SE 4G06 (Capstone Design Project)

Fall & Winter 2024–2025

Verification and Validation Plan

Dr. Spencer Smith

Faculty of Engineering, McMaster University

October 24, 2024

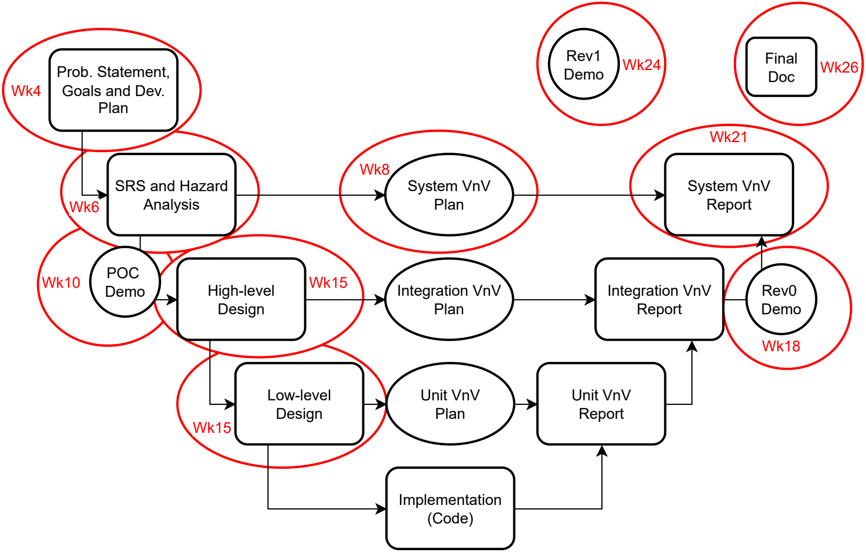
Verification and Validation Plan

- Administrative details
- Review big picture view of our process
- Feedback on Problem and Goal State, POC and Dev. Plan
- PoC Demos
- Motivating VnV questions/discussion
- Verification and validation review
- VnV "Template"
- VnV rubric
- Sample VnV plans

Administrative Details

- Hazard analysis due **October 25**
- Informal VnV Plan presentations scheduled by TAs
- VnV Plan due Fri, Nov 1, 11:59 pm
- VnV issues due by Nov 4 (+ 2 day grace period)
- Team Avenue survey avail Oct 30, due Nov 6
- Team contribution report, due Nov 6
- PoC Demos Nov 11 – Nov 27, location TBD
- GenderMag tutorial, Mon, Nov 4, 2:30 pm (online)
- Extras folder in capTempate
- Keep all documents alive
- Don't just focus on the next deliverable
- Recommend looking at final doc rubric (on Avenue)
  - Use issue tracker for changes
  - One commit for one change
  - Plan for extras

Review Our Process



Feedback on Prob State, Goals and Dev Plan

- Overall well done
- POC plan problems
  - Plans were too ambitious
  - Plans not based on risks (hurtles/pain points)
  - Feasibility risk, not project risk
  - If there was a problem, update your plan!
- Grammar problems for some teams
- Charter does not include quantified expectations with consequences
- Review the assignment rubric
- Revise the documents now!

Proof of Concept Demonstration

- 15 minutes total time per team
- Each team informally demonstrates
- Demos only for TA and instructor
  - Give team name and number, introduce members
  - No need to introduce the project itself
  - Show what is proposed in the development plan
  - You can change what is proposed, update dev plan
  - Required to show some running code
  - You will throw out this code
  - No slide shows — emphasis is the demo!
  - Fine to gather around a laptop screen for demo
- Rehearse your demo in advance
  - Know which device or devices you are using
  - Have the correct branch pulled
  - Practice what you will show us
  - Be prepared to deviate from your demo plan

POC Cont'd

- We'll ask questions
- Prepare to be interrupted
- We'll also look at commit logs
- We'll discuss potential changes to project scope
- Discuss extras
- We may discuss team communication issues
- Feedback may seem harsh at times
- If you have any concerns with your feedback, let the instructor know
- Grading following the rubric on Avenue
- PoC checklist

A Priori Feedback on POC Demos

- Update README files
- Give your repo/project a proper name
- Use issue tracker
- Use project board
- Use co-author commits
- Attendance in lectures, meetings
- Distribute work between team members

Verification Questions

- How might you verify usability?
- How might you validate the SRS?
- Should you list every testing technique you know?
- How do you do stress testing with a web-app?
- How might you verify the verification?
- Do you have to test external libraries?
- How would you verify a video game?
- How do you verify when there is no test oracle
- How might you verify your software isn't gender biased?

VnV Review

- Verification versus validation
- Not just testing
- Dynamic versus static testing
- Manual versus automated testing
- Regression testing
- Coverage metrics
- System testing
- Integration testing
- Unit testing
- Testing can show presence of bugs, not their absence
- Oracle problem
- Testing generally cannot prove correctness
- Random testing is often an inappropriate choice

VnV Review Continued

- White box versus black box testing
- Stress testing
- Model checking
- Fault testing to estimate number of bugs
- Stubs and drivers (test scaffolding)
- Rubber duck testing
- etc. L29–L34 from 2AA4 (among many possible resources)

VnV "Template"

- Suggest basing your document off our VnV Template
- Don't have to follow exactly, but follow rubric
- SRS Verification ideas
  - Review by teammates
  - Review by stakeholders
  - Review by supervisor
  - Review by classmates (already done)
  - Reviews can be ad hoc, or even better structured
    - Task based inspection
    - SRS walk through
    - Checklist
- Similar for design verification, VnV plan verification
- Implementation verification will involve dynamic and static techniques

VnV "Template" Continued

- Automated testing and verification tools — can point to your dev plan if you already covered this
- Software validation plan — how will you ensure that you got the right requirements?
- Can fill in system tests now, but details of unit tests will have to wait
- Specific details on each test
  - Control - manual or automatic
  - Initial state
  - Input
  - Output
  - Test case Derivation (justify expected output value)
  - How test will be performed

VnV “Template” Continued

- Nonfunctional requirements
  - Be specific!
  - Usability tests should include your survey questions (in the Appendix)
  - Tests may compare to other existing products
- Traceability table between requirements and test cases
- You will eventually show traceability between test cases and modules
- Self reflection appendix
  - What knowledge and skills will you need to acquire for VnV?
  - What options do you have to acquire these skills?
  - What option have you selected for acquiring each skill?
  - As for SRS reflection

VnV Rubric

- System tests specific! Make decisions!
- Tests should be aimed at trying to break the system
- Test cases are unambiguous
- Not just testing
- Not just code
- Plans for automation and tool use
- Learning outcomes
  - Selects appropriately from relevant knowledge base to plan appropriate data collection methods and analysis strategies
  - Reflects on one's own educational needs and opportunities for growth
- Available in Avenue
- Include plan for extras
- Include reflection

Sample VnV Plans

- Several examples are in our repo
- PyERT
- The samples overlap with our needs, but past students had a different template, different rubric