

SE491+CS494 Capstone Design Winter 2021

Prof Derek Rayside

April 16, 2021

Overview

drayside@uwaterloo.ca

NOTE: please include 'capstone' in email subject, or consider using your team's channel in MSTeams. Meetings: <https://calendly.com/derek-rayside/30min>

SE491 COURSE DESCRIPTION

Final implementation, testing, and communication of the design project started in SE390. Technical presentations by groups. Analysis of social, legal, and economic impacts. Final release of the project. Project retrospective.

<https://ucalendar.uwaterloo.ca/2021/COURSE/course-SE.html>

CS494 COURSE DESCRIPTION

Continuing from CS493, student teams continue development of their project, update project plans, explore design alternatives, perform testing, and analyze experimental results. Teams prepare and deliver technical presentations and demonstrations of their projects, and analyze ethical and legal aspects of their work.

<https://ucalendar.uwaterloo.ca/2021/COURSE/course-CS.html>

LEARNING OBJECTIVES The capstone project is intended to embody essentially all of the learning objectives of the undergraduate degree. Learning objectives have been identified by the Canadian Engineering Accreditation Board (CEAB), the Canadian Information Processing Society (CIPS), and the UWaterloo Software Engineering Curriculum Committee. These are listed in the *Handbook* §1.

Handbook §1

URLs & Repositories

Course Website	https://ece.uwaterloo.ca/~se_capstone/
Project metadata	https://git.uwaterloo.ca/secapstone/se491-2021 (<i>you need to clone this</i>)
Course Discussion	MS Teams (same as last term)
Meetings	https://calendly.com/derek-rayside/30min (<i>when in the meeting, click on Participants and invite Derek or he won't know that you are in the meeting</i>)

Contribute Cover Art!

Calling all artists! Please contribute your artwork for the cover of our class abstract booklet. We have used student artwork on the cover for the last few years.

See past abstract booklets on the course website.

Project Evaluation

Project evaluation is aligned with the learning objectives and is discussed in *Handbook §14*. Generally speaking, capstone projects should, in all relevant ways:

- exemplify the learning objectives;
- demonstrate the skills expected of a graduate;
- make appropriate trade-offs and judgments; and
- not suffer serious oversights.

A central premise is that the software should perform its intended function properly and in a unified way.

GRADE WEIGHTS are organized as follows:

<i>Facet</i>	<i>Weight</i>
Reflection (learning outcomes, feedback, impact, IP, etc.)	15%
Requirements & Specifications	15%
Design, Implementation, & Deployment	25%
Verification & Validation (testing & results)	25%
Teamwork	10%
Communication (abstract, demo, presentation)	10%

DEDUCTIONS FOR OVERSIGHTS are a possibility. Part of being a professional is knowing what needs attention, covering the bases appropriately, and not making serious oversights.

There will also be a list of some specific minor deductions for relatively small matters.

DEADLINES ARE ENFORCED BY DEDUCTIONS. All grades are assessed at the end of the term. But there are deadlines along the way. Because no grades are assessed at those intermediate deadlines, they are enforced by deductions. Deductions can be waived for extenuating circumstances or prior arrangements.

The deduction is 1% off the final grade for each day late.

Handbook §14

Handbook §14

Grades are generally assigned to teams rather than individuals. The *teamwork* grade may involve peer evaluation that produces some individual variation in final grades.

In extreme cases, the contributions of individual students might be assessed individually.

Three main strategies for avoiding oversights are *peer interactions*, *learning activities*, and *formative assessments*.

1% off the final grade for each day late.

DELIVERABLES & DEADLINES

Class reps can propose changes to the dates. The upload to the uw website is the only externally-imposed deadline.

