CST 316 Proposal

October 22, 2014

**Prepared For:**

Daniel Freeman

Todd Breedlove

**Prepared By:**

Stewart Taylor

Jacob Neal

Aaron Costner

Lewis Sanchez

Table of Contents

[Introduction 2](#_Toc401678445)

[What is our proposal? 2](#_Toc401678446)

[What is the purpose of this proposal? 2](#_Toc401678447)

[Architecture 2](#_Toc401678448)

[Requirements 3](#_Toc401678449)

[Development Process 4](#_Toc401678450)

[Product Backlog 4](#_Toc401678451)

[Release Backlog 4](#_Toc401678452)

[Sprints 4](#_Toc401678453)

[Progress 4](#_Toc401678454)

[Burndown Charts 4](#_Toc401678455)

[Kanban 4](#_Toc401678456)

[Gantt Charts 4](#_Toc401678457)

[Daily Scrum Meetings 4](#_Toc401678458)

[Weekly Status Reports 4](#_Toc401678459)

[Code Reviews 4](#_Toc401678460)

[Technologies 5](#_Toc401678461)

[Repository 5](#_Toc401678462)

[Source control 5](#_Toc401678463)

[Client for Source Control 5](#_Toc401678464)

[Continuous Integration Server 5](#_Toc401678465)

[Scrum tools 5](#_Toc401678466)

[GUI Designer 5](#_Toc401678467)

[IDE 5](#_Toc401678468)

[Documentation 6](#_Toc401678469)

[CG Framework 6](#_Toc401678470)

[Libraries 6](#_Toc401678471)

[File Structure (doc, src, lib, bin) 6](#_Toc401678472)

[Naming Convention 6](#_Toc401678473)

[Static code analysis 6](#_Toc401678474)

[Code complexity 6](#_Toc401678475)

[Testing 6](#_Toc401678476)

[Unit Tests 6](#_Toc401678477)

[Integration tests 6](#_Toc401678478)

[Acceptance Tests 6](#_Toc401678479)

[Automated tests 6](#_Toc401678480)

[Project Delivery 6](#_Toc401678481)

[Expenses 7](#_Toc401678482)

[STRs 7](#_Toc401678483)

[Verification and Validation 7](#_Toc401678484)

[Contact Details 8](#_Toc401678485)

[Team Members 8](#_Toc401678486)

[Product Owner 8](#_Toc401678487)

[Stakeholders 8](#_Toc401678488)

# **Introduction**

This document details our proposal for CST 316 (Software Process Management).

### **What is our proposal?**

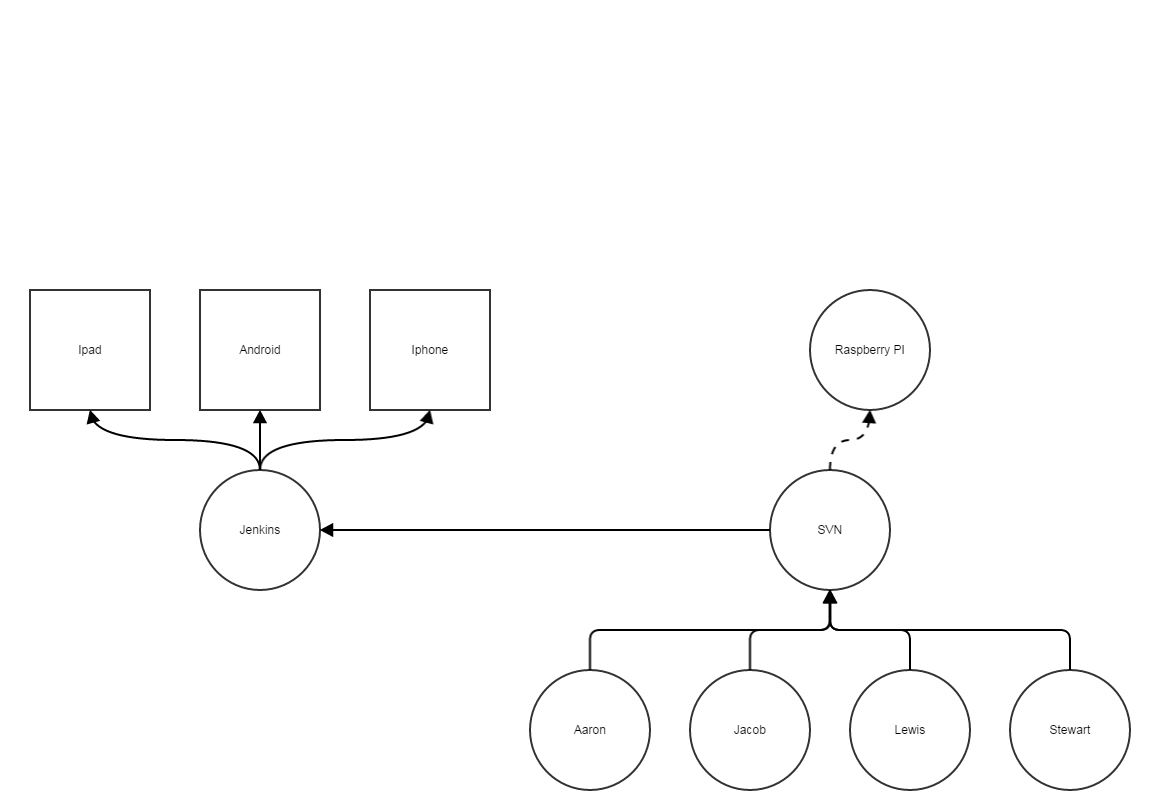
Our team is proposing to build a mobile internet chess server that will enable chessgames.com members to play chess on the Android and iOS operating systems.

