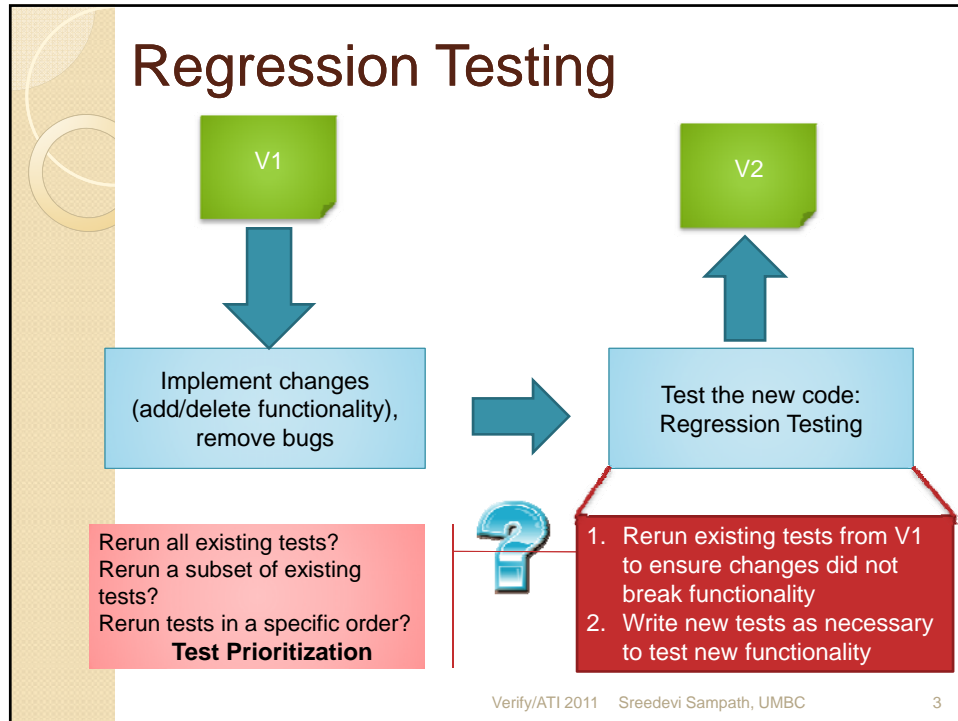


CPUT – Combinatorial-Based Prioritization for User-Session-Based Testing of Web Applications

Sreedevi Sampath, University of Maryland, Baltimore County
Renee Bryce, Utah State University
Sachin Jain, University of Maryland, Baltimore County
Schuyler Manchester, Utah State University
Richard Kuhn and Raghu Kacker, NIST

Overview of CPUT

- Automated and general framework to
 - Create and manage regression test suites
 - Designed for web applications
 - Logs usage data and converts to test cases
 - Prioritizes test cases and creates test orders
- Test cases can then be replayed and used to test web systems

