

TESTING WEB APPLICATIONS

ASSESSING WEB QUALITY

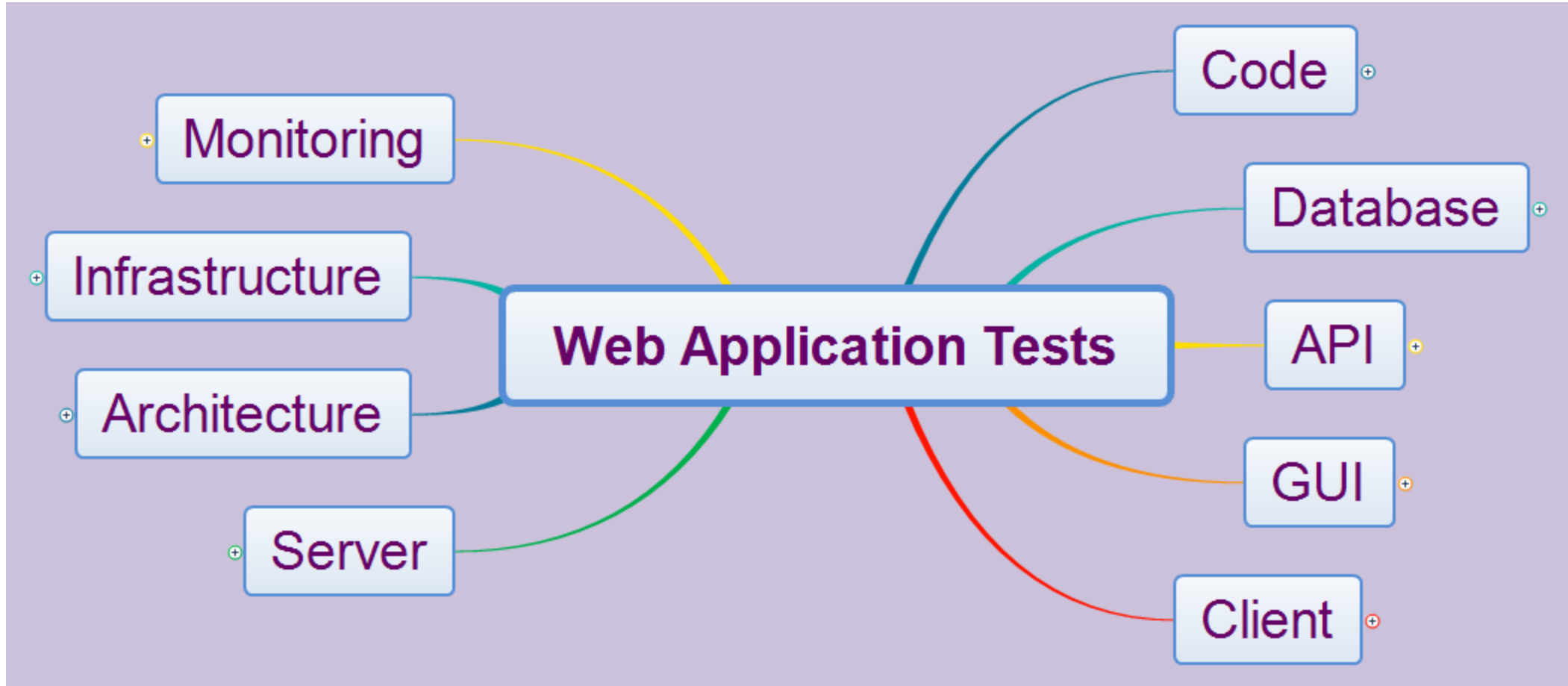
- **CONTENT** EVALUATED AT BOTH SYNTACTIC AND SEMANTIC LEVELS
- **FUNCTION** TESTED TO UNCOVER LACK OF CONFORMANCE TO REQUIREMENTS
- **STRUCTURE** IS ASSESSED TO ENSURE PROPER CONTENT AND FUNCTION ARE DELIVERED
- **USABILITY** IS TESTED TO ENSURE THAT EACH CATEGORY OF USER CAN BE SUPPORTED AS NEW CONTENT OR FUNCTIONALITY IS ADDED
- **NAVIGABILITY** IS TESTED TO ENSURE THAT ALL NAVIGATION SYNTAX AND SEMANTICS ARE EXERCISED