### What is the purpose of this proposal?

The purpose of this document is to give the product owner (i.e. Daniel Freeman) and any stakeholders (e.g. Todd Breedlove) a high-level overview of the project. It details how the project will be designed, constructed and delivered for the product owner.

# **Architecture**

Below is the development architecture for the project.



# **Requirements**

The requirements below have been specified by the product owner. The following are the minimum requirements for the mobile internet chess server.

1. Connect to User Login System
   1. Allow user to create username
      1. Restrict username between 2 to 20 characters
      2. Restrict to same specifications as chessgames.com usernames
      3. Username cannot be the same as another username in the system.
   2. Save the username
   3. Save the password
      1. Restrict password to same specifications as chessgames.com
2. Elo Rating System
   1. Implement Elo rating system into application
      1. Change player’s Elo rating on their skill level compared to others
      2. Update Elo rating after each finished game, or resignation from a game
3. Record game in algebraic notation
   1. Allow users to swipe through notation, or game
   2. Allow users to peruse game with GUI arrow keys
4. Implement two Clock Objects into each game
   1. Assign one clock to each player
   2. A clock only counts down when it is that player’s turn
   3. Game ends when time on any clock reaches zero
   4. Control LED on screen to determine which player’s turn it is
5. Implement Game Pools
   1. Place users in queue for a game
      1. Queues divided into 1 minute, 5 minute, and 30 minute games
      2. First move decided randomly between the two players in a game.
6. Allow users to Resign or Abort
   1. Restrict Abort feature to first three moves of a game
      1. A user who aborts from a game does not lose Elo rating points
   2. Allow users to Resign after first three moves of a game
      1. A user who resigns also loses Elo rating points
7. Allow users to Offer Draw to competitor
   1. Change Offer Draw GUI button to Accept Draw on competitor view
8. Record Game State (Win, Lose, Draw) in chess database
9. Allow users to under promote pawn upon reaching the eighth rank
10. Record all games played on application to database via PGN format.
11. Allow users to log in through Facebook and Google.
12. Allow user to play chess against another player on application.
    1. Allow users to perform any legal move in application.

# **Development Process**

The software development lifecycle for this process will revolve around the scrum methodology. We will continue to release iterations of the product through sprints until the product is accepted by the product owner.

## **Product Backlog**

Our team will work with the product owner to complete a product backlog (wish list of features).

## **Release Backlog**

We will then divide the product backlog into a release backlog (realistic wish list).

## **Sprints**

Sprints will have a two week development cycle. They will be divided into features that can be completed in less than 24 hours and each task will be assigned to a team member. An ETA will be associated with each feature in the sprint to give a rough estimate of

## **Progress**

Progress will be measured through burndown, kanban and gantt charts. These tools will give any stakeholders an idea of if the project is on schedule or not.