<i>Deliverable</i>	<i>Deadline</i>	
meta.tex contents	January 31	
Web blurb draft (< 85 words)	January 31	
Abstract draft (~ 300 words)	January 31	
Initial status sheet	January 31	
Team picture(s) draft	January 31	
uw website upload (blurb + pics)	February 4	
Societal Impact Report draft	February 7	
Peer Activity #1	February 12	<i>changed from February 7</i>
Abstract revised (~ 300 words)	February 28	<i>must incorporate feedback</i>
Peer Activity #2	February 28	
Teamwork Quiz	March 7	<i>March 14 extension</i>
Teamwork Health Assessment	March 7	<i>March 14 extension</i>
Teamwork Process Assessment	March 7	<i>March 14 extension</i>
Teamwork Reflection	March 14	<i>March 21 extension</i>
Feedback to SE2022	March 21	<i>March 28 extension</i>
Peer Activity #3	week before video	<i>probably a practice talk</i>
Peer Activity #4	day before of video	<i>probably a practice talk</i>
Video of Presentation + Demo	April 17	<i>first day of exams, per class request</i>
Final abstract	same as video	
Final status sheet	same as video	
Teamwork feedback	same as video	<i>superseded by above</i>
Video viewing + Q&A	after video is submitted	

University Policies

Intellectual Property: UWaterloo has the (fairly unique) policy that intellectual property is owned by its creators (rather than by the university). The university has resources to help you commercialize your project (if desired), as well as local incubators such as Velocity.

<https://uwaterloo.ca/secretariat/policies-procedures-guidelines/policies/policy-73-intellectual-property-rights>
<https://uwaterloo.ca/secretariat-general-counsel/faculty-staff-and-students-entering-relationships-external>

Academic Integrity: In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility.

<http://uwaterloo.ca/academicintegrity/>

AccessAbility: AccessAbility Services collaborates with all academic departments to arrange appropriate accommodations for students without compromising the academic integrity of the curriculum. If you require academic accommodations, please register with AccessAbility Services at the beginning of each academic term.

<https://uwaterloo.ca/accessability-services/>

Grievance: A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

Policy 70, Student Petitions and Grievances, §4, <http://secretariat.uwaterloo.ca/Policies/policy70.htm>

Discipline: A student is expected to know what constitutes academic integrity to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (*e.g.*, plagiarism, cheating) or about rules for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean.

Appeals: A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals).

Reconciliation

We acknowledge that the University of Waterloo is on the traditional territory of the Neutral, Anishinaabeg and Haudenosaunee peoples. The University of Waterloo is situated on the Haldimand Tract, the land promised to the Six Nations that includes ten kilometres on each side of the Grand River.

<http://uwaterloo.ca/academicintegrity/>
For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, <http://secretariat.uwaterloo.ca/Policies/policy71.htm>
For typical penalties check Guidelines for the Assessment of Penalties, <http://secretariat.uwaterloo.ca/guidelines/penaltyguidelines.htm>

<http://secretariat.uwaterloo.ca/Policies/policy70.htm>
<http://secretariat.uwaterloo.ca/Policies/policy71.htm>
<http://secretariat.uwaterloo.ca/Policies/policy72.htm>

<https://uwaterloo.ca/arts/about-arts/territorial-acknowledgement>



Figure 1: Contemporary map of the original Haldimand Tract and the remaining Six Nations Territory (red).

Announcements

Weekly TA Meetings Can Be Short

Your weekly meetings with your TA can be short. The primary purpose of the meeting is to touch base to make sure that you are aware of course announcements and deliverables, and to identify any questions that you might have. Especially this year it is important for us to stay in touch.

Of course, your TA is also interested to have deeper conversations about your project with you.

January 31 Deliverables

GETTING STARTED

1. Look at the compiled abstract booklet on the course website.
2. Clone the course metadata repo.

https://ece.uwaterloo.ca/~se_capstone/
<https://git.uwaterloo.ca/secapstone/se491-2021>

META.TEX

1. Update your team's meta.tex file appropriately.
2. Submit a merge request.
3. There are mark deductions for not doing this on time.

