Overview of Visual Studio 2005 Team Edition for Software Architects

Ken Garove
Visual Studio Technology Specialist
Microsoft Federal
http://blogs.msdn.com/federaldev

Topics
- Dynamic Systems Initiative Overview
- Systems Definition Model
- Domain Specific Languages
- Visual Studio 2005 Team Edition for Software Architects

Increased Complexity
- Distributed systems increase complexity, and complexity spans application life cycle

Towards Self-Managing Dynamic Systems
- Organizations find cost and complexity of distributed systems operation high
- Pervasive inflexibility
  - brittle infrastructure
  - labor-intensive management
  - inefficient capacity utilization
- Dynamic and self-managing IT systems mitigate distributed systems complexity

Capturing Knowledge
- Comprehensive Solution
  - Model-based software development tools help IT managers and software development teams capture system knowledge for operations
  - Operationally-aware Windows platform with virtualization and management
  - Model-based management tools that cover entire management scope
  - Application health models for production monitoring
Leveraging Knowledge for Operations Efficiency

- Root Cause Analysis
- Health Model

Balanced Workloads

- The ability to decouple workloads from the resources needed to execute them makes dynamic resourcing possible

Designing for Operations

- SDM encourages the creation of reusable, prescriptive models with embedded best practices
- SDM helps to close the gap between how administrators think and how they operate
- SDM model provides a single point of integration, coordination, and policy enforcement across a distributed system
- A system model aids in a design's deployment and management
- SDM is a live model that can evolve and be enhanced