AGILE & KANBAN IN COORDINATION
Team Background & History

- 18 Engineers
- Relatively mature and expansive codebase
  - C# / .Net
  - MS Team Foundation Server (TFS)
- System – 5.0
  - Over 4 years in development.
  - Large scale feature upgrades = 60% of the work.
- Over the last 2 years we have worked to transition from a “Laissez-faire” waterfall team to a simple and well tuned Lean / Agile team.

**MY ROLE:** My role in the team is both as one of the development team managers and as Agile coach for the company as a whole.
How We Got Here

- What was previously called agile was essentially the team holding a 15 minute standup daily.

- We started with three teams of 5, 6, and 7 developers respectively.

- Eventually we started to work as one large team.

- Create 2 Groups - defined by the size and type of work they would commit to.
Defining Agile - Adding in Kanban

Iterative Agile
1. User Stories
2. Acceptance tests
3. Iterative Development
4. Burn Down Charts
5. Story Boards
6. Daily stand-ups
7. TDD / Unit Tests
8. Continuous integration

Kanban
1. User Stories
2. Acceptance tests
3. Iterative Development
4. Burn Down Charts
5. Kanban Boards
6. Daily stand-ups
7. TDD / Unit Tests
8. Continuous integration
Organization & Product Ownership

- **Product Owners Team**
  - Comprised of one of the team managers and two developers/systems engineers.

- **Random Sampling – User Story Creation / Estimation**
  - More XP style approach to User Stories, we involve portions of both the Iterative and Kanban teams in the story creation and estimation process.

- **Standing Meeting**
  - Every other Wednesday for 2 hours of story planning and estimation.

- **Team Synchronization**
  - Since the teams worked together to estimate and define User Stories, we are able to keep both teams in sync.
Iterative / Kanban Development

- **Iterative Team**
  - Responsible for all large-scale projects.
  - Architectural Roadmap / Structural Changes
  - Limited to 2 or 3 WIP Projects.
  - Iterations are 2 weeks, and managed using a typical project board.

- **Kanban Team**
  - Small feature requests along with bugs and change requests.
  - This team uses a Kanban board that manages development process only.
  - The team maintains an evolved Kanban focused 15 minute standup daily in the same area as the Iterative team.
  - They maintain a cycle time and lead time metric integrated with our story point system.
WIP Limits – With Story Points

- **Board WIP limits**
  - Constraining the number of stories allowed in each queue.
  - Imposing a limit to the maximum amount of points in certain queues.
  - The two primary queues we were concerned with were the Development and Verify / Accept queues.

- **Team WIP Limits – By Story Size**
  - Only stories of a certain size (8 Points) and below would be allowed on the board.
Metrics – Pseudo Metrics

- **Velocity / Pseudo-Velocity**
  - Velocity – Calculated as normal for the Iterative team.
  - Pseudo-Velocity – Calculated from time slices of the Kanban team’s board.

- **Cycle Time / Pseudo Cycle Time**
  - Cycle Time w/ Story Points – Calculated using a weighted average approach for each story point size.
    - ie. 1 pt Avg. = .35 days, 5 pt. Avg. = 3 days weighted average = (0.35 + 3) / 6
  - Pseudo Cycle Time
    - Calculated using iteration start date and end date.
Release Trains

- Quarterly Potentially Shippable Increments (PSI)
- 5 full iterations per PSI
- No “Hardening Sprint” / Kanban Team Instead
- Kanban teams Pseudo-Velocity Used for Planning
“In 9 months we have seen a steady improvement in cycle time and pseudo-velocity of the Kanban team bringing them in line with the performance of the Iterative team.”

Results
- Teams within close parity in 7 to 9 months.
- Dramatic in numbers but the amount of motivation and energy it has provided for the team has been immeasurable.
- Team self organized around commonly created goals.
- The team reached out beyond their charter asking for more work and more complicated work.
Iterative & Kanban - A Model

- Kanban, Scrum-ban and all kinds of other -Ban ideas.
- Mixing and blending processes is often suggested.
- Running both processes in synchronization, and not blending them, has been highly valuable for our organization.
- Adding synchronized Kanban alongside an Iterative team.
  - Even out our iterations
  - Create a productive and healthy work environment where we are able to meet our customers’ needs.
Questions?
Email: rpolk@rallydev.com
Blog: http://www.spryyeti.com
Twitter: spry_yeti
LinkedIn: rpolk – http://linkedin.com/rpolk
Resources & References


