Student Co-Op Evaluation System

Patrick Flanagan
Tom Small
Whitney Sorenson
Dan Volpe
Chris Woodbury
Outline

- Project Overview
- Requirements
- Process Plan
- Testing
- Risks Faced
- Metrics
- Lessons Learned
- Project Status
Project Overview

• Online student co-op evaluation system
• Built on top of existing employer evaluation system
• Customer
  – RIT Office of Co-Operative Education and Career Services
• Target Audience
  – Engineering and Engineering Technology
Benefits of System

• Remove the need for paper evaluations

• Provide sophisticated methods of reporting and analysis

• Convenient and easy access to both student and employer evaluations
Requirements

- Extend existing system
- Online student evaluation submission
- Analysis of student evaluation data
- Student access to past evaluations
- Integration with employer evaluation system
- Notification of users by email of important events
Process

• Agile development
  – Multiple short iterations
  – No major up-front design
  – Refactoring of existing system
  – Frequent access to customers
• Weekly status meetings
• Bugzilla
• Testing
Iteration Cycle

1. Identify goals of iteration
2. Requirements drill down
3. File feature requests into Bugzilla
4. Assign features to developers
5. Implementation
6. Integration testing
7. Deployment
8. Stakeholder feedback session
## Project Schedule

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Weeks</th>
<th>Dates</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>042</td>
<td>5 – 8</td>
<td>1/10/05 – 2/4/05</td>
<td>Iteration 1</td>
</tr>
<tr>
<td>042</td>
<td>9 – 11</td>
<td>2/7/05 – 2/25/05</td>
<td>Iteration 2</td>
</tr>
<tr>
<td>043</td>
<td>1 – 3</td>
<td>3/7/05 – 3/25/05</td>
<td>Iteration 3</td>
</tr>
<tr>
<td>043</td>
<td>4 – 6</td>
<td>3/28/05 – 4/15/05</td>
<td>Iteration 4</td>
</tr>
</tbody>
</table>

- Deliver in 6th week
- Planned 2 week buffer
### Iteration Tasks

- **Iteration 1**
  - Student view of system, system & development environment configuration, email system

- **Iteration 2**
  - Refactoring, polish student view, dynamic form generation/editing

- **Iteration 3**
  - Reporting

- **Iteration 4**
  - Bug fixes, new feature requests, further polishing
Testing

- Informal, small-scale testing performed continuously throughout development
- End-of-iteration integration testing
  - Code freeze
  - Developers tested each other’s features
- Stakeholder testing
  - After each iteration, throughout development
- Full-scale test plan
- Pilot testing
Risks Encountered

- Existing design and implementation
- Existing documentation
- Feature creep
- Changing requirements
- Configuration
- Final deployment
Metrics

- Earned value
- Defects
- Requirements volatility
Earned Value
Defects

- Employer enhancements: 3%
- Employer bugs: 20%
- Student enhancements: 56%
- Student bugs: 21%
Volatility

![Volatility Chart]

- Time (Weeks)
- Quantity
- Minor Reqs. Changes
- Moderate Reqs. Changes
- Major Reqs. Changes
- New Minor Features
- New Moderate Features
- New Major Features
Deliverables

• System code
• Test plan
• Online, integrated user manual
• Technical overview of code
Lessons Learned

• Agile process
  – Flexibility in responding to new requests
  – Frequent customer communication, feedback
  – Feature creep; no up-front sign-off on requirements
    • Had to decide which requests we could accept
    • Had to make sure we could do what we agreed to

• Difficult to extend inherited system
  – No contact with former team
  – Poor internal documentation
  – Constraining design
Future

• OCECS has indicated desire to use system for entire Institute
  – System should be able to support that now, but will need testing

• Significant refactoring and/or re-implementation may be necessary
Project Status

• Delivered to OCECS
• Team is available to assist in deployment, troubleshooting and bug-fixing
• OCECS is testing internally and running pilot program
• Roll-out planned for this summer
Questions