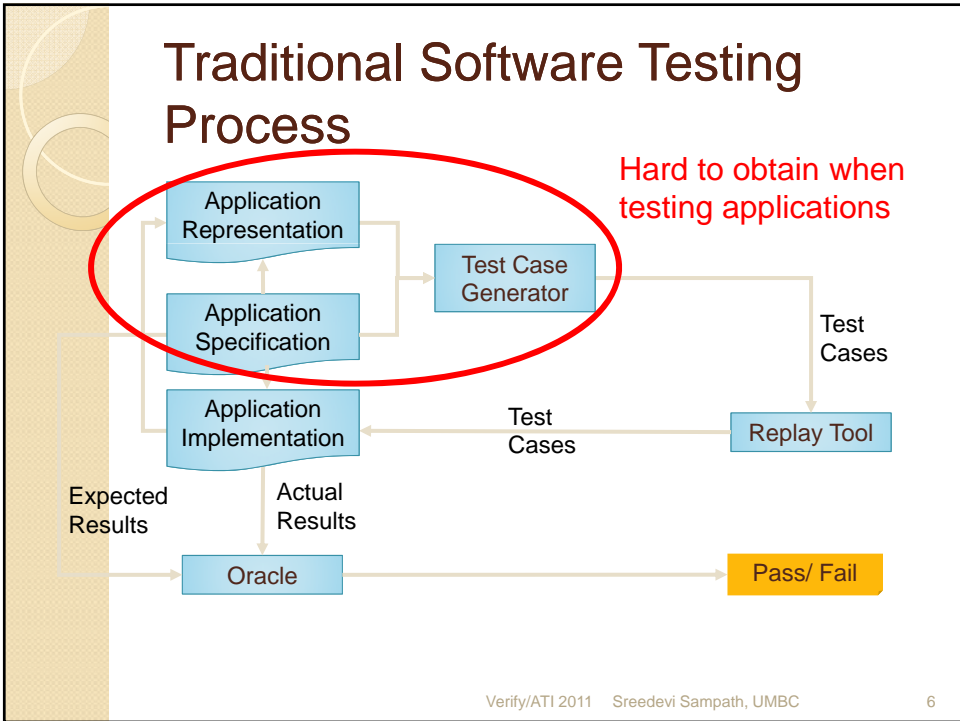
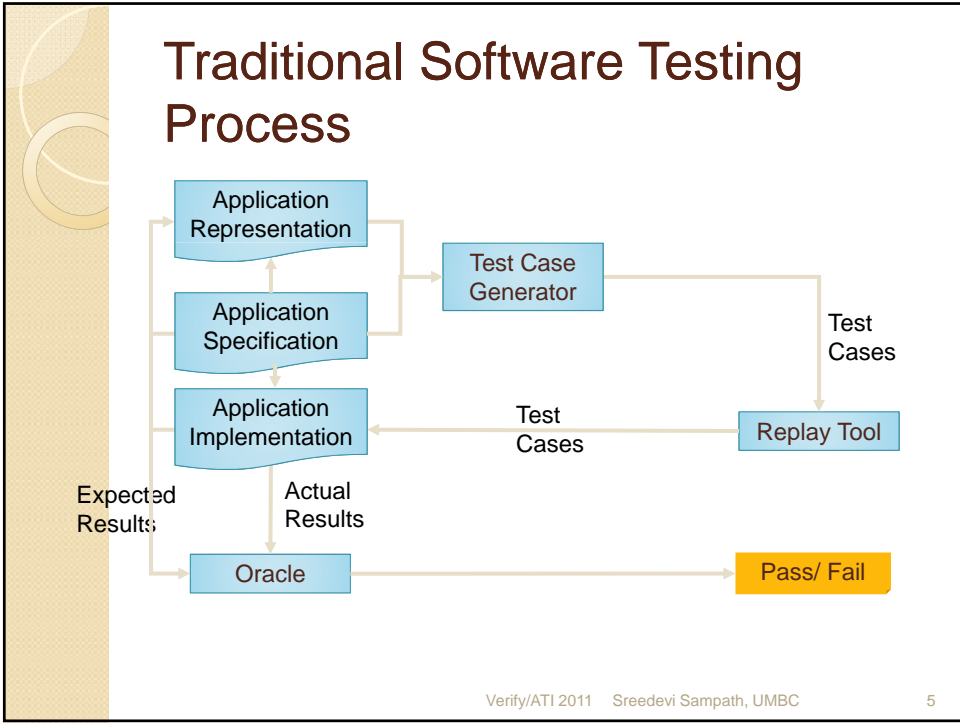