ABSTRACT.TEX

1. Read the *Handbook* §13.9 section on revising your abstract.
2. Update your team's abstract.tex accordingly.
3. About ~300 words — fits on one page of abstract booklet.
4. Submit a merge request.
5. Get feedback on your revised abstract from your TA.
6. There are mark deductions for not doing this on time.
7. There are mark deductions for doing a negligently poor job.
8. You will revise your abstract at least one more time this term.
9. Your abstract will not be marked until the grading deadline.

Handbook §13.9

Grading deadline = when you submit your final presentation video, towards the end of the term.

BLURB-WEB.TEX

1. 85 word summary of your project that you will upload to the university website (next week — not this week).
2. Write this after you have revised your abstract.
3. Submit a merge request.
4. Discuss your blurb with your TA.
5. There are mark deductions for not doing this on time.
6. There are mark deductions for doing a negligently poor job.

...continued on next page ...

TEAM-PHOTO.JPG

1. This doesn't need to be a traditional team photo. Options include, but are not limited to:
 - traditional team photo
 - collage team photo
 - logo
 - project infographic
 - illustrative screenshot
 - architecture diagram
 - project-related art
2. Create a merge request.
3. Discuss with your TA.
4. There are mark deductions for not doing this on time.
5. There are mark deductions for doing a negligently poor job.
6. You can update this at any time until the grading deadline.
7. It will be evaluated as part of your communication grade.

STATUS-SHEET-1.PDF

1. Find a PDF program that lets you fill out the form and save it. Test out your candidate program before you get serious about filling out the form. Ensure that others can read the data you record in the form.
2. Use CLOC to measure lines of code. Not Git or wc.
3. Complete status-sheet-1.pdf.
4. Create a merge request.
5. Discuss your status sheet with your TA.
6. There are mark deductions for not doing this on time.
7. There are mark deductions for doing a negligently poor job.

The status sheets have been significantly revised since last term.

STATUS SHEET CLARIFICATIONS

- 2.4 Responses to Feedback: You will have several peer interaction activities this term (see syllabus), from which you should get some feedback. You could also put prior feedback from last term here.
- 2.5 Feedback Given: You will have several peer interaction activities this term (see syllabus), which will give you a chance to give feedback.
- 2.6 Societal Impact Report Highlights: You haven't written this report yet, so might not have much to say. But if you are seeking a patent, or trademark, or working with a vulnerable user population (*e.g.*, children, seniors), or something else that you know will be interesting here, then you could mention it now.

Thank you Team Beam!

The status sheet is a snapshot in time. You'll do another one at the end of the term. Some parts of the status sheet are more or less important for some projects. It is intended to be broad, covering everything that might be important to some project. Going over this breadth, you might discover something interesting for your project that you had not previously been focused on.

- 2.7 Graduate Attributes Self Assessment: Take a look at the handbook chapter on *Learning Objectives*, in particular the section *SE Curriculum Committee Intended Graduate Attributes*. Let's take Team Beam as an example. Their project is a distributed load tester. Let's consider the first line, Knowledge Base.

In the handbook we see that the SE Curriculum Committee has identified 7 intended graduate attributes here, but only deemed 3 of them as relevant to capstone projects. Those 3 are listed in this section of the handbook, and they are (paraphrasing):

- understand software systems
- know how to program
- apply discrete math

All of these are *relevant* to Team Beam's project. Team Beam has currently *achieved* the first two — but they haven't applied discrete math yet. So now applying discrete math is something they can put on their list of things to consider doing. Maybe they will decide that it's more valuable to apply statistics instead — and give some feedback to the SE Curriculum Committee that this intended graduate attribute should be rephrased to also include statistics. Or maybe they'll ultimately decided that other objectives are more important for their project and they won't apply any math. The purpose of this graduate attributes self-assessment has been achieved though: they have identified applying math as an intended graduate attribute relevant to their project that they have not yet done.

