On Demand Insurance Business Problems

1. We lose customers because we process new policy applications too slowly.
2. Our claims processing is time-consuming and inefficient.
3. We need to do a better job of attracting new customers.
4. We need our employees to be more productive.
5. We need to grow revenue in existing accounts, and by partnering.
6. Our development teams are always behind schedule and over budget.
7. We need to gain maximum efficiency out of our IT infrastructure.

On Demand Insurance
CEO
On Demand Insurance – Business Imperative

**GOAL**
*Deliver high quality applications on time and under budget*

**REWARD**
*Achieve the most efficient Software Development Process*

---

**On Demand Insurance – Needs to Deliver Applications More Efficiently**

I need to be more productive. Many of the tools we have do not work together and my project managers are frustrated as they have little control of our projects.

The IBM Software Development Platform incorporates over 30 years of industry best practices that will make your development team operate at peak efficiency.

---

On Demand Insurance
CEO

On Demand Insurance
CIO

IBM
Software Development Challenges

Why Software Development Teams Need to Improve

- Cancelled prior to completion: 30%
- Were delivered over budget: 54%
- Were NOT considered successful: 66%

Source: THE STANDISH GROUP 2003

Waterfall Development Lifecycle – “Old Fashioned” Way of Doing Development

Application

Sequential Process Limitations
- Requirements can never change
- Design has to be 100% correct
- Not well suited for On Demand changes
- High degree of failure for large projects

Goodness:
- This model did formalize the development roles

Winston Royce, 1971
Follow these 30 years of Best Practices - Operate at Peak Efficiency

IBM Software Development Platform

Integrated Lifecycle Development
Integrated Development Tools
Unified Modeling Language (UML)
Systematic Testing
Team Processes
Iterative Development

Software Development Process

Four Imperatives – State of the art Software Development!

1. Develop Iteratively
2. Focus on Architecture
3. Continuously Ensure Quality
4. Manage Change and Assets

Iteratively = to perform multiple times
Iterative Development – Reduce Risk and Achieve Better Productivity

Application

How do you handle a big project?
- One bit at a time!

Iteration 1
- Analyze
- Design
- Code
- Test
- Integration

Iteration 2
- Analyze
- Design
- Code
- Test
- Integration

Iteration 3
- Analyze
- Design
- Code
- Test
- Integration

- Earliest iterations address greatest risks
- Each iteration includes integration and test
- Ensures Quality early in the process

Iterative Development Reduces Risk

Risk reduction
Iterative Insures success

Waterfall risks project delay

Increasing TIME
Iterative Software Development Platform at On Demand Insurance - A LifeCycle

Business Analysts
- Analyze
  - Analysis & Requirements
  - Document requirements

Architects
- Design
  - Model application and data

Developers
- Code
  - Study impact
  - Check out source code
  - Make changes
  - Testing
  - Mark complete

Testers
- Test
  - Manual Test
  - Functional Test
  - System test
  - Document results

Project Managers
- Start project and track progress
- Assign resources
- Enforce Rational Unified Process

Executives
- Align investments with business objectives
- Analyze and monitor project portfolios

Fatal Flaw for Mercury – No Iterative Integrated Development Lifecycle

Business Analysts
- Analyze

Architects
- Design

Developers
- Code

Testers
- Test

No iterative integrated development lifecycle

Mercury

- Mercury Quality Center
  - No iterative development process
  - No focus on quality early in the cycle
  - No Integrated Development Environment
  - No managing change and requirements early in the cycle
  - No UML Modeling

Source: IBM Competitive Project Office Research, 2005
Fatal Flaw for Microsoft – No Iterative Integrated Development Lifecycle

- Microsoft Visual Studio.NET
  - No iterative development process
  - No focus on quality early in the cycle
  - No managing change and requirements early in the cycle
  - No UML Modeling

Source: IBM Competitive Project Office Research, 2005

The Software Development Platform – First Start with Rational Unified Process (RUP)

Best Practices in a box!

