**Features Overview**

**(Convert these to Agile Stories)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Core Features for Beta Release** | **Core Features V1.0 Release** | **Planned V2.0 Features** | **Under Evaluation** |
| Azure  DevOps  Processes | Use Azure DevOps tools to automate software development and deployment, source control, versioning & releases | Documented procedures and commitment to continuous improvement in development processes |  |  |
| Resume using the Azure DevOps Scrum or Kanban based development process |  |  |  |
| Use package manager for all the open source software distribution & updates needed | Use a package manager to install and update SportsPip specific code |  |  |
| Need an installer for Linux w/ Docker; Strapi, the MEAN stack, common utilities, and the SportsPip apps and services |  |  |  |
| Implement an automated test capability using Azure DevOps |  |  |  |
| Start a Microservices implementation for all the software we are using or building | Deploy release with a complete Microservice architecture |  |  |
|  |  |  |  |  |
| Linux  Platform | Start with basic system logging and reporting capability | Use experience to take the logging and reporting to production level |  |  |
| Remote diagnostics and support to troubleshoot and repair customer systems | Some type of 2FA capability (to provide secure access to on premises server) |  |  |
|  |  |  |  |
|  | Need GUI app and console scripts for basic end user system administration |  |  |
|  | Some type of watchdog and recovery capability for Linux server and Docker |  |  |
| Understand the security requirements to protect system from internal and external users | Network security including firewall and authentication |  |  |
|  |  |  |  |
|  |  |  |  |  |
| Long Running  Video Content with Metadata  Capture | Create PIP with IP camera capture service  (*migrate from Windows service to Linux*) | Strapi scheduled events to start automated IP camera recording |  |  |
|  | Support dynamic allocation of IP cameras to location groups |  |  |
| Evaluate using other types of RTSP streaming devices, such as camcorders | Add support for RTSP streaming from camcorders with Wi-Fi capabilities  (*possibly same as Windows IP camera*) |  |  |
| Create PIP & metadata with iPhone and Android, add iPad and features like unlimited clips (*migrate from Windows to Linux)* | Need to expire and delete transferred files due to potential for larger packages when recording long running events |  |  |
| Add support for multiple contributors to a single PIP, allow content to be added over extended period of time (hours/days) | Strapi scheduled events to enable iPhone and Android upload to a shared PIP with ability to upload photos and audio |  |  |
| Add real time tags to events so they can be reviewed by coaches during the game or in between periods or matches | Use iPad to insert timestamped metadata in shared PIP to match video and play for coaches review |  |  |
|  |  |  |  |  |
| Real-Time  Live Game  Distribution Options | Add real-time file transfer from server to coaches on sideline with tagged data  (for game time review on iPad, Win10) |  |  |  |
|  | Live streaming video and tags to coaches on sidelines from Linux server (for game time review on iPad, Win10) | Allow live streaming video to fans from Linux server using app during game |  |
|  | Console preview screen showing active streams and system utilization |  |  |
|  |  |  |  |  |
| Local Campus Content Access to Win10 and Streaming  Video | Reliable PIP upload and download transfer service between Linux & Win10 editing  (*refactor based on Linux open source*) |  | Reliable PIP upload and download transfer service between Linux & iPad app |  |
|  | Reliable PIP streaming between Linux server & client apps using the local campus streaming service |  |  |
|  |  |  |  |  |
| Cloud Based  Content  Access to Win10 and Streaming  Video |  | Reliable PIP upload and download transfer service between Linux Server or Win10 and the Azure cloud (*refactor of the current Azure service*) | Reliable PIP upload and download transfer service from Linux and Azure to iPad |  |
|  | Automated email and app notification with links to download PIP to Win10 player (or watch streaming video content when available) | Streaming highlight videos to fans from cloud archive after game is completed |  |
|  | Client admin app to manage and maintain Azure content |  |  |
|  |  |  |  |  |
| General  Video Streaming |  |  | Live streaming via integration with OBS or similar services or tech |  |
|  |  | New streaming production console app for broadcast style graphic effects |  |
|  |  |  |  |  |
| Basic PIP  Admin | Add retention metadata to PIP at creation based on category | PIP backup, restore and cleanup of expired content services using NAS | Transcode video from 4K to 1080P video using NAS |  |
| Display PIP in Strapi template by category and filtering (current sprint) |  |  |  |
|  |  |  |  |  |
| PIP Player and Editing  Application | Win10 client with playback features, radial menu and canvas objects with minor compatibility updates  (*enhancements to current app*) | Add new radial menu features to support advanced tagging with voice narration and view any client history |  |  |
|  | Use tagging data to playback video on timeline like Dartfish or Hudl apps | Use radial menu to generate game summary and statistics for level 1, 2 sports analytics for a few selected sports | Create custom radial menus for each sport with scorekeeping and results reporting to sports management sites |
|  |  | Compress PIPs to remove footage that is not included in edited timeline |  |
|  |  |  |  |  |
| Community Features |  |  | Notification and data updates between coaches and players via mobile app |  |
|  |  |  | Social media integration with Twitter, Facebook, and YouTube |
|  |  |  |  |  |
| Advanced  Sports Analytics |  |  |  | API to support sensor data from standard IoT devices and wearables |
|  | Create advanced sports analytic PDF reports which are populated by the data from the tagging application |  |  |
|  |  |  |  |  |
| Licensing and Subscriptions | Allow beta app users to continue use after subscription licensing begins | Some type of licensing features for paid subscriptions and support |  |  |
|  | Support for home and away versions of the Linux server |  |  |
|  |  |  |  |  |
| Graphics Design | First draft graphics design for Strapi, console and Win10, iOS and Android client applications | Production release graphics design for Strapi, console and Win10, iOS and Android client applications |  |  |