The Army Game Studio’s Agile Process: A Retrospective

Jay Olive
Development Lead, Army Game Studio
The Army Game Studio adopted Scrum because of a growing Team, growing Project sizes and the creation of a common Product line.
The Past

• Small Team
  – 15 People

• Half Dozen Concurrent Projects

• Tasking by Email or Over-the-Shoulder
  – Project Leads would go straight to Developer
  – This “process” remained largely unchanged as we continued to grow in size and project scope.
CBRN Dismount

- CBRN Dismount Prototype
- A Couple of Handheld Sensors
- Multiplayer
- Customer Wanted UE3
- Very Quick Turnaround
TRICT

- RG33 Simulator
  - Crew Skills
    - CROWS II
    - FBCB2
    - Driver
    - Commander
  - Rollover
    - Motion Platform
    - Egress Safely
TRICT / Origins

- Bulk of Requirements were given after we were funded
- Customer wanted to use Unreal Engine 3
  - No multiplayer AGS reuse platform to build from
  - No large environments to reuse
  - No CROWS II weapon simulation
- From napkin to drivable prototype in 18 months
TRICT / Initial Development

• Broken into Modules
  – Environment
  – Weapon Simulation
  – Multiplayer / Instructor Workstation
  – Vehicle Simulation

• Functional Leads were responsible for their own teams
  – No one person was ultimately responsible for a module...
TRICT / 6 Months In

• Behind Schedule
  – Environment
  – Multiplayer / Instructor Workstation
  – Vehicle Simulation

• On Schedule
  – Weapon Simulation
    • One Developer

• Poor Demonstrations from Developers
• Inconsistent Feedback from Project Leads
• Team was not communicating well
  – Code was not being reused between teams
TRICT / It’s not You, It’s Me...

• Not Developer’s Fault
  – Their features were not being critiqued often enough
  – They were not being time-boxed appropriately
  – They didn’t know who to communicate with

• Not Area Lead’s Fault
  – They were managing requirements
  – They were managing pieces of Modules that were in their specific “functional” area
AAVP3

- America’s Army Visualization Platform
  - Standardize UE3 Product Line
  - Common Baseline
  - Easier Asset Reuse
- Multiplayer
  - Instructor Workstation
- Vehicles
- Dismount
AAVP3 / Scrum

- Scrum was a process that fit our development style:
  - Requirements are never in stone, and most of the time are not completely known when funding occurs (design late)
  - Iterative, supported our customer feedback loop.
- Area Leads
  - Only role would be to critique work in their area.
- Developers
  - Could be time-boxed. This gave them a goal and sense of urgency for even small tasks.
  - Bugs were triaged quicker and in a standardized fashion.
  - Tasking could only come from Product Owner.
- Product Leads
  - Could see frequent demonstrations and could give these builds to customers for feedback.
  - Could effectively communicate what they wanted, straight to the developers, through User Stories.
AAVP3 / Sprint Teams

• Two teams
  – Not based on functional areas or specific projects, instead based on product features within a common product line.
  – Instructor Workstation / Vehicle Team
    • Programmers / Artists / Testers
  – Environment / Pawn Team
    • Programmers / Artists / Level Designers / Testers
AAVP3 / Bug Reporting

- Bugs are reported and end up in the product backlog.
- They are triaged by Scrum-master every day and the appropriate Sprint Team is asked whether they can take it on
  - Bugs for features in development the current sprint are sent directly to that developer.
Tools

• We need software to track:
  – User Stories
  – Developer Progress
  – Sprint Cycles
  – Bugs

• We need software that allows us to communicate:
  – Design
  – Customer Feedback / Meeting Notes
Tools / JIRA

A screen shot of the JIRA interface showing various issues and tasks managed within the project. The interface includes a planning board with specific tasks like "As a Level Designer, I would like to wrap up most of the environmental work on the Urban Map," and "As a CIVCROWS player, I don’t want the mouse wheel to change positions in the vehicle." The sprint 2012.4 dashboard is also visible with details such as the start date, end date, and time estimate for the project. The technical task% is also highlighted with a value of 83.
Tools / Confluence

## AAVP 2010 - Sprint Goals & Artifacts

Past sprint artifacts for future sprint planning purposes.