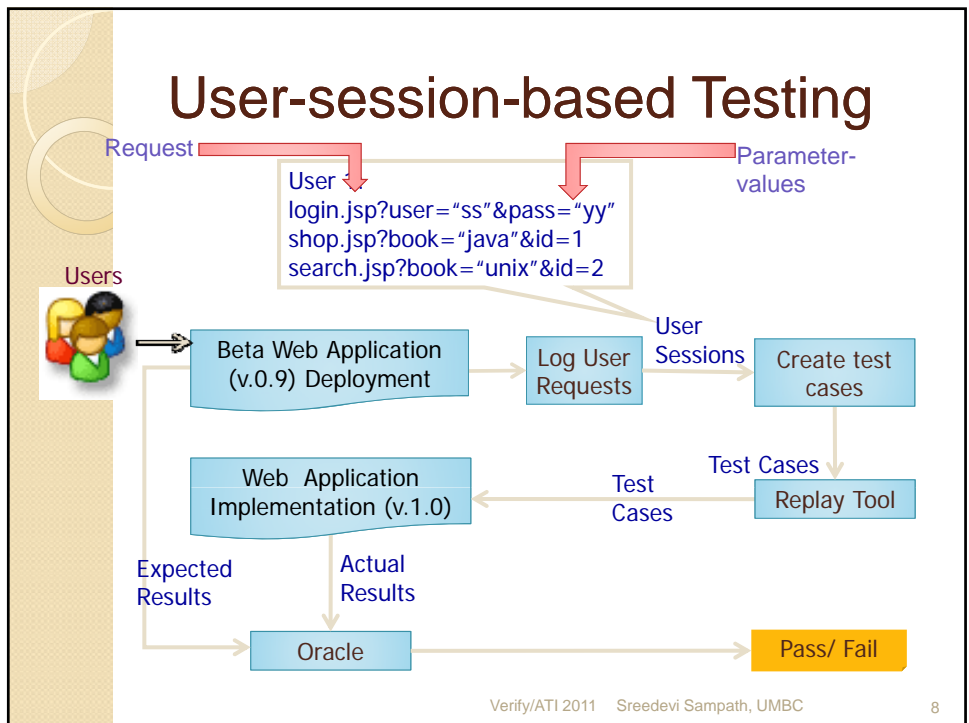
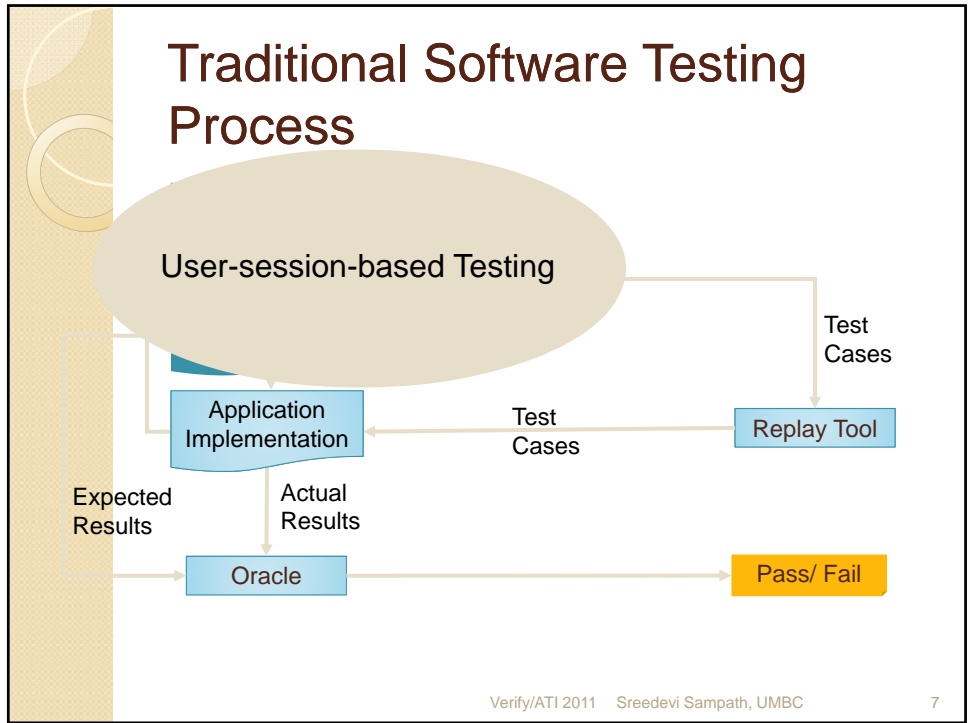
Need for Reliable Web Applications

- Increasing shift of applications to the web
 - E.g., Google Docs
- Huge losses on web site failure: tune of millions of dollars per hour¹
- Large number of failures during maintenance²

1. Web Application Development - Bridging the Gap between QA and Development by Michal Blumenstyk
 2. Causes of Failures in Web Applications by Solia Pertet and Priya Narsimhan. December 2005

