


CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: TeamBandit Teams Management Portal	
Sponsor Information: 	Dr. Eck Doerry, Professor Collaborative Computing Lab Eck.Doerry@nau.edu School of Informatics, Computing, and Cyber Systems Northern Arizona University

Project Overview:

Small agile teams have become the dominant paradigm in corporate computer science practice (as well as in many other areas) over the last two decades. Small teams minimize cost, can easily be composed, deployed and reconfigured, and minimize organizational overhead within the team. This makes them ideal operational unit in fast-moving high-tech sectors like software engineering that require agility and fast time-to-market.

Working effectively on small teams imposes more demands on and requires higher skills in individual team members, however. There is little room for “specialists”; while team members may “lead” on some project aspect, all team members typically have to have some expertise across the spectrum, allowing the team to shift its resources between project areas as the team evolves. All team members must be able to quickly explore, evaluate, and assimilate new technologies needed for novel projects. Team communication and strong collaboration is also vital, as team members must work closely under short deadlines to move projects ahead.

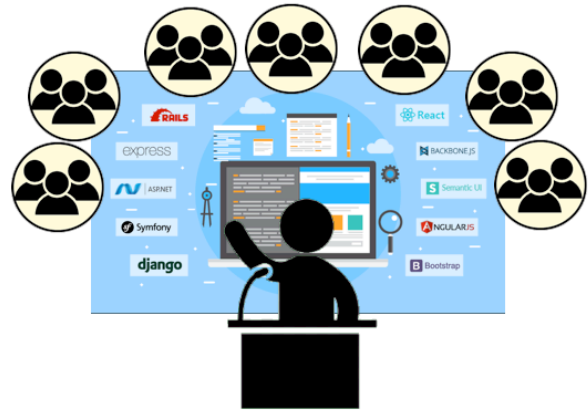


Learning to work effectively as a small team therefore requires special training and exposure, meaning that educational training programs serious about preparing graduates for rapid success in the software engineering workforce must provide specialized educational experiences aimed at honing these skills. Fortunately, NAU was an early leader in this area, launching the “Design4Practice” initiative in 1996, for which the Capstone Design sequence is the apex and most visible aspect. In the intervening years, many CS programs nationwide have followed suit, adding some sort of team-based software design and development course to their curriculum.

The Problem: A persistent challenge, both to existing team-based courses and to those considering adding them to their curricula, is the high overhead it requires to organize and manage a team-based course. After project options are collected, they must be posted, course members must somehow express their project preferences, faculty must evaluate responses and form teams. Once the teams are formed, the challenges switch to course management: a public website must be created to advertise teams and projects, and various classroom activities (like Design Reviews) must be organized, peer evaluations must be deployed and organized, and project websites for teams must be created.

Current “solutions” to these challenges are essentially non-existent: faculty tasked with running project courses must invent “manual” solutions to all aspects of team course management; the large overhead leads to overloaded faculty, poor course organization, or both. The client, Dr. Doerry, has nearly 20 years of experience organizing Capstone courses at NAU, as well as leading the Collaborative Computing Lab (CCL), which specializes in developing innovative informatics solutions to support groups and group work.

Solution Vision: A powerful team-based course management web application that significantly reduces the overhead for both faculty organizing and running a Capstone course, and for students/teams taking part in the course. A key characteristic of a successful solution will be flexibility and robustness: the design should carefully leverage existing technologies (e.g., HTML, webservers, XML, and other standard well-established tools) to avoid re-inventing existing wheels, and also making the product easy to maintain and extend over time. Although we will focus on the Capstone sequence here at NAU as a concrete test case, the product will be highly configurable to support project-based courses of all sorts, at all levels in the curriculum, at any university, and even in other non-CS disciplines. Some specific features that the TeamBandit product will support can be prioritized as follows:



Level 0: Minimum Viable Product.

