Computer Science MS Project

CSCI-788 Section 01
Prof. Matthew Fluet

MoWeFri 11:15am – 12:05pm
GOL-1445 / Synchronous Zoom

COVID-19 Reminders:
► Wear a mask (covering nose and mouth)
► Choose an available seat (spread out, obey “do not sit” designations)
   ► Seat chosen today will be assigned to you for remainder of semester
► Complete location check-in (QR code)
► No food or drink during class
Computer Science MS Project

Scientific Method and Engineering Design\textsuperscript{1}

\textsuperscript{1}Adapted from slides by Joe Geigel.
Initial Thought for Today

Master of **Science** in Computer **Science**

What is “science”?

*Science is a systematic enterprise that builds and organizes knowledge in the form of testable explanations and predictions about the universe.*

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

Undertake your MS Capstone Project as a scientific endeavor.

- A means to undertake your project
- A means to approach writing a technical paper
- A means to approach preparing an academic paper

How is an MS Capstone Project different from a homework assignment?
Master of Science in Computer Science

It is true that, considered as a science, computing is difficult to categorize. The underlying theories – in particular, information theory and computability – appear to describe properties as eternal as those of physics. Yet much research in computer science is many steps removed from foundational theory and more closely resembles engineering or psychology.

J. Zobel, Writing for Computer Science, Ch. 3

MS Capstone Projects fall along a broad spectrum, from basic research to research-and-development to product development.

Let’s briefly consider two “approaches” along this spectrum.
Scientific Method

1. Ask a Question
2. Do Background Research
3. Construct a Hypothesis
4. Test with an Experiment
   - Troubleshoot procedure. Carefully check all steps and set-up.
   - Procedure Working?
     - No
     - Yes
5. Analyze Data and Draw Conclusions
   - Results Align with Hypothesis
   - Results Align Partially or Not at All with Hypothesis
6. Communicate Results

Experimental data becomes background research for new/future project. Ask new question, form new hypothesis, experiment again!
Engineering Design

Engineering Method

Define the Problem

Do Background Research

Specify Requirements

Brainstorm, Evaluate, and Choose Solution

Develop and Prototype Solution

Test Solution

Solution Meets Requirements

Solution Meets Requirements Partially or Not at All

Communicate Results

Based on results and data, make design changes, prototype, test again, and review new data.
Scientific Method vs. Engineering Design

Both contribute to the world of human knowledge:

▶ scientists: study how nature works
▶ engineers: create new things
Scientific Method and Engineering Design

Scientific Method

1. Ask a Question
2. Do Background Research
3. Construct a Hypothesis
4. Test with an Experiment
5. Procedure Working?
   - No
   - Yes
   - Troubleshoot procedure, carefully check all steps and set-up.
   - Analyze Data and Draw Conclusions
   - Results Align with Hypothesis
   - Communicate Results

Engineering Method

1. Define the Problem
2. Do Background Research
3. Specify Requirements
4. Brainstorm, evaluate, and Choose Solution
5. Develop and Prototype Solution
6. Test Solution
7. Solution Meets Requirements
   - Solution Meets Requirements Partially or Not at All
   - Communicate Results

Experimental data informs background research for new/future project. Ask new questions, form new hypotheses, experiment again!

Not mutually exclusive!
Most “real scientific/engineering work” integrates elements of both. Notice much similarity and correspondence between steps.
Scientific Method and Engineering Design

Where to begin?

▶ Ask a Question
▶ Define the Problem

In either case, should be specific, precise, and unambiguous.

Should also be testable:

▶ A negative result should be possible (even if unlikely)
▶ Scope of project may be limited to feasibly test
  ▶ Beware of drawing conclusions outside the tested limits/assumptions
▶ Even in engineering design:
  ▶ test that solution meets specification
  ▶ test that solution improves over alternatives
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

▶ Proof
▶ Model
▶ Simulation
▶ Experiment
▶ Implementation
▶ Case Studies
▶ User Studies
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:
▶ Proof
▶ Model
▶ Simulation
▶ Experiment
▶ Implementation
▶ Case Studies
▶ User Studies

Example: Merge sort is faster than insertion sort.
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

- **Proof**
  A formal, mathematical argument that hypothesis is correct (or incorrect).

