DEVELOPERS AND VALIDATORS WORKING REMOTELY TOGETHER ON A SINGLE KANBAN BOARD

Becky Berndt
Intel Corporation
INTRODUCTION

Becky Berndt
Software Engineer / ScrumMaster
Intel Corporation
Becky.Berndt@Intel.com
About a year ago:
• Development (BSTEs) had a scrum team
• Validation each had a separate scrum team
• Poor communication
• Developers didn’t know what was being tested or what tests were still needed
• Validators were frustrated that features unexpectedly changed
• Resentment, lack of trust
• Took as much as a year to test new features
• Defects filed on old code
• Task-switching between old and new code
• Validation not willing to take this risk to achieve better communication
Dream team:

- Developers and validators on the same team, communicating, planning, giving status, talking about priorities, improving, and succeeding together
- Shifting timelines left
- Still BSTEs
KANBAN

Why?

• React quicker to change
• Reduce overhead time
Kept what worked:
• Scrum roles
• 3 week sprint cycles
• Standups (Mon-Thurs)
• Team member led retrospectives

Added:
• Custom Kanban columns
• WIP limits
• Acceptance Criteria for each column
Team restructured, validation work was moved to a team in Poland

Challenges:
- New members of our team, new to product and agile practices
- 8 hour time difference
- Majority of existing functionality had no associated tests
- Tests that did exist did not run
- Rumors
However they were eager and willing to take on the challenge. I set them up as part of the team now consisting of both developers and validators!

**Change:**
- Added “test in progress” and “test done” columns
- Set some decent sized WIP limits
- Overviews of our product and processes by several team members
- Shared documentation
- Shared stand-ups and retrospectives
- Freedom to ask anything, any time – still holds today
- Off we went
<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Do</td>
<td>Exit Agreement: Acceptance criteria defined using acceptance criteria template with sign-offs. Tasks defined. Tests defined.</td>
</tr>
<tr>
<td>Analysis Done</td>
<td>Exit Agreement: Tests written and passing.</td>
</tr>
<tr>
<td>Development In Progress</td>
<td>Exit Agreement: Tested and verified by validation in end of sprint build.</td>
</tr>
<tr>
<td>Development Done</td>
<td>Exit Agreement: Tested and verified by validation.</td>
</tr>
<tr>
<td>Test In Progress</td>
<td>Exit Agreement: Tested and verified by validation.</td>
</tr>
<tr>
<td>Test Done</td>
<td>Exit Agreement: Tested and verified by validation.</td>
</tr>
<tr>
<td>Accepted</td>
<td>Exit Agreement: Tested and verified by validation.</td>
</tr>
</tbody>
</table>
Problem solved, right?

Our goal is to have validation test code changes before developers check the code changes into the master repository.
• Validation was new to the project
• Things happened fast
• Validation didn’t want to flood the board
• Backlog of validation work
• Leadership changes
• Dev had to keep up with project stack changes
• Confusion on priorities
• Hard to get to know remote team
• Differences in expected work item behavior
• We were not behaving as one team
• Cycle time was too high
• Validation jumped right into the team member led retrospective rotation
• Discuss cycle time & throughput at retrospectives
• Talk often
• Break down work items
• Communicate better around what it means for a work item to be done
• Video retrospectives early on, and recently added video stand-ups
• Spreading the knowledge
• Product Owner holds weekly sync meetings
• Hot pink work items
Team agreed to the following:

- Talk about hot pink work items in stand-up
- <= 2 work items at a time per person
- Work item is moved to analysis ONLY when someone is ready to start working on it for more than 20 minutes
- Work right to left across the Kanban board, focus on finishing
- If your work item is blocked by another team, it goes back to “todo” until it is unblocked
- If your work item is blocked by a team member, talk to them
ARE WE THERE YET?
Continue to improve:

- Get better at analyzing work items early
- Back off of functionality changes
- Force the change to validate code changes before check-in
  Bug free?
- Continue to be available to answer questions
- Think and act like we are all one team
- Cycle time of 14 days
**FUTURE KANBAN BOARD**

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Development In Progress</th>
<th>Development Done</th>
<th>Test In Progress</th>
<th>Test Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 of 4</td>
<td>9 of 10</td>
<td>12 of 20</td>
<td>18 of 20</td>
<td>0 of 4</td>
</tr>
</tbody>
</table>

- Exit Agreement
- Code written.
- Unit tests passing.
- Development build passing.
- Code reviewed - "Ship It!"
- Submitted to master

- Exit Agreement
- Tests written and passing.
- Tested and verified by validation
- Submitted to master

- Exit Agreement
- In end of
• Communication!
• Adapt to change
• Respect
• Make full use of the window of shared work hours
• Break down stories
• Share the knowledge across the entire team
• Always experiment and strive to improve
• Share work items
• Share resources
• Show appreciation
Experience Report:
agilealliance.org/resources/experience-reports/
developers-and-validators-working-remotely-together-on-a-single-kanban-board/
Becky.Berndt@Intel.com

MORE DETAILS AND QUESTIONS
KUDOS TO YOU