CST 316 Proposal

October 18, 2014

**Prepared For:**

Daniel Freeman

Todd Breedlove

**Prepared By:**

Stewart Taylor

Jacob Neal

Aaron Costner

Lewis Sanchez

Table of Contents

[Introduction 2](#_Toc401428437)

[What is our proposal? 2](#_Toc401428438)

[What is the purpose of this proposal? 2](#_Toc401428439)

[Architecture 2](#_Toc401428440)

[[Insert Block Diagram here] 2](#_Toc401428441)

[Requirements 3](#_Toc401428442)

[User Login Systems 3](#_Toc401428443)

[User roles (sysops, moderators, users) 3](#_Toc401428444)

[Username verification 3](#_Toc401428445)

[Rating System 3](#_Toc401428446)

[Algebraic Notation 3](#_Toc401428447)

[Flags 3](#_Toc401428448)

[Clocks 3](#_Toc401428449)

[Follow 3](#_Toc401428450)

[Game Pools 3](#_Toc401428451)

[Friends List 3](#_Toc401428452)

[Challenge Friend 3](#_Toc401428453)

[Sounds 3](#_Toc401428454)

[Audience 4](#_Toc401428455)

[Resign (Abort) 4](#_Toc401428456)

[Offer Draw (Accept) 4](#_Toc401428457)

[Flip board 4](#_Toc401428458)

[Chat 4](#_Toc401428459)

[Censor 4](#_Toc401428460)

[Profiles 4](#_Toc401428461)

[Events 4](#_Toc401428462)

[Simultaneous Exhibitions 4](#_Toc401428463)

[Master Challenges 4](#_Toc401428464)

[KO Tournaments 4](#_Toc401428465)

[Team matches 4](#_Toc401428466)

[Audio broadcasts 4](#_Toc401428467)

[FEN (Forsyth-Edwards Notation) 4](#_Toc401428468)

[PGN Parser 4](#_Toc401428469)

[Game states 5](#_Toc401428470)

[Broadcasts 5](#_Toc401428471)

[Premove 5](#_Toc401428472)

[Automatically promote to queen 5](#_Toc401428473)

[Underpromotion 5](#_Toc401428474)

[Game database 5](#_Toc401428475)

[Social media logins 5](#_Toc401428476)

[Development Process 6](#_Toc401428477)

[Product Backlog 6](#_Toc401428478)

[Release Backlog 6](#_Toc401428479)

[Sprints 6](#_Toc401428480)

[Progress 6](#_Toc401428481)

[Burndown Charts 6](#_Toc401428482)

[Kanban 6](#_Toc401428483)

[Gantt Charts 6](#_Toc401428484)

[Daily Scrum Meetings 6](#_Toc401428485)

[Weekly Status Reports 6](#_Toc401428486)

[Code Reviews 6](#_Toc401428487)

[Technologies 7](#_Toc401428488)

[Repository 7](#_Toc401428489)

[Source control 7](#_Toc401428490)

[Client for Source Control 7](#_Toc401428491)

[Continuous Integration Server 7](#_Toc401428492)

[Scrum tools 7](#_Toc401428493)

[GUI Designer 7](#_Toc401428494)

[IDE 7](#_Toc401428495)

[Documentation 8](#_Toc401428496)

[CG Framework 8](#_Toc401428497)

[Libraries 8](#_Toc401428498)

[File Structure (doc, src, lib, bin) 8](#_Toc401428499)

[Naming Convention 8](#_Toc401428500)

[Static code analysis 8](#_Toc401428501)

[Code complexity 8](#_Toc401428502)

[Testing 8](#_Toc401428503)

[Unit Tests 8](#_Toc401428504)

[Integration tests 8](#_Toc401428505)

[Acceptance Tests 8](#_Toc401428506)

[Automated tests 8](#_Toc401428507)

[Project Delivery 8](#_Toc401428508)

