Research Methods and Software Engineering Workshop for Computational Humanities and Knowledge Systems

BIO/HPS 498/591, CSE 598 — 3 Credit Hours

Professor Manfred Laubichler
President's Professor, School of Life Sciences/Center for Biosocial Complexity
mlaubic@asu.edu

Julia Damerow, PhD
Scientific Software Engineer, Nexus Lab
jdamerow@asu.edu

Erick Peirson, PhD
Postdoctoral Research Associate, Office of the University Provost
erick.peirson@asu.edu

Course Description

Computer technology is changing the way that we represent knowledge and generate meaning in the 21st century, creating new and exciting opportunities for contact between computer science and the humanities. This course brings together graduate and advanced undergraduate students from the humanities and computer science for a hand-on, intensive workshop that advances methods and tools for computational research in the humanities. Computer science students will develop advanced competencies in software development principles and practices while carrying out research and development projects as part of an interdisciplinary team. Humanities students will learn new methods for computational research, gain experience in advanced computer technologies, and participate in the software development process. This course is an opportunity for CS graduate students to develop advanced skills in an applied research setting, and a stepping-stone into computational humanities research and software development at ASU and beyond.

Course Objective & Goals

This course is designed to provide hands-on interdisciplinary research and software development experience for graduate and advanced undergraduate students in computer science and the humanities. Students will gain experience conducting team-based interdisciplinary research and software development. Graduate students will explore and develop an in-depth knowledge of a specific topic of their choosing (e.g. NoSQL databases, semantic web, network analysis/visualization), which will prepare them to carry out independent research in computer science or computational humanities. Humanities students will gain knowledge regarding the design and execution of digital humanities research projects, as well as experience in the software development process. Computer science students will work on bigger software development projects that consist of several parts. They will gain hands-on

experience in an agile software development environment, and will learn technologies such as software frameworks (e.g. Django or Spring) and Continuous Integration software (e.g. Jenkins or Travis CI). We will conduct continuous code reviews to ensure high code quality and improve the students' coding skills. Instruction will be tailored to the interests and goals of the students as the semester progresses.

Course Structure & Mechanics

This is a project-based workshop/laboratory course, with all credit hours dedicated to in-class activities. Students will attend three x three-hour workshop sessions per week, for a total of nine hours per week. There will be no additional homework or other activities outside of class. Students will attend the Monday morning session (A), one B-session (Tues or Weds), and one C-session (Thurs or Fri).

Monday (Session A):	9am–12pm
Tuesday (Session B):	1:30pm–4:30pm
Wednesday (Session B):	9am–12pm
Thursday (Session C):	1:30pm–4:30pm
Friday (Session C):	9am–12pm

Covered Topics

- Agile Software Development (especially Scrum)
- Version Control Systems (especially GitHub)
- Documentation

Computer Science Specific

- Object-Oriented Programming and Design
- Software Design Patterns
- Software Architecture
- Software Testing
- Continue Integration
- Software Frameworks (for example Spring, Rails, Django)

Digital Humanities Specific

- Digital Collections, Repositories, & Metadata
- Introduction to Text Mining
- Digital Data Management
- Network Analysis & Visualization
- Intro to Scripting Languages (e.g. Python, Ruby)

Course Outline

The first two weeks of the course will completely be comprised of workshops. We will cover topics such as agile software development, version control systems, and quality assurance. After the first two weeks, students will be separated into groups that each work on a specific

project. Each group will have certain goals regarding software features and research outcomes that should be achieved by the end of the semester.

There will be a workshop every Monday morning. In those workshops we will discuss the topics listed above. Workshops will be split into two intertwined tracks: one emphasizing software development topics, the other one emphasizing digital humanities topics. The rest of the week will be used for team-based project work.

The last one or two weeks at the end of the semester will be used for project presentation. Each group is expected to present their project and results to the class. Graduate students will give short presentations describing their focal research area.

Prerequisites

Instructor permission required.

Computer Science Students (CSE 598)

It is expected that undergraduate students from computer science are comfortable programming in an object-oriented language such as Java or C#. A basic knowledge of software design patterns, testing principles, and concepts such as REST are required.

We require graduate students to have a good understanding of software engineering principles such as software design patterns or testing. They should be comfortable using frameworks such as Spring or Django. It is expected that graduate students will make independent intellectual contributions to the software development projects.

Humanities Students (BIO/HPS 498/591)

Open to upper-level (Junior or Senior) undergraduates and master's students in the humanities. Previous experience with digital humanities is not necessary, but an adventurous spirit is required! Graduate students will be expected to make independent intellectual contributions to computational humanities research projects.

Selection of Projects

Students can choose from a variety of ongoing computational humanities projects, which we will present at the beginning of the semester. Here are just a few examples:

- **Quadriga** (http://diging.github.io/quadriga/)
  Quadriga is a Java web-application for storing and managing text annotation projects. Students can choose to work on the extension of its functionality to connect it to specific repository solutions and its user interface to visualize annotation networks. CS graduate students can choose to focus on the implementation of graph transformation mechanisms.
- **Conceptpower** (http://diging.github.io/conceptpower/)
  Conceptpower is an authority file service based on WordNet. It can be used to uniquely identify concepts in text annotations or metadata entries. Students would work on

implementing a federation functionality to combine different Conceptpower instances to allow users to query several instances at once and to connect other authority services such as VIAF or Geonames. CS graduate students can choose to focus on implementing algorithms that decide if two entries are the same to improve query results from Conceptpower instances with overlapping entries.

- **Tethne** (http://diging.github.io/tethne/)
  Tethne is a Python module for bibliographic network analysis. Students can choose to implement a web application that allows users to run bibliographic network analysis online rather than on their own computers. CS graduate students can choose to focus on implementing an efficient storage/retrieval mechanism for sparse matrices.
- **Knowledge Architecture for History of Science**
  This project seeks to transform existing historical knowledge in the secondary research literature, in the form of stylized prose, into computationally useful data structures. This involves applying a variety of text-mining and machine-learning approaches to history of science journals and other texts.
- **Genecology Project** (http://devo-evo.lab.asu.edu/?q=genecology-project)
  The Genecology Project is a collaborative, interdisciplinary, student-run investigation of the agro-ecological field of genecology, also known as experimental taxonomy, in post-WWII Britain. We use computational methods in conjunction with archival materials, oral histories, and social network analysis to reconstruct and contextualize the shifting patterns of collaboration and discourse among plant ecologists who engaged in genecological research.

Evaluation & Attendance

- In-class exercises: 45%
- In-class participation: 45%
- Final presentation: 10%

Participation includes being on-time to all workshop sessions. You are expected to arrive in time to begin our workshop sessions promptly at their scheduled times, and arriving late will negatively affect your final grade. You are allowed to miss two classes without explanation, but we ask that you let us know at least 24 hours in advance (except in emergencies). If you miss more than 2 classes without our approval, it will negatively affect your final grade.

Disabilities

If you have a learning, sensory, or physical disability and feel that you need special assistance in lecture, reading, or assignments, please contact us immediately. The first week of the semester is the best time for these discussions. All matters will be held in strictest confidence.

Academic Integrity

The ASU Student Academic Integrity Policy (i.e. Honor Code) applies to all aspects of this course. This policy includes, but is not limited to, doing all your own work on assignments and appropriately acknowledging all sources used in written work. Refer to the ASU Student

Academic Integrity Policy if you have questions, or talk with us. Remember that it is the responsibility of the student to clarify ambiguous situations. In other words, if you are not sure, ask us. For more information, see: https://provost.asu.edu/index.php?q=academicintegrity