Computer Science Senior Capstone
CS 476 – Requirements Engineering

ECOders

Project: *Colorado Plateau Cooperative Ecosystem Studies Unit (CPCESU) Project Management System*

Team Standards

Overview: The purpose of this document is to outline the organization and inner workings of ECOders. This includes Code of Conduct, organization of meetings, roles and positions of team members, and more.

Team Members:

Colton Nunley
Joseph Remy, Jr.
Jasque Saydyk

Clients:

Dr. Todd Chaudhry
Laurie Thom

Northern Arizona University
School of Informatics, Computing, and Cyber Systems

Cooperative Ecosystems Studies Unit
Colorado Plateau

Version: September 23, 2018

This page is left blank intentionally.

About

General Overview

ECOders is a Senior Computer Science Capstone formed under the School of Informatics, Computing, and Cyber Systems at Northern Arizona University for the 2018-2019 school year. Our group is tasked with helping out our client, the Colorado Plateau Cooperative Ecosystem Studies Unit (CPCESU), with their problem and delivering the final product according to the specifications of the CS476 and CS486C courses.

This document specifies how the group will organize itself and ultimately function. This Team Standards is a living document and is subject to change as noted in Section 14: “Changes to this Document.”

Team Members

Colton Nunley
Joseph Remy, Jr.
Jasque Saydyk

Clients

Dr. Todd Chaudhry
Laurie Thom

Team Mentor

Ana Steinmacher

Capstone Lead Organizer

Dr. Eck Doerry

Section 1: Roles and Duties

Defined below are the various roles of the project and the duties they entail. The duties of any given role maybe larger than is stated here, and if so, this document should be updated to reflect that.

These roles do not dictate that your contribution to the project is solely defined by your role, they are just general areas of responsibility that the role holder is tasked with leading, overseeing, and delegating.

Team Leader

Joseph Remy

- Coordinates task assignments
- Developing software project plan
- Ensure work is progressing
- Makes initial effort to resolve conflicts
- Coordinates and conducts client communications
- Provides regular updates to senior management
- Ensures work is turned in and deadlines are met
- Determines Coding Standards for the project

Recorder

Jasque Saydyk

- Responsible for taking notes during client, team, mentor meetings, and general conflict resolution
- Creates agendas to the meetings
- Maintains detailed meeting minutes
  - Notes the minutes when agenda items are reached
  - Notes general course of Discuss items
  - Notes each member’s vote in Action items
  - Notes each member’s thoughts on Proposal items
  - Notes each member’s assigned Task items
- Posts minutes in a public repository available to all team members

Architect

Colton Nunley

- Determines and maintains awareness of the scope of the project
- Determines system layout, functionality, and paradigms used in it’s construction
- Determines gaps between the requirements and system functionalities
- Determine solutions to bridge gaps between requirements and system functionalities
- Create charts, diagrams, and materials necessary to accurately convey and discuss architecture details to the team, management, and client

Business Analyst

Jasque Saydyk

- Creates, maintains, and oversees documents for producing and updating requirements
- Creates procedures to handle changes to the requirements
- Creates visual diagrams to help present requirements to clients and management
- Communicates with the client to determine requirement priorities

Editor-in-Chief

Joseph Remy

- Reviews all documents to ensure proper and relevant information
- Establishes the formatting and style of all documents to ensure good document design that looks professional and is consistent with previous documents
- Proofreads all documents and sends the final version to the Team Leader to be turned in
- Ensures project documentation is kept up to date and is presentable to management and clients
- Determines look and feel of documents and documentation that is put online

Release Manager

Jasque Saydyk

- Responsible for proper versioning of the project
- Responsible for maintaining main project repository as per the Version Control Section and Coding Standards document
- Responsible for reviewing pull requests to the support and release candidate branches as outlined in Team Standards & the Coding Standards document
- Responsible for setting up continuous integration for the project and maintaining it

