Agile, Scrum and Kanban for Video Game Development

A tour of what agile is and what can be applied (or not) to video game development.
Clinton Keith - Background

• Full-time agile trainer and coach for video game development
• 20 Years of Video Game Development Experience
• Introduced the Video Game Industry to Scrum and Kanban
• Author of “Agile Game Development with Scrum”
What is Agile?

- It’s a set of values and principles (link below) for developing products using short iterations, which
  - Are like short projects
    - Include design, code, art and testing
  - Use “inspect and adapt” practices to adjust the project plan and development practices
  - Focus on adding features in a value prioritized way (fun first)
  - Include frameworks such as Scrum and Kanban that best fit the complexity and uncertainly level of work

http://agilemanifesto.org
What is Scrum?

- It’s an implementation of the agile values and principles
- It defines three roles, four meetings and a few artifacts used to kickstart an agile adoption
- It focuses on cross-discipline teams iterating in “sprints”
Why Scrum for Video Game Development?

- Focuses on “Finding the Fun”
- Reduces wasted effort
- Eliminates death marches
- Engages developers
- Creates transparency

Why Scrum for Video Game Development?

Focuses on “Finding the Fun”
Reduces wasted effort
Eliminates death marches
Engages developers
Creates transparency

Chapter 3
The Scrum Cycle

Sprint goal (features)
- Jump
- Fly

Sprint planning

Sprint backlog (tasks)

Daily Meeting

Sprint 1-3 weeks

Improved Game

Progress is tracked through iterations (sprints) that demonstrate real progress every 1-3 weeks.

Planning is captured in a “Product Backlog” that allows the plan and game to be continually synchronized.
Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the players
- Prioritized by the Product Owner
- Reprioritized at the start of each sprint

This is the product backlog
Scrum roles

- Scrum Team commits to accomplishing sprint goals, with quality.
- The stakeholders outside the team commit to letting them do that.
The Developers

- Commit to a sprint goal with Product Owner and does everything necessary to achieve that goal
- Autonomous on how to achieve their goal.
- Remove most impediments
- Intensely collaborative.
- Most successful when working in one team room with long-term, full-time membership.
- 7 ± 2 members.
- Attend all sprint meetings
- Grow their ability to self-organize
The Scrum Master

- Facilitates the Scrum practices, meetings and artifacts.
- Helps resolve impediments
- Creates an environment conducive to team self-organization
- Captures empirical data
- Shields the team from external interference and distractions to keep it in “the zone”.
- Enforces time-boxes
- Keeps Scrum artifacts visible
- Promotes improved practices
- Has no management authority over the team
- Supports and guides the Product Owner role
- Coaches & guides the team on agile/Scrum principles
- Challenges the organization to approach the agile values

© 2008–2016 Clinton Keith
Product Owner Duties

- Establishes a shared vision between the team and stakeholders.
- Is responsible for the long-term schedule of the game’s development, using the metrics from team output and the product backlog (more below). Continuously prioritizes and refines the product backlog.
- Conveys a shared vision.
- Represents the players and stakeholders.
- Participates in all Scrum meetings.
- Is a member of the Scrum team and can take on team tasks.
- Accepts or rejects sprint results.
- Communicates status externally.
- Terminates a sprint if needed.
- Ensures that the items in the backlog are relevant to their features, or that their feature is dependent on, are tracked, sized and prioritized accurately.
Sprints

- Scrum projects make progress in a series of sprints
- During the sprint, the team does
  - Animation
  - Coding
  - Testing
  - Level design
  - and so on
- After each sprint, the improved game can be played / demoed
Always deliver

• You must have a potentially demoable / playable game at the end of each sprint

• Do not miss the end of the sprint
  • The deadline is sacred
  • Functionality may vary
Reciprocal commitments

The team commits to delivering some amount of functionality

The business commits to leave priorities alone during the sprint
No changes during a sprint

• What the team commits to—and what the Product Owner agrees to—during sprint planning should be what is delivered

However, keep in mind that

• We start with vague requirements
• Our understanding of those requirements is refined during the sprint
Sprint length

- Most common lengths:
  - 1, 2 or 3 weeks
- You can change your sprint length, but not every sprint

Factors to consider...

