A Practical Approach to Large Scale Agile Development

Gary Gruver
August 5, 2013
Gary Gruver:

- 24 year veteran in high tech
- Led HP’s 400+ developers on a journey from waterfall to agile development
- Currently VP of QE, Release, and Operations at Macys.com
4+ Year Large-Scale Agile Journey

400+ engineers around the world

Over 10M lines of code

- High-end LaserJet printers and MFPs
- Embedded SW & FW
- Digital Sending and HP open Extensibility Platform

- 6 weeks + to get through a complete testing cycle (mainly manual)
- Build integration taking 15-20% of resources a week to get fixes to main
- Manual testing a key driver and constraint for adding products

Lengthy Build Integration & Testing Cycles

- Ongoing customer issues with consistency and lack of features
- Marketing had essentially given up asking for FW innovations

Products Lagging the Competition

- Development costs growing 2.5X from 2004-2008 and the business was still constrained
- Up to 10 different branches (driven by each product release window) in MFP
- CPE driving millions/year in CPE investments

Costs out of control

- 80-90% of resources just porting existing FW to new products and qualifying
- Unable to add new products to the plans due to lack of FW resources
- 20% of resources developing plans that quickly became obsolete

Couldn’t Add Enough Resources
Firmware Development Transformation

Consistent Dev Environment

Integrated Tools

Agile Development with Mini Milestones (Sprints)

Organizational Change Management

Architected for product variability

Fully automated unit and system test

Continuous integration and test system

One branch for all products including CPE

Organizational Change Management
Breakthrough Capacity for Development

New Customer Capabilities

FutureSmart FW Large Scale Agile Development Engine

- 400+ developers
- 10+M LOC
- 75,000-100,000 LOC turmoil
- 100-150 Commits
- 10-15 builds /day
- 15,000 hours/day of testing (90% pass rate)

Defect Fixes
Cycletime Driver Improvements

2008

- Build Bosses **1 Week**
- Number of Builds **1-2**
- Feedback on Main ~**10 Commits/Day**
- Full Manual Registration **6 Weeks**

2011

- Continuous Integration **3hrs**
- Continuous Integration **10-15/Day**
- Autorevert ~**100 Commits/Day**
- Auto Regression Testing **24 Hrs**
Development Cost Driver Improvements

**2008**
- Code Integration 10%
- Detailed Planning 20%
- Porting Code 25%
- Current Product Support 25%
- Manual Testing 15%
- Capacity for Innovation ~5%

**2011**
- Continuous Integration 2%
- Agile Planning 5%
- One Main Branch 15%
- One Branch CPE 5%
- Most Testing Automated 5%
- Capacity for Innovation ~40%
State of the art FW development model

2008

- Costs out of control
- Couldn’t add resources fast enough
- Lengthy build, integration and testing cycles
- Products lagging the competition

2011

- ~70% reduction in FW development cost per program
- 50% reduction in FW headcount
- Cont. integration, daily automated regression
- Vintage chart unleashed and capacity for innovation
Making an Enterprise Agile

VS.

Enabling Small Agile Teams in the Enterprise
Scrum ≠ Agile
Improvements Best Driven at the Enterprise Level

- Business Objectives/Priorities
- Enterprise Level Continuous Improvement
- CI/CD and test automation infrastructure
- Planning Process
Business Objectives (Don’t “Do Agile”)

Define your value proposition

Understand your cost & cycle-time drivers

Either automate, eliminate, or engineer out the drivers that aren’t key to the value prop
Having real time metrics is essential for the speed of agile & aligning the org. But don’t manage by metrics.

Use the metrics to understand where to have conversations about what is not getting done.
Finding the offending code

What Code? When? Are you sure it wasn’t Shawn?
CI/CD and Test Infrastructure

How much of the system do you put together how often and in what order?

How do you build up the system?

Automated testing is as hard or harder than writing good code.

How do you create frameworks that improve stability and productivity?

Where do you create test harnesses/simulators/emulators?

Where do you turns builds red and stop the train versus logging defects and tracking passing rates?
Building up a Large SW System

Agile Comp 1
Agile Comp 2
Agile Comp 3
Agile Comp 4
Agile Comp 5
Agile Comp 6

Legacy Waterfall IT 1
Legacy Waterfall IT 2
Legacy Waterfall IT 3

Interface Test Simulator
Value Per Iteration

Project 1

Number of iterations

Value

Project 2

Number of iterations

Value
One of the biggest challenges with Agile Planning at the enterprise level is getting the organization to accept the uncertainty in SW development and appreciate the flexibility and opportunity.
Long Term Predictability for SW Schedules
Do we really need the predictability of our current planning processes?
Are our current planning processes really that accurate?

Accuracy

Planning Investment
Taming the Planning Beast

The old way:

Final feature list 12 months in advance

Planning resources and development resources one in the same

Always in “locked in” mode, had to say no to late-breaking requests or start over

Today’s Philosophy:

Every hour we spend planning a feature is an hour we don’t spend delivering it (the real goal).
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initiative A</td>
<td>21</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Initiative B</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Initiative C</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Initiative D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Initiative E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>Initiative F</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>Initiative G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Initiative H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>Initiative I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Initiative J</td>
<td>20</td>
<td>27</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>Initiative K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Initiative L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Initiative M</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

|                                           | 29 | 25 | 51 | 30 | 20 | 25 | 23 | 12 | 74 | 26 | 38 | 59 | 401 |   |       |

Limit long-term commitments to 50-60% Capacity
FutureSmart Firmware User Stories per Sprint

Input Queue of user stories (not in WIP yet; being evaluated)
Now on quarterly releases (2 “feature sprints” plus 1 “final qual” sprint), so Input Queue is ~2 releases of backlog (80-100 User Stories per release)

2nd year throughput (big drive to complete architecture; not sustainable)

1st year throughput (25 user stories/ sprint)

Latest throughput (need more data to know new steady-state; probably 40-50/ sprint)

Final qual for 1st release

Final qual for 2nd release

Final qual for 3rd release
Getting Mgmt/Mktg Buy-in to Agile Planning

“FW will still commit to basic new product support one year ahead”

Means prioritizing “product turn-on and delivery/qualification” ahead of new features

“You get to decide what we work on first”

Establish a “1-N feature request list” and the combined marketing teams decide the order

“You will get 20% more features this way”

Easy to explain the 20% of resources previously used to estimate

“We’ll actually listen to your last-minute requests”

Just put it at the top of the list, ahead of all the other “input queue” features
Making an Enterprise Agile

VS.

Enabling Small Agile Teams in the Enterprise

E-mail: gbgruver@gmail.com
Blog: largescaleagile.com
Twitter: @GRUVERGary