Verify/ATI 2011 Sreedevi Sampath, UMBC 4





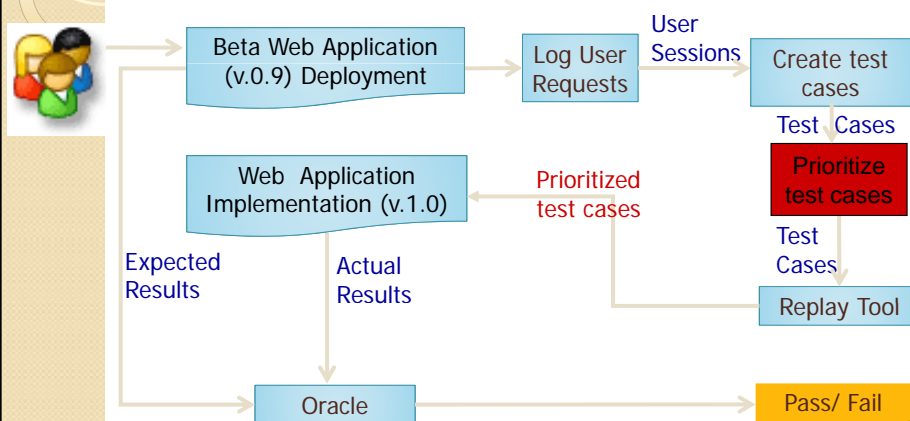
Test prioritization

- Order existing tests based on some criterion to achieve a performance goal
 - Examples of traditional prioritization criteria: total statement coverage, total method coverage
 - Performance goal: find faults quickly in test execution cycle

Verify/ATI 2011 Sreedevi Sampath, UMBC

9

User-session-based Test Case Prioritization



Verify/ATI 2011 Sreedevi Sampath, UMBC

10

Our test prioritization criteria

- Combinatorial-based
 - 2-way
- Count-based
 - Number of requests
 - Number of parameter-values

Renee Bryce, Sreedevi Sampath, Atif Memon, **Developing a Single Model and Test Prioritization Strategies for Event-Driven Software**, IEEE Transactions on Software Engineering, Vol. 37, Issue 1, pp 48-64, 2011

Verify/ATI 2011 Sreedevi Sampath, UMBC

12

Combinatorial-based: 2-way

Test Case 1:

Catalog.java, item_name="shirt", item_weight="2"
View_Cart.java, ship_type="air", zip="21250"

1

2

3

4

- 2-way interactions:
(1,3) (1,4) (2,3) (2,4)
- **Intuition:** Interactions of parameters set to values on different windows expose faults
- **Criterion:** Give higher priority to tests with larger number of 2-way interactions

Verify/ATI 2011 Sreedevi Sampath, UMBC

13

Count-based: Request length

Test Case 1:

`Catalog.java`, item_name="shirt", item_weight="2"
`View_Cart.java`, ship_type="air", zip="21250"

- Number of requests in test: 2
- **Intuition:** tests that contain more requests are more likely to reveal faults because they cover a large part of the underlying code
- **Criterion:** Give higher priority to tests with larger number of requests

Count-based: Parameter-Value length

Test Case 1:

`Catalog.java`, item_name="shirt", **item_weight="2"**
`View_Cart.java`, **ship_type="air"**, zip="21250"

- Number of parameter-values in test: 4
- **Intuition:** tests that set more parameters to values are more likely to reveal faults
- **Criterion:** Give higher priority to tests with larger number of parameter-values