- **Model**

- **Simulation**

- **Experiment**

- **Implementation**

- **Case Studies**

- **User Studies**

Example: Merge sort is faster than insertion sort.
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:
▶ Proof
▶ Model
  A mathematical description of the hypothesis, with a demonstration of the correspondence between them.
▶ Simulation
▶ Experiment
▶ Implementation
▶ Case Studies
▶ User Studies

Example: Merge sort is faster than insertion sort.
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

- Proof
- Model
- Simulation
  A (partial) test of a simplified form of hypothesis, with omissions or approximations and/or tested with artificial data.
- Experiment
- Implementation
- Case Studies
- User Studies

Example: Merge sort is faster than insertion sort.
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

▶ Proof
▶ Model
▶ Simulation
▶ Experiment
  A (full) test of hypothesis, with no omissions or approximations and tested with real data.
▶ Implementation
▶ Case Studies
▶ User Studies

Example: Merge sort is faster than insertion sort.
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

- Proof
- Model
- Simulation
- Experiment
- Implementation
  - Software or system built to solve problem, demonstrated to meet the requirements.
- Case Studies
- User Studies
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

▶ Proof
▶ Model
▶ Simulation
▶ Experiment
▶ Implementation
▶ Case Studies
   A special kind of experiment that involves applying hypothesis/solution in a (small) number of realistic examples.
▶ User Studies
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

- Proof
- Model
- Simulation
- Experiment
- Implementation
- Case Studies
- User Studies

A special kind of experiment that involves human subjects. (Note: Speak to advisor early if you need to conduct one.)
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

▶ Proof
▶ Model
▶ Simulation
▶ Experiment
▶ Implementation
▶ Case Studies
▶ User Studies

(Distinction between some of these is often blurry.)
Evidence

Convince yourself (and others!) that hypothesis answers the question or solution solves the problem:

- Proof
- Model
- Simulation
- Experiment
- Implementation
- Case Studies
- User Studies

Non-evidence

- Intuition
- Opinion
- Isolated or Non-Representative Examples
Measurement

Most projects will need to take measurements to be used as evidence:

- execution time
- precision and recall
- lines of code
- user opinions
- ...

Measure what is important, not (only) what is easy.

- Most projects have a *qualitative* aim
  (e.g., make something “better”/“faster”/“easier”)
- Measurements are *quantitative*
  (i.e., reduced to a number/quantity)

Also see *WfCS* Chs. 14 (Experimentation) and 15 (Statistical Principles)
Questions (to keep in mind)

▶ What are the area, topic, and question / problem of your project? (How are these concepts distinct from each other?)
▶ Is your project of the correct scale to be completed in one semester, matching your skills and interests?
▶ What skills will you need to develop?
▶ What resources will you need to acquire?
▶ What forms of evidence will you gather or produce?
▶ How will you measure and analyze or evaluate the evidence? Distinguish between qualitative aims and quantitative measures.
▶ How will you recognize success? failure? What are the consequences of each?
▶ What are the potential obstacles and challenges?
▶ What are the important milestones to complete your project?
Consider the last few steps.

What to do with “negative results”?

- Results align paritially or not at all with hypothesis; Solution meets requirements partially or not at all
- Always communicate results!
- Still contributes knowledge
Consider the last few steps.
What to do with “negative results”?

- Results align partially or not at all with hypothesis; Solution meets requirements partially or not at all
Consider the last few steps.
What to do with “negative results”?

- Results align partially or not at all with hypothesis; Solution meets requirements partially or not at all

*Always* communicate results!
- Still contributes knowledge
Academic/Technical Papers

Most papers follow a structure that (somewhat) aligns with scientific method and/or engineering design:

▶ Introduction
▶ Background
▶ Body (most variation)
  ▶ Hypothesis / Requirements
  ▶ Evidence / Description
  ▶ Analyze Data / Evaluate Solution
▶ Conclusions
▶ Future Work
▶ Acknowledgements
▶ References
▶ Appendices (much more common in MS Capstone reports)

*The written work rests on a program of activity that begins with interesting questions and proceeds through a sound methodology to clear results.*

*J. Zobel, Writing for Computer Science, Ch. 3*
Introduction

Every paper answers a question or solves a problem

- Describe *what* the question / problem is
- Describe *why* the question / problem is important
- Describe (at a high-level) the *contribution*:
  - core idea that is novel
  - methodology employed to validate

A preview of the main body and conclusions of the paper.

