Some Assembly Required: Using Agile Methodologies to Develop an Interactive Software User’s Guide
Presentation Agenda

- **Some Assembly Required: Agile Methodologies**

  - **Introduction / Problem Statement**
    - Why pursue a new technical document development platform?

  - **Part 1… Background: Enabling Technologies, Software Architecture and Development Platforms**
    - Object-oriented programming (OOP)
    - Rich Internet Application (RIA) / Rich Desktop Application (RDA)

  - **Part 2… SUG Development**
    - Agile and Technical Documentation
    - The Pilot Project
    - User stories

  - **Part 3… SUG Lessons Learned**
Why Pursue a New Development Platform?

- Current development platform (Authorware) is ‘abandonware’
  - Vendor (Adobe) discontinued development in August of 2007

- Strengths
  - Broad acceptance in the marketplace; rapid prototyping; particularly well suited to creating e-learning content

- Weaknesses
  - XML handling is sub-optimal; web delivery and desktop delivery is problematic; ‘Protection Mode’ and runtime errors with IE7 and Vista

- Need a development platform that’s ‘future-proofed’
  - Leverage Broadband proliferation and enabling technologies
  - Accommodate multi-channel delivery: web, desktop, mobile, and print
What is an Object Oriented Programming language?

- Programming oriented around objects - self-contained collections of computational procedures and data structures.

A computer language can be said to be Object Oriented if it provides support for the following:

<table>
<thead>
<tr>
<th>Class</th>
<th>A class is a blueprint, or prototype, that defines the variables and the methods common to all objects of a certain kind.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>An instance of a class. More than one instance of the same class can be in existence at any one time.</td>
</tr>
<tr>
<td>Encapsulation</td>
<td>The act of placing data and the operations that perform on that data in the same class. The class then becomes the container for the data and operations.</td>
</tr>
<tr>
<td>Inheritance</td>
<td>The reuse of base classes to form derived classes (subclasses). Methods and properties defined in the superclass are automatically shared by any subclass.</td>
</tr>
<tr>
<td>Polymorphism</td>
<td>Ability of objects belonging to different types to respond to method, field, or property calls of the same name, each one according to an appropriate type-specific behavior.</td>
</tr>
</tbody>
</table>

Why use it?

- OOP is an efficient methodology for minimizing development time and maximizing code reusability.
- OOP creates a modular design that is easily modified without having to restructure the entire system.
Rich Internet Application (RIA)

Web-based application that approaches the speed and elegance of a local (installed) application

- Traditional desktop features
  - Drag and drop, windows, wizards, and panels

- Cross Platform
  - Cross Browser
    - IE, Mozilla Firefox, Safari
  - Cross Operating System
    - Windows, Linux, MAC OS

- No installation, and accessible just about anywhere

- MVC pattern on client and server
  - MVC pattern(s) can manage the interaction between user and the user interface, and requests to the server
Rich Desktop Application (RDA)

Platform-independent web applications that can be run on a user's desktop

- In contrast to RIA’s, RDA’s are able to run off-browser and off-line
- Applications have network connection awareness
- Cross Platform... browsers and operating systems
- Desktop security model - applications can be packaged and digitally signed
## RIA in the Browser / RDA on the Desktop

<table>
<thead>
<tr>
<th>Feature</th>
<th>RIA in the Browser</th>
<th>RDA on the Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application delivery</td>
<td>Applications can be easily discovered, explored, and used.</td>
<td>Installed applications have more persistence, power, and functionality.</td>
</tr>
<tr>
<td>Installation</td>
<td>No application installation is necessary.</td>
<td>Applications install seamlessly from the browser or install like a traditional application</td>
</tr>
<tr>
<td>Background capability</td>
<td>RIAs can only run in a visible browser window</td>
<td>Applications can run in the background and provide notifications like traditional desktop applications</td>
</tr>
<tr>
<td>Persistence</td>
<td>Activity is limited to the browser session. When the browser is closed, information is lost.</td>
<td>RDAs are installed on the desktop. They store information locally and operate offline.</td>
</tr>
<tr>
<td>Desktop integration</td>
<td>Applications are sandboxed, so desktop integration is limited.</td>
<td>Applications can access a desktop file system, clipboard, drag and drop events, system notifications, and more.</td>
</tr>
<tr>
<td>User interface control</td>
<td>RIAs run within a browser window that has its on controls, branding, and integration with the desktop.</td>
<td>RDAs have a customizable user interface and desktop integration, enabling branding experiences.</td>
</tr>
<tr>
<td>Data storage</td>
<td>Applications have limited local storage, which the browser can destroy.</td>
<td>Applications have unlimited local storage and access to a local database, plus encrypted local storage</td>
</tr>
</tbody>
</table>
RIA and RDA Development Platforms