ASSESSING WEB QUALITY

- **PERFORMANCE** IS TESTED UNDER A VARIETY OF OPERATING CONDITIONS, CONFIGURATIONS, AND LOADING TO ENSURE A REASONABLE LEVEL OF USER RESPONSE
- **COMPATIBILITY** TESTED BY EXECUTING WEBAPP USING A VARIETY OF CLIENT AND SERVER CONFIGURATIONS
- **INTEROPERABILITY** TESTED TO ENSURE PROPER INTERFACES TO OTHER APPLICATIONS AND DATABASES
- **SECURITY** IS TESTED BY ASSESSING POTENTIAL VULNERABILITIES AND TRYING TO EXPLOIT EACH OF THEM

WEB ERROR CONCERNS

- MANY TYPES OF WEBAPP TESTS UNCOVER PROBLEMS EVIDENCED ON THE ***CLIENT SIDE*** USING AN SPECIFIC INTERFACE (E.G. MAY BE AN ERROR SYMPTOM, NOT THE ERROR ITSELF)
- IT MAY BE DIFFICULT TO REPRODUCE ERRORS ***OUTSIDE OF THE ENVIRONMENT*** IN WHICH THE ERROR WAS ORIGINALLY ENCOUNTERED
- MANY ERRORS CAN BE TRACED TO THE ***WEBAPP CONFIGURATION, INCORRECT DESIGN, OR IMPROPER HTML***
- IT IS HARD TO DETERMINE WHETHER ERRORS ARE CAUSED BY PROBLEMS WITH THE ***SERVER, THE CLIENT, OR THE NETWORK ITSELF***
- SOME ERRORS ARE ATTRIBUTABLE TO PROBLEMS IN THE ***STATIC OPERATING ENVIRONMENT*** AND SOME ARE ATTRIBUTABLE TO THE ***DYNAMIC OPERATING ENVIRONMENT***



TESTING WEB APPS

- **WEBAPP CONTENT** MODEL IS REVIEWED TO UNCOVER ERRORS.
- **INTERFACE MODEL** IS REVIEWED TO ENSURE ALL USE-CASES ARE ACCOMMODATED.
- **DESIGN MODEL** FOR WEBAPP IS REVIEWED TO UNCOVER NAVIGATION ERRORS.
- **USER INTERFACE** IS TESTED TO UNCOVER PRESENTATION ERRORS AND/OR NAVIGATION MECHANICS PROBLEMS.
- SELECTED **FUNCTIONAL COMPONENTS** ARE UNIT TESTED.

TESTING WEB APPS

- **NAVIGATION** THROUGHOUT THE ARCHITECTURE IS TESTED.
- WEBAPP IS IMPLEMENTED IN A VARIETY OF DIFFERENT ENVIRONMENTAL CONFIGURATIONS AND THE **COMPATIBILITY** OF WEBAPP WITH EACH IS ASSESSED.
- **SECURITY** TESTS ARE CONDUCTED.
- **PERFORMANCE** TESTS ARE CONDUCTED.

TESTING WEB APPS

- WEBAPP IS TESTED BY A MONITORED GROUP OF SELECTED END-USERS LOOKING FOR
 - CONTENT ERRORS
 - NAVIGATION ERRORS
 - USABILITY CONCERNS
 - COMPATIBILITY ISSUES
 - PERFORMANCE

WEB APP TESTING PROCESS

- CONTENT TESTING
 - TRIES TO UNCOVER CONTENT ERRORS
- INTERFACE TESTING
 - EXERCISES INTERACTION MECHANISMS AND VALIDATES AESTHETIC ASPECTS OF UI
- NAVIGATION TESTING
 - USE-CASES BASIS FOR DESIGN OF TEST CASES THAT EXERCISE EACH USAGE SCENARIOS AGAINST THE NAVIGATION DESIGN (INTEGRATION TESTING)
- COMPONENT TESTING
 - EXERCISES THE WEBAPP CONTENT AND FUNCTIONAL UNITS (INTEGRATION TESTING)

WEB APP TESTING PROCESS

- CONFIGURATION TESTING
 - ATTEMPTS TO UNCOVER ERRORS TRACEABLE TO A SPECIFIC CLIENT OR SERVER ENVIRONMENT
 - CROSS-REFERENCE TABLE IS USEFUL
- PERFORMANCE TESTING TO ASSESS
 - WEBAPP RESPONSE TIME AND RELIABILITY UNDER VARYING SYSTEM LOADS
 - WHICH WEBAPP COMPONENTS ARE RESPONSIBLE FOR SYSTEM DEGRADATION
 - HOW PERFORMANCE DEGRADATION IMPACTS OVERALL WEBAPP REQUIREMENTS
- SECURITY TESTING
 - TESTS DESIGNED TO EXPLOIT WEBAPP OR ENVIRONMENT VULNERABILITIES