### **Burndown Charts**

These are used to track daily updates to see if a project is on schedule. The chart uses Calculus to produce tangent lines to get a project back on schedule or show how far off the project is off schedule.

### **Kanban**

These charts are used to see what is tasks are currently in progress, done or not started. It also shows which developers are assigned to each task.

### **Gantt Charts**

These charts show what has to be done and when. These will be used to show if an individual is on schedule with their assignments.

## **Daily Scrum Meetings**

These meetings will be held every day for at least 15 minutes. Minutes will be taken and a form will be signed by each group member to document that the content in the form reflects what happened in the meeting.

### Weekly Status Reports

These reports will consist of seven daily scrum meeting forms and a summary of the week’s progress. This will be kept in a binder that could be checked at random from any stakeholder (e.g. Todd Breedlove).

### Code Reviews

Reviews of each member’s code will be done on an as needed basis. Code will be refactored to be more efficient, easier to read and easier to maintain.

## **Technologies**

### Repository

A raspberry pi will be our repository server. It can be accessed through everyone remotely as long as they have internet access.

### Source control

Subversion (SVN) will be our source control for the project. It will be installed on the raspberry pi.

### Client for Source Control

TortoiseSVN will act as a GUI for SVN, so our team members will not have to memorize SVN command line arguments to access the repository.

### Continuous Integration Server

Jenkins will serve as our CI tool that will perform static code analysis, code coverage, bug tracking and nightly builds. Cron expressions will be used to schedule builds at any given time.

### Scrum tools

Scrum wise is an online tool that we will use to keep track of our product backlog, kanbans, burndown charts and gantt charts.

### GUI Designer

The Qt Framework will be our GUI designer for the project.

### IDE

QtCreator will be the IDE in which everyone will be working in.

## Documentation

This project will be heavily documented because it will eventually be handed over to the product owner who will have to figure out what is going on with the code.

### CG Framework

All the libraries (dll’s) will be wrapped up inside a reusable framework called CG (chessgames).

### Libraries

Each library will be documented and prefixed with a lib\_. All classes and methods inside the library will also be detailed in the documentation.

### File Structure (doc, src, lib, bin)

The codebase will be divided into doc, src, lib and bin folders. The doc folders will contain the documentation for the codebase. The src folders will be the source code, the lib folders will be the libraries and the bin folders will be the actual executables.

## Naming Convention

A naming convention will be in place to ensure all variables, functions, etc. will be the same across the codebase.

### Static code analysis

Tests to ensure these naming conventions are adhered to will be performed before code is allowed to be checked-in.

### Code complexity

This service will test if any branch of code has not been tested.

## **Testing**

Testing will be extensive throughout the course of the project. Although testing will happen from the beginning of the project, the majority of the testing will occur during spring term.

### Unit Tests

These are small tests that will be run every time code is checked in. These tests can include moving the pieces or logging into the system.

### Integration tests

These tests will be run nightly to ensure the product runs smoothly under extreme pressure. We will use load tests to test bring the product to the brink of breaking, in order to test its limits.

### Acceptance Tests

There are two types of acceptance tests that will be performed during the project: Reliability Acceptance Tests (RAT) and Stability Acceptance Tests (SAT). RAT tests will ensure the product can run for thirty days straight with no defects. The SAT tests are an onsite test that will be performed in front of the product owner to see whether or not he accepts the product as finished.

### Automated tests

Scripts will automate unit, integration and acceptance tests during the course of the project.

### Project Delivery

Project will be released on the Android and iOS app stores after project passes a SAT test.

### Expenses

All expense reports will be documented and given to the product owner for reimbursement.

### STRs

Every defect or bug will spark a software test report (STR), in order to remedy any bugs.

### Verification and Validation

Team will ensure that the mobile app meets the specifications and that fulfills its intended purpose.

# Contact Details

### Team Members

Stu Taylor

stewart.taylor@oit.edu

(530) 638-7205

Jacob Neal

jacob.neal@oit.edu

(541) 890-4103

Aaron Costner

aaron.costner@oit.edu

(541) 643-0190

Lewis Sanchez

lewis.sanchez@oit.edu

(971) 570-9494

### Product Owner

Daniel Freeman

chess@chessgames.com

(954) 494-3638

### Stakeholders

Todd Breedlove

todd.breedlove@oit.edu

(541) 885-1577