2... Don't forget to use:

CSC 405 Lecture Notes Week 9

OCU/Manet Testing Decisions

I. Items III - XVI correspond to the "best practices" points discussed during lecture.

II. These are the decisions we made on Wednesday of Week 9.

III. Harnesses & Frameworks

Harness 1: OCU teams use QT Test

Harness 2: Manet teams use Cpp Unit

Harness 3: Drive with automated QT GUI harness

Harnesses 4-5:

  The football field across from Vons

Harness 6:

  Protocol simulation with HW sim

IV. Coverage

postpone, do if easy

V. Number of of tests

It is what it is

VI. Mem tests

Postpone, do if easy (with valgrind)

VII. I/O Tests

Not our problem

VIII. Crash and power loss tests

Ditto

IX. Fuzz tests

Yes, with random packet data, random data parm val

X. Boundary parms

Yes to all, and add inactivity time
Need to determine spec-level boundaries for each

XI. Optimization Tests

Not our problem

Tim said "he does special compilation"

XII. Regression

Of course of course.

XIII. Malformed data tests

See Fuzz tests above.

XIV. Asserts and exceptions

Bats and OCU arm wrestle; policy to be determined

XV. Valgrind

Determine policy of use by end of quarter.

XVI. Re-visit repo structure

Do this this week.

III. **Original Action Items for this Week**

A. Agreed testing framework for OCU teams.

B. Agreed testing framework for 80211s teams.

C. Committed testing for 80211s and OCU teams.

D. Project-wide regression test makefile.

E. Agreement, as appro, on preceding practices.

IV. OAdditional Action Items for this Week

A. All 4 teams create testing makefiles.

B. All 4 teams run valgrind, at least once.

C. All 4 teams run lcov, or equivalent, at least once.

V. **Suggested repository updates.**

A. Populate `testing` subirs for all 4 subprojects.

B. Move `batmobile/implementa-tion/.../*Tests*` to `batmobile/test-ing/implementation/.../*Tests*`.

C. Code tests and install for `80211s`, `kareem-nassar`, `ocunited`.

Repository additions and modifications, cont'd

D. Add `manet-ocu/testing` dir, with `Makefile` for project-wide test build and execute.

E. Install bug-tracking supporting infrastructure (if not already there).

F. Add `requirements` dir and put SRS there.

G. Add `administration` dir and put project-wide admin docs there.