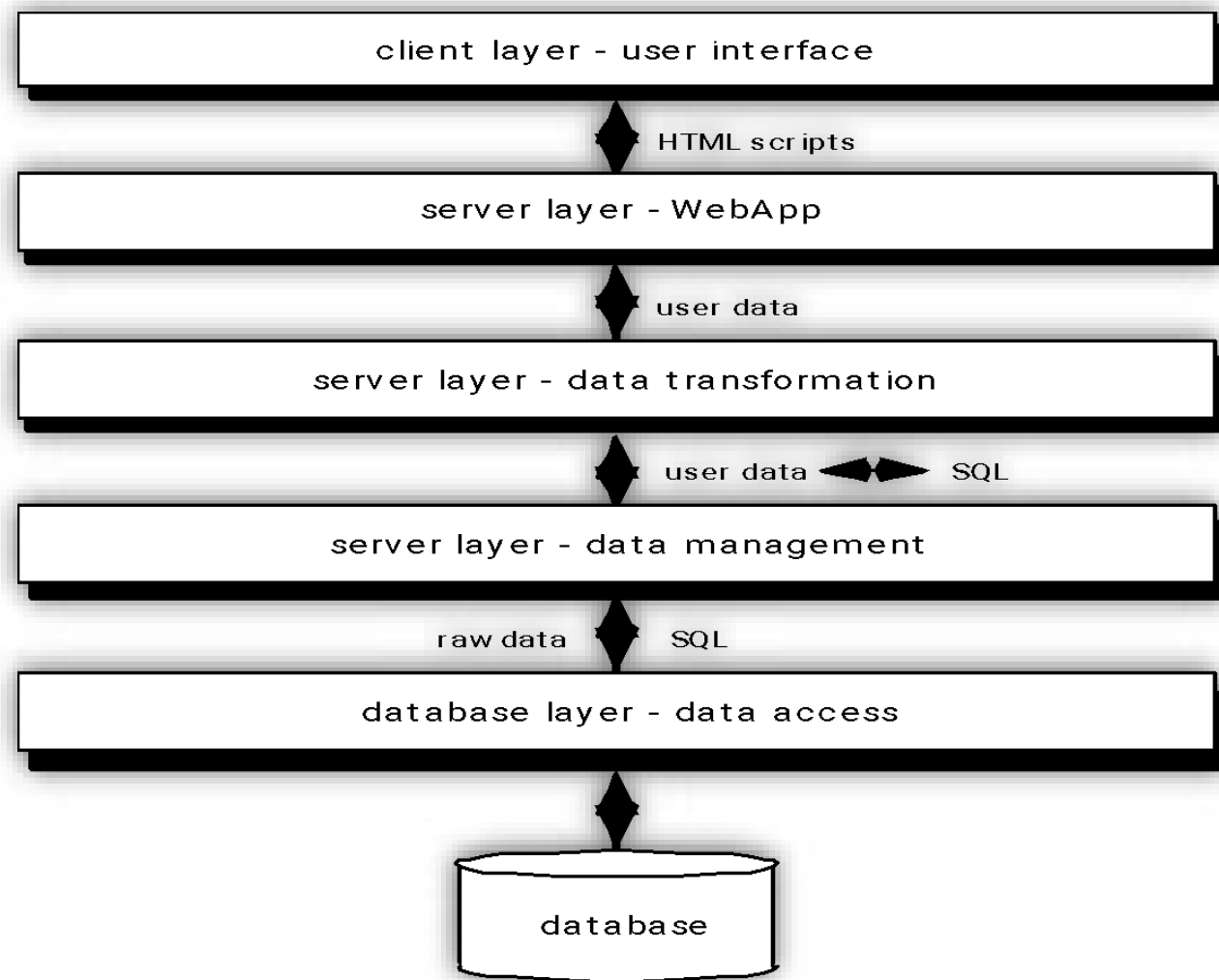
CONTENT TESTING OBJECTIVES

- UNCOVER SYNTACTIC ERRORS IN ALL MEDIA
 - (E.G. TYPOS)
- UNCOVER SEMANTIC ERRORS
 - (E.G. ERRORS IN COMPLETENESS OR ACCURACY)
- FIND ERRORS IN ORGANIZATION OR STRUCTURE OF CONTENT PRESENTED TO END-USER

DATABASE TESTING

- THE **ORIGINAL QUERY** MUST BE CHECKED TO UNCOVER ERRORS IN TRANSLATING THE USER'S REQUEST TO SQL
- PROBLEMS IN COMMUNICATING BETWEEN THE **WEBAPP SERVER AND DATABASE SERVER** NEED TO BE TESTED.
- NEED TO DEMONSTRATE THE **VALIDITY OF THE RAW DATA** SENT FROM THE DATABASE TO THE WEBAPP AND THE VALIDITY OF THE TRANSFORMATIONS APPLIED TO THE RAW DATA.
- NEED TO TEST **VALIDITY OF DYNAMIC CONTENT OBJECT** FORMATS TRANSMITTED TO THE USER AND THE VALIDITY OF THE TRANSFORMATIONS TO MAKE THE DATA VISIBLE TO THE USER.