- How long the business can go without changing its mind
- Amount of uncertainty on the game
- Ability to reliably predict effort on tasks three weeks out
- The overhead of planning, executing and reviewing once a week
- Pick a length that spreads intensity appropriately
Intensity varies over time

By delivering value prioritized features and addressing debt every 1-3 weeks, Scrum creates a sustainable and measurable pace that can eliminate death marches.
The Sprint Cycle

- Sprint Planning Meeting
- Daily Scrums
- Sprint Review Meeting
- Sprint Retrospective
- Sprint Planning Meeting
The “overhead” of sprint planning, review and retrospectives should be less than 10% of total team time.

These meetings should be engage everyone, not run by one person.
Sprint Planning Meeting

• Team selects items from the product backlog they can commit to completing
• Sprint backlog is created
  • Tasks are identified and each is estimated
  • Collaboratively, not done alone by the Scrum Master
• Very high-level design is considered

As a player I want punches, reactions and blocks synchronized, so that fighting looks natural and realistic

Create close punch animations (12 hours)
Tune attack percentage in AI (4)
Remap controls so attacks are on free buttons (4)
Tune block and reaction animations to be same length (2)
Managing the sprint backlog

- Individuals sign up for work of their own choosing
  - Work is never assigned
- Estimated work remaining is updated daily
- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time
  - Break it down later
- Update work remaining as either
  - More is known
  - Items are worked on
## Task boards

<table>
<thead>
<tr>
<th>Story</th>
<th>To Do</th>
<th>In Process</th>
<th>To Verify</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>As a user, I... 8 points</em></td>
<td>Code the... 8</td>
<td>Code the... LC 8</td>
<td>Test the... MC 8</td>
<td>Test the... SC 8</td>
</tr>
<tr>
<td></td>
<td>Test the... 8</td>
<td>Test the... SC 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code the... 4</td>
<td>Test the... SC 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test the... 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>As a user, I... 5 points</em></td>
<td>Code the... 8</td>
<td>Code the... DC 8</td>
<td>Test the... SC 8</td>
<td>Test the... SC 8</td>
</tr>
<tr>
<td></td>
<td>Code the... 8</td>
<td>Test the... SC 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test the... 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test the... 8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Burndown charts

- Primary method of tracking progress
- A burndown chart shows how much work is left as of various dates
The Daily Scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up

- Not for problem solving
  - Whole world is invited
  - Only Developers, Scrum Master and (a well behaved) Product Owner, can talk

- Helps avoid other unnecessary meetings
Everyone answers 3 questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are not status for the Scrum Master
- They are commitments in front of peers
The Sprint Review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world
Typical review results

• Restore unfinished functionality to the Product Backlog
• Remove functionality from the Product Backlog that the team unexpectedly completed
• Reformulate the team
• Re-prioritize the Product Backlog based on what we find is fun (or not)
• Expand or cut features
Sprint Retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - Scrum Master
  - Product owner
  - Developers
  - Possibly customers and others (invitation only)
Sprint Resets

• If change cannot be kept out of a sprint...
  • The sprint may be reset
  • Don’t substitute one feature for another in the existing sprint!
• An extreme circumstance, not done very often
• Raises visibility of priority changes

After resetting...

• All work-in-progress from the current sprint is set aside
• Work might revert to where it was at the end of the prior sprint
• Next step is to plan a new sprint
Abnormal Terminations

- Team can abnormally terminate if...
  - They feel they cannot meet the overall sprint goal
- Management can abnormally terminate if...
  - Business priorities change
Releases

- Small or live games can release to the market every sprint, if desired.
- Larger games will bring larger features to a “magazine demo” state at least once every three months.
Higher priority features are broken down into sub-features that can be finished in a sprint.