The Introduction sets the rules of the game:

- A good paper is often good because it delivers on its promise: “We will show XYZ” and, indeed, XYZ is shown in the paper.
- A bad paper is often bad because it fails to deliver on its promise; easily disappointed by a paper that promises much and delivers little.
Background

- No question / problem or methodology is ever brand new
- Almost all work builds on existing work
  - Reference and briefly describe this other work
  - Give credit where credit is due

- Sometimes includes a longer, more detailed description of the *what* and *why* of the question / problem
  - All the details that the reader needs to know to understand the contribution
  - Might introduce a motivating example

- Sometimes includes a detailed analysis of related work, especially if this work directly builds on previous paper(s) or if this work directly addresses limitations of previous paper(s)
  - Interestingly, in my sub-field (Programming Languages), it is most common to have an explicit Related Work section immediately before the Conclusions section.
Body

Tell the story of your work.

*The body of a good paper — everything between the introduction and the conclusions — should have a logical flow that has the feel of a narrative. A strong paper has a story-like flow, with a sequence of concepts building from a foundation of knowledge assumed to be common to all readers up to new ideas and results. Thus an effective paper educates its readers. It leads readers from what they already know to new knowledge you want them to learn.*

*J. Zobel, Writing for Computer Science, Ch. 3*

Because every project is different, the body of each paper is a little different. Usually comprised of three or more sections, each telling a different chapter of the story.

How you present your project is different from how you executed it.
Body

Tell the story of your work.

Because every project is different, the body of each paper is a little different. Usually comprised of three or more sections, each telling a different chapter of the story.

How you present your project is different from how you executed it.

Many different organization strategies. Typically:

- the “key ideas”
- followed by “technical meat”
- followed by “evidence and analysis”

In “quotes”, because never actually title sections with these terms.

More in a few weeks; for now, make progress on your project.
Conclusions / Future Work

Conclusions
▶ Based on the analysis / evaluation, what can be concluded:
   ▶ Was the hypothesis confirmed? Did the solution work?
   ▶ Be honest about any limitations with analysis / evaluation.

Future Work
▶ No work is ever complete:
   ▶ What planned work was not completed (and why).
   ▶ New questions / problems identified (especially by negative results).
   ▶ Limitations that could be addressed.
   ▶ Invitation to others (and yourself) to continue the work.
Acknowledgements / References / Appendices

Acknowledgements

- Thank the supporters of the work (including funding agencies)

References

- Proper citation of work is essential

Appendices (much more common in MS Capstone reports)

- Additional supporting evidence that would detract from main narrative:
  - proofs of supporting lemmas
  - per-benchmark execution time statistics
  - full details of user surveys (e.g., included text or images)
Academic/Technical Papers

Most papers follow a structure that (somewhat) aligns with scientific method and/or engineering design:

- Introduction
- Background
- Body (most variation)
  - Hypothesis / Requirements
  - Evidence / Description
  - Analyze Data / Evaluate Solution
- Conclusions
- Future Work
- Acknowledgements
- References
- Appendices (much more common in MS Capstone reports)

*The written work rests on a program of activity that begins with interesting questions and proceeds through a sound methodology to clear results.*

*J. Zobel, Writing for Computer Science, Ch. 3*
Presentations

Multiple presentations during semester:
- Project Description (8min + 2min questions)
- Milestone Presentations (× 3; 5.5min + 1.5min questions)

Why present work in progress?
- Practice communication skills
- Receive feedback and suggestions
Presentations

General advice:

▶ Note that these presentations are short (5-10min)
  ▶ Think carefully about what is essential to present and what can be omitted
  ▶ Explain what the work is about, but not the details of the work itself

▶ Prepare slides (PowerPoint, Keynote, Beamer, Google Slides)
  ▶ Approx. 1 slide per minute of talk (plus title)
  ▶ Simple is best (avoid unnecessary fonts, colors, animation, . . .)
  ▶ Content should be brief, but not meaningless
  ▶ Pay attention to line breaks and never break words between lines
  ▶ Use figures when appropriate (“a picture is worth a thousand words”)
Presentations

General advice:

▶ Delivery
  ▶ Speak slowly (approx. 100 words per minute) and clearly
  ▶ In-person: make eye-contact; Remote: look into camera
  ▶ Never read slides to the audience
    ▶ Audience can read faster than you speak
    ▶ Text on slides should be reminders of topics to discuss
  ▶ Give audience time to think
  ▶ Strive to be natural

▶ Preparation
  ▶ Rarely helpful to prepare a transcript of full presentation
  ▶ Rehearse the presentation
    ▶ Check time (of full presentation and individual sections)
    ▶ Know what to say and how to say it (but don’t recite from memory)
    ▶ Know what will be on the next slide
Project Description Presentations (Weeks 3 - 4)

- 8 minutes (+ 2 minutes questions)
- For a general computer science audience
- Title: Project title, student, faculty advisor
- Summary of project
  - Motivation:
    Why is this topic (bigger picture, long term) and project (narrower focus, short term) important?
  - Concrete Example(s):
    Give an instance of the problem/solution or "vision" of working system.
  - Proposed Work: What is the (ideal) final deliverable?
  - Results and Evaluation:
    What evidence will be gathered? How will success be measured?
- Background and Related Work
- Briefly list milestones
Project Description Presentation (Weeks 3 - 4)

▶ Presenters
  ▶ Prepare slides (PowerPoint, Keynote, Beamer, Google Slides), but must submit to myCourses in PDF format (Sep. 2 (Mon) @ 8:00AM)
    ▶ In-person presentations will be given from instructors laptop
    ▶ Remote presentations will share screen (but must still submit slides)
  ▶ 8 minutes (+ 2 minutes questions)
    ▶ Practice and time presentation

▶ Audience
  ▶ Live feedback (questions and suggestions on project)
  ▶ Peer review of presentations
    ▶ Simple surveys on myCourses
      to be completed immediately after presentation
    ▶ Survey results will be shared (anonymously) with presenter
Research Paper Review (due Sep. 9 (Mon.) @ 5:00PM)

- Read a research paper relevant to your project
  (Note: you may not choose a survey paper for this assignment.)
- Write a one- to two-page critical review:
  - Summary
    - For a general computer science audience
    - Convey all the main ideas and contributions
    - Will necessarily omit many/all technical details
    - Almost entirely the “what” and the “why”, almost none of the “how”
    - Use your words; do not quote or repeat authors’ description of the work
  - Analysis
    - Originality: Does this paper present novel work?
      Do the authors properly relate their work to previous results?
    - Significance: Does this paper present a significant advance to the field?
    - Technical quality: Is the work technically correct?
    - Presentation: Is the paper well-written? Is there anything that could be added? Is there anything that could be omitted?
- Also see WfCS Ch. 3 and Exercises 4–5 (pp. 268–269)
A successful reader can identify the contributions and value of a paper, while recognizing its flaws; and uses critical scrutiny to identify the extent to which the flaws in a paper are serious.

Contribution is the main criterion for judging a paper. In broad terms, a paper is a contribution if it has two properties: originality and validity.

- The originality of a paper is the degree to which the ideas presented are significant, new, and interesting. When evaluating the significance of a contribution, it is helpful to consider its effect, or impact: that is, to judge how much change would follow from the paper being published and widely read.

- The validity of a paper is the degree to which the ideas have been shown to be sound. Good science requires a demonstration of correctness, in a form that allows verification by other scientists.

J. Zobel, Writing for Computer Science, Ch. 3
Research Paper Review (due Sep. 9 (Mon.) @ 5:00PM)

- Read a research paper relevant to your project
  (Note: you may not choose a survey paper for this assignment.)
- Write a one- to two-page critical review:
  - Summary
  - Analysis
- Also see WfCS Ch. 3 and Exercises 4–5 (pp. 268–269)

A successful reader can identify the contributions and value of a paper, while recognizing its flaws; and uses critical scrutiny to identify the extend to which the flaws in a paper are serious.

J. Zobel, Writing for Computer Science, Ch. 3

Note: You will be reading peer reviewed and published research papers, so it is unlikely that the papers will have serious flaws; nonetheless, all research has limitations.
Next steps

Schedule:
- Friday (no class): Work on Project Description Presentations
- Project Description Presentations (Weeks 3 - 4 (class each day))
  - Project Description Presentation due Monday (Sep. 2 @ 8:00AM)
  - Schedule of presentations to be posted on Friday
- Research Paper Review due next Monday (Sep. 9 @ 5:00PM)