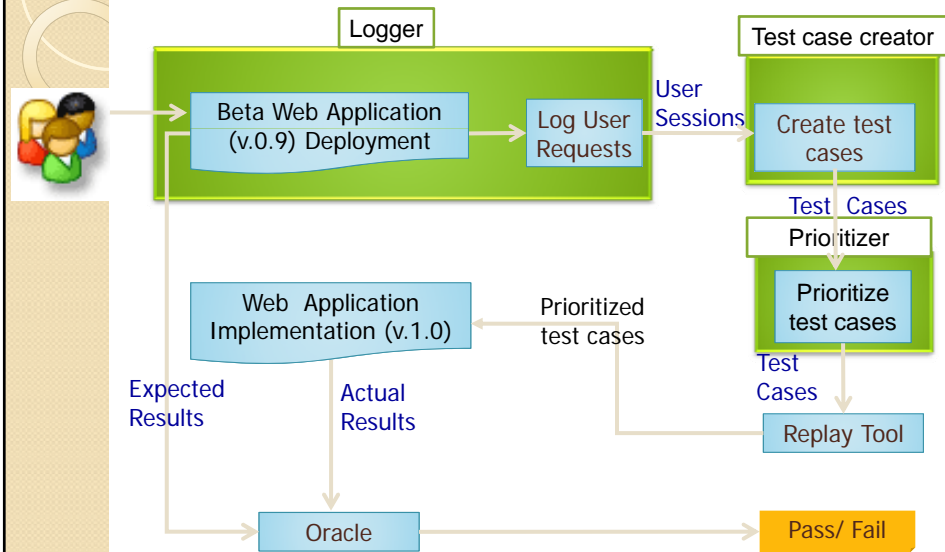
Empirical study: Results

- Measure the rate of fault detection

Application	Best 3
Calc	{ <u>PV-LtoS</u> , <u>2-way</u> , <u>Weighted-Freq</u> }
Paint	{ <u>PV-LtoS</u> , <u>2-way</u> , MFPS}
SSheet	{UniqWin, <u>2-way</u> , <u>1-way</u> }
Word	{ <u>PV-LtoS</u> , <u>2-way</u> , MFPS}
Book	{APS, <u>1-way</u> , <u>2-way</u> }
CPM	{ <u>2-way</u> , <u>1-way</u> , <u>PV-LtoS</u> }
Masplas	{ <u>Weighted-Freq</u> , <u>Action-LtoS</u> , <u>2-way</u> }

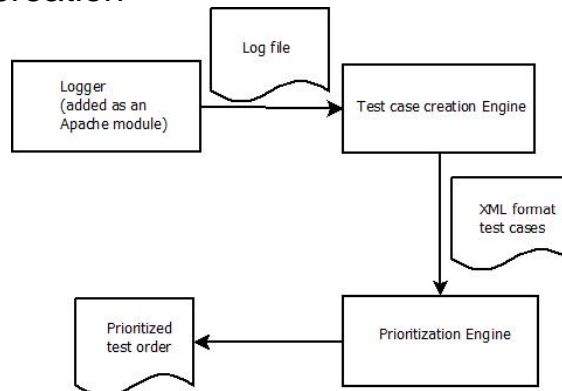
Prioritization criteria that were the best in all the subject applications (3 web and 3 GUI applications)
Underlining indicates criterion that is always among the best

CPUT components: 3 engines



CPUT components

- Three main engines
 - Logger
 - Test case creation
 - Prioritizer



Verify/ATI 2011 Sreedevi Sampath, UMBC

18

Logger

- Enables capture of HTTP GET and POST requests and associated data
- Developed as a module for Apache web server
- Will work on both Linux and Windows platforms
- Minimal performance overhead
- Easy integration with rest of the web server

Verify/ATI 2011 Sreedevi Sampath, UMBC

19

Deploying the Logger

- Server administrator must place module with other Apache modules
- Enable the module in Apache's configuration file (can specify path and filename). Log written to a text file
- Format of entry in log file

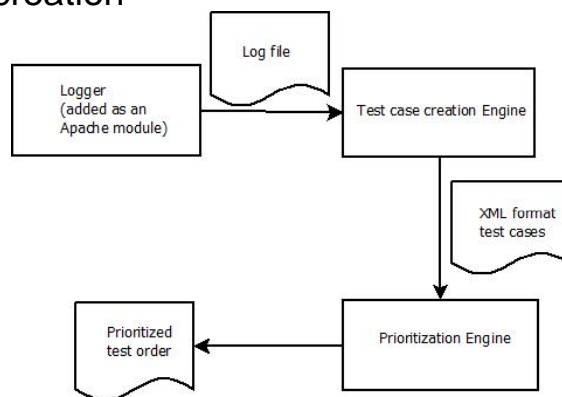
```
[Fri Jan 08 13:33:01 2010] [# 127.0.0.1]# POST]#  
/schoolMateApplication/index.php]#  
PHPSESSID=a9099cd16db2e4134200cd69f6dc87cf]#  
http://localhost:8000/schoolMateApplication/index.php]#  
PostData:page2=5&logout=1&page=1
```