QA Manager

Colton Nunley

- Responsible for ensuring software meets quality standards, non-functional requirements, and follows code standards set by team
- Ensures software is meeting the functional requirements and delivers it's intended value to the client
- Communicates, designs, and conducts usability tests for software with focus groups
- Reports and documents issues and recommends possible improvements to the system

UI Designer

Joseph Remy

- Focused on the web portal to the clients website, not team website or online project documentation
- Responsible for creating and maintaining a consistent look and feel to the user interface
- Responsible for creating mockups of the UI for approval by the team
- Responsible for creating sketches and actions flows on how the UI looks and moves, acts, and transitions for the team to implement
- Responsible for creating professional looking designs of the UI that can be realistically implemented by the team to present to management and the client
- Responsible for ensuring Coding Standards are upheld in the UI

Capstone Website Developer

Rotational[1]

- Manages/maintains external Capstone website
- Manages/maintains CEFNS Capstone website
- Manages/maintains the project documentation website/wiki

Customer Communicator

Vacant

- If customer communication or a section of customer communication that does not include the clients of the project gets out of hand, the Team Leader may decide to develop this role

General Coding Jobs

To Be Determined

- Each member of the team is responsible for contributing to the codebase of the project. General coding roles may be changed as the project requires, team members may be required to take more than one role if it is necessary. Below is a list of general coding jobs, further specification will be required as the project develops.
    - Frontend Developer
    - Backend Developer
    - Database Administrator
    - Tester

[1] Rotation is reassigned every 2 months to a different team member

Section 2: Meeting Times and Organization

This lists out the meeting times, dates, and places along with the various communication tools the team will use and their purpose. Any deviation from below will be announced to the team by email. Client meetings will be announced to the team by the Team Leader. Impromptu meetings can be called and organized by the team members as needed.

Team Meeting	**Mentor Meeting**
Mondays, 4:00 pm to 5:00 pm	Wednesdays, 4:00 pm to 5:00pm
Engineering building, Room 102	SICCS, Room 227

Methods of Communication

- Discord - For normal text communications and impromptu meetings within the team
- Hangouts/Skype or phone call - For remote meetings with the client
- SMS messaging - For emergency communications or team wide updates
- Email - For client communication, official memos within the team, and team updates

Section 3: Agenda Structure

An agenda will be created for every team meeting to help guide discussion. Agenda's may be created for client meetings and impromptu meetings as needed. The agenda's structure will be as follows:

- **Header**
  - Meeting number
  - Date
  - Time
  - Location and room number
- **Pre-Meeting**
  - Any tasks that need to be brought to the meeting
- **Agenda Beginning**
  - Roll call
  - Reading of the agenda
  - Changes to the agenda
  - Short report - On activities since the last meeting
- **Agenda Main**
  - **Discuss:** items that require discussion and possible action
  - **Action:** items that require a vote by the team
  - **Proposal:** item that a team member brings forward to the group to discuss
- **Agenda End**
  - Closing remarks
  - Meeting adjourn

Section 4: Minutes Structure

The minutes will be maintained to ensure decisions, assigned tasks, and general discussions in meetings are remembered outside of the meeting.

For meetings with agendas, the minutes structure will just be the agenda, noted with

- The minutes when agenda items are reached
- The general course of Discuss items
- Each member's vote in Action items
- Each member's thoughts on Proposal items
- Each member's assigned Task items

The minutes will be italicized and task declarations will be highlighted yellow in the final version.

For meetings without agendas, minutes will be recorded with general shifts in the conversation and general notes will be recorded, abiding by the list above to the best of the recorder's ability.

Section 5: Decision-Making Process

While it is important and ideal that team members work together to make decisions on the project unanimously, sometimes there may be impasses that require a majority vote to just get through agenda items, assign tasks, and move ahead with the project.

If general consensus cannot be met, the following procedures are to take place:

- Each side of the disagreement will be given ample time[2] to discuss their point.
- If the disagreement is over a Proposal or Action item on the agenda, the measure can be brought to vote.
- If the disagreement is over an item added to the agenda during the "Proposed changes to the agenda" item or came up over the course of discussion during the meeting:
  - A vote maybe taken to decide if the item requires further research.
    - If it passes, it is added to the agenda of the next meeting
    - Otherwise, the item is discarded
- For any vote to pass, it requires a 2/3 majority.

