Using Perforce to Facilitate Agility

Victoria Hall
Sr. SW Engineering Manager
Bio-Rad Laboratories

Topics

• Bio-Rad & system backgrounder
• Agile introduction
• SCRUM focus
• Software engineering best practices
• Continuous integration
• Perforce in a real world agile implementation
**Backgrounder**

- **Bio-Rad**
  - 1.7 B laboratory device and diagnostic company
  - Best known for delivering high quality medical testing equipment and consumables to diagnostic laboratories and academics involved in biological research and clinical practice

- **Who am I?**
  - 20 years software development & management
  - Certified Scrum Master/Practitioner and agile proponent
  - Released numerous multi-tiered web and enterprise systems, concentrated in last ten years on bioinformatics and biological sciences

**SELDI**

- **The instrument PCS 4000** — Surface Enhanced Laser Desorption Ionization (SELDI) Mass Spectrometer
- **Applications** — proteomics, life science biology, drug discovery, food safety
Data analysis software

- Instrument Control
- Data aggregation
- Data analysis

What do we build

Instrument client –
Linux / Apache Tomcat
Java, C++, JESS

Data manager server –
Win XP / Java
Jetty
mySQL/Oracle/Derby

Data manager clients –
Win XP / Java
HASP licensing, 3rd party libs
Tools we use

- Junit (www.junit.org)
- Silk (www.borland.com)
- Perforce (www.perforce.com)
- Ant (ant.apache.org)
- Shell scripts
- Rally (www.rallydev.com)

Agile Processes: a brief history

Taylor – 1911 – The principles of scientific management
Ford – 1915 JIT, DFM
1948-1975 - Toyota Production System
Deming – 1950 - Statistical methods of process control

Six sigma 1986
CMM 1989
RUP 1995
UML 1997
Extreme Programming, 1999
SCRUM - 2000
Agile manifesto - 2001
Waterfall

As a reference point

- Information known up front
- Manage and reduce risk
- Change is expensive
- Contractual ("sign off")
- Document-centric

Why Change?

Source:
Standish Group - Chaos Reports
**Process Theory**

**Defined vs. Empirical processes**

- "It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood.

- "When the process is too complicated for the defined approach, the empirical approach is the appropriate choice."

---

**Agile — Project Vision Drives the Features**

*Project vision drives features*

<table>
<thead>
<tr>
<th>Waterfall</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Plan creates cost/schedule estimates</td>
<td>The Vision creates feature estimates</td>
</tr>
</tbody>
</table>

**Fix These**

<table>
<thead>
<tr>
<th>Features</th>
<th>Cost</th>
<th>Schedule</th>
</tr>
</thead>
</table>

**Plan Driven**

**Value / Vision Driven**

**Estimate These**

<table>
<thead>
<tr>
<th>Cost</th>
<th>Schedule</th>
<th>Features</th>
</tr>
</thead>
</table>

---

*Process Dynamics, Modeling, and Control*

Ogunnaike and Ray, Oxford University Press, 1992
Agile Manifesto

We value

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiations
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

www.agilemanifesto.org

Many Agile Flavors

- Test Driven Development (TDD)
- Feature Drive Development (FDD)
- Extreme Programming (XP)
- SCRUM
- Crystal
- RUP
SCRUM?

An implementation of ceremonies, roles and documents for process control

**Roles**
- Team members
- SCRUM Master
- Product Owner

**Ceremonies**
- Sprint Planning
- Daily Stand-ups
- Sprint Review

**Documents**
- Product backlog
- Sprint backlog
The SCRUM Framework

**Daily Scrum**
- Done yesterday
- Plan for today
- Roadblocks?

**Vision**

**Sprint Planning**
- Review Product Backlog
- Choose / Estimate Sprint Backlog
- Commit to iteration

**Backlog tasks**
Expanded by team

**Sprint Backlog**
Features assigned to Sprint
Estimated by team

**Sprint Review Meeting**
- Demo features to all
- Retrospective

**Potentially Shippable**
Product Increment

**Backlog:**
Prioritized list of Features & Activities

Agile Principle:
Satisfy the customer through delivery of valuable software
Agile Principle:
Deliver working software

Potentially Shippable Product Increment

2 weeks

Agile Principle:
Deliver working software frequently
Agile Principle:
Welcome change

Vision

Sprint Planning
• Review Product Backlog
• Choose / Estimate Sprint Backlog
• Commit to iteration

Sprint Backlog
Features assigned to Sprint
Estimated by team

Backlog tasks
Expanded by team

Agile Principle:
Individuals and interactions

Vision

Daily Scrum
• Done yesterday
• Plan for today
• Road blocks?

Meet every 24 hours
Agile Principle:
Reflect regularly on process and product

Sprint Review Meeting
• Demo features to all
• Retrospective

The Scrum Framework

Vision

Backlog:
Prioritized list of Features & Activities

Sprint Planning
• Review Product Backlog
• Choose / Estimate Sprint Backlog
• Commit to iteration

Daily Scrum
• Done yesterday
• Plan for today
• Roadblocks?

2 weeks

Potentially Shippable Product Increment

Sprint Backlog
Features assigned to Sprint
Estimated by team

Backlog tasks
Expanded by team

Sprint Review Meeting
• Demo features to all
• Retrospective
One SCRUM implementation

- 4 week cycles
- RallyDev project management
- Junit/Silktest for automation
- Perforce for continuous integration in conjunction with Ant/batch scripts
- Test automation/execution is done in parallel with development

Best Practices

SW engineering best practices enable successful agile implementations

What are the important ones?

- Short cycle iterative development.
- Propagating the notion of DONE.
- User story driven development.
- Infrastructure to enable easy change.
  - Automated testing
  - Code/ design reviews
  - Continuous integration
Infrastructure is important!

especially for
successful agile implementations

Continuous Integration Overview

Wash, rinse, repeat

If changes are checked in, do build. If build completes, install and smoke test.

If smoke test passes, fully regression test.
Continuous Integration – In Detail

- Pull system - poll perforce for changes
- Deploy the build candidate using installer and smoke test.
- Full regression tests are run on the latest daily build.
- SILK test scripts are self documenting and a pass/ fail web page is generated.
- The whole process takes several hours.

Agile practices made real

Development

- Check in early and often.
- Test as you go.
- Keep the build line clean, that is buildable and passing all unit tests.
- Work in a sandbox but integrate.
It’s not done until it passes all tests!

Full source get

Integrate other’s changes

Local build using actual build scripts

Run unit tests locally and fix any defects found

Finally – check in

Agile practices made real

- **Integration**
  - Formal builds fully regression tested. We do these daily at night.

- **Test**
  - Separation of hardware and roles for build, test, and development.
  - Develop and implement testing as you go.
  - Expand your regression suite over time if you are starting at a deficit as we were.
Deployment

- Code freeze
- Manual tests emphasizing actual (not simulated) instrument data acquisition and processing.
- Defect fix cycle (as needed).
- Post acceptance tests, label based on change list build number as release candidate.
- Post alpha and beta testing, update the label to RELEASED. Launch to our internal manufacturing processes.

Other important uses of Perforce

- Track design documents, test results, test source changes, installation and documentation source changes.
- Control change and add predictability and sanity to your development and release process.

A good thing
Scripting

- Extensive library of custom developed Windows shell scripts.
- Same thing can be done with Unix/Linux shells, with Perl and with Python.
- Automation alternatives.
  - Anthill, Hudson, or CruiseControl

Challenges

People – getting the right people with the right skills can be challenging.

Culture changes slowly.

- Key hires
  - Experienced build/deploy person not afraid of scripting.
  - Experienced automated tester – in our case experienced with Silk4Test language.
  - Developers who can learn to appreciate, or who already appreciate, building quality in as you go.
Challenges

Seamless transitions

- Ensuring development is using the same scripts/libraries/tools as the final build, install, test, release cycle.
- It can be tricky to get developers to change their behavior. Use regression tests as a check on progress.
- Quality focus for delivery helps appreciably with this attitude migration.

Summary

- Infrastructure is important for facilitating rapid specification, development, release and incorporation of customer feedback.
- Perforce is easy to integrate with and very easy to build a continuous integration system around. You can integrate with any build/test system using minimal scripting.
- Perforce is robust and fast. It behaves as promised without corrupting data.
Recommendations

- **Facilitate agile practices daily through continuous integration**
- **Be disciplined and patient**
- **Get some training if you have budget; teach yourself if not**
- **Executive or other champion can help**

Resources

- **Home of Scrum** [www.controlchaos.com](http://www.controlchaos.com)
- **The SCRUM Alliance** [www.scrumalliance.org](http://www.scrumalliance.org)
- **Agile University** [www.agileuniversity.org](http://www.agileuniversity.org)
- **Agile Project Leadership Network** [www.apln.org](http://www.apln.org)
- **Agile Alliance** [www.agilealliance.org](http://www.agilealliance.org)
- **Yahoo Groups**: scrumdevelopment, XP, XPUK, agiletesting
### Suggested Reading

- Agile Software Development with Scrum, Ken Schwaber and Mike Beedle, Prentice Hall 2002
- Agile Project Management with Scrum, Ken Schwaber, Microsoft Press, 2004
- Iterative and incremental developments. a brief history, Larman, C., Basili, V.R. - Computer Publication Date: June 2003, Volume: 36, Issue: 6, pp 47-56

### Thank you’s

**Bio-Rad Laboratories:**

Iztok Marjanovic, John Dunaway, Niels Thomsen, Sangeeta Ahuja, Conan Nado, and Brian Robertson

**Santa Cruz Software Release Specialists:**

Curt Patrick, Tom Greer, & Robin Patrick