Sport Impact Assessment System Proposal

Simon Codrington III

Master of Science in Software Engineering Capstone Project Proposal

Grand Canyon University

Instructor: Professor Mohamed Mneimneh

Revision: Version 1.0

Date: 5/27/2024

**ABSTRACT**

Coaches in team sports at all levels are charged with evaluating the impact a player has on the organization, and making hierarchical decisions based on said impact. Because most players, by design, do not register a statistic during any given play, attempting to judge said impact based on these raw data is not sufficient. This is especially true in America’s highest revenue generating team sport of Football. There are 22 players on the field at any given period and only 3-4 players at most will register a trackable statistic. How much of an impact a player has on a game is then left up for interpretation which leads to inconsistent evaluations and non-data driven roster decisions. I propose a system of web applications that aim to allow football organizations to manage their player data (such as grades, on field statistics, strength, and conditioning, etc.), save game film, and provide an algorithmic framework for evaluating on and off field performance. Doing so will allow the organization to track player progression over time, in a unified and common way, and make the best data driven roster decision when they matter most, after the season.

|  |
| --- |
| **HISTORY AND SIGN-OFF SHEET** |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**Project Approval**

*<Insert name of instructor here>*

**TABLE OF CONTENTS**

Project Overview and Project Objectives 4

Project Scope 5

Project Success Measures 6

Project High-Level Solution 7

Project Controls 8

Project Cost and Schedule 9

Appendix A – References 10

Appendix B – Copyright Compliance 11

Project Overview and Project Objectives

**State the Problem and Background**

The purpose of this project is to give football and potentially other sport teams/organizations a tool/framework to be able to take a data driven and systematic approach to player evaluation and roster management while being able to share player data with outside organizations for recruiting purposes.

In my experience as a football player and football coach, I can attest that coaches are not the best judge of talent. This is because most of the roster decisions that are made are not data driven ones. They are mostly judgment calls, and this normally leads to misevaluations. In college football, this could cost organizations millions in scholarships and name/image/likeness deals. This system of applications aims to give teams the tools to remedy this by normalizing the evaluation and data collection process regardless of position and to allow for the convenient sharing of this data.

**Christian Worldview**

The Bible discusses that we are all equal in the eyes of the Lord. The Bible is a framework to live by and be judged. When using said framework as a guide it is easy to determine how much a person is living with God. This can too be in team sports. If everyone is operating under the same framework, it is easy to determine who has the most impact and who is reliable when compared to others.

**Project Objectives**

1. A team will be able to onboard and set up an account.
2. A recruit will be able to register and sign up for the system.
3. A team will be able to allow coaches and players to sign up and register with their team.
4. A team or coach will be able to create and add a player.
5. A team will be able to add a recruit to their team.
6. A team or recruit will be able to share player data and film with other teams.
7. And team and recruit will be able to store game film composed of small video clips or “plays.”
8. A team will be able to add custom data to be stored.
9. All users will be able to stream game films that they have access to.
10. The system can provide dashboards for players and user-created groups.
11. The system provides coaches a uniform way to evaluate their players using the same language, regardless of position.
12. The system allows coaches to evaluate multiple players at once for each game clip.
13. The system keeps track of each play clip and each player involved in each clip and the evaluations for that clip.
14. The system generates a series of assessment scores that normalize player evaluations on and off the field.

**Challenges**

Security, scalability, performance of video streaming, and ease of use are the main priorities for this project. Because this is an enterprise application, users can be expected to scale quickly with each new client. If there is ever a security breach, coaches will cease to use it. If the expected workflow is not easy and convenient, it will never be used by coaches. Only professional football employs people who are completely dedicated to roster management. At all other levels this is taken on by the coach. This tool must work in their workflow for it will be them using the system the most.

**Benefits and Opportunities**

Giving coaches the tools to make data driven decisions will benefit programs in both the short and long term. In the short term, making game decisions based on data will lead to more wins and better player growth. Conversations are centered around data, transparency, and everyone being judged in the same manner week to week. In the long term, roster decision of who to keep and what recruits to target are now much more informed due to them being data driven. Less mistakes on talent are made.

Project Scope

In Scope Features:

* Save game film consisting of multiple games and practice video clips to cloud database.
* Save player stats, grades, and assessments into cloud database.
* Generate dashboards for the team and individual players.
* Allow the transfer of player data between teams and players.

Out of Scope Features:

* Anything with team operations or team centered data outside of dashboards.

[Use the table below to list all known stakeholders and contacts including self. For some projects, 'self' may be the only name listed. If not applicable, remove.]