- 2.10 Process: Something about your team's process. Kanban? Sprints? Scrum? Some words describing how you work together.
- Process Alternatives. Other team processes that you are familiar with (perhaps via co-op) that you are choosing to not use on your capstone team for some reasons. For example, maybe you worked at Intel, which uses OKR, but you think that's too heavyweight for your capstone team.
- Process Assessment: This lists some well-known software engineering team process assessments. You could perhaps apply one (or more) of them to your team's process, to discover opportunities for improvement. **Scrum Checklist** by Henrik Kniberg should be on this list, and is probably the most relevant process assessment for most of your projects. But it is good for you to be familiar with all of these process assessments.
- 5.2 Summative User Activities: This might not be relevant to your project. For example, Team Beam's project is not particularly user-focused. But if you are in the New Product category and planning to show *results* by user activities, this box is for you.

<https://en.wikipedia.org/wiki/OKR>

<https://www.crisp.se/gratis-material-och-guider/scrum-checklist>

§5 here is *Verification & Validation*. This is where you describe your *results*. How do we know that your software actually does what you claim it does?

End of Term Presentations / Video

Historically we had a *symposium day* where everyone made live presentations in the same physical space on the same physical day, and judging happened right then and there. Back in those olden days, you used to attend physical lectures in physical classrooms. On co-op terms you would make presentations in the same physical room with your colleagues.

This year things are different. Now your lectures are videos. Now your co-op presentations are remote. So this year, instead of a live presentation, you will make a recorded video presentation. Recorded video presentations have suddenly become an important professional skill in the world of work.

This year we will have a deadline day towards the end of the term when your videos are due.

Let's chat about when this deadline should be.

Then, some time after that deadline, we will schedule a viewing and Q&A time that works for your team (and your selected referees, if any). Scheduling live symposium day with concurrent sessions is hard. We use a SAT constraint solver to do it. This year the scheduling will be a bit easier because we won't have to worry about scheduling conflicts between teams.

Societal Impact / Intellectual Property Report

Due February 7th

- Template is in your team directory.
- A compiled version of the template is on the course website:
https://ece.uwaterloo.ca/~se_capstone/society-report/
- Presentation from an IP lawyer is in Learn:
<https://learn.uwaterloo.ca/d2l/le/content/633126/viewContent/3524806/View>
- Most questions are boring for most teams.
- Some questions are very interesting for some teams.
- Discuss the interesting bits with your TA or instructor.

Peer Activity #1

You pick what you want your partner team to give you feedback on. Some examples might include:

Due February 12 — extended from February 7th.

The table of team pairings is in our MS Teams discussion.

- usability of your software
- code/architecture review
- social impact report
- selecting and designing user activities (which will then be carried out with actual users)
- feedback on your plan to achieve your results
- *etc.*

Free vs Open Source vs Public Domain

Some of your society/impact/IP reports have phrases like ‘as permissive as possible’ or ‘as open-source as possible’ or ‘nobody will own it.’ These phrases are too fuzzy. You need a better understanding of software licensing.

All software licenses depend on ownership of the source code under copyright law. Copyright law is the legal basis for software licenses. It is possible to relinquish ownership of the source code and release it under the public domain, but that is different than applying a license. Licenses have terms that need to be respected. Anyone can do whatever they want with things in the public domain.

The Free Software Foundation defines four freedoms. In some cases, it might be worthwhile to identify which of these freedoms you are trying to preserve.

0. The freedom to run the program as you wish, for any purpose (freedom 0).
1. The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
2. The freedom to redistribute copies so you can help others (freedom 2).
3. The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

The FSF also takes the position that free software should never be made unfree. The BSD/MIT style licenses are a bit more flexible on this point. Who wants to make software unfree? Companies. Can your startup take this source code, modify it, and keep those modifications closed-source (*i.e.*, unfree)? Can your competitor do this?

Phrases like ‘as permissive as possible’ are controversially interpreted differently by people who prefer GPL-style licenses *vs* BSD/MIT-style licenses because of this important distinction on whether the software can be made unfree in the future. You should be clear about what permissions you are trying to preserve and why.