Process Definition for Iterative Development

All team members share:
- One knowledge base
- One process framework
- One common language (UML)
- Web-based delivery
On Demand Insurance Began by Defining a Software Development Process with RUP

- This process definition will become their foundation
- Roadmap of how value will be delivered to the business

- Consistency
  - Help team members understand their responsibilities and their relationship with other team members

- Predictability
  - Help identify what resources are needed and when
  - Enables metric development to support future planning
  - Defines decision points to reduce surprises

- Quality
  - Built on the four Imperatives
  - Focus on risk reduction – Iterative Development

DEMO: Rational Unified Process (RUP)

A process definition to help you to decide
- Who…
- Does what…
- To produce something…
- When they do it…
- And how that artifact gets used…
- To build and maintain a quality software system

What did you just see?

A pre-defined and configurable process that enables best practices in your development organization and processes  

Productive and fewer mistakes
On Demand Insurance Next Looks at Architecture and Modeling

Model the Architecture
- Communicate with a common language
  - Unified Modeling Language (UML)
- Reduce complexity
  - Create and manage abstract representations
- Improve quality
  - Seamless model integration reduces translation errors
- Manage change
  - Maintain traceability across artifacts

Create Use Case model

Create Class Diagram

Generate code from model and synchronize code to model

Accelerate results with model-driven development

Why is Modeling Important?

1. Could you build this without a blueprint?
2. Could you teach someone else how to build this?
3. Could you change this without a blueprint?

Modeling provides a blueprint for your application
**DEMO: Architecture and Modeling Overview**

- On Demand Insurance uses modeling
  - Modeled their architecture and their applications
    1. Use Case diagrams
    2. Sequence diagrams
    3. Deployment diagrams
    4. Class diagrams

What did you just see?

**Model Driven Architecture (MDA)**

- A well designed and architected approach to their applications
- Productive and fewer mistakes

**On Demand Insurance – Needs to Improve Insurance Application**

We need to change our Insurance Web application. I want to add a home insurance discount for preferred customers.

On Demand Insurance
CIO

Use the IBM Software Development Platform to do this.

IBM
Software Development Platform – Enhancing the Application

Roles, activities and artifacts coordinated by IBM Software Development Platform

Roles:
- Business Analyst
- Project Manager
- Architect
- Developer
- Tester

Activities:
- Requirements document
- Change request
- Update model
- Create code
- Test cases
- Defect report

Artifacts:
- IBM RequisitePro
- IBM ClearQuest
- IBM ClearCase
- IBM Rational Software Architect
- IBM Rational Software Modeler
- IBM Rational Manual Tester
- IBM Rational Functional Tester
- IBM Rational Performance Tester

Start with Managing Requirements – IBM Rational RequisitePro

Effective Requirements Management is Critical to Iterative Development
RequisitePro integrates the entire software development process

- Understand and track requirements
- Model user interactions; validate requirements
- Manage changes to requirements
- Define test cases on requirements

Access RequisitePro from
1. Native Interface
2. Eclipse plug-in
3. Browser access

Integrated with the IBM Software Development Platform
Next Manage Activities, Change and Workflow-
IBM Rational ClearQuest

- Provides activity-based management
- Manages defects, enhancements, issues and workflow process
- Provides predefined configurations and automatic e-mail notification and submission
- Scales regardless of team size, location or platform
- Integrated with IBM Software Development Platform

Access ClearQuest from:
1. Native Interface
2. Eclipse plug-in
3. Browser access

Roles, activities and artifacts coordinated by IBM Software Development Platform

DEMO: Business Analyst Enhances Request

Business Analyst will:
1. Use UML to understand the application
   - Define new requirement using RequisitePro
   - Analyze impact of new requirement
2. Use ClearQuest to start a new activity
What Did You Just See?
Communicate Requirements & Activities Across Lifecycle

Single Requirements & Activity & workflow Repository for the entire software team!