[Expenses 8](#_Toc401428509)

[STRs 8](#_Toc401428510)

[Quality Assurance 8](#_Toc401428511)

[Verification 8](#_Toc401428512)

[Validation 8](#_Toc401428513)

[Common Problems 8](#_Toc401428514)

[Deliverables and Beneficiaries 9](#_Toc401428515)

[Contact Details 10](#_Toc401428516)

[Team Members 10](#_Toc401428517)

[Product Owner 10](#_Toc401428518)

[Stakeholders 10](#_Toc401428519)

# **Introduction**

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

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

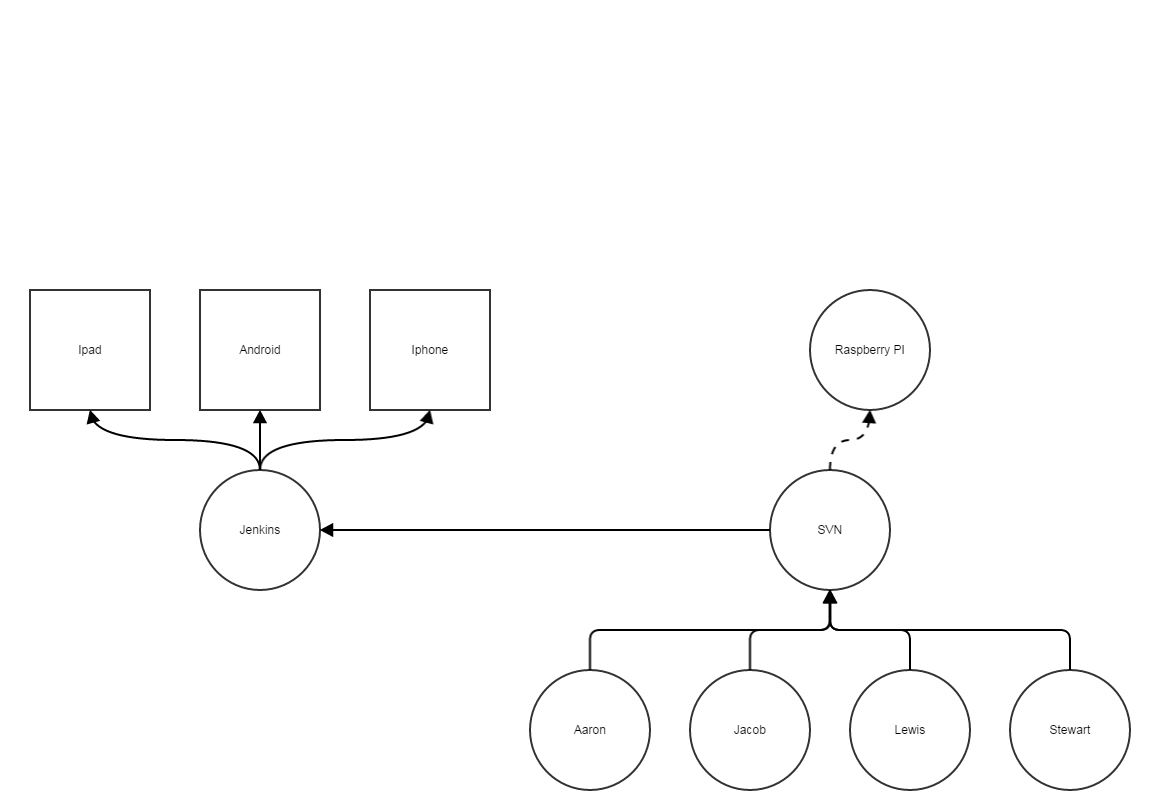
Our team is proposing to build a mobile internet chess server that will enable chessgames.com members to play chess on the Android and iOS operating systems.

### 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.

### User Login Systems

User will have the ability to create usernames and passwords. They will use this as their login information to access the server.