<https://medium.com/@fokusman/the-easiest-way-to-check-all-your-npm-dependency-licenses-753075ef1d9d>

There exists EULA generators online ...

<https://www.ic.gc.ca/opic-cipo/cpd/eng/search/basic.html>

<https://patents.google.com>

February 10

Your report should reflect that you understand these concepts.

<https://www.gnu.org/philosophy/free-sw.en.html>

Note that the GPL does prohibit selling the code. It just says that the source code also needs to be available. Some companies make a business out of having the previous version of the code available GPL, but having a closed-source license for the current version (*e.g.*, Ghostscript). Or always having the source code available and selling support/consulting/customization (*e.g.*, BestPractical.)

Public Domain / CCo / Unlicense

Societal Impact Report Drafts Discussion

February 10

Some common themes and questions from a first quick review of your societal impact reports:

- Consider applying *Privacy by Design*, and especially such questions as:
 - Do you really need to store email addresses? Could you store cryptographic hashes of email addresses instead? Are you actually going to send email to that address?
 - How much user data do you really need to store?
 - Can users delete their data?
 - Can users download their data?
- Risks of optimizing for user engagement: Optimizing for anxiety?
- In some cases it might be worthwhile to look to the future and speculate about possible positive or negative outcomes. For example, when social networks started twenty years ago, it was not common to think that they were going to play a significant role in election outcomes.
- Some interesting reports for you to check out (alphabetical order). Being on this list does not mean that their reports are perfect, just that they have some interesting aspect for you to look at.
 - AutoDash
 - Bery
 - GeoCache
 - HCW
 - JKKody

Handbook §8.5

<https://www.ipc.on.ca/wp-content/uploads/resources/7foundationalprinciples.pdf>

https://en.wikipedia.org/wiki/Privacy_by_design

https://ece.uwaterloo.ca/~se_capstone/tmp/society-report/

We are still in the formative feedback stage of the course. Grades are assessed at the end. You still have the opportunity to revise and improve your own work based on what you learn from your peers. It is important that, by the time you graduate, you understand how the law applies to your work, and how your work impacts society.

Joke: what do you call the person who graduated last from medical school?
Answer: Doctor. Moral: we expect all professionals to have a baseline of competency in their field.

New Product User Engagement: Total User Minutes

February 10

A way to think about the New Product user engagement metric is in total user minutes. The rubric tries to express this concept, but evidently doesn't do a good job of it with the current phrasing.

For example, if you have 1000 users who each use the software for 5 minutes, then that's 5000 user minutes total. If you have 3 users who use the software as a main part of their full time job (say 30 hours per week), then that's 5400 user minutes per week, so in the same ballpark.

Also note that this rubric needs to be interpreted by an expert in context. Consider, for example, the team from SE2020 that made a smart pillbox for seniors. We want them to have zero users, because the ethical and medical risks of messing up granny's medicine are too high for coursework. It's an awesome project to pursue, but we aren't going to penalize them for having zero users. They should instead do user activities with their peers to test out the technology.

Some Notes on Grading and Software Engineering at Google

Feb 20

This note is primarily extra explanation based on student questions. It also provides extra information about *Teamwork* and *Communication* evaluation. Let's start this discussion with a quote from Google:

We can also say that software engineering is different from programming in terms of the complexity of decisions that need to be made and their stakes. In software engineering, we are regularly forced to evaluate the trade-offs between several paths forward, sometimes with high stakes and often with imperfect value metrics. . . . With those inputs in mind, evaluate your trade-offs and make rational decisions.

— Titus Winters, *Software Engineering at Google*, 2020

Titus Winters is the lead engineer for Google's C++ code base and co-author of the recent book *Software Engineering at Google*, published by O'Reilly. He met with Patrick and Derek and gave us a copy of the book to share with you, which you can find in LEARN.