1. Analyze and design
2. Construct / develop
3. Unit Test

Pre-production Testing
Integration
Deploy

Business Requirements
Workflow & Activities

RequisitePro - Requirements
1. Use Case Documents
2. Eclipse plug-in
3. Auto e-mail for team
4. Relationships and traceability
5. Distributed support

ClearQuest - Activities
1. Manage activities and workflow
2. Eclipse plug-in
3. Auto e-mail for team
4. Distributed support

What Did You Just See?
Communicate Requirements & Activities Across Lifecycle

Fatal Flaw for Mercury Quality Center

- No requirements & activity management across the Software Development Lifecycle

Activity
Mercury provides some activity tracking designed for testers. It does not do activity tracking for the software development team

Requirements
Mercury provides requirements designed for testers not the software development team

46% of required functions do not make it into released products

Source: Gartner
Fatal Flaw for Microsoft Visual Studio.NET

- No requirements & activity management across the Software Development Lifecycle

Activity
Microsoft provides no activity tracking across the software development lifecycle

Requirements
Microsoft provides no requirements tracking across the software development lifecycle

46% of required functions do not make it into released products
Source: Gartner

Source: IBM Competitive Project Office Research, 2005

DEMO: Project Manager Starts Project and Assigns Resources

IBM ClearQuest

Business Analyst
Project Manager
Architect
Developer
Tester

1. Use ClearQuest to understand the activity
2. Use ClearQuest to determine workload assignment and assign someone to work on the enhancement

Roles, activities and artifacts coordinated by IBM Software Development Platform
What Did You Just See?

**A change workflow state model in action**

1. Enables a structured, measurable workflow
2. Project Manager and Team communicate effectively
3. Workloads are managed

**Bottom Line:** Change tracking

1. Enables iterative development
2. Reduces project risk

---

DEMO: MODEL the Enhancement

Architect will:
1. Receive notification from ClearQuest
2. Check the requirements using RequisitePro
3. Use UML to understand the application and enhancement and Model the enhancement

Roles, activities and artifacts coordinated by IBM Software Development Platform
**What Did You Just See?**

**Architect used UML Modeling**
1. Use case diagram
2. Sequence diagram
3. Class diagram

**Productivity is enhanced**
1. Requirements are quickly identified
2. Easily identify objects to change
3. Visual UML diagrams document system architecture
4. Communication complete between Business Analyst and Architect

---

**Unified Change Management**

**Manage Change and Complexity**
- Simplify and control change
- Associate artifacts with activities

**IBM Rational ClearQuest**
Manages activities

**IBM Rational ClearCase**
Manages artifacts

Build the system

Which activities in build
1. Calculate Premium
2. Create a new class called RiskFactor
3. 10% discount for preferred customer

Which source code modules

Use activity association to include appropriate source code
**DEMO: MAKE the Enhancement**

Roles, activities and artifacts coordinated by IBM Software Development Platform

- **Business Analyst**
- **Project Manager**
- **Architect**
- **Developer**
- **Tester**

**Developer will:**
1. Receive notification from ClearQuest
2. Use the model to understand
3. Make the enhancement
4. Synchronize model with code

**What Did You Just See?**
**Developer Makes the Enhancement**

**Improve Effectiveness:**
*Manage Change and Accelerate Productivity*

- Developer receives assignment (ClearQuest) and understands enhancement (UML) → Improve Communication
- Developer uses built-in modeling tools to understand how to make enhancement → - Increase Quality
  - - Reduce errors
- 1. Developer makes the change in the model, and synchronizes model and code.
- 2. Changes to source code are associated with an activity.

**Productivity is enhanced**
- Requirements are quickly identified
- Easily identify objects to change
- Source code is associated with activities

**Developers**
- Software Architect
- Business Analyst
- Project Manager
- Architect
- Developer
- Tester
Recent Developer Productivity Study

Recent Branham Group Study

Developer Productivity Study - Comparing IBM Developer Tools to Microsoft Developer Tools

IBM Development Environment VS Microsoft Development Environment

Which one is more productive?