- **Sprint 2010.5**
  - Sprint Goals 2010.5

- **Sprint 2010.4**
  - Sprint Goals 2010.4

- **Sprint 2010.3**
  - Sprint Goals 2010.3

- **Sprint 2010.2**
  - Sprint Goals 2010.2

- **Sprint 2010.1**
  - Sprint Goals 2010.1

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Story Points (Ideal Man Days)</th>
<th>Duration (Calendar Days)</th>
<th>Duration (Work Days)</th>
<th>Enviro / Pawn Team (Time Estimate - Ideal Man Days)</th>
<th>UI / Vehicle Team (Time Estimate - Ideal Man Days)</th>
<th>Release Date</th>
<th>Release CLI#</th>
<th>Sprint Notes / Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010.5</td>
<td>218</td>
<td>39</td>
<td>25</td>
<td>125</td>
<td>109</td>
<td>01/10/11</td>
<td>37552</td>
<td></td>
</tr>
<tr>
<td>2010.4</td>
<td>178</td>
<td>35</td>
<td>25</td>
<td>74(Enviro) / 104(Pawn)</td>
<td>88.1</td>
<td>12/10/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010.3</td>
<td>256</td>
<td>38</td>
<td>26</td>
<td>87(Enviro) / 96(Pawn)</td>
<td>78.5</td>
<td>11/05/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010.2</td>
<td>NA</td>
<td>30</td>
<td>22</td>
<td>80(Enviro) / 99(Pawn)</td>
<td>67.5</td>
<td>09/30/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010.1</td>
<td>NA</td>
<td>30</td>
<td>22</td>
<td>176(Enviro) / 99(Pawn)</td>
<td>75.7</td>
<td>08/31/10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AAVP3 Development Lifecycle Model Overview
AAVP3 Development Lifecycle Model Overview
FOX

- FOX Simulator
  - Crew Skills
  - Sensor Simulations
  - Desktop and Fixed Simulator

- Reused everything in the AAVP3
  - Environments
  - Vehicle Models
  - IWS
  - Sensor Detection Models
TRICT / 18 Months / Hardware

- Motion Platform
- 180 Degree Roll in Both Directions

- All Doors / Hatches Open
- Interior Identical to Real RG33
TRICT / 18 Months / Hardware
TRICT / 18 Months / Software

- Desert Valley Environment with Scenarios of varying difficulty.
- Instructor Workstation
  - Curveballs
  - AAR
- Vehicle Simulation
  - Controlling Motion Platform and Safety Features
- CROWS II Simulation
CBRN Dismount

- CBRN Dismount Prototype
- Four Handheld Sensors
- Joint Service Training
- Three Environments
- User-Defined Scenarios with Custom Contaminants
What Did We Learn?

• Retrospectives are extremely important
  – Act on the feedback immediately
• Experiment with different sprint lengths until one feels right
  – We always let the Developers determine the sprint length
• Don’t appoint leaders in sprint teams, let them just appear on their own
  – This feels very counter-intuitive
• The daily meetings are crucial to communication
• Ideal days are way shorter than anyone thought they would be. We usually estimate ideal days at 2/3 of available days in the sprint. This has helped keep expectations in check with Project Leads.
What Did We Learn?

• Our Developers have taken ownership of our Products - they feel in control.
  – Product Proponents

• Code has become more reusable.

• Sprint Demos are a point of pride.

• We started giving story points to our bugs and it has helped in tracking velocity.

• Our velocity has steadily increased over time because we are forced to reevaluate ourselves at the end of every Sprint.

• Scrum allows Developers to just worry about developing.
The Present

• Large Team
  – Programmers: 31
  – Artists: 15
  – Level Designers: 4
  – Game Designers: 2
  – Support: 6
  – Test: 2
  – Project Leads: 12
  – Offsite Misc: 9

• Scrum is used on all software development.
  – AAVP
  – Mobile
  – Outreach
  – Prototype & Sustainment

• Dozens of Concurrent Projects.
Questions?

Jay Olive
Development Lead, Army Game Studio
256.842.9412
jay.olive@us.army.mil
Acronyms!

• **AAR** - After Action Review
• **AAVP3** - America's Army Visualization Platform 3
• **AGS** - Army Game Studio
• **CBRN** - Chemical, Biological, Radiological and Nuclear
• **CROWS II** - Common Remotely Operated Weapon Station, 2nd Generation
• **FBCB2** - Force XXI Battle Command Brigade and Below
• **IWS** - Instructor Workstation
• **TRICT** - Transportable Reconfigurable Integrated Crew Trainer
• **UE3** - Unreal Engine 3