[2] Ample time will be defined as more than 5 minutes but less than 15 minutes

Section 6: Attendance

This section lists out the policy for attendance and tardiness for meetings and any events the team is required to attend.

Attendance Policy

- If a team member cannot make a meeting more than 24 hours in advance, they are obliged to communicate this with the team in an effort to reschedule the meeting.
- If a meeting is missed without prior notice, an email must be provided to the team within 24 hours containing:
  - The reason for the absence
  - A written report of activities taken with the project over the past week
- If two meetings are missed within a four week time span:
  - If the reason for missing the meeting is time-based:
    - The meeting time is to be rescheduled to a new time with potential changes to meeting format if necessary.
  - If the reason for missing the meetings is personal-relation based:
    - General conflict resolution procedures are to be taken.

Tardiness

- If a team member is consistently and meaningfully[3] tardy:
  - A discussion must take to rectify the issue. This may include adjusting the meeting time or place.
  - If the issue continues, general conflict resolution procedures are to take place.

Section 7: Conduct

This notes the various processes that will be used to rectify bad behaviour that endangers the group and project. While conflict is to be expected, all issues should be resolved within the team as a team.

Subsection 1: General Conflict Resolution Process

- The conflicting parties and a third party, if a third party exists within the team, are to meet at a scheduled time outside of the weekly meeting to discuss the dispute.
- This is how the meeting is to be conducted:
  - The conflicting parties will begin by allowing each other to tell their story on the conflict so far.
  - From this point, the conversation should move to addressing how to handle each parties' grievances.
  - A plan is then made to address the grievances and fix the situation.
- If the discussion fails to address each parties' grievances, or the plan fails to rectify the issue, it is to be followed with a team meeting with the CS Capstone Mentor.

[3] Consistently and meaningfully being defined as greater than 15 minutes late for three meetings over a four week time span

- If the results of this meeting are unable to rectify the issue, then official Capstone Firing procedures are to then take place.

Subsection 2: Meetings

- Meetings, and the preceding discussions, are to be professional.
- The agenda will be followed during the meetings:
    - Derailments from the agenda will be add as a topic later on in the meeting or on the agenda for the next meeting.
- While some discussions will be heated, it is important that it doesn't turn into a verbal attack. If a discussion does get to this point, the following procedure is to take place:
    - A third party to the discussion is to quell the verbal attacks, if possible.
    - The third party then goes through the points of the two parties in an effort make both sides understand the other's merits and demerits.
        - This, in turn, will hopefully validate both sides of the debate and will create a more constructive atmosphere.
    - If the third party is unable to quell the verbal attacks, or the discussion relapses into verbal attacks :
        - The meeting is to be cancelled immediately.
        - The rest of the agenda items are rescheduled to be discussed later that week.
        - General conflict resolution procedures are to take place outside of the meeting.

Subsection 3: Nonparticipating members

- If a team member isn't participating or pulling their weight, a "heads up" occurs where the team talks to the team member to figure out what is going on.
    - If it is a knowledge/confidence issue, knowledge sharing sessions may need to take place.
    - If it is a personality conflict with the team that can't be solved trivially, general conflict resolution procedures are to take place.
    - If it is a scheduling/crunch time issue, assignments may need to adjust to account for it in a specified time span.
        - If it continues past the time span, general conflict resolution procedures are to take place.
    - If it is a lack of will issue, general conflict resolution procedures are to take place.

Subsection 4: Changing design without team consent

- Presenting solutions to the client that was not previously agreed on by the team that conflicts with the team's presentation is not to be tolerated.
    - It will be followed by an immediate meeting following general conflict resolution procedures.
- For any code or document change, the team is allowed to revert the project back to a point were that change is not included, if the change meets any two of the following:
    - Is before a deadline

  - Is of significant[4] size
  - Violates requirements

[4] Significant being on the level of a shift in architecture design or scope

Section 8: Version Control System

For general, robust version controls, the team will be using Git, with GitLab as the collaborative interface for performing pull requests, storing prototypes, documents, and anything else relevant to the project.

