TUTORIAL 10

10.Study and document working of WinRunner and prepare a comparison with other testing tools.

Study & Documentation: WinRunner

What is WinRunner?

WinRunner is an **automated functional testing tool** developed by **Mercury Interactive**, which was later acquired by **HP**. It is designed for testing GUI-based applications and automating user interactions to validate software functionality.

Primary Use: Functional Regression Testing of applications.

Features of WinRunner

Feature	Description
Scripting Language	Test Script Language (TSL) – C-like scripting
Recording Modes	Context Sensitive and Analog
Application Support	Windows-based desktop applications
Integration	Works with TestDirector/Quality Center
Object Repository	Stores GUI objects for reuse
Checkpoints	Allows validation of GUI objects, databases, and text

How WinRunner Works (Step-by-Step)

Step 1: Launch WinRunner

- Open the WinRunner application.
- Choose the application under test (AUT).

Step 2: Record Test

- Use **Context-Sensitive Mode** to record user actions like clicking buttons, entering text, etc.

- WinRunner records actions in **TSL (Test Script Language)**.

Step 3: Insert Checkpoints

- Add verification points (e.g., Text Checkpoint, GUI Checkpoint).
- Example: Validate whether a login button appears.

Step 4: Save Test Script

- Save the recorded script for future test runs.

Step 5: Play Back the Script

- Replay the script to simulate user actions.
- Compare actual vs expected results using checkpoints.

Step 6: Analyze Test Results

- WinRunner generates a test execution log.
- Errors and mismatches are highlighted for review.

Advantages of WinRunner

- Easy to use with record & playback.
- Strong support for Windows applications.
- Integrates well with Quality Center.
- Customizable scripts using TSL.

Limitations of WinRunner

- Only supports Windows desktop applications (not web/mobile).
- Deprecated – no longer maintained or supported by HP.
- Licensing is costly compared to open-source tools.
- Lacks support for modern browsers and frameworks.

Comparison: WinRunner vs Other Tools

Feature	WinRunner	Selenium	UFT/QTP	TestComplete
Developed By	Mercury (HP)	Open Source Community	HP / Micro Focus	SmartBear
Supports Web Apps	❌	✅	✅	✅
Supports Desktop Apps	✅	❌	✅	✅
Script Language	TSL (Proprietary)	Java, Python, C#, etc.	VBScript	JavaScript, Python
Record & Playback	✅	✅ (via IDE)	✅	✅
Free / Paid	Paid (discontinued)	Free	Paid	Paid
Browser Support	Limited (old IE only)	Chrome, Firefox, Edge, etc.	Chrome, IE, Firefox	Chrome, IE, Edge
Mobile Testing	❌	✅ (via Appium)	✅	✅
Active Support	❌ (discontinued)	✅	✅	✅

Summary for Students

Parameter	Details
Tool Name	WinRunner
Use Case	GUI automation testing for Windows desktop apps
Script Language	TSL
Main Features	Recording, Checkpoints, Integration with QC
Limitations	No web/mobile support, discontinued tool
Best Alternatives	Selenium (web), UFT/QTP (desktop + web), TestComplete (UI-rich testing)

Suggested Student Activities

Task	Description
Practical 1	Study TSL scripting syntax (search archived documentation)
Practical 2	Watch legacy demo of WinRunner (YouTube or archived site)
Assignment	Prepare comparison report between WinRunner and Selenium
Report	Submit documentation with screenshots of tools compared
Viva Questions	What is TSL? What is a checkpoint in WinRunner?

Conclusion

WinRunner was a pioneer in GUI automation testing but is now obsolete. Students should study it for **historical understanding** and compare it with **modern tools** like Selenium and UFT for practical applications.