Asset Based Development
Grant Larsen
gjlarsen@us.ibm.com
November 2003
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
- Getting Started
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
- Getting Started
**Business Challenges**

**Cost**
- Reduce business & IT costs
  - Decrease complexity
  - Decrease blank sheet
  - Increase productivity
  - Increase quality
  - Increase consistency
  - Increase maintainability

**Agile**
- Organization is nimble and responsive to market forces

**Leverage**
- Resources effectively applied to business problem

**Collaboration**
- Integrate end-to-end business processes

**Alignment**
- Business Technology
- Technology delivers business solutions

**ROI**
- Increased returns from IT investments
Organizations’ Needs

To decrease the time-to-value and produce systems in competitive time

Cost
- Decrease business & IT costs
- Decrease complexity
- Decrease blank sheet
- Increase productivity
- Increase quality
- Increase consistency
- Increase maintainability

ROI
- Increased returns from IT investments

Agile
- Organization is nimble and responsive to market forces
- Software architectures which are extensible and easily integrate the enterprise

Leverage
- Resources effectively applied to business problems
- To leverage their scarce resources to a broader audience

Collaboration
- Integrate end-to-end business processes

Alignment
- Business
- Technology
- Technology delivers business solutions

Process & alignment guidance and seamless tooling
How is this done?

- Asset-based Development (ABD) delivers the process guidance, tooling, standards, and assets for delivering these results
  - It is developing software solutions (re)using cohesive, documented software artifacts
  - It is organizing software development in a way that leverages previous investments and influences the nature of future investments

These results

- Teams know what to do and how to solve business problems, using...
- architected solutions, models, artifacts, which are...
- organized for use throughout the enterprise...
- delivering timely value.

Are delivered by

- **Process**
  - RUP ABD guidance
- **Tooling**
  - Rational XDE, Rose, ClearQuest, ClearCase
- **Standards**
  - UML, MDA
  - RAS
- **Assets**
  - Leveraged development
What’s an Asset?

- An Asset is
  - a collection of Asset Artifacts
  - which provide a solution to a problem
    - for a given context
    - with rules for usage
    - and variability points
- What are Asset Artifacts?
  - Workproducts from the software process
    - Requirements, Models,
      Source code, Binary files, Tests, Plans, and so on…
- Kinds of assets
  - Components, patterns,
    web services, frameworks, templates, …
Asset Qualities

- Should be easy to use, customize and should declare the context wherein it should be reused
- Tight cohesion, loose coupling
- Purpose and intent should be easy to understand
- Should be easy to determine the asset’s match to a particular context
Barriers & Keys To Effective Reuse

- **Reuse Barriers**
  - A lack of incentive
    - Teams focus on delivering to their project goals rather than investing energy into creating reusable things
  - A lack of trust
    - Manifested in a "not invented here" mentality
  - An architecture that isn't organized to support a relatively easy insertion of new items
  - Difficulty finding and using the right asset

- **Effective Reuse**
  - Need engineering support, tooling and education so teams can build and use something that is re-usable
  - Organization must support a reuse culture
  - Software development processes must be able to work with asset-based development techniques
  - Costs and returns on investments must be managed

- Ivar Jacobson (*Software Reuse*)
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
ABD Process

- Asset-based Development Key Workflows
  - Reuse Program Management
  - Asset Management
  - Candidate Asset Identification
  - Asset Production
  - Asset Consumption
Where Does It Fit?

Asset-based Development process workflows can occur within any phase and across multiple disciplines.
ABD Process Spans Multiple Levels

**Organization-level**

- Asset Development Team
  - Focus is on reuse program and asset management, asset identification and production, but may perform asset consumption

**Project-level**

- Application Development Team
  - Focus is on asset consumption, but may perform candidate asset identification and asset production

- Asset production
- Candidate asset identification
- Determining the cost/benefit of harvesting or producing reusable assets
- Management of reusable asset libraries
- Initiating projects to harvest/produce reusable assets
- Certifying and accepting assets into asset libraries

- Application development
- Asset consumption
- Candidate asset identification
- Asset production
Asset Production Process

Harvest Asset Artifacts

Develop Asset Artifacts

Asset Artifact

Package Reusable Asset

Reusable Asset

Reusable Asset Library

Asset Librarian

responsible for
Asset Consumption Process

1. Identify Candidate Asset
2. Provide Reuse Feedback
3. Locate Asset
4. Apply Asset
5. Reuse Feedback
6. Reusable Asset

Asset Librarian responsible for

Locating Assets Using Rational XDE
Applying Assets Using Rational XDE
Re-using Assets

