How to Agile

Goals

- Create value for the player early
- Don't be afraid to change project direction
- Deliver a near shippable game every week and as soon in development as possible
- Only do what is necessary to accomplish our goal
- Own your work and features, be responsible for them
- Analyze our approach to work and improve it often

Terminology

Story — kind of like a feature but from the side of the user. Is not bound to one discipline. Can be split into multiple smaller stories if its too big to fit in one sprint. Every story has an owner that is responsible for its quality upon completion.

Examples:

- As a player i want to control a character that moves around the environment.
- As a writer i want the dialogue editor to allow me to create quests and item checks quickly.

Task — things that have to be done to execute a story. Can be bound to a discipline. Are always bound to and make up a story.

Examples:

- Write code that makes the player move when a key is pressed
- Draw the animation of player movement

Backlog — an ordered list of stories that make up the game. It is regularly sorted so that stories that have the biggest bang for their buck(least effort to develop and high value to the player) are at the top. When you have an idea for a thing that would be cool to add to the game it goes into the backlog.

Sprint — a weekly TO DO list. Sprint contains stories that have to be done this week by the team. With rare exceptions all stories within a sprint must be complete by the time it ends.

Epic — a way to group sprints by themes. Independent of sprints and releases.

Release — a way to group sprints by deadlines. Independent of sprints and epics. Releases happen once every few sprints and are the deadlines for having the game look and feel complete (not all features are there but the ones that are there are finished with no placeholders).

Story points — an abstract measure of story/task complexity. The values of tasks and stories are estimated by the team before sprints. Are used to determine when the sprint backlog is full (the sum of story points of all stories and tasks must not exceed what is doable by the team)

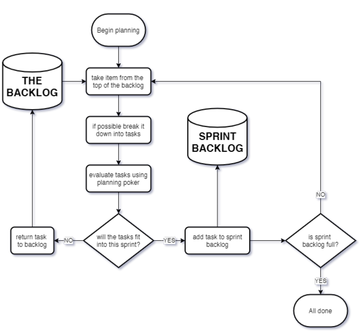
Week Structure

Start of week:
Evaluate upcoming tasks using planning poker
Fill up sprint backlog

Every day:
Do daily standups

End of the week:
Do sprint review at the end of week
Do sprint retrospective

Filling up the sprint backlog



After reaching the end you can stop worrying about planning and get to work! Repeat at the beginning of each weekly sprint.

Planning poker

This is done to evaluate the complexity of tasks when filling up the backlog. This helps ensure the team does not bite off more than what they can chew when planning a new weekly sprint.

Story points

Before you understand planning poker you have to understand story points. They are an abstract and relative measure of task complexity.

Why we use them?
Its easy to estimate how *complex* or *difficult* something is *relative* to something else. However, we usually have trouble when trying to estimate how *long* something will take in *hours of work*.
We don't have perfect days/weeks of work, we get distracted, we encounter bugs, we have to iterate, multiple people can work on one thing, and also everyone takes different amounts of time to accomplish things. Story points try to eliminate the hour counting by being one step removed from them and only count the amount of work relative to a real task people know.

Baseline task
As a baseline we use a task that everyone has experience completing from start to finish. Assign it a static number of story points that the team can agree on. The baseline task must not change throughout the project to ensure the size of a story point remains constant.

Example:
Team decides the baseline task is worth 5 story points.
That means that when the team decides a task in a story is worth 3 story points it will be approximately 60% as hard to finish and be 60% as complex as the baseline task.

Planning poker itself

0. The person in charge of evaluation posts a task/story in the chat and prompts everyone to respond to it.
1. Everyone comes up with a relative point value for the item being evaluated. You can only choose numbers from the fibonacci sequence (reference and explanation below).
2. Everyone posts their value prediction in the planning poker discord chat with ||spoiler tags||
3. When everyone has posted their prediction look at what other people posted. Discuss why you chose the value.
4. Go to step 1. and repeat it until a consensus is reached
5. Write down the agreed upon value in the backlog item card. Go to step 0. and repeat until items that will be picked up in the next few sprints are all evaluated.

remember to reevaluate the point value of a story when it gets broken down into tasks/smaller stories

> The fibonacci sequence numbers are:
> 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

we use them because people are bad at accurately estimating bigger numbers and there is no point in arguing over small details (there is significant difference between 1 and 2, but almost no difference between 34 and 35), we use a sequence that separates the numbers more and more as it goes up.
most other sequences either grow too fast, too slow, are obscure or produce awkward non integer values.

Daily standup

Happens every working day, a bot will ping you when its time to do the standup.

Write the following in the appropriate discord channel:

1. What you did on the project yesterday
2. What are you gonna be doing today
3. What is standing in your way (things from teammates/life events)

Sprint review

- Everyone plays the game in its current state, says what they think about its current state and if the stories in the sprint were successfully executed.
- Showcase you features to the rest of the team
- Discuss what is the next direction the development should take.
- Reorganize items in backlog, add new ones.

Sprint retrospective

- Examine the work process that took place during the sprint.
- What could be done better?
- What should we stop doing?
- What should we continue doing?
- What should we start doing?
- How can we improve the way we work together during a sprint.
- Write down a few things to do next time that result from the discussion above.
- Do them during the next sprint.