BEST TESTING PRACTICES

Developing Test Cases:
- Determine the focus of the test before beginning!
- Create test scenarios – Follow the application functionality when creating.
- Scenarios should have multiple test cases tied to each.
- Positive Test Cases – Acceptable Input data - Expected to work.
- Negative Test Cases – Unacceptable Input data - Expected NOT to work.
- Creative Brainstorming Test Cases – Helps detect unexpected results.
- Consider Risk Factors – Increase number of test cases for volatile areas of the systems, complex code, newly created code.
- Number of test cases – Average is 2 to 3 per function (for Low to Mid risk factors).

Documentation:
- Test case information should be extremely detailed.
- Number each test case – See format of the Test Plan Template.
- Expected Results should be documented prior to testing for EACH Test Case.
- Document run times for all reports and processes.
- Note the user role being utilized during testing.
- Copy errors exactly as they appear – Screen shots are the best way to capture the error message. Include the test case number in the screen shot’s file name. (Print Key or Snagit works well).
- Errors should stand out on test plan (use red to highlight).
- Test environment - Database instance and version should be noted on test plan.
- Web testing - Document URL, Browsers and Versions used in test.
- Be consistent with documentation throughout test plan(s).
- Document logistics if appropriate.
- Document tester’s name on test plan.

Web Testing:
- Always test security – i.e. Log-on process, password management, encryption, etc.
- Test connectivity and response times.
- Browser compatibility – Should be tested on at least one version of Internet Explorer and one version of Netscape. Best to use versions supported by the GT Campus.
- Test on multiple operating systems such as Windows NT, Windows 2000 and Mac OSx.
- Check all hyperlink connections – Document broken links. (Link Checkers tools are useful.)
- Verify usability.

General Testing Practices:
- Whenever possible, tests should be isolated and without system or user conflicts.
- Retests should always be executed and fully documented when significant errors are found.
- Review all scenarios and test cases with a functional expert prior to testing.
- Explain findings and provide backup documentation.

For more information on testing, refer to the EIS Testing Standards and Documentation document.