DATABASE TESTING



USER INTERFACE TESTING

- INTERFACE FEATURES ARE TESTED TO ENSURE THAT DESIGN RULES, AESTHETICS, AND RELATED VISUAL CONTENT IS AVAILABLE FOR USER WITHOUT ERROR.
- INDIVIDUAL INTERFACE MECHANISMS ARE TESTED USING UNIT TESTING STRATEGIES.
- EACH INTERFACE MECHANISM IS TESTED IN THE CONTEXT OF A USE-CASE OF NAVIGATION SEMANTIC UNIT (E.G. THREAD) FOR A SPECIFIC USER CATEGORY
- COMPLETE INTERFACE IS TESTED AGAINST SELECTED USE-CASES AND NAVIGATION SEMANTIC UNIT TO UNCOVER INTERFACE SEMANTIC ERRORS
- INTERFACE IS TESTED IN A VARIETY OF ENVIRONMENTS TO ENSURE COMPATIBILITY

TESTING INTERFACE MECHANISMS

- LINKS
 - EACH LINK IS LISTED AND TESTED
- FORMS
 - CHECK LABELS, FIELD NAVIGATION, DATA ENTRY, ERROR CHECKING, DATA TRANSMISSION, MEANINGFUL ERROR MESSAGES
- CLIENT-SIDE SCRIPTING
 - BLACK BOX TESTING AND COMPATIBILITY TESTS
- DYNAMIC HTML
 - CORRECTNESS OF GENERATED HTML
 - COMPATIBILITY TESTS

TESTING INTERFACE MECHANISMS

- CLIENT-SIDE POP-UP WINDOWS
 - PROPER SIZE AND PLACEMENT OF POP-UP
 - WORKING CONTROLS
 - CONSISTENT WITH APPEARANCE OF WEB PAGE
- STREAMING CONTENT (DEMONSTRATE EXISTENCE, ACCURACY, AND CONTROL OVER CONTENT DISPLAY)

TESTING INTERFACE MECHANISMS

- COOKIES
 - CHECK THAT SERVER CONSTRUCTS COOKIE CORRECTLY
 - COOKIE TRANSMITTED CORRECTLY
 - ENSURE PROPER LEVEL OF PERSISTENCE
 - CHECK TO SEE WEBAPP ATTACHES THE CORRECT COOKIES
- APPLICATION SPECIFIC INTERFACE MECHANISMS

USABILITY TESTING

- DEFINE SET OF USABILITY TESTING CATEGORIES AND IDENTIFY GOALS FOR EACH
- DESIGN TESTS THE WILL ENABLE EACH GOAL TO BE EVALUATED
- SELECT PARTICIPANTS TO CONDUCT THE TESTS
- INSTRUMENT PARTICIPANTS' INTERACTIONS WITH THE WEBAPP DURING TESTING
- DEVELOP METHOD FOR ASSESSING USABILITY OF THE WEBAPP

USABILITY TESTING GOALS

- INTERACTIVITY
 - INTERACTION MECHANISMS ARE EASY TO UNDERSTAND AND USE
- LAYOUT
 - NAVIGATION, CONTENT, AND FUNCTIONS ALLOWS USER TO FIND THEM QUICKLY
- READABILITY
 - CONTENT UNDERSTANDABLE
- AESTHETICS
 - GRAPHIC DESIGN SUPPORTS EASY OF USE

USABILITY TESTING GOALS

- DISPLAY CHARACTERISTICS
 - WEBAPP MAKES GOOD USE OF SCREEN SIZE AND RESOLUTION
- TIME SENSITIVITY
 - CONTENT AND FEATURES CAN BE ACQUIRED IN TIMELY MANNER
- PERSONALIZATION
 - ADAPTIVE INTERFACES
- ACCESSIBILITY
 - SPECIAL NEEDS USERS