Report available at: http://www.branhamgroup.com/tools_study

Source: Branham Group Inc.

Developer Productivity Study – Top 10 Productivity Lessons

1. The IBM tools were more productive in 7 out of 8 applications built
2. Almost twice as many Microsoft tools and support software was required to build the 8 applications
3. The IBM tools met all of the application requirements; Microsoft tools only met 4 out of 8
4. Microsoft had 3X the number of lines of code that were manually generated by hand
5. IBM provided wizards to build all the applications. Microsoft provided wizards in 4 of the 8 applications.
6. IBM provided built in test environments in all 8 applications. Microsoft had to build test harness and forms in 6 of the eight applications.
7. Microsoft was unable to build a robust human workflow application
8. Microsoft was unable to simulate and do cost analysis of workflows
9. Microsoft was unable to easily build multiple related pages in a portlet
10. Microsoft was unable to build a UML sequence diagram from existing code. Also unable to synchronize UML code with C# code

Source: Branham Group Inc.
## Tools Productivity - Recent Study Proves IBM is More Productive

### IBM Development Environment is more productive for Server Side programming

<table>
<thead>
<tr>
<th>Application Scenario</th>
<th>IBM</th>
<th>Microsoft</th>
<th>Slower by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Build a Simple Web Application</td>
<td>251</td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>2. Build Web Service from Scratch</td>
<td>19</td>
<td>40</td>
<td>2.1X</td>
</tr>
<tr>
<td>3. Build Web Service from Existing Code</td>
<td>9</td>
<td>22</td>
<td>2.6X</td>
</tr>
<tr>
<td>4. Implement Distributed Transactions</td>
<td>27</td>
<td>41</td>
<td>1.5X</td>
</tr>
<tr>
<td>5. Build &amp; Deploy Human &amp; Automated flows</td>
<td>145</td>
<td>Unsupported</td>
<td></td>
</tr>
<tr>
<td>6. Build Portlet Function for a Portal</td>
<td>44</td>
<td>96</td>
<td>2.2X</td>
</tr>
<tr>
<td>7. Connect Portlet to Enterprise Systems</td>
<td>23</td>
<td>74</td>
<td>3.2X</td>
</tr>
<tr>
<td>8. Model Key Components of Application</td>
<td>2</td>
<td>5</td>
<td>2.4X</td>
</tr>
</tbody>
</table>

Note: All time is in minutes

Source: Branham Group Inc.
How Was this Integration Achieved?

*Raising the level of abstraction, transforming development into one integrated lifecycle process*

Eclipse Model Framework (EMF)
- Meta Data Storage for the IBM Software Development Platform
- Point of integration (like a single repository)

Eclipse 3.0 Core Function

Software Lifecycle Development – Competitive Landscape

<table>
<thead>
<tr>
<th>Function</th>
<th>IBM</th>
<th>MERCURY</th>
<th>Microsoft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated Software Development Lifecycle</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Design and Modeling</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Integrated Developer Environment</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Managing Change across Lifecycle</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Heterogeneous Support</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

**LEGEND**
- ✗ No Function
- ✓ Function
- ✓Weak Function

Source: IBM Competitive Project Office, 2005
Software Lifecycle Development Platform

Customer Success

Standardized on the IBM Software Development Platform

Challenge
- Differentiate Unisys in the systems integrator marketplace

Solution
- 3D Visual Enterprise
  - Unique consulting framework

Results
- Accelerated client success
  - 25-40% cycle time
  - 75-100% productivity
  - 25-60% cost savings

http://www.ibm.com/software/success

"IBM Rational forms the kernel of our 3D Visual Enterprise solution. Using Rational tools and best practices has helped us achieve a 6 to 18 month market lead over our competitors."

Ed Ferrara, 3D-VE Program Development Manager

Are You as Productive as You Could Be?