**Table <Insert #>. Stakeholders**

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
| Simon Codrington III | Project Lead | Oversee everything related to project |
| Melanie Smith | Head Assistant | To help with any necessary research |
| Sia Codrington Walker | Head Assistant’s Assistant | To learn what she can from project lead |

SEE PROJECT SCHEDULE

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |

Project Success Measures

|  |
| --- |
| Project Completion Criteria |
| 1. User can log in and create an organization |
| 1. User can upload game footage and attach to a game |
| 1. User can stream game footage that is attached to a game |
| 1. User can add players to each play and grade them |
| 1. When user logs in they can view data based on players |
| 1. User can add players to their organization |
| 1. When user clicks on a player in organization, they will be able to view all of their data. |
|  |

[Use the table below to list the project assumptions and constraints, if applicable. An assumption is an educated guess that a likely condition or circumstance is presumed to be true. A constraint is a limiting condition or circumstance that defines the project boundaries. Assumptions allow the project to succeed. Constraints restrict or limit the project execution.]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |

Project High-Level Solution

**Introduction**

[Describe in detail the nature of the topic or challenge addressed. Adjust the title of this section accordingly. Be very clear when you describe what is given/known, what the objective is, and what the characteristics of the solution/answer sought are. Include diagrams and illustrations to clarify your narrative. Include a detailed description and examples of the data/input to this project. State any assumptions you made and explain why.]

I would like to be clear I am a former national championship winning football player, and also a state runner up offensive coordinator. There are very few things that I know better than I know football. One thing that I have learned throughout all of my time is that the evaluation system of talent is broken. This is due to the fact that coaches lack the tools and knowledge to be able to adequately assess their talent and truly compare their talent over time. This project seeks to change that. During my time as an offensive coordinator, I invented an algorithm that I called the player production index. It is a tool to grade players, regardless of what position they played, on three different aspects. Using this tool, I can calculate and aggregate data on players’ performance, that way I could be comparing everyone under the same language and keep my evaluations completely and unequivocally unbiased. I achieved this through a somewhat elaborate Microsoft Excel workbook. Before I knew a single thing about software or coding, I knew my statistics and I knew how to use Excel. Therefore, I used Excel and coded it to be able to run my calculations on a play-by-play basis using data cleansing techniques. I applied certain weights to certain attributes per play. I then ran a regression of what the expectation was and subtracted it from the actual player’s value, based on the number of plays they participated in throughout the entire game. This residual value (the difference between expectation and reality) was their score. I then used Excel to print out a report. Because I was dealing with high school varsity athletes, it was very important that the reports be easy to understand, easy to read, and easy to articulate what was going on. I was able to find success with players that everyone else discarded. I was able to prove and show actionable measurable growth. The players looked forward to their scores, and the data-driven conversations we were able to have allowed them to exceed expectations all the way to a state championship game. My goal has always been to take this concept wrap it into a software and allow coaches to be able to do the same thing that I did. During this time as a football coach I found myself interviewed on the news due to my fervor in looking for software solutions that could help me on the field. I knew nothing of software at the time but my heart had me searching for someone who could turn my idea into an application. I did not really find anyone. The thought that, six years later, I would become the person that I was looking for is touching. At the end of the day, I am building this application for myself and all the other people who used the algorithm. Three of which will be suiting up with the Florida State Seminoles this season. I will do my best to get other coaches and organizations to use my product when it is complete. But if every other coach on planet Earth says no, I will be able to look myself in the mirror with my head held high, knowing I’ve fulfilled the goal of that former version of myself.

**Solution**

[Describe in detail the nature of your solution, both in theoretical terms (principles, concepts) and technical terms (UML, flowcharts, pseudocode, code snippets). If the project is entirely theoretical/mathematical, prove every point you make and anchor in external references. If the solution is code or another type of software, provide an architecture of the solution (diagram), clearly labeling and explaining the function and operation of each component. Detail the type of input, output, and the nature of data/information processing. Provide screenshots of correct execution of your code. Include key code snippets and comment on their role and approach to implementation. Detail and reference any external resources used. Summarize this section with a reminder of how your answer/approach/solution addresses the objectives.]

Outside of training, the most time-consuming task a football coach will do is breakdown film. Game film is broken up into small clips called plays, and each play has up to 22 players involved. Coaches tend to watch this game film docked at a desktop computer or maybe sometimes on a cell phone and then they take notes about the game. With this in mind, I believe it is best to make this software a web application that will allow seamless upload of video clips and streaming data collection and data visualization. As it pertains to the architectural makeup of the system it is as follows.

Out of all of the different software architectures that I have researched, I believe the domain driven design approach is the best path forward. This design approach, matched with repository implementation and clean architectural patterns, will give me the best of both worlds as it pertains to plan driven development versus agile. Domain driven design emphasizes decoupling and ease of modifiability and testability. Modifiability, testability, scalability, and ease of use are the key quality attributes that I will be focusing on. Domain driven design, clean architecture, and repository design, accompanied by a code first approach, will allow me to tackle all of these attributes.

