

1. Course number and name

CptS 423: Software Design Project II [CAPS]

2. Credits and contact hours

3 credits, 3 lecture hours

3. Instructor's or course coordinator's name

Aaron Crandall

4. Textbook, title, author, and year

M. Poppendieck and T. Poppendieck. 2006. *Implementing Lean Software Development from Concept to Cash* (1st ed.). Addison-Wesley. ISBN: 978-0321437389. (Optional)

A. Page, K. Johnston and B. Rollison. 2008. *How We Test Software at Microsoft*. Microsoft Press. ISBN: 978-0735624252. (Optional)

R. Pressman. 2005. *Software Engineering: A Practitioner's Approach* (6th ed.), McGraw Hill. ISBN: 978-0071238403. (Optional)

Berkun. 2008. *Making Things Happen: Mastering Project Management (Theory in Practice)*, O'Reilly Media. ISBN: 978-0596517717. (Optional)

P.C. Jorgensen. 2008. *Software Testing: A Craftsman's Approach* (3rd ed.). Auerbach Publications. ISBN: 978-1466560680. (Optional)

Other supplemental materials

IEEE Standards for Software Engineering

Scott Berkun's Blog on software engineering and project management addressing creativity, leadership, philosophy, and speaking:

<<http://scottberkun.com/blog/>>.

5. Specific course information

- a. *Catalog description:* Laboratory/group design project for large-scale software development, requirements analysis, estimation, design, verification techniques.
- b. *Prerequisites or corequisites:* CPT S 421 with a C or better; certified major in Computer Science, Computer Engineering, Electrical Engineering, or Software Engineering.

6. Specific goals for the course

By the end of the course, students will be able to

- Identify, formulate, analyze and solve complex computing and software engineering problems by applying principles of engineering, computing, science, mathematics, and other relevant disciplines (1a-e).
- Design, implement and evaluate computing solutions that meet specified requirements with consideration of public health, safety, and welfare concerns, as well as global, cultural, social, environmental, and economic factors (2a-g).
- Communicate effectively with a range of audiences in a variety of professional contexts (3a-f).

- Recognize ethical and professional responsibilities in software development and make informed judgments based on legal and ethical principles, and with consideration of broader impacts (4a, b, d, f).
- Function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (5a-g).
- Apply appropriate computing and engineering approaches, theories, and fundamentals to conduct appropriate experimentation, analyze and interpret data, use engineering judgment to draw conclusions, and produce solutions (6a-d).
- Acquire and apply new knowledge as needed, using appropriate learning strategies (7a-g).

7. Brief list of topics to be covered

- Project management
- Engineering solutions for real world open-ended problems
- Communication with clients, mentors, teammates, and collaborators
- Team-based software development
- Tools for software development and testing
- Collaboration tools for professional software development such as Git
- Writing technical documents
- Product packaging and delivery to clients
- Job search processes, developing contacts, and communication