**Organization: 3 Bins and a Board**

1. **Bin1: For MVPs w/ Deadline <1 Year**
   1. *Requirement to Stay:*
      1. Must have MVP name and short description of purpose.
   2. *Note:*
      1. Indefinite projects are automatically given a max deadline 1-year from that day.
2. **Bin2: For MVPs w/ Deadline <6 Months**
   1. *Requirement to Stay:*
      1. Must have detailed MVP description and purpose.
      2. Must have defined MVP features.
   2. *Note:*
      1. Possibly will have scoring method for features
3. **Bin3: For MVPs w/ Deadline<3 Months**
   1. *Requirement to Stay:*
      1. Must have detailed MVP description and purpose.
      2. Must have defined MVP features.
      3. Must have defined Stories, Story Points, and Exit Criteria for each Feature.
   2. *Note:*
      1. Looking into feature scoring method..?
4. **Kanban Board**
   1. Has columns:
      1. Stories
      2. To Do
      3. In Progress
      4. Integration
      5. Testing
      6. Complete
   2. It is the goal for the team to swarm on every task to complete them as quickly as possible
      1. If a task fails integration or testing, it falls back to the development column and goes again
   3. When the tasking is complete, the story must pass the same 5 gates as its tasks did

**Planning**

1. **Address Changes in Bins**
   1. That is, if any deadlines are closer enough to have moved items from one bin to another OR if there are new items thrown into each bin
   2. **Bin 1: MVP Submission**
   3. **Bin 2: Feature Addition**
   4. **Bin 3: Story Addition**
      1. Stories are big picture steps to produce a feature.
      2. Creation:
         1. Stories should be:
            1. Independent
            2. Negotiable
            3. Valuable
            4. Estimable
            5. Small
            6. Testable
         2. Story Examples
            1. User Stories

Card: ‘As a <user role>, I want <activity> so that <value of even doing action>

Conversation: ‘Yeah, the robot needs to lift heavy things too’

Success Criteria:

The phone is silver

The robot lifts heavy things

* + - * 1. Enabler Stories

Test <entities> in <location> because <purpose>

Do <action> for <entity> so that <purpose>

* + 1. Scoring:
       1. Scored relatively on a Fibonacci scale, i.e. 1,2,3,5,8
       2. The model is calibrated by taking the smallest story, one that takes a half-day to build and a half-day to calibrate, and marking that as a ‘1’
          1. Everything else is scored
    2. Exit Criteria:
       1. Exit criteria = Success Criteria
       2. When a story’s tasking is completed, the story is only closed out if the exit criteria is satisfied.
          1. Test that <criteria>
          2. Demonstrate that <this happens>
          3. Verify that when <a role> does <some action> they get <this result>
          4. Given <a context> when <this event occurs> then <this happens>
       3. Stop writing criteria when there is enough to size a story, testing will be too convoluted, or there have already been 2-3 revisions.

1. **Dash Planning**
   1. Pull in stories from the 3-month bucket only
      1. Use capacity and estimation to fill a dash
      2. If there’s any close calls, use value vs effort, RICE and ICE to make a decision.
   2. Give an estimate of how long you think this dash will take
      1. This, along with total hours, establishes burn-down line
         1. This burn down line shows where your work is concentrated and more importantly, how close you were to predicting accurately
      2. Speed\*estimate will give a prediction of how many points should be taken over this dash
   3. Calculate capacity and fill dash with stories until capacity is reached.
      1. Capacity = (full time workers\*8 – each day off)
      2. The ‘estimated’ story points are basically a recommended average, which can be ignored if this dash is particularly slow/fast
   4. Build Sprint Goals
      1. Address risks and present possible mitigations (1-5)
      2. Take a team-wide confidence vote on finishing this sprint’s goals (1-5)

**Execution**

1. Every morning, go over last 24 and next 24 [15 mins]
2. There are 5 automatic events triggered by the board:
   1. Planning Trigger:
      1. Counts tasks on To-Do and if falling below a certain threshold (4-6 hours: half a day of work left), a planning session will be triggered (see above)
         1. All remaining tasks are not counted for in the capacity as it is relative.
   2. To-Do limit:
      1. Limits number of to-do tasks on the board
      2. Is determined by capacity at the time of planning
   3. Work-In-Progress limit [Default: 3 per worker]
      1. Limits number of ‘in progress’ tasks, which includes integration and testing
   4. Feature Freeze
      1. If a deadline is within a specified amount of time [1 week], no other to-dos can be added
      2. Also triggers a ‘Triage’, where product owner selects which features can continue to final product and which ones will be covered in next release
         1. That is, only complete must-haves and drop all nice-to-haves
3. If all tasks are done and the story also makes it through integration and testing, it can only be removed from the board if it passes its exit criteria

**Important Notes**

1. Regarding the bins, if the bin is a location where an entity is added, that specific entity is liable to change/be removed/be amended.
   1. this also means that during planning, a project in the 1-Year bin can be upgraded to the 3-month bin (but not yet to the board) if the proper ‘Stay’ requirements are met
2. There’s no requirement for something entering a bin, like having defined diagrams/calculations. This will be taken care of in relevant tasking, though it’ll probably help make for better designs and save some time if you were to include them with the initial submission.