### User roles (sysops, moderators, users)

Users will be divided into three roles on the server: sysops, moderators and users. Sysops will be able to have administrator privileges such as removing users from the server. Moderators will be able to schedule events, censor users and make updates to the database. Users will only have the basic functionality of the server such as playing chess, participating in events and following friends.

### Username verification

Usernames will not be duplicated.

### Rating System

Users will have ratings attached to their usernames. The rating system that will be is used is the ELO system. Users’ ratings will be updated after finished every game.

### Algebraic Notation

Users will have every game recorded via algebraic notation. Users will also be able to use arrows to review a game using this notation.

### Flags

Users will be able to choose the countries they want to play for. Their flag will be next to their username during games.

### Clocks

Games will be timed and kept track of. The clocks will contain an LED signifying whose turn it is. After each move, the clock is stopped while the other clock starts. The two clocks will never run simultaneously. The purpose is to keep track of the total time each player takes for his or her own moves, and ensure that neither player overly delays the game.

### Follow

Users will be able to follow their friends’ games as well as anyone of their choosing.

### Game Pools

Users will be placed into queues while waiting for a game. These queues are divided into the following time controls: 1 minute, 5 minutes and 30 minutes.

### Friends List

Users will have the ability to keep of list of friends on the server. Users will be notified whenever their friend logs online.

### Challenge Friend

Users will be able to challenge their friends or anyone to a game and time control of their choosing.

### Sounds

Various sounds for events such as winning, losing or drawing a chess game will be implemented.

### Audience

Users will be able to see the users that are currently viewing their (or any) game on the server.

### Resign (Abort)

Users will have the ability to resign (quit) a game without playing until checkmate. If the number of moves is below three moves, the server will consider this an abort instead of a “resign” and will not deduct rating points from the user.

### Offer Draw (Accept)

User will be able to offer draws during the course of a game. If a draw is being offered, the offer draw button will turn into an accept draw button for the player receiving the draw offer.

### Flip board

The ability to flip the board to view the game from the white or black side will be implemented.

### Chat

Users will have the ability to chat with anyone on the server that isn’t censoring them.

### Censor

Users will have the ability to censor anyone from chatting with them.

### Profiles

Users will be able to customize their profiles and upload pictures as avatars.

### Events

Administrators will be able to host events for users. A banner at the top of every users screen will show the current event that is being hosted on the server.

### Simultaneous Exhibitions

Titled players will be able to play multiple users at once during simul events.

### Master Challenges

Titled players will be able to give odds to a queue of users.

### KO Tournaments

Users will be able to participate in knockout tournaments.

### Team matches

Moderators will be able to set up matches such as Hungary vs Armenia or Russia vs. The Rest of the World.

### Audio broadcasts

Moderators will have the ability to give users the power to create a channel for audio broadcasts. This is useful for commenting tournament events or talking during a master challenge.

### FEN (Forsyth-Edwards Notation)

Users will be able to paste a string using the FEN protocol to view any given chess position.

### PGN Parser

Users will be able to paste pgns using a pgn parser in order to review games.

### Game states

Users will be able to offer a draw or resign during the course of a game. The result will then be updated to the database.

### Broadcasts

Users will be able to broadcast packets to multiple clients. This feature will be useful when users want to watch multiple games at once or multiple clients viewing the same game.

### Premove

Users will have the ability to make a move before their turn. User will be able to premove up to 20 moves in advance.

### Automatically promote to queen

Upon reaching the eighth rank with a pawn, users will have the option of the pawn always promoting to a queen.

### Underpromotion

Users will be able to underpromote to any piece of their choosing upon reaching the eighth rank with their pawns.

### Game database

All games played on the chessgames.com mobile app will be stored in a database via PGN format.

### Social media logins

Users will be able to login via Facebook and Google.

# **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 (wishlist of features).

## **Release Backlog**

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

## **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 (dlls) 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 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 is 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