- Assessing asset applicability is essentially performing a fit-gap analysis to see if an asset will work in the required context
  - Business fit
    - Financial, legal and maintenance
  - Organizational fit
    - Acceptance, who will use it
  - Process fit
    - Support
  - Engineering fit
    - Understanding, completeness, reliability, error handling, cohesion/coupling, portability
Asset Reuse Scope and Formality

- Asset-based Development process adjusts as reuse scope and formality changes

- Intra-team
  - Sharing assets amongst team members

- Inter-team
  - Sharing assets between teams

- Inter-project
  - Sharing assets between projects
RUP Asset-based Development Configuration

Available from the Rational Developer Network

Concept: Asset-Based Development in RUP

Topics
- Introduction
- RUP and the Organization-Level Reuse Process
- Asset-Based Development on RUP Projects

Introduction

Asset-based development is developing software solutions (re)using cohesive, documented software artifacts, or workproducts. It is organizing software development in a way that leverages previous investments and influences the nature of future investments. It is speeding up development, and reducing cost and risk by reusing assets and artifacts to solve recurring problems.

For more information on reusable assets, see Concepts: Reusable Asset.

RUP and the Organization-Level Reuse Process

An effective reuse program involves the interaction of two separate processes. One is the process of managing reuse at the organizational level. The other is the process of using and producing assets on individual projects.

Organization-level concerns include:
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
UML, MDA

- UML models are used to represent various views of the business and the architectures
- MDA is used to leverage models to produce and maintain systems faster, cheaper, better
  - The OMG MDA® separates the fundamental logic behind a specification from the specifics of the particular middleware that implements it.
- MDA drives model transformation techniques to develop systems
- MDD is a development approach where the models drive the entire development lifecycle

Asset-based Development implements systems using assets that realize MDA techniques
RAS

- Reusable Asset Specification (RAS)
  - Describes the structure and nature of assets
  - Reduces the friction on development transactions thru standard, consistent packaging
  - Each asset is described using these sections
    - Classification
    - Solution
    - Usage
    - Related Assets
Packaging RAS Assets

RAS

Asset
Classification
Solution
Usage
Related-Assets

RAS is realized in rasset.xml

Artifacts & RAS XML file

rasset.xml is the manifest & points to each file

Zipped into a .ras file

assetx.ras

ClearCase VOB
File System
RAS Repository
RAS Profiles

- RAS describes the general characteristics of software assets
- RAS is extended to represent specific kinds of software assets
- RAS Profiles describe a specific type of asset
  - Current profiles available
    - Default profile version 2.1
    - Default component profile version 1.1
    - Default web service profile version 1.1
  - Create your own profile
    - Companies are creating their own profile to describe their assets – XDE addins can be created to handle these asset types
**UML, MDA, RAS**

**UML**
Language to specify components and systems

**MDA**
Specifies model organization for business-driven component architectures

**RAS**
Packages patterns, components and other artifacts as assets to leverage the business
RAS To The OMG

- RAS was developed in a Consortium of IBM Rational, Microsoft, and Component Source, and others

- RAS is being submitted to the OMG
  - Co-submitters
    - Adaptive (http://www.adaptive.com)
    - Blueprint Technologies (http://www.blueprinttech.com)
    - ComponentSource (http://www.componentsource.com)
    - Flashline (http://www.flashline.com)
    - IBM (http://www.ibm.com)
    - LogicLibrary (http://www.logiclibrary.com)
    - OSTnet (http://www.ostnet.com)
  - Supporters
    - ABB, Aetna, Borland Software, Cap Gemini Ernst & Young, Caterpillar, Fujitsu, IconMedialab, Iocore-7n, Jaczone, Kantega, Martin Griss Associates, OSTnet, Praxis Engineering Technologies, RDA Corporation, Telstra, Unisys, USPTO, Volvo, Xansa
The RAS RFC document can be found at

The supporting documents that accompany the RAS RFC can be found at
    - Rose models of the existing XML schema and the recently inserted MOF 2.0 XMI representation
    - RAS RFC XML schema and MOF 2.0 XMI XML schema documents which realize the UML models
    - WSDL document for the lightweight RAS Repository Service
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
- Getting Started
ABD Tools

Reuse Management

Asset Production
- XDE
  - modify/refine
- CC Development Repository
  - check-in

Asset Management
- Less formal
- CC Production Ready Repository
  - check-in
  - deliver

Asset Consumption
- XDE
  - Map to multiple repositories
  - apply & customize
- CC Production Ready Repository
  - deliver
  - search
- RAS Repository Service
  - search

Coordination, metrics, reporting to mgmt

Coordination, metrics, reporting to mgmt

Asset Certification
- ClearQuest
  - Packaged
  - Submitted
  - RWG Reviewed
  - Reviewed
  - Librarian Reviewed
  - Production Ready
  - Reviewer Logins