Lower priority features are not broken down until later, as we learn more.
User Stories

Some developers use “User Stories” to capture player-facing features
These drive conversation and keep us focused on the player

One or more sentences in the everyday language of the end user that captures what the user wants

Use this template

“As a <user role>, I want <goal> so that <reason>.”
Some sample user stories

As a player I want punches, reactions and blocks synchronized, so that fighting looks natural and realistic.

As a player I want to know which of my friends are playing this game.

As a content creator, I want the asset validation process to recompile scripts so that I know if some of them reference deleted assets.
A project is a series of releases

- Larger (or pre-deployed) games will often have stages of development (especially pre-production and content production)
- These are managed through releases
Release planning on long projects

- On a multi-year game, break the total project into a series of shorter interim internal “releases”
  - Three months is a good horizon
- For each release, establish one or a few major feature deliveries
  - “Epic” user stories work well for this, such as:
    - As a player I want online multiplayer so I can connect to the internet and play against other players online.
    - As a player I want to engage enemies in hand-to-hand combat.
    - As a player I want to drive a car around the city.
Scaling Scrum for Large Teams

- Scalability comes from teams of teams
- Practices that work
  - “Scrum of Scrums” meeting
  - Synchronized Sprints
  - Product Owner hierarchies
  - Guilds
Scrum of scrums
Running the scrum of scrums

Attendees
- Each team sends an individual contributor
- If four or fewer teams, it’s OK to send a Scrum Master also
- Rotate based on whose skills are needed most

Frequency
- Some say daily
- I usually do these MWF or TuTh
- These are problem-solving meetings
- Not time-boxed to 15 minutes

Agenda
- Everyone answers four questions
- Attendees discuss the product backlog for the scrum of scrums
The four questions

1. What has your team done since we last met?
2. What will your team do before we meet next?
3. What’s in your team’s way?
4. What are you about to put in another team’s way?

Rule:
No names during this discussion
Synchronize Sprints

• Don’t stagger sprints like this:

• Synchronize sprint starts instead
A hierarchy of Product Owners

Chief Product Owner
- Visionary for the game
- Prioritizes major features, which leads to which teams form
- Manages release schedules

Feature Product Owner
- Works with teams implementing features daily
- Can work with up to three teams
Augment with Guilds

- Beyond a certain project size, augment the team structure with orthogonal, virtual teams (Guilds)
  - Programming guild
  - Audio guild
  - AI guild
  - Scrum Master guild
- Informal or semi-formal at best
- Meet periodically
- Discuss and resolve issues related to their specialty
<table>
<thead>
<tr>
<th>Team</th>
<th>Team</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmers</td>
<td>Animators</td>
<td>Testers</td>
</tr>
<tr>
<td>Audio Engineers</td>
<td>Scrum Masters</td>
<td></td>
</tr>
</tbody>
</table>
Kanban for Content Production

- While sprints work well for developing new features, content production in the later part of developing a game doesn’t fit well within a sprint time-box
  
  - Content is more predictable and the flow of work more manageable

  - Assets don’t fit well into a 1-3 week time-box
A kanban is a visual mapping of a flow of work for a class of assets (e.g. models, characters, levels, etc.). Lean practices, such as limiting work-in-progress, can help teams find continuous improvements in the flow of work and how they work together.

<table>
<thead>
<tr>
<th>On Deck</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Texture</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bug</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Asset</th>
</tr>
</thead>
</table>

© 2008–2016 Clinton Keith
Scrum and Kanban

• Scrum and Kanban are compatible with one another

  • What’s the same:
    • Both have Product Owners to prioritize the work and Scrum Masters to coach and support the team.
    • Teams will establish cadences to regularly hold reviews and retrospectives

  • What’s different:
    • Instead of Sprint planning, teams will plan on demand when they need to pull in more work
    • Instead of measuring & optimizing the amount of features added every sprint, we measure & optimize the total amount of time an asset takes from start to finish.
More Info

Clinton Keith
Clint@ClintonKeith.com
www.ClintonKeith.com
@ClintonKeith