## 26/03/2021

To abstract away the difference between Scrum and Kanban and make switching between them at a later time possible, the game time will pass in periods that can be mapped to a week in Kanban or a Sprint in Scrum.

I have decided to defer the decision about whether the player takes on the role of CTO or multiple roles, and I think this can be accommodated easily by either automating some decisions in the CTO role or making them available via a role notification bar if the player is taking on multiple roles.

The most important thing is to further refine the scenarios of gameplay. All player interaction will be represented by events that they need to respond to via a popup or notification icon.

The events will be categorized as:

* Project start events. A series of actions/decisions to perform before development begins. Such as:
  + Accept job offer.
  + Meet the team.
  + Meet the customer.
  + Project setup (pipelines, tests, etc.)
  + Prioritisation.
  + Agile training.
* Scrum/Kanban events
  + 3 Amigos
  + Refinement
  + Sprint Planning - can also be used to represent weighing up customer priorities against others
  + Standup
  + Review – customer feedback, bug generation
  + Retrospective - this could be an opportunity to introduce new concepts as if they were introduced by the team.
  + Firefighting
* Intermittent events
  + Hire new staff
  + Go to a conference, or send employees
  + Whiteboard sessions

Game variables:

So far, the game has 1 property, developer skill level. This is averaged across the team to calculate how much work is done. Some further property ideas:

|  |  |  |  |
| --- | --- | --- | --- |
| Property | Game Entity | High | Low |
| Skill | Team member | More work complete, fewer bugs | More bugs, leading to firefighting. |
| Quality mindset | Team member | Slight decrease in output, but fewer bugs. Increases test coverage. | Quick and dirty approach, more bugs. |
| Collaboration / communication | Team member | Team becomes more efficient and scrum events more useful. Flow improves. | Less efficient etc. |
| Psychological safety | Organisation / team | Better refinements, standups, design sessions and retrospectives, as communication is improved. | Poor communication, design |
| Number of bugs | Project | Less feature work, more firefighting. | Team can focus on delivering features. |
| Test coverage | Project | Fewer bugs | More bugs, leading to firefighting. |
| Flow (could just be an average of team members’ collaboration properties) | Team | Team work together to move stories across the board, rather than working separately. Increases chances of getting stories to Done. | Team work independently on different stories. More likely to not complete work. |
| WIP (could simply be the inverse of flow) | Team |  |  |

Next steps:

* New game properties:
  + Introduce bugs at end of sprint, represented by new property: **number of bugs**.
  + Introduce new property: **test coverage**.
  + Introduce new property: **quality mindset**, which impacts **number of bugs** and **test coverage**.
  + Introduce new property: **psychological safety**, which for now will determine how good a Retrospective was with a simple score value. Later this can be used to generate Retrospective outcomes.
  + Introduce new property: **collaboration**, which impacts **flow**.
  + Introduce new property: **flow**, which for now will be a simple score value for the sprint. Later this can be used to modify the way stories are completed.
* Add views to display the above properties as the sprint is in progress or after it has ended.
* Come up with ideas about how a prioritisation session can be represented in the game.