Verify/ATI 2011 Sreedevi Sampath, UMBC

20

CPUT components

- Three main engines
 - Logger
 - Test case creation
 - Prioritizer



Verify/ATI 2011 Sreedevi Sampath, UMBC

21

Test case creation (parser)

- Efficient storage and retrieval of web usage logs
 - Store usage log and test cases in PostgreSQL database
- Apply previously proposed heuristics to create test cases [Sampath TSE07, Sprenkle ASE05]
- Extensible and generic format of test cases to enable replay
 - Design an XML format for tests

Test case creation (parser)

- Storing usage log in database
 - Create new table
 - Append to existing table
 - Overwrite existing table

Test case creation: XML test case format

Example test case

```

1 <testSuite>
2 <session id="1.XML">
3 <url>
4 <request type>POST</request type>
5 <baseUrl>/SchoolMate/index.php</baseUrl>
6 <param>
7   <name>book_name</name>
8   <value>java</value>
9 </param>
10 <param>
11   <name>book_author</name>
12   <value>savitch</value>
13 </param>
14 </url>
15 </session>
16 </testSuite>

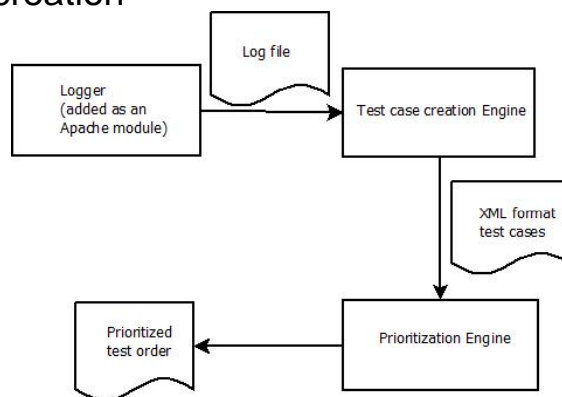
```

Verify/ATI 2011 Sreedevi Sampath, UMBC

24

CPUT components

- Three main engines
 - Logger
 - Test case creation
 - Prioritizer



Verify/ATI 2011 Sreedevi Sampath, UMBC

26

Prioritizer

- Three prioritization criteria
 - 2way, request-length, parameter-value length
 - Random prioritization
- Accommodate two types of web systems
 - Unique URL: each page is identified by a unique base URL
 - Non-unique URL: application has the same base URL for all its pages, and the value of one or more of parameters is used to determine which page to load next

Unique vs. Non-unique URL

Unique URL

- Each page in web application has a unique base URL
- Example test case

```
index.php
registration.php?name=tom
adduser.php?name=henry&pass=joy
```

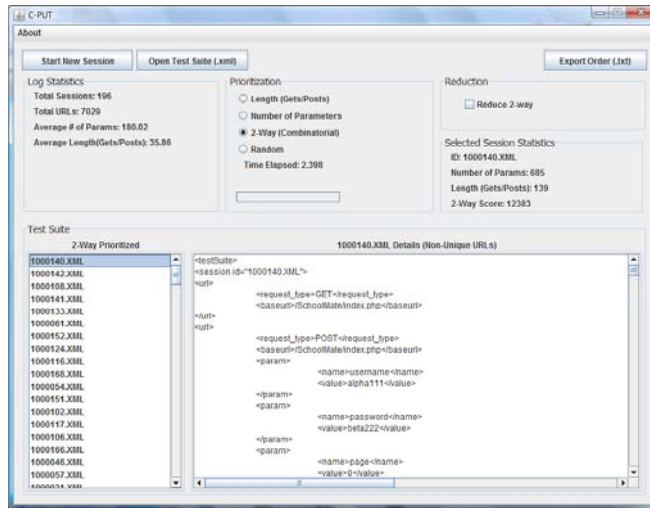
Non-unique URL

- Same base URL for all the web pages. Value of parameter-values determines unique page
- Example test case

```
index.php
index.php?name=tom&page1=0&page2=1
index.php?name=henry&pass=joy&page1=1&page2=1
```

Tool Demonstration

CPUT main screen



Verify/ATI 2011 Sreedevi Sampath, UMBC

29