The group will include specific details and guides on version control, how the repositories and trees are managed, software engineering principles, issue tracking, and more within the Coding Standards document, which supersedes this Section and Section 9. Below is a general overview of how the processes will work.

Subsection 1: Overview of GitLab groups:

- We will maintain three separate Git repositories (and additional ones as needed):
    - Main Project Deliverable - to deliver to client
    - Capstone Website - labelled as "website", for changing CEFNS / external capstone website
    - Documents - labelled as "gdrive", contains documents, presentations, task reports, and other relevant documents backed up from Google Drive

Subsection 2: Overview of Main Project Repository

- Stable branch - "master"
    - This will contain the current stable version of the project
    - Updated via passing release candidate (RC) and support branches
        - RC and Support branches are managed by the Release Manager via pull requests
        - Only the Release Manager may merge RC and Support branches into "master" after final approval from Team Leader
- Support branch - "support"
    - This will contain bug fixes and small changes that hinder, break, or expose the project. Tasks are determined by the issue board with the label "hotfix".
    - This branch is only updated via pull requests from "hotfix" branches attached to issue cards.
    - Hotfix branches - "hotfix-<ISSUE NUMBER>"
        - Hotfix includes a quick remedy to the stable version of the codebase.
        - These are created by the assigned team member once an issue card is made and assigned.
- Release candidate branch - "release_candidate"
    - This will contain a collection of features and commits that are passing tests, QA, and Coding Standards. The commits in this branch are to be treated as final and ready for external testing without hindering the stable release.
    - This branch is only updated via pull requests from the "dev" or experimental branch and must be reviewed by the whole team.
- Experimental branch - "dev"

  - This will contain features and commits that are approved to be merged in by the assigned code reviewer(s) via pull request after the team member completes the relevant issue(s) to the best of their ability.
  - One or two reviewers are to review the pull request to ensure it follows the Coding Standards and passes the testing suite.
  - This branch is only updated via pull requests from “feature” branches attached to issue cards.
  - Pull requests may be “near final” with unanimous approval (or through the decision making process), but team members should strive for completeness.
    - This allows the opportunity to prevent deadlock if multiple developers are work on the same area, feature, etc.
  - Pull requests must pass all unit tests and meet QA and Coding Standards.
  - Feature branches - “feature-<SHORTHAND FEATURE NAME>”
    - Features include relevant issue card(s)
    - These are created by the assigned team member once an issue card is made and assigned.

Subsection 3: Overview of Archival Repository

- Sole branch - ‘master’
  - The current documents within the Google Drive will be backed up once a week after the weekly team meeting.

Subsection 4: Overview of Capstone Website Repository

- Stable branch - ‘master’
  - Current working version of the capstone website
  - Updated via pull requests from individual team member’s branches
  - Approved by Editor-in-Chief
- Team branches - ‘colton’, ‘jasque’, and ‘joseph’
  - Each team member’s workspaces are independent branches

Section 9: Issue Tracking

Issue tracking is more than just the tracking of bugs in the code, it is the team wide to-do list. As such, the issue tracker will be used in the assignment, updating, and resolution of all tasks in the project, from humble documentation to the creation of robust code.

- GitLab is the issue tracker for all tasks in the team

Procedures for issue creation:

- Issues and tasks are created at team meetings
  - Impromptu cards can be made after quick discussion and approval with team members. This allows for things not covered, missed, or are suddenly needed.
- Issues and tasks are then labelled and assigned
  - Milestones are a collection of like card (capstone assignments, collective tasks, etc.)

    - Labels are used to quickly describe the issues and tasks like “hotfix”, “feature”, “capstone_assignment”, “capstone_task”, etc.
- Issues are logged into the system (GitLab issue boards to the relevant repository)
- Issues status and life are updated through the system

Section 10: Word Processing and Presentations

Various presentations and documents will need to be created throughout this project. Below are the tools that will be used to perform these tasks.