A major theme of evaluation in the capstone project is for you to explain your engineering decisions, to explain how you have allocated your resources in the most appropriate way for your project, and why the things you have chosen to do reflect best practices and are better than alternatives. Let's revisit the grading outline in this light:

<i>Facet</i>	<i>Weight</i>
Reflection (learning outcomes, feedback, impact, IP, etc.)	15%
Requirements & Specifications	15%
Design, Implementation, & Deployment	25%
Verification & Validation (testing & results)	25%
Teamwork	10%
Communication (abstract, demo, presentation)	10%

THE STATUS SHEET is a broad *formative* assessment — not a summative assessment. It attempts to ask every question that might be relevant to any project — in a shallow manner. This is a tool to help you identify your blindspots and your balance of resource allocation:

- Are there questions that are worthy of your consideration but which have heretofore escaped your notice?
- Are you allocating your resources in appropriate balance for your project?

So the specific answer to any specific question on the status sheet isn't the main point. The point is in examining that answer in light of the objectives and needs of the project, and in comparison to answers.

Situation to avoid: a referee thinks that something is important for your project, and you ignored it — didn't even comment on why it's not actually important, or why it was less important than other things you did do.

One good preparation strategy is to gather formative peer feedback on your status sheet and your presentation.

REQUIREMENTS & SPECIFICATIONS (15%)

- Do you really know what your project's opportunity is? Is your view of this grounded in evidence beyond your own intuition?
- Do you really know what problem you are solving? Can you give it a clear definition?
- Have you applied appropriate software engineering techniques?

Again, the status sheet attempts to give a broad overview of common questions in this area. It's a structuring mechanism, to help you identify what's important and possible oversights.

DESIGN, IMPLEMENTATION & DEPLOYMENT 25%:

- How normal/novel is the design? Why is that appropriate?
- Fitness for purpose: Does the design/implementation/deployment meet the requirements? In principle? In practice? What is the evidence?
- Fitness for future: What changes are anticipated? Is the design intended to be flexible on these points? Why or why not?
- Accumulated technical debt? Does it matter?
- Are there significant discrepancies between the design and implementation? What is the evidence either way?
- Have the appropriate foundations been applied?

VERIFICATION & VALIDATION (TESTING & RESULTS) 25%:

- How should your effort be allocated between testing, user activities, and real world results? Are those even distinct categories for your project, or is there significant overlap?
- What are the appropriate techniques for testing, user activities, *etc.*, in your project? Can you explain (briefly) why those techniques follow best practices and/or are better than other good alternatives?

Have you done the TCPS2 training? (if appropriate)

Different projects will have different answers to these questions. For some projects, it might be a good and sensible choice to do no user activities. For other projects, user activities will be the main focus. The status sheet gives you (and your referees) a high-level summary of how you've allocated your effort here. Your presentation gives an in-depth explanation of what's important and why — you tell the referees how to interpret the data, and why your choices make sense.

REFLECTION 15%:

- learning outcomes
- peer feedback
- societal impact
- intellectual property
- lessons learned
- *etc.*

COMMUNICATION (ABSTRACT, DEMO, PRESENTATION) 10%:

- General criteria: clear, correct, complete, concise
- Appropriate application of abstract-writing guidance
- Appropriate selection of narrative structure for presentation
- Following presentation guidance from at least one identifiable source, such as: Jean-Luc Doumont (Trees, Maps, Theorems), Nancy Duarte, Edward Tufte, Patrick Henry Winston, TED.

Teamwork Assessment 10%

February 21

The teamwork assessment will have four components:

1. 4% Quiz in LEARN (~~March 7th~~ 14th)
 - A summative assessment of your knowledge of teamwork concepts, as described in §3 of the handbook.
 - Open book.
 - Mostly questions low on Bloom's taxonomy of learning: *remember* & *understand*. (Higher levels include *apply* and *evaluate*.)
2. 2% Team Health Assessment (~~March 7th~~ 14th)
 - The handbook has five different team health assessments.
 - Each individual will do two of them in LEARN:
 - One that the entire team agrees to do. (To have a *common* point of discussion for the upcoming team reflection. It is especially interesting if different team members give different answers to the same question.)
 - One that others are not doing. (To have a *unique* point of discussion for the upcoming team reflection. If your team has more than 4 members then there will inevitably be some overlap here, which is fine.)
 - A *formative* assessment.
 - Full marks for honest effort.
3. 2% Team Process Assessment (~~March 7th~~ 14th)
 - The handbook has five different team process assessments.
 - Highlights of them have been combined into one LEARN quiz.
 - Will be done in LEARN by each individual.
 - A *formative* assessment.
 - Full marks for honest effort.
4. 2% Team Reflection (~~March 14th~~ 21st)
 - Team reflects on their health and process assessments.
 - Discuss the answers you gave individually. Note differences and similarities. Discuss interesting points raised from the varied health assessments.
 - Submit in MSTeams channel files tab.
 - A *formative* assessment.
 - Full marks for honest effort.

<https://uwaterloo.ca/centre-for-teaching-excellence/teaching-resources/teaching-tips/planning-courses-and-assignments/course-design/blooms-taxonomy>

Final Presentation Discussion

April 4

- ~20 minute video presentation (17–23 minutes)
- Due April 17th.
- Upload to YouTube unlisted channel (details later).
- Public viewing with referees and live Q&A scheduled according to team and referee availability (April 26th is last day of exam period).
- Audience targets:
 - expert referee in the area of your project
 - SE alumni who have seen lots of other projects
- *Goal:* provide referees with the right information to properly assess your project according to the course marking scheme.
- *Status Sheet:* Referees will also have your final status sheet to review before your presentation, to help them have a broad (but shallow) overview of the project. The presentation brings focused depth in the important areas.

In the SE490 final presentations last summer there was often a disconnect between what was on the status sheet and what was in the presentation. That creates confusion and wastes time, and makes you look disorganized.

Course marking scheme (reproduced here for convenience):

<i>Facet</i>	<i>Weight</i>
Reflection (learning outcomes, feedback, impact, IP, etc.)	15%
Requirements & Specifications	15%
Design, Implementation, & Deployment	25%
Verification & Validation (testing & results)	25%
Teamwork (assessed by course staff outside of presentation)	10%
Communication (abstract, demo, presentation)	10%

PROJECT-ORIENTED MARKING rather than artifact-oriented marking. Most undergraduate courses have artifact-oriented marking: report is worth X, poster is worth Y, etc. But this is not a communications course: it is an engineering design course. Communication is an important aspect of the course, but it is not the whole thing. We are assessing your project. The artifacts that you produce along the way are tools that you use to communicate the project.

So you will not see societal impact report, status report, presentation, etc., explicitly in the marking scheme. It doesn't mean that those things aren't important. They are important tools to help you communicate your project. We are marking your project. The *reflection* mark, in particular, will draw on your societal impact report and the learning outcomes part of the status sheet.

THE STATUS SHEET IS A BROAD BUT SHALLOW ASSESSMENT. It asks almost every question that might be relevant to any project. Many questions will not be relevant for your project, or might not be phrased in a way that best aligns with your project. The purpose is to help you stand back and get a comprehensive view of your project. This serves as an important input to your presentation.

- obviously irrelevant — no further discussion necessary
- obviously relevant — elaborate in presentation as necessary
- hmmm? would a referee think this is relevant?
 - *misconception?* help the referee understand your project by (quickly) explaining why one might think X is relevant, but actually it isn't. See *near-miss* learning in AI.
 - *tradeoff?* Maybe it is actually relevant, but you had to make a prioritization tradeoff against other objectives. Let the referees know that you are not ignorant: you are explicitly exercising your professional judgment.

SEE ALSO: Note above from Feb 20 titled *Some Notes on Grading ...*

DEMO: The historic presentation videos do not include a demo of the software — the referees would have seen the demo live out at the poster booth before coming in to the presentation.

Your presentation videos may also include a brief demo.