For a structural overview of the system by clean architectural standards, the system will be split into layers, and each layer will be its own project. By domain driven design, the layers will be the domain layer, the infrastructure layer which will house our repository implementation and interfaces, the database layer which will be generated based off of the infrastructure, the API layer that will be used to communicate with the database, the domain services layer which would be responsible for back-end logic that is attached to only one singular domain, the application layer which will house the complex application services that cannot be contained in the domain service layer, and of course the user interface layer. The data pipeline and movements throughout each layer will be conducted by way of inversion of control and dependency injection. This detail is crucial for making sure that the system maintains clean, testable, modifiable quality attributes.

To accomplish all of this I will be using a full Microsoft suite of technologies. Asp.net is the framework of choice, alongside Visual Studio and Azure for the cloud platform. The entire back end and each layer will be built in C#, with the database for the structured data going inside of Microsoft SQL Server and the unstructured video data being saved in Azure BLOB storage. Finally, the user interface layer will be built with either Blazor web assembly or Blazor server. If I go with Blazor server, then the API becomes unnecessary. However, Blazor web assembly creates an added layer of security by forcing any outside threats to go through the browser. Blazor server, on the other hand, will give me the absolute best performance. However, I am concerned about the load of streaming, especially as the application scales. For example, having thousands of people using the Blazor Server SSR channel to stream the BLOB storage video proved troublesome. While Azure DevOps offers a solution to scale my servers both horizontally and vertically, this option can be quite costly and may not be the best choice. Most of the application will be tied in back-end logic and, therefore, the last thing that needs to be done will be building the front end. That is a decision that can come at a later date and can even potentially done and implemented in both technologies.

A group of rectangular white rectangles with black text

Description automatically generated

A diagram of a computer

Description automatically generated

A black screen with white text

Description automatically generated

A black screen with white text

Description automatically generated

* Dependency List
  + EntityFrameworkCore
  + EntityFrameworkCore.Tools
  + EntityFrameworkCore.Design
  + EntityFrameworkCore.SqlServer
  + MudBlazor
  + AspNetCore.Cors

Project Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RISK MANAGEMENT | | | | |
| **Event Risk** | **Risk Probability**  **(high, medium, low)** | **Risk Impact** | **Risk Mitigation** | **Contingency Plan** |
| What is the risk? | What is the probability? | What is the impact if the risk occurs? | What can be done to minimize the risk? | What can be done to minimize the impact of the risk? |
| Unauthorized attack | High | Critical | Separation of data with API and use of azure security measures | Use redundant data storage to replace incase of attack |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ISSUES LOG | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | What is the issue? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| 2 | Lack of video storage/streaming service implementation knowledge | The application purpose if to allow users to stream and upload videos. This must work properly and be fast | Use community support and build a separate project to learn how to implement | Simon Codrington | CRITICAL |  |  |  |
| 3 |  |  |  |  |  |  |  |  |

Project Cost and Schedule

* Each sprint to take 2-4 weeks
* Sprint 1: Project setup and develop Domain and Infrastructure layer
  + Generate new projects for each layer
    - All will be class libraries except for the UI (Blazor) and API (ASP.Net REST API)
  + Build all needed models for domain (Player, Game, Organization, Coach, Play, Etc)
  + Define data access repository interfaces in infrastructure
* Sprint 2: Set up API and database layers and complete infrastructure layer
  + Set up Azure database resource
  + Implements interface using entity framework and set migrations of models
  + User interface to build API
  + Implement API consumption in infrastructure
  + Test API and infrastructure layer
* Sprint 3: Implement Azure Blob Storage Service
  + Set up blob storage resource
  + Implement video streaming and upload service
  + Test service
* Sprint 4: Document and finalize use cases
  + Document all models and services
  + Define needed use cases and services for application (add player to organization, grade player game, get player data, etc.)
* Sprint 5: Develop Domain and application services and complete back-end logic
  + Implement all use cases in Domain and Application services projects
  + Test both layers
* Sprint 6: Implement system authentication and documentation
  + Implement authentication system and session identifier
  + Update documentation of system
* Sprint 7: UI development
  + Building of Blazor UI and using MudBlazor components
* Sprint 8: UI Testing and Deployment
  + Test UI and prepare resources on Azure to host
  + Fully test using end to end and regression test
* Sprint 9: Beta Test
  + Onboard neighboring High School to test system

Appendix A – References

[*List all references using APA style.*]

Sommerville, I (2015). *Software engineering* (10th ed.). Pearson.

Bass, L., Clements, P., & Kazman, R. (2021). Software architecture in practice (4th ed.). Addison-Wesley Professional.

Lenzo, M. [Marco Lenzo]. (2024, Jan 4). The Onion Architecture Explained | Should we Use It? [Video]. YouTube. <https://www.youtube.com/watch?v=example>

Appendix B – Copyright Compliance

[For each external technical tool or code used, provide a reference to its copyright policy, clearly showing your right to use it. For each external technical tool or code used, detail how you used it, how you adapted it, how you modified it (if permitted), and why did you use it as opposed to write your own. Only a small portion of your project may rely on external code. When code libraries/packages are used, explain why this was necessary/required/recommended. Seek instructor approval for using external resources prior to beginning to work on the project.]