Abstract - Any credible software engineering program must thrive to connect college education with the realities of commercial software development. In most cases, the study of software processes takes place in the lecture halls but it is not practiced in student projects. Students graduate into the commercial world driven by software and business processes without any hands-on experience and thus without an in-depth understanding of the significance of a process to the industrial setting. In order to close this gap, we have chosen to use a commercial grade process framework: Rational Unified Process® (RUP) for the backbone of our newly introduced software engineering capstone project. The capstone project is required for all students and offered through the senior year of our ABET-accredited BS in Computer Science program. In our opinion, this will prepare students better for the competitive job market.

Index Terms – Software process, Capstone project, Rational Unified Process, RUP

PROJECT DESCRIPTION
Starting in the fall of 2007, a two-semester capstone project replaced a required classical one-semester software engineering course. Previously semester-long projects conducted in this course utilized agile-type development by self-organizing student teams working for campus-based customers. Projects were successfully completed but the students’ learning experience did not match the reality of software industry. The implementation of the capstone project was based on an idea of a software enterprise conducting projects for corporate customers [2] [3]. The project acceptance process involves the preliminary feasibility evaluation and scope negotiation between the instructor and the customers. The first week of the semester, students submit their resumes via a learning portal. Instructor selects team leads based on their qualifications to manage the available projects. Team leads are provided with the project description and, through resume reviews and interviews, select the team members for their projects. Conflicts in the selection process are resolved by the leads through negotiations. Each team consists of three to four students and no migration between teams is allowed through the project’s duration.

Due to the fact that the capstone project replaced the existing software engineering course, the teaching of software engineering material had to be integrated into the fabric of hands-on software development. The project became the driving factor for the delivery of course material in the lecture portion of the course. Such a shift into the project-centric teaching of software engineering required not only the redesign of the contents and order of the material but also a careful selection of a more rigid software process as the backbone for the projects. The reorganization of the course material to meet the needs of the project involved moving topics such as risk management and software architectures to the early parts of the course. Topics normally covered in the early chapters of the software engineering textbooks, including software process models, had to be shifted to the later part of the course. These challenges were negligible. The major challenge was in the selection and customization of the software process to replace the agile-type approach previously used by the student teams. After careful consideration, Rational Unified Process (RUP) [4] was chosen as the process framework for our capstone project. RUP is widely accepted in software industry, is well-defined, and very well-documented. It promotes requirements engineering based on use cases (our students are well versed in Unified Modeling Language) and it can be customized with respect to the level of ceremony. RUP is role-based and defines the process in terms of four phases (inception, elaboration, construction, and transition). Each phase can be further structured into several iterations. The roles are very well defined and mapped to specific tasks/activities in the four phases of the process. Artifacts are linked to specific roles and activities organized into well-structured work flows [1] [5] [6]. The RUP framework provides an adequate set of artifact templates.

PROJECT STATUS
Using the RUP framework for software engineering projects was first tested in a one-semester software engineering course in the fall semester of 2007. The current academic year of 2007/2008 is the first year of phasing-in the new two-semester capstone project. Only three of eight teams were able to work on two-semester projects. The remaining teams consisted of students, who in order to graduate, needed the phased-out software engineering course but were not able to accommodate the new two-semester capstone project. In the next academic year of 2008/2009, all students will take the capstone project as part of their program of study.
PROJECT’S EXPECTED OUTCOMES

The main expected programmatic outcomes of using RUP in the capstone project are:

- Improvement of the product (software) quality and increase in the overall project success rate
- Improvement of customer satisfaction leading to even more challenging projects being commissioned by software companies
- Further convergence of the “virtual reality” of our BS Program with the reality of the software industry

The specific educational outcomes for students participating in the project are:

- To utilize project artifacts for improved, effective communication with all stakeholders
- To organize project’s activities into workflows linked to schedules and artifacts
- To better assess and mitigate the risks (in feasibility analysis and then through the entire project)
- To better integrate risk management and quality management into the software development process
- To learn resource management in context of a software process
- To learn about the value of compromise in project management
- To gain an appreciation of software process and its significance in developing a quality product

PROJECT EVALUATION PLAN

The capstone project improvement process is part of our continuous curriculum improvement. The current goal is to evaluate the impact of using the RUP framework in the capstone project. The main questions are:

- How did the team use the RUP framework in the entire project cycle?
- How did RUP impact the team dynamics, communication, and work distribution/scheduling?
- What was the impact of RUP on the interaction with the customer?
- What was the value-added of the specific artifacts to the overall success of the project?
- How did the use of the RUP framework contribute to risk mitigation?
- Did use of RUP have a positive impact on the quality of software developed by the teams?

The tools for data collection used though each project cycle are as follows:

- Weekly written progress reports and progress review meetings (team, instructor)
- Formal review meetings at all milestones (team, instructor, graduate assistant)
- Monitoring of team activities in laboratories (instructor, graduate assistant)
- Project acceptance meeting (all stakeholders)
- Postmortem evaluation tools (designed by the instructor) – feedback from team lead, team members, customer, graduate assistant

Collected data for each project cycle is analyzed and the results are used in the project’s continuous process improvement integrated into our curriculum improvement process. Specifically:

- Instructor formulates best practices and lessons-learned based on collected data.
- Instructor performs comparative analysis and correlation based on previous cycles.
- Results are presented to the Undergraduate Curriculum Committee and the Undergraduate Accreditation Committee. Potential changes to the implementation of the capstone project course and to the BS in Computer Science curriculum are made.
- Changes to the capstone project course are implemented by the instructor. Customization of the RUP process framework to meet the new needs is straightforward – another reason for choosing RUP!

PRELIMINARY RESULTS

Utilizing the documentation of the RUP framework, teams are able to jump-start the projects with minimal guidance from the instructor. Projects are completed with fewer interventions from the instructor due to better risk management, task management, and scheduling projections done by the students. The overall team autonomy has improved. Students use RUP framework and its documentation on a daily-basis as a reference for all project related activities. By experiencing hands-on a specific software process (RUP), students demonstrate better in-depth understanding of all aspects of software development and a drastically better understanding of software process models and capability maturity models such as CMM and CMMI discussed in the late part of the course. These conclusions are supported by data collected in the preliminary stages of the project over the past two years.

REFERENCES