Craeg Strong
Experience Report:
Converting A Successful Criminal Justice Program
From Waterfall To Agile
Software Development since 1988
Large Commercial & Government Projects
Turned Around Projects With Agile
Apache Ant Open-Source Contributor
Started with the Program in 2003
Partnership with ECS Federal in 2010

CTO, Ariel Partners
CSM, CSP, CSD, PSM, PMI-ACP, PMP

www.arielpartners.com
cstrong@arielpartners.com
@ckstrong1
Agenda

About the Program
Plan For Moving To Agile
Key Decisions and Their Ramifications
Year One
  • Challenges We Encountered
  • Benefits We Observed
Year Two
  • New Challenges
  • Vision and Roadmap For Moving Forward
Lessons Learned
Q & A
Success Story

1990: Kidnapping, sexual assault, and attempted murder
8 year old victim survives 10cm throat laceration
Composite sketch put together based on victim’s description
No leads

Evidence: Men’s underwear, T-Shirt, Victim’s panties and shirt
Male DNA profile recovered from waistband of men’s underwear
Evidence resubmitted to DNA lab in May 2008

2008: DNA hit to Arkansas Offender Dennis Earl Bradford
Never a suspect: Lived 2 miles from Jennifer’s house

Why Do We Need To Be Agile?  
Fight Crime Faster, Better
How Does DNA Get Into the System?
Where Does the Software Operate?

1 National Lab
54 State Level Labs
137 Local Labs

United States

5 Continents
56 Countries ↔ Interpol, Prüm
Evolution of the System

1990: Pilot

1994: DNA Identification Act

1998: Client-Server: VB6

2002: Multi-Language

2008: SOA Re-Architecture

2012: 2M Lines of Code
From Traditional Waterfall

<table>
<thead>
<tr>
<th>Year One</th>
<th>Year Two</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>M1</td>
</tr>
<tr>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>M3</td>
<td>M3</td>
</tr>
<tr>
<td>M4</td>
<td>M4</td>
</tr>
<tr>
<td>M5</td>
<td>M5</td>
</tr>
<tr>
<td>M6</td>
<td>M6</td>
</tr>
<tr>
<td>M7</td>
<td>M7</td>
</tr>
<tr>
<td>M8</td>
<td>M8</td>
</tr>
<tr>
<td>M9</td>
<td>M9</td>
</tr>
<tr>
<td>M10</td>
<td>M10</td>
</tr>
<tr>
<td>M11</td>
<td>M11</td>
</tr>
<tr>
<td>M12</td>
<td>M12</td>
</tr>
</tbody>
</table>

- PRR
- PDR
- CDR
- TRR
- TRR
- OAR

Change Control Board Meeting

- Requirements
- Design
- Code And Unit Test (CUT)
- System Test
- GAT
- UAT
Key Decision #1: Automation Support Team

- Business Requirements
- Customer
- Technical Requirements
- Architecture Review Board

Development Scrum Team:
- Developers
- Analysts
- Testers
- Doc / Training / QA

Support Services:
- Enhanced Automation Capabilities
- Technical Requirements

Automation Support Scrum Team:
- Agile DBA
- DevOps Engineer
- SharePoint Administrator
- Automated Build Engineer
- Test Automation Specialist
- CM Engineer

Technical Requirements
Enhanced Automation Capabilities
### Key Decision #2: Quantify Everything We Do

#### User Stories

<table>
<thead>
<tr>
<th>Category</th>
<th>Story Points</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Requests</td>
<td>Story Points</td>
<td>Better Usability, Improved DNA Matching Algorithms</td>
</tr>
<tr>
<td>Problem Reports</td>
<td>Story Points</td>
<td>Defects We Inherited, Latent Defects</td>
</tr>
<tr>
<td>Technical</td>
<td>Story Points</td>
<td>Test Automation Frameworks, Custom Scrum Templates, Test Data Generation</td>
</tr>
<tr>
<td>Documentation</td>
<td>Story Points</td>
<td>Design Documents, On-Line Help, CBT</td>
</tr>
</tbody>
</table>