PRACTICE TALKS / PEER ACTIVITIES. You have two more peer activities left in the term:

- Peer Activity #3, due 1 week before video (*i.e.*, April 10th)
- Peer Activity #4, due same day as video (*i.e.*, April 17th)
- Recommended that you use these for practice talks, but you can do something else with them if you want.
- You can choose the groups to do these with. You can repeat groups from previous peer activities if you want.

You want to avoid a situation where a referee asks: why didn't you think about X? X seems like an important and obvious thing. That is not a good situation. It is much better to proactively say: we considered X, but actually it's not relevant, or it was lower priority than Y, *etc.*

Similarly, you also want to avoid a situation where a referee asks: why are you using algorithm A with data D? Algorithm B is known to be better for data like D. Answer questions like this before they get asked.

Presentation Tips

April 9

- *Slide Numbers*: will aid discussion.
- *Pointing at Slides*: This is a challenge in the virtual world, but turns out to be an important part of your presentation strategy in some cases. Different people are using different techniques, none of which (so far) are great. Has anyone seen a prof who does this well? What technology is good for this? It is possible to use your slide program to highlight the parts you want to point at, but that can be a bunch of work and planning ...
- *Outline slide*: Should be unique to your presentation. Should not be able to be transplanted into someone else's talk. Items like "introduction", "problem", "solution", etc. are totally generic. Try adding a few words after the generic term that contextualizes it to your project.
- *Quantify if possible*. Do not say "we have lots of tests". Just measure it. The status sheets are intended to help you with this.
- *Avoid value judgments*. When you say "lots of tests", it's not only a failure to quantify, it's also a value judgment. Leave value judgments for the judges. Report the facts. Quantify. Compare your numbers to others, to standard metrics, to best practices, etc. Let the judge make the judgments. Both because that is your job (and not yours), but also because it's something they can argue with, which wastes discussion time. Report facts.
- *What is the point of each slide?* What is the thing the viewer is supposed to learn? The most common offender here are slides titled "experimental results" or something like that. What is the experiment showing? For example, instead of "empirical observations" you might say "birds can fly" for the title, then on the body document how you observed 73 species of local birds (including geese), and noted that all of them can fly. So, from your evidence (excluding penguins, emus, ostriches, etc), you conclude that birds can fly. This is not a value judgment: it's a conclusion from your evidence. Evidence is never perfectly complete, and you can talk about the limits of your experiments, etc.
- *Variable+function names*. It's important to pick good names for code samples. Not everyone needs code samples, but if you have them, pick good names. Avoid name collisions between variables and functions. Avoid (potential) name collisions between names and language keywords. And so on.
- *i.e. vs e.g.*: "e.g." means "for example". "i.e." means "that is" (as in, going on to clarify by saying the same thing in a different way).

More Presentation Tips

- *Handbook on Accessible Writing.* The links in the new handbook section on accessible writing might help you tighten up the phrasing on your slides. April 15
Handbook §13.11
- *Course Announcements.* Especially those from February 10th and 20th also give you guidance for the final presentation. Some people have recently asked me questions that were answered in those announcements.
- *Narrative structure:* How is your talk organized? Think about it. Handbook §13.7
- *Presentation guidance.* The course guidance on February 20th encouraged you to learn from the recommended sources of presentation guidance. Practice talks so far are showing opportunities to apply those skills. Handbook §13.1–5
- *Saying important things that are not on the slides.* Update your slides to cover the important things you are saying verbally but that don't have visual presence on slides. A lot of times this is around problem and opportunity identification. If your presentation outline has a list of three key concepts, then each of them should get at least one explicitly identified slide.
- *Give evidence for your claims.* For example, it's not enough to say 'we applied the CAP theorem.' Say how you applied it. The theorem says that engineers must make a tradeoff between consistency, availability, and partitioning. So what tradeoff did you make and why? This is just one example of many unsubstantiated claims in practice talks.
- *Use illustrations* where possible. There are many opportunities for illustrations in the practice talks so far.