- A secure web application that supports login and personalized accounts for faculty and students.
- Student accounts can be created via uploaded spreadsheet/CSV. Sends “activation” invites to email, to allow students to create a password to log in.
- Ability for students to manage templated profiles, e.g., fill in bio, skills, interests, headshot, etc.
- Ability for faculty to create new “Project Objects” to contain some particular team-based experience. Examples might be “Capstone-2021” or “CS477-final proj2022”.
- Ability to configure and edit Project Objects, includes:
 - Ability to upload PDF project descriptions, to populate it with projects.
 - Supports bulk upload, but also individual addition.
 - Supports full range of editing of project objects.
 - Ability to associate registered students with project objects, to populate the class.
 - Ability to associate students with projects (to form teams).
 - Fixed layout Teams page, dynamically generated with project/team/student information for DB
 - Ability for students to enter a URL to an external team webpage for each team.
 - Faculty can enter a URL for an external course website
- Ability for faculty to post/hide Teams page.
- Mechanism for students to enter project preferences. Builds on current 1-2-3 approach...but could also support other rating schemes.
- Ability for faculty to view/download CSV/Excel of teams, students, and preferences for external processing aimed at forming teams.
- A basic “dashboard” page for faculty: view Project Objects they own, create new ones, perhaps sort into “Folders” to keep things organized.

Level 1: A well-developed software product

- A templated “course website” facility to allow faculty to easily create a basic course website. Ability to create top-level “organizations” entity as a wrapper for all the above. This allows infinite institutions/faculties to use the site to support their programs. Allows self-signup and/or system admin creation of new organizations.

- A configurable but standardized template for team websites: allows selection of tabs to show/hide and other options.
- Automated team website generation: teams simply upload items to display in the standardized template. Could allow some cool customization options to allow aesthetic differentiation.
- Ability to configure a website template for a Project Object to customize content/layout of Teams Page. Viable external URL for Teams Page automatically generated.
- Improved and refined GUIs for all MVP features, to optimize aesthetics and usability.

Level 2: Advanced features (stretch goals)

- A team activities manager tool: One thing about team courses is that you need to manage presentation days where each team delivers something. Allows faculty to create and configure an “activity” (e.g. Design Review 1). Faculty can publish a specified presentation order, publish randomly generated order, or have a “wheel of fortune”-like selector to click to choose next presenter live in class!
- API to allow (authorized) outside software products to request certain information from the system. For instance, a report generator could harvest specified content about projects on the site via some form of targeted (by year, discipline, institution, etc.) query.
- A dynamic “Capstone Program Archive”. A templated website facility so that departments can easily create a “program archive” for the team-based courses in their programs. A splash page intros the program, and the faculty owner can specify which projects in the system are targeted in the site; search/filter/browse of the resulting page to find/view project websites.
- Rotating “featured projects” panel based on the “project pitch videos” mentioned above. Embedded in the “Capstone Program Archive” to show off cool projects.

These features are expected to evolve as the team brings its creativity and ideas to bear on the project. For robustness and future-proofing, the entire system will be deployed in the virtual AWS cloud infrastructure to support accessibility and easily scalable compute resources to provide consistent performance as popularity grows and more team-based courses nationwide are supported.

If successful, this product could increase both the quality and quantity of team-based courses offered at institutions across the nation and beyond. By providing lower overhead for both faculty and participating teams, many more institutions would add significant team-based courses to their curriculums, while at the same time increasing consistency of the experience and quality of outcomes.

Knowledge, skills, and expertise required for this project:

- Understanding of cloud-based computing; deploying/managing DB and compute instance on AWS
- Basic understanding of modern web2.0 web application technologies for building high quality web-based GUIs. Includes web app and front end frameworks and related languages and concepts.
- Understanding of how team courses work and challenges involved, from both faculty and team perspectives.

Equipment Requirements:

- Access to Amazon Web Service (AWS) account and products; will be provided.
- No other specialized equipment should be needed, outside of standard (free) IDEs and a development station.

Software and other Deliverables:

- The TeamBandit software product, as described above, deployed on a provided AWS infrastructure and tested successfully, e.g., using the data for this Capstone course. Must include a complete and clear Admin Manual for configuring and operating the software.
- A User Manual or integrated help system for end-users, i.e., faculty and students. Backstop for when clear and obvious GUI design falls short.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- Complete professionally-documented codebase, delivered as a Github repository owned by the client.