Word Processing

- Google Docs for collaborative document creation
- PDF for final turn in

Presentations

- Reveal.js
- PDF as a back up

Documentation

- GitLab Wiki or Independent Wiki
- Team Website
- User Manual - How to work the system from an administrator point of view
- Technical Handbook - How to setup and work the system from a technical standpoint
- Coding Standards - For future programmers to use as reference

Section 11: Coding Standards

Coding Standards are necessary for the maintenance of the code, not just for this team, but also for the team that will inherit this project to update it’s codebase after this capstone experience.

- The team will work collaboratively to find Coding Standards to reference and/or customize standards to the client and/or team’s needs where applicable.
    - Examples for existing style guides and Coding Standards include pep8 and JsLint.
- The Coding Standards include areas such as style guides, programming styles, best practices, software engineering principles, workflows, etc.
- After team approval of any additions, removals, or alterations to the Coding Standards via general consensus or the decision making process, the Team Leader will document these changes to the team’s Coding Standards document.

Section 12:
Composition and Review Process

The composition process is designed such that it provides ample time for the Editor-in-Chief to do their job, with the more time away from the deadline, the more time the editor-in-Chief gets to to their job. The following are recommended guidelines and may need to be adjusted to individual situations on a case by case basis.

- Assignments are made as far in advance as possible by the Team Leader.
- General rough drafts for assigned sections must be made :
    - 2 weeks from the deadline for assignments given 1 month or more in advance
    - 1 week from the deadline for assignments given 2 weeks in advance
    - 2 days from the deadline for assignments given 1 week in advance
- Final drafts to be submitted to the Editor-in-Chief must be given:
    - 1 week from deadline for assignments given a month or more in advance
    - 3 days from deadline for assignments given 2 weeks in advance
    - 1 day from deadline for assignments given 1 week in advance
- Editor-in-Chief is to then create the final document within the remaining timespan, which is to be given to the Team Lead for submission when finished.

Section 13: Team Self Review

The team self review is an opportunity for the team to review themselves and the general progress of the team. Issues are brought to light and corrected for future months.

- A team self review will take place at the end of software sprints (as determined by the Coding Standards document).
- The review will begin with a summarization of the work done over the past month.
- Team members will take turns in a round table fashion examining and critiquing themselves on the following subjects, which will then be open for discussion by everyone at large, respectfully:
    - Attendance
    - Successes
    - Weaknesses
- This is followed by a team analysis of the following subjects with regard to the entire team:
    - Reflect on previous sprint write-up
    - Communication
    - Current Progress
    - What has worked well and the group should continue doing
    - Any problems that occurred outside the group’s control
    - Any issues at large hindering the group that we can change
- Notes about what is said by team member will be taken by the recorder and placed into a separate file from the minutes for the team review.

Along with the team self review, during weekly meetings, team members should quickly summarize their progress on issues (the good and bad) and anything items that need to be addressed.

- This happens when the team reviews the issue board and a team member will present when their issue card is called.

Section 14: Changes to This Document

This document is a living document that will be added to and removed from throughout the project. As such, listed below are mechanisms to make changes to this document so it retains its original content for reference along with any addendums.

- The Team Standards must be kept in its original condition and any additions and/or modifications must be written as a new Amendment in the Amendments section at the end of the Team Standards document.
- All new Amendments must be approved unanimously by the team.
  - If consensus cannot be reached, use the Decision Making Process
- New Amendments can only be approved of during team meetings with full attendance of the members:
  - Formal weekly team meetings
  - Mentor meetings
  - Impromptu meetings agreed upon by the team
- New Amendments must include the date of the revision and the content the revision changes.
- If an Amendment changes or supersedes any of the above sections, including this one, then there will be a single line at the end of the section stating the Amendment number(s).
  - Example: “Revised and Altered by: Amendment 1”
  - Example: “Revised and Altered by: Amendments 1, 4, 5”

Amendments

None at present and this line will be removed once the first Amendment is made.