More formal
DEMO
Case Study: Volvo IT

- Objectives for JNX Framework Project
  - Simplify the workflow for application development
  - Achieve consistency and maintenance benefits
  - Leverage OMG’s MDA PIM and PSM as the strategy for application development
  - Achieve a robust and reliable architecture that will shield the programmers from the actual plumbing

- Target Audience
  - 400 – 600 developers
    - Have not had training in UML
    - Use waterfall software process
  - 200 developers
    - Have had training in UML and RUP
  - 10 – 20 Architects
    - Experienced with UML
    - Largely use iterative development
Volvo IT – JNX Framework - Applying MDA in XDE

Rational XDE

- Platform-independent Model
  - Domain Model
  - Platform-specific Model
    - Design Model
    - Implementation Model

RAS Repository Service

- Reusable asset (.ras file)
- Reusable asset (.ras file)
- Reusable asset (.ras file)

- Model templates
- Patterns
- Patterns
- Patterns
- Patterns

- Model templates extend Domain Model
- Patterns transform Platform-specific Model
- Patterns integrate Design Model
- Patterns validate Design Model

- Code & Test
- Business logic

- Requirements synchronize
- Requirements validate

IBM Software Group | Rational software
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
- Getting Started
Assets To Get You Started Now

- Rational XDE – patterns in the box
  - GoF patterns (Java, C#, C++)
  - J2EE patterns
- Rational Developer Network
  - Patterns and other assets on RDN ready for use in Rational XDE
When To Create Assets

- The question to ask: “Is it worthwhile for me to create this reusable asset?”

- Items to consider when answering that question:
  - What recurring problems do you see?
  - Is there a solution to a recurring problem?
  - Is there an area needing enhanced productivity or consistency of deliverables?
  - What’s the impact on performance and memory usage?
  - What’s the stability of the domain and requirements?
  - What’s the viability of the technology? Is obsolescence a concern?
  - Have the target consumers been identified?
  - Who will own and maintain the asset?
Business Processes & IT Solutions

Business processes are intertwined with the IT infrastructure – impacting the organization’s ability to respond quickly to market forces.
Patterns: Aligning The Business & IT Solutions [1]

Patterns can describe the business structure and the component architecture to support adaptability and predictability.
The business patterns are mapped to a component architecture which realizes the business functions and is deployed on a runtime infrastructure.
Patterns: Aligning The Business & IT Solutions [3]

- Patterns address specific architectures as well as cross-cutting concerns for the system.
- Pattern can be applied in succession.
- Patterns need to be specified and created for each category.
Patterns: Aligning The Business & IT Solutions [4]

1. Business process / Entity / function modeling

2. High-level solution architecture modeling [IBM patterns for e-business]

3. Use cases for Business processes

4. Use Case realizations

5. Behavioral modeling Entity modeling

6. Component specification
    7. Component design and workflow design

8. Component creation, Database generation mapping

9. Code generation

10. Deploy and monitor

Shortened time to production. Fewer Mistakes. Lifecycle traceability.
Patterns: Aligning The Business & IT Solutions [5]

- Patterns, and other assets, are easily accessible through the RAS Repository Service for XDE [Eclipse/WSAD/VS.NET] users
Kits

- A Kit is a comprehensive, structured collection of assets, guidance, examples and tools addressing anticipated needs of a target audience, in a specific development context and domain.

![Diagram showing Kits and their components.](image)
Agenda

- Introduction
- Asset-based Development
  - Process
  - Standards
  - Tooling
  - Assets
- Getting Started
Getting Started

First Things First

1. Management support for asset-based development

2. High quality, relevant assets must be identified
   - Must be critically needed by the target asset consumer and has the highest probability for early return on investment. The asset consumer must be well understood.

3. Policies to identify, understand, and control reinvention in the organization and to measure progress

4. Dedication to recognize and encourage all forms of software assets that have net benefits to the organization

Begin Pilot Project

1. Start small, with narrow scope, such as for one project
   - Set up tooling
   - Harvest & create assets and deliver to asset consumers
   - Measure & report & refine

2. Increase scope, such as for two projects
   - Refine policies, procedures, guidelines
   - Review/refine architectures to support assets
   - Harvest, create, certify assets and deliver to asset consumers
   - Measure & report & refine
Summary

- Companies need to be more nimble to meet their customers’ needs and to address market changes.
- Asset-based Development (ABD) delivers the process guidance, tooling, standards, and assets for delivering these results.

These results

- Teams know what to do and how to solve business problems, using...
- Architected solutions, models, artifacts, which are...
- Organized for use throughout the enterprise...
- Delivering timely value.

Are delivered by

- Process
  - RUP ABD guidance
- Tooling
  - Rational XDE, ClearQuest, ClearCase
- Standards
  - UML, MDA
  - RAS
- Assets
  - Leveraged development
Thank You