COMPATIBILITY TESTING

- GOAL IS TO UNCOVER EXECUTION PROBLEMS THAT CAN BE TRACED TO CONFIGURATION DIFFERENCES
 - DEFINE A SET OF COMMONLY ENCOUNTERED CLIENT-SIDE COMPUTING CONFIGURATIONS AND THEIR VARIANTS
 - ORGANIZE THIS INFORMATION (COMPUTING PLATFORM, TYPICAL DISPLAY DEVICES, OPERATING SYSTEM, AVAILABLE BROWSERS, CONNECTION SPEEDS)
 - DERIVE COMPATIBILITY VALIDATION TEST SUITE FROM EXISTING INTERFACE TESTS, NAVIGATION TESTS, PERFORMANCE TESTS, AND SECURITY TESTS

COMPONENT LEVEL TESTING

- BLACK BOX AND WHITE BOX TESTING OF EACH WEBAPP FUNCTION
- USEFUL TEST CASE DESIGN METHODS
 - EQUIVALENCE PARTITIONING
 - BOUNDARY VALUE ANALYSIS (ESP. FORM FIELD VALUES)
 - PATH TESTING
 - FORCED ERROR TESTING

NAVIGATION TESTING

- **SYNTACTIC NAVIGATION TESTING**

- ✓ LINKS

- ✓ REDIRECTS

- ✓ BOOKMARKS

- ✓ FRAMES AND FRAMESETS

- ✓ SITE MAPS

- ✓ INTERNAL SEARCH ENGINES

NAVIGATION TESTING

- **SEMANTIC NAVIGATION TESTING**

- ✓ NAVIGATION SEMANTIC UNITS ARE DEFINED BY A SET OF PATHWAYS THAT CONNECT NAVIGATION NODES
- ✓ EACH NSU MUST ALLOW A USER FROM A DEFINED USER CATEGORY ACHIEVE SPECIFIC REQUIREMENTS DEFINED BY A USE-CASE
- ✓ TESTING NEEDS TO ENSURE THAT EACH PATH IS EXECUTED IN ITS ENTITY WITHOUT ERROR
- ✓ EVERY RELEVANT PATH MUST BE TESTED
- ✓ USER MUST BE GIVEN GUIDANCE TO FOLLOW OR DISCONTINUE EACH PATH BASED ON CURRENT LOCATION IN SITE MAP

CONFIGURATION TESTING

- **CONFIGURATION TESTING SERVER-SIDE**

- ✓ COMPATIBILITY OF WEBAPP WITH SERVER OS
- ✓ CORRECT FILE AND DIRECTORY CREATION BY WEBAPP
- ✓ SYSTEM SECURITY MEASURES DO NOT DEGRADE USER SERVICE BY WEBAPP
- ✓ TESTING WEBAPP WITH DISTRIBUTED SERVER CONFIGURATION
- ✓ WEBAPP PROPERLY INTEGRATED WITH DATABASE SOFTWARE
- ✓ CORRECT EXECUTION OF WEBAPP SCRIPTS
- ✓ EXAMINATION SYSTEM ADMINISTRATION ERRORS FOR IMPACT ON WEBAPP
- ✓ ON-SITE TESTING OF PROXY SERVERS

CONFIGURATION TESTING

- **CONFIGURATION TESTING CLIENT-SIDE**

- ✓ **HARDWARE**

- ✓ **OPERATING SYSTEMS**

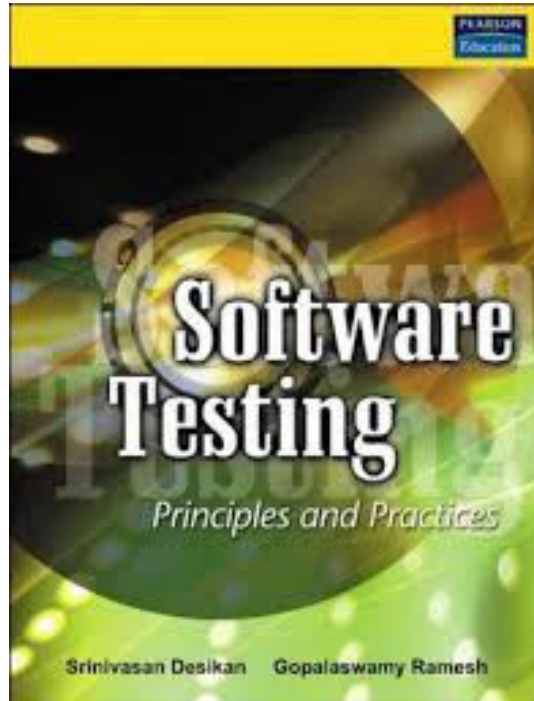
- ✓ **BROWSER SOFTWARE**

- ✓ **USER INTERFACE COMPONENTS**

- ✓ **PLUG-INS**

- ✓ **CONNECTIVITY**

REFERENCES



GURU99

ðG

edureka!

Software Testing Help