#### Overhead

<table>
<thead>
<tr>
<th>Category</th>
<th>Hours</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugs</td>
<td>New feature fails a test</td>
<td></td>
</tr>
<tr>
<td>Administrative Tasks</td>
<td>Onboard new staff member, Re-cable blade server, reconfigure SharePoint</td>
<td></td>
</tr>
</tbody>
</table>
Key Decision #3: Technical aka “Testing” Sprints

No New Business User Stories During Technical Sprint

**PROS**

- Faster Technical Debt Reduction
- Faster Progress Towards Continuous Delivery
- Additional Testing
- Fewer Regressions

**CONS**

- Reduced Business Value Delivery
- Additional Complexity
- Inconsistent Metrics
Key Decision #4: When Do We Start Sprint 1?

1. People
   - Staffing Reaches Critical Mass (83%)

2. Hardware
   - Every Team Member Has An Imaged PC (GFE)

3. Software
   - Source Code Manager Installed
   - Source Code Loaded (GFI)

4. Product Backlog Items
   - “Top Twenty” List of Customer Priorities
   - Team Brainstormed List of Technical User Stories

<table>
<thead>
<tr>
<th>Centralized Build</th>
<th>...Five Months</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Integration</td>
<td>...Seven Months</td>
</tr>
<tr>
<td>Automated Reporting</td>
<td>...Eleven Months</td>
</tr>
</tbody>
</table>
Challenges: Automated Testing Coverage

Goal: 80%

(1) Identify The Constraint
Dependent On Manual Testing For Foreseeable Future

(2) Subordinate Everything Else
- Reduce Testing Cycle Time
- Eliminate False Positives
- Validate Configurations
- Improve Test Case Quality
Challenges: Working in a Time Box

On a fixed price project, what happens if you don’t complete all “must do for release” items within the time box?

Solution: We Added Another Sprint...

Test Readiness Review

• All Documentation Updated
• All System Tests Completed
• Test Documentation Completed
• No Priority One or Two Bugs

Sprint 10

...to the release,

and reduced the next release by one sprint.
Benefit: Able to Adjust to Unforeseen Circumstances

SAN Failure

Out of warranty

All Testing Environments Lost

SAN Failure

RIP

Out of warranty

All Testing Environments Lost
Benefit: Customer Can Reset Priorities

SP3 Backlog Items Delivered

Emergent 50%  Planned 50%
Transition: Base Year to Option Year One

Achievements

- Agile Is Now “Norm”
- Buy-In: Changed “Hearts and Minds”
- Team Members Growing Outside of Narrow Specialties
- Consolidated Management Tools
  - Legacy Requirements
  - Defects
  - Tier III tickets
  - Manual Test Cases
- Established CI Build
- Successful Release To Production
- Achieved CMMI Level Three Certification...While Adopting Scrum!

Challenges

- Team Not Meeting Sprint Commitment
- Team Not Keeping Up With Documentation Requirements
- User Stories Too Big
- Automated Unit Testing Requires Extensive Refactoring
- GUI Controls Not Compatible with Automated User Interface Testing Framework (CodedUI)
- Manual Testing Tools Require Network Reconfiguration
- Test Deployments Lengthy, Manual, Error Prone
- DevOps Tools (Salt) Poor Fit for Microsoft Environment
Challenge: Meeting Definition of Done

User Story

As a user working in a state or local lab, I want to perform moderate match estimation on my DNA profile, So that I know it will be accepted by the national lab

Task
Task
Task
Task
Task
Task

Separate task for every item in “definition of done” checklist

Lesson Learned

✓ Maximizes Parallel Efforts
✓ No Complex Workflows or Approvals
✓ Coordinate at Daily Scrum

Task Audit During Sprint Review
Challenge: Maintaining Technical Documentation

Why Document Design in Agile?
- Reduce Program Risk
- Every Five Years Team Disbands, Mission Continues

EXISTING
- 50-100 pages
- “Tick The Box”
  - Fill In The Template
  - Applicable? Maybe.
  - More Is Better
- Lots of Details
  - Long API Listings
  - Busy Complex Diagrams
  - Every Process Step In Detail

NEW
- 10—25 pages
- As Simple As Possible (But No Simpler)
  - Its All About Context
  - History And Background
  - Trade-offs and Design Decisions
- Explain Why. Give Examples
  - High-Level Architecture
  - How Do Pieces Fit Together
  - Pitfalls, Best Practices
- Opportunities for Improvement
  - Rated By Risk, Opportunity
What About DevOps?

Development:
- Source Code
- Awesome Software Application

DevOps:
- Source Code
- Fully Configured System

Superior Microsoft Support

Desired State Configuration
Challenge: Automated Testing

**Build Testing Capability**
- Technical Frameworks
  - NUnit
  - SpecFlow
  - Moq
- Reporting Capability
  - Test Results
  - Code Coverage
  - Functional Coverage
- Encouraging Adoption
  - Examples Everywhere
  - Mentoring
  - Pair Programming

**Make The System Testable**
- Upgrade Dot Net Framework
- Drop Support for Old Operating Systems
- Upgrade Third Party Libraries
- Refactor Code To Isolate Dependencies
- Simplify Project Structure
- Different Components May Need Different Strategies

Both Pieces Are Required For Success
Challenge: Producing an RTM

What Exactly is A Requirements Traceability Matrix?

- Inventory of all requirements
- ...traced to features implemented in the software,
- ...traced to test cases that cover the functionality.

Why Do We Need A Requirements Traceability Matrix?

- Thousands of complex features
- ...developed over many years
- ...with a brittle code base
- How can we know we have tested everything?
Can An Agile Team Meet That Challenge?

- All system functions expressed as user stories
- All user stories have acceptance criteria
- Each criterion translated to an automated test using structured English (Gherkin)
- Customized Report matches Epics and User Stories to automated acceptance tests
- Test fails unless software is implemented correctly
Transitional Strategy

- Everything Captured In Microsoft TFS
- Legacy Requirements linked alongside new User Stories
- Today
  - 5000 “Shall” Requirements
  - 700 Manual Test Cases
  - 500 User Stories
  - 200 Automated Tests

Plan
Gradually Migrate legacy “Shall” requirements to User Stories and Acceptance Criteria
Looking Ahead To Year Three

Achievements

- Automated Reporting
- Third Party Library Upgrades
- HW & SW Upgrades
- Automated HW Configuration (DevOps)
  Dramatically Reduced Configuration Time
- Automated Unit Test Framework
  Reached Critical Mass
- Documentation Quality Has Stabilized
- Second Successful Release To Production Imminent

Challenges

- Tier III Support Workflows Don’t Fit Scrum Iterations Well
- Meeting Sprint Commitment Is Still Difficult
- Lots Of Ongoing Upgrades & Refactoring Work Still Required
- Automated Unit Testing Not Yet Ubiquitous
- Test Deployments Still Lengthy, Manual, Error Prone
Lessons Learned

1. Agile benefits can be realized even on successful waterfall projects
2. No silver bullet
3. Planned, incremental adoption of agile practices
4. Get an expert tool smith
5. Start scrum without everything in place
6. Some practices are better done by the book
7. Adopt systems thinking approach to addressing technical debt
Your Next Step

US Government Agencies increasingly interested in and ready to move to Agile

WE WANT YOU!
Questions?
Session Feedback

Please provide feedback on this session!

You can do so in 3 ways:
1. Visit this session on the Mobile App. Click Session Feedback.
2. Scan the unique QR Code for this session located at the front and back of the room.
3. Visit the unique URL for this session located at the front and back of the room.

Thank you for providing your feedback 😊