CS846 Advanced Topics in Software Engineering (0.50) SEM Requirements
Engineering (RE) Spring/Summer 2019

Instructor: Daniel M. Berry (dberry@uwaterloo.ca)

Class Meetings: Tuesdays and Thursdays, 13:00–14:50, DC2568,
but for only 36 class-hours over about 18 of the 24 sessions
so that it ends up being equivalent to a normal course.

Course Web Site: http://se.uwaterloo.ca/~dberry/ATRE/

The Reason For the Strange Schedule

The School of Computer Science asked me to teach this graduate seminar in Spring to shift the course from a term
with too many graduate courses to one with not enough. I explained that I normally travel a lot in the Spring, includ-
ing up to two conferences. Therefore, I would be gone about half the weeks of the term. I told them that I could
do it only if we could meet longer per time, but for only fewer times. They agreed.

Course Structure

I will give most the first few lectures, two based on my famous Requirements Iceberg slides, describing RE and its
research, and several based on research that I and colleagues have done. Occasionally, we may have a guest lec-
turer. Besides giving an overview of RE, these lectures will show some experiments, case studies, and research
topics. The rest of the lectures will be given by students, reporting on what they are doing. We might use part of a
meeting to conduct an experiment for a project. I hope that you will agree to be a subject. Any such experiments and
at least watching them being done in class is considered part of this course.

Course Requirements

Each student will either

1. do some project, e.g., an experiment with volunteering subjects,
2. do some research, e.g., solving some open RE problem,
3. explore RE issues in your own research area,
4. do a case study, e.g., describe an RE experience at some job in the past, or even now, from which we can
   learn something, or
5. research a topic, i.e., read a bunch of research papers on a topic,

and then write it up and present it to the class in a 25-minute (the standard conference presentation time) presenta-
tion. This year, because we are fewer students, we may double the presentation time.

Any of these could end up being the basis for a master’s essay or a master’s or doctor’s thesis.

To avoid potential problems with a too-small or a too-big project, study, or topic, I strongly suggest running it by me
before going too deeply in it. Also if more than one want to work on the same project, study, or topic, we can ensure
that there will either be cooperation or no overlap.

To encourage people to volunteer to present early, let it be known that the earliest presentations will be evaluated
Initial Reading Assignment

Please read four papers that give overviews of RE and describe one experiment: Please go to the library site so that you are allowed to download these files for free.

1. “Requirements engineering in the year 00: a research perspective” by Axel van Lamsweerde at ICSE 2000 in the IEEE Xplore site:
   http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=870392

   (This is publically available, and you don’t have to be at the library site to get this.)

3. “Research Directions in Requirements Engineering” by Betty H.C. Cheng and Joanne M. Atlee at Future of Software Engineering (FOSE’07) Workshop at ICSE 2007 in the IEEE Xplore site:
   http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4221627


Course Themes

The two overriding themes of the class are:

1. requirements engineering: everything covered in the course must be related to requirements engineering.

2. empirical evidence: try to find empirical evidence for the claims being made about requirements engineering and its artifacts, methods, tools, etc.

A Nice Case Study Topic

A nice topic for a case study is anecdotal evidence that up-front RE pays off big:
RE research has shown how useful RE is for clearing up problems in a system’s concept before implementation begins so that implementation proceeds quicker and with fewer problems. However, few managers really believe this in their guts. When a deadline looms, most managers will move on to coding rather than finish RE, because they feel that the sooner they start coding, the sooner they will finish coding. A good case study gathers anecdotal evidence from a real-life software development showing that when the crunch comes, finishing the RE is a better choice than plunging directly into coding.