**The Next Generation Competition System**

V1

Infrastructure:

* Laptops connected real-time:
  + At least 3 or 4. These are used for
    - bot analysis on the bots we’re going to play against (usually only 3-4 per match)
    - General ratings for all
    - Holistic analysis of game play, bot selection
    - Communication with field in selection
    - Print out rap sheets (need portable printer)
    - Fix data.
* Tablets:
  + Used to gather specific stats described below
  + Need real-time connection and must be fed by master server. This way the tablets contribute constantly to game play.
  + Are configured using Red1, Red2, …
  + Receive json file of template
  + Return json file of data
  + Record lock on data they are taking, and include “break lock” functionality
  + **Key:** UI is completely data-driven from server so it can match server (see data-driven below. We need to build a UI where the widgets can be totally driven from json data provided by the server. This gets rid of all the coordination of data model. The data can either come in the template or in a separate template from the server. Would be nice if it came and configured in every match analysis so we know it’s always up-to-date.
* Communication:
  + hotspots at the top of the stands away from the field worked great ran on 2.4G and were far enough away from the field not to be detected. Cloud server is preferred because everyone has access and it can be used in scouting meetings in the evening.
  + Another option is tethered and Bluetooth access, also using a cloud server. A laptop is USB tethered to the internet, a Bluetooth hub connects the tablet and PC devices
  + Real-time communication is needed for our data to be effective. Generally, the tablet-gathered data was difficult to sync and use.

Information:

* We need a more disciplined way of designing the information model and needs
  + General idea: less data, and only gather data that matters directly for an end use
  + For all data, we need to think and plan its ultimate use. It’s ultimate use will generally be informing game play or informing selection
    - Example: does it need to go through an averaging function or compare function before it provides value
      * Examples:
        + Defenses breached goes to defenses rating for entire alliance
        + Scoring characteristics are fed to “contribution to win” algorithm that calculates score the bot produced. It doesn’t need to be perfect but count the things we care about.
        + Reliability characteristics go to final alliance selection
        + Other stats compilers so we can see strength of robot **in a specific role**
  + Analysis: We have most of this. The recommendations of each bot.
    - How do we get ratings consistent between raters?
    - Most of this should be rating or describing performance characteristics that matter (not just characterisitics)
* Categories of data we gather
  + Analysis:
    - for/with recommendations, we have most of this now
    - Ratings
  + Stats (generally collected on a per-bot basis on the tablets)
    - The general idea is to collect data we can’t get from first and helps us know the bot exactly. The FiRST seed ratings, as an example, often don’t yield the strongest bot.
    - Offensive contribution stats
      * Calculate the specific contribution to score
        + Example in 2016 game:

Only could breaches that contribute to score

High shots

Low shots

On platform

Hang at end

Anything else?

Things we don’t need to have:

Missed shots

* + - Defensive contribution stats
      * Specific defense plays or things it does to contribute
        + Examples:

Blocked shots

Impaired shots

Blocked moves

* + - * For defense, we might also need to rate its abilities as many bots won’t be placed in defensive position until finals
    - Robot reliability and ability to perform
      * An example would be breakdowns or % of match out of commission
      * Our compiler could count the numbers of games it broke down or didn’t show and % of time played. These would help us rank it lower

Schema and App Changes:

* May want to separate fields further into sections:
  + Match: offense, defense, reliability
  + Bot: Selection criteria vs. gameplay?
* May want to rethink through stats compilers. Do we want a separate file? How to specify rankings.
  + On rank form and team forms, do we want our own stats compiled?
* Is include and params file too long? Doesn’t seem to impact performance, but it is long

Little things:

* Red1, Red2, order for tablets should match 1,2,3 display in team eval
* Use Practice round data for a while, but have a feature that turns off its contribution once we get better data