

## Chapter 25

### Testing Web Applications

#### Testing quality dimensions:

Content- function- structure- usability- navigability- performance- compatibility- interoperability- security.

Errors in a WebApp:	WebApp testing strategy:
<p>1-often see a symptom of the error, not the error itself.</p> <p>2-it's impossible reproduce the error outside the environment that occurred the error inside it.</p> <p>3-Many errors can be traced to the configuration and regardless of the cause of the error.</p> <p>4-errors can be difficult to trace across three architectural layers: the client, the server, or the network itself.</p> <p>5-Some errors are due to the static operating environment, while others are attributable to the dynamic operating environment.</p>	<p>1-reviev the content model.</p> <p>2-reviev the interface model.</p> <p>3-reviev design model.</p> <p>4-test user interface.</p> <p>5-Selected functional components are unit tested.</p> <p>6-Navigation throughout the architecture is tested.</p> <p>7-test the compatibility for each configuration to WebApp.</p> <p>8-test the security.</p> <p>9-test performance.</p> <p>10-test the WebApp by the end users.</p>

#### Testing interface mechanisms:

Links-forms-client-side scripting-dynamic HTML-client side pop-up windows-CGI scripts-streaming content-cookies-application specific interface mechanisms.

<b>Usability tests:</b>	design by Web team .. executed by end-users.
<b>Compatibility testing:</b>	to uncover errors that cusses configuration differences.
<b>Component-level testing:</b>	to uncover in WebApp functions.
<b>Navigation Testing:</b>	Navigation links-Redirects-Bookmarks-Frames and framesets-Site maps-Internal search engines.
<b>Configuration Testing:</b>	from server-side and client-side.
<b>Security Testing:</b>	to check out weaknesses of client and server side.
<b>Load testing:</b>	to test number of users, transactions and data load.
<b>Stress testing:</b>	check the system after override the allowed value from the variables.. N,T,D.