- Adobe Flash/Flex/AIR
- Sun JavaFX
- Microsoft Silverlight
- Google GWT
- OpenLazzlo
- AJAX

"By 2010, at least 60 percent of new application development projects will include RIA technology, and at least 25 percent of those will rely primarily on RIA."

Source: Gartner RAS Core Research
Flash Builder and Adobe AIR

**Flash Builder**
- Adobe’s platform for developing and deploying RIAs for the **browser**
  - Leverages the nearly ubiquitous Flash Player (98% penetration level)
  - Flex applications can be built using the free open source Eclipse framework

**Adobe AIR (Adobe Integrated Runtime)**
- Adobe’s platform for deploying RIAs for the **desktop**
  - Allows you to transform a RIA into a Rich Desktop Application (RDA)
  - Air applications run outside of the browser, on multiple operating systems
Flash Builder and Adobe Air... Who’s Using It?

- Flash Builder and Adobe AIR are reaching critical mass
  - SAP, HP, Google, NASDAQ, eBay, AOL, Yahoo, MINI...
Agile and Technical Documentation

- Traditionally, Agile methodologies focused on software development or product engineering

- Recently, Agile practices have extended to other areas such as technical documentation

- Agile methodologies encourage/allow:
  - A means to accommodate content that is more and more malleable and requires more touch points
  - Effective communication and collaborative spaces for technical communicators and Subject Matter Experts (SMEs)
  - The use of Scrums and user stories to create a platform by which to develop customer-facing documentation
Attributes of the Ideal Pilot Project

http://blog.mountaingoatsoftware.com/four-attributes-of-the-ideal-pilot-project
Attributes of the Ideal Pilot Project

- **Duration**
  - A project that could be completed in 8 weeks

- **Project Size**
  - A project that could be supported by one team

- **Importance**
  - Medium priority

- **Business Sponsor Engagement**
  - Reasonably strong
Agile and User Stories

- Substitute of formal requirements documents
- Summaries of functionality that leave space for expansion and refinement
- Prioritized based on costs and their value

As a designer, I can customize the look and feel of my deliverable

As a user, I can drill down to specific information using multi-level menus

As a user, I can display/use the SUG on multiple platforms
User Stories

Skinning and styling

As a designer, I can customize the look and feel of my deliverable.
User Stories

Software User’s Guide (SUG) … Prototype

- Versatile Menu System
  - As a user, I can drill down to specific information using multi-level menus
  - As a designer, I can pull data from external XML collections
User Stories

Software User’s Guide (SUG) … Prototype

- Data Grids and Tables
  - As a user I can resize, and sort column layouts
User Stories

Software User’s Guide (SUG) … Prototype

- Multi-channel delivery
  - As a user, I can display/use the SUG on multiple platforms
Lessons Learning

Software User’s Guide (SUG) … Prototype

- Team
  - Team was not co-located
  - Could have been better formed

- Short stories
  - Need to be short
  - Need to have clear value to the user
  - Divide the feature (story) into smaller pieces and add value incrementally

- Momentum
  - When the team sees visible progress, progress and creativity increase
  - Relates back to keeping “stories short” and fostering team communication and demanding whole team collaboration
Lessons Learning… cont.

Software User’s Guide (SUG) … Prototype

- Feedback
  - Rapid feedback cycles make for quicker adjustments and improvement
  - Easy to fall into a vacuum – little feedback to sanity check decisions
  - Relates back to keeping “stories short”

- Acceptance testing
  - In hindsight – ad hoc
  - Different hardware/software platforms led to “faux” behavior/performance/scenario testing

- Conclusions
  - Practice iterative, incremental development
  - Encourage active customer participation and demand whole team collaboration
  - Deliver business value at regular intervals at a sustainable pace