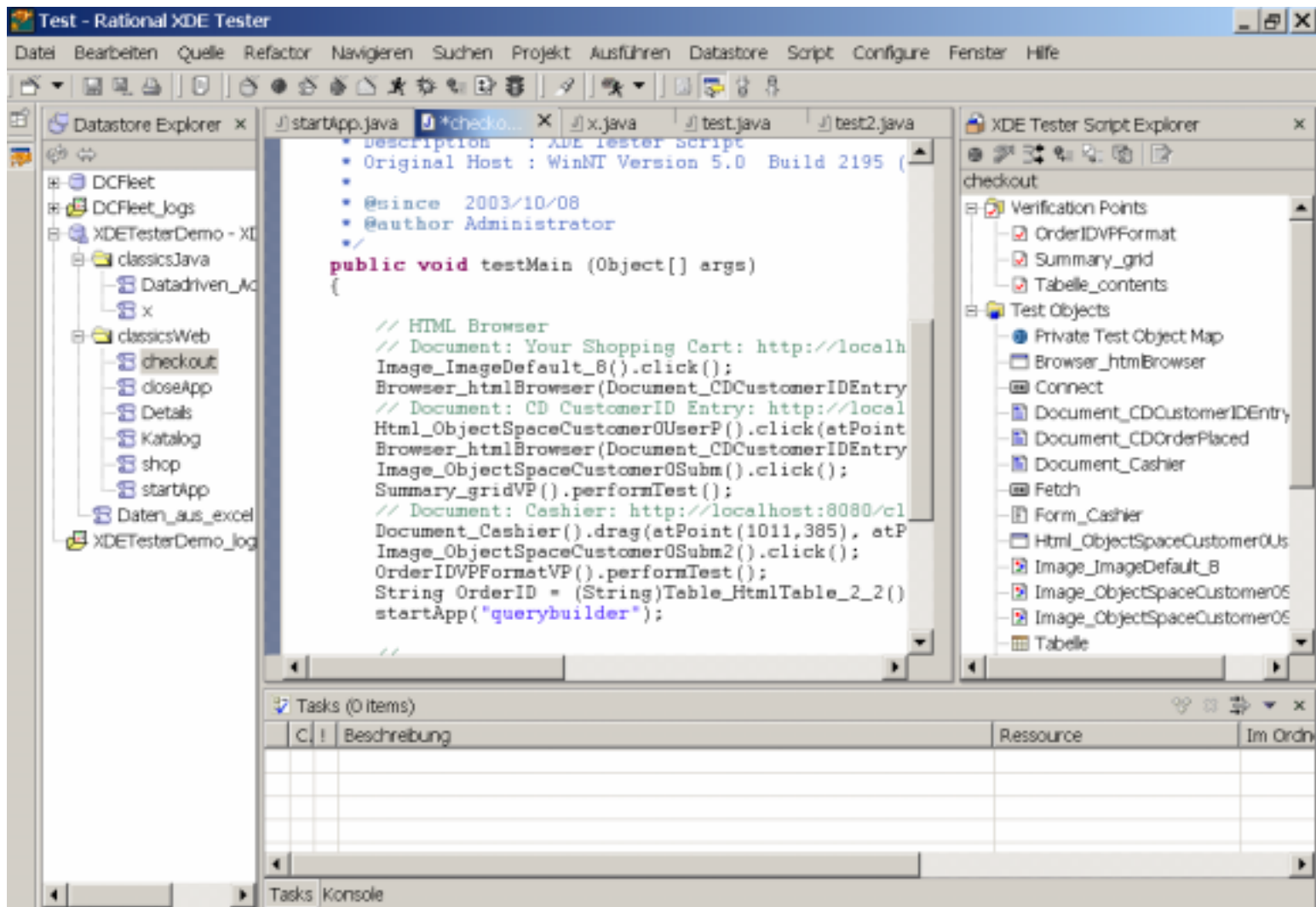
Dann läuft der Testfall durch (ohne Checkpoints, denn die sind ja noch nicht implementiert)



Programmier-Arbeit!



Oberfläche XDE Tester



Wartung von Testskripten

- Problem mit dem GUI-Map-Ansatz



A screenshot of a web form titled 'Version 1'. It contains two input fields: 'Account # / Log In ID' and 'Password'. Below these fields is a 'Log In' button. At the bottom, there is a link 'Click here to save your start page!' followed by a checkbox.

Version 1



A screenshot of a web form titled 'Customer Log On' with a blue background. It contains two input fields: 'User Name:' and 'Password:'. Below these fields is a 'Log On' button.

Version 2

- Anpassung von Testskripten
 - manuell (unter Umständen viel Aufwand!)
 - automatisch, fuzzy matches (Forschungsbedarf)

GUI-Testorakel

- **Verdikte:** *pass*, *fail*, inconclusive, undefined, error
- Auswertung dynamisch oder nach dem Durchlauf
- Vergleich von
 - Ein- und Ausgabeströmen
 - Dateien (Text- oder Grafikdateien)
 - Bildschirmdaten (Screenshots bzw. Texten)
- Testbeurteilung ggf. durch externe Komparatoren



- MS Notepad, leeres Fenster
- Text „Hallo Welt!“ einfügen, markieren, Bearbeiten-Rückgängig klicken
- Was passiert:
 - Markierung wird aufgehoben,
 - Text wird wieder gelöscht, oder
 - Ganz was anderes?

:-)

Testplanung

- Erstellung eines detaillierten Dokumentes für folgende Punkte:
 - Testziele (welche Qualitätskriterien sollen eingehalten werden)
 - Teststufen (in welchen Projektphasen sind welche Aktivitäten auszuführen)
 - Testtypen (welche Tests sollen durchgeführt werden, welche Werkzeuge)
 - Randbedingungen (Hardware / Softwareumgebung)
 - Rollen und Verantwortlichkeiten
 - Meilensteine und Deliverables

Muster eines Testplanes (1)

1. Einführung

- 1.1 Zielsetzung
- 1.2 Geltungsbereich
- 1.3 Definitionen und Abkürzungen
- 1.4 Referenzen
- 1.5 Übersicht

2. Teststrategie

- 2.1 Testtypen
 - 2.1.1 Benutzbarkeitstest
 - 2.1.2 Modultest
 - 2.1.3 Integrationstest auf Komponentenebene
 - 2.1.4 Annahmeprüfung
 - 2.1.5 Systemtest
 - 2.1.6 Abnahme
- 2.2 Test der einzelnen Releases

Muster eines Testplanes (2)

3. Testtools

- 3.1 Testumgebung
- 3.2 Testmanagement und Fehlerverfolgung
- 3.3 Funktions- und Regressionstest
- 3.4 Last- und Performancetests
- 3.5 Organisation

4. Rollen

5. Testumgebungen

- 5.1 Entwicklungsumgebung
- 5.2 Systemtestumgebung
- 5.3 Pre-Productionumgebung
- 5.4 Produktionsumgebung

Muster eines Testplanes (3)

6 Verantwortungen und Akzeptanzkriterien

- 6.1 Modultest
- 6.2 Integrationstest auf Komponentenebene
- 6.3 Annahmeprüfung
- 6.4 Systemtest
- 6.5 Abnahme

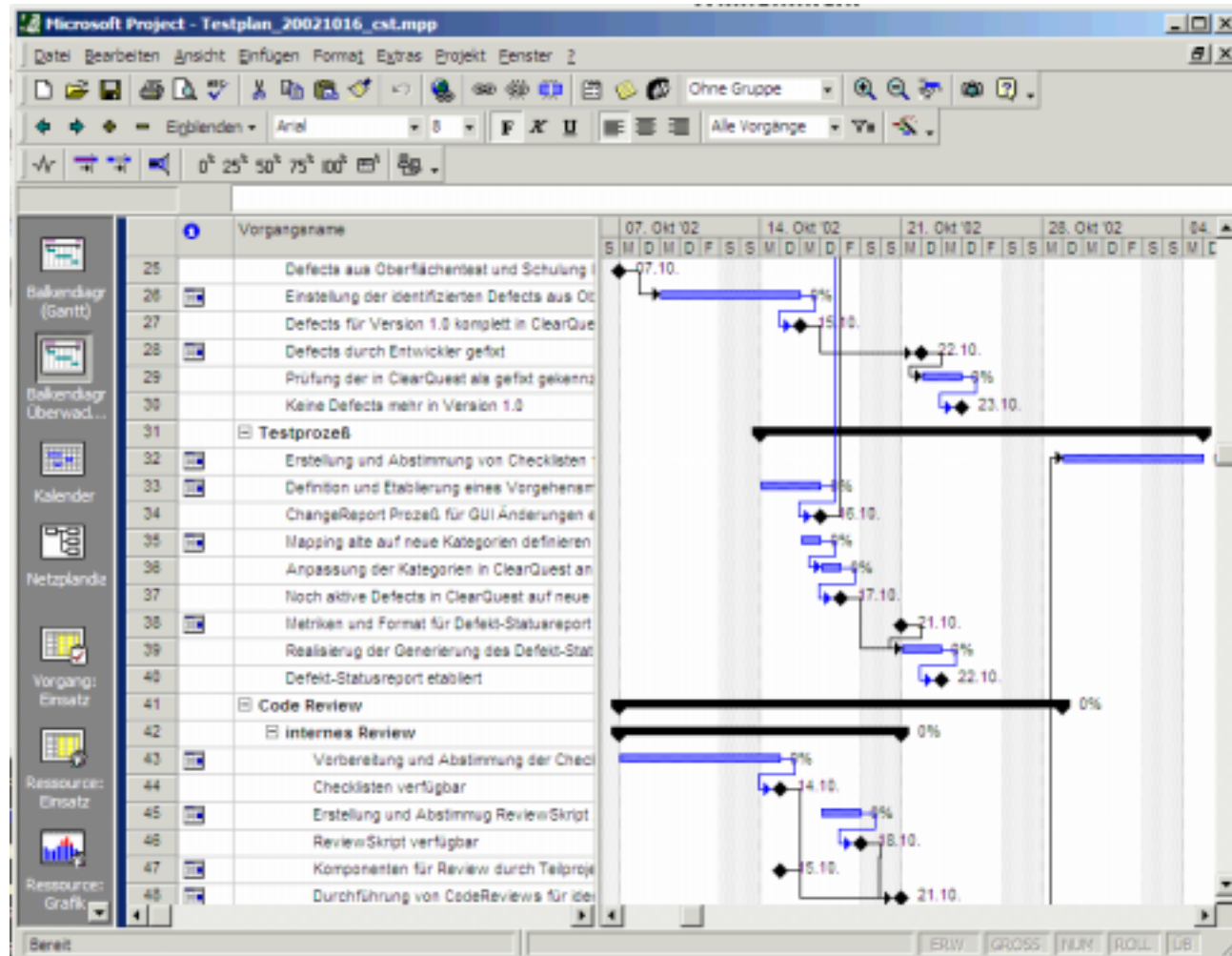
7 Dokumentation

- 7.1 Testberichte
- 7.2 Fehlerverwaltung

Beispiel Testrollen

Rolle	Aufgaben	Personen
Testleiter	Koordinierung Testaktivitäten Zuständigkeit für Ressourcen Erstellung Managementreports abschließende Bewertung der Ergebnisse	HXS
Testdesigner	Identifikation, Implementierung der Testfälle Erstellung des Testplanes Beurteilung der Effizienz des Testaufwandes	EKM, RSC
Tester	Durchführung der Tests Protokollierung u. Bewertung der Ergebnisse	MAF, EMM, RSC
Testautomatisierer	Erstellung von Testskripten Umsetzung der GUI-Map	HXS, MAF
Testsystem-administrator	Installation und Verwaltung des Testsystems Datenbankadministration und –management	EKM

Beispiel Testplan



Testdokumentation

- Problem: ggf. lange Archivierung der Ergebnisse (z.T. über 20 Jahre)
- IEEE/ANSI 829 Teil 1: Testdokumente:
 - Testplan
 - Testspezifikation
 - Testskript
 - Testfall
- IEEE/ANSI 829 Teil 2: Berichtsdokumente:
 - Software-Übergabe
 - Testprotokoll
 - Problemmeldung
 - Abschlussbericht

Testdokumentation: Testdokumente

- Testplan
 - Inhalt siehe oben
- Testspezifikation Komponenten
 - Zu testende Funktionen, Testverfahren
 - Testskripte und Testfälle, Pass / Fail Kriterien
- Testskripte
 - Testziel, spezielle Anforderungen, Einzelschritte
- Testfall
 - Zu testende Funktionen, Eingaben, Ausgaben
 - Umgebung, Besonderheiten, Abhängigkeiten

Testdokumentation: Berichtsdokumente

- Software-Übergabe
 - Inhalt der Lieferung, Übergabepunkt, Zustand
- Testprotokoll
 - Beschreibung, durchgeführte Testfälle, Ergebnis, unerwartete Ereignisse
- Problemmeldung
 - Beschreibung, Auswirkungen
- Abschlussbericht
 - Abweichungen, Umfang, Bewertung, Maßnahmen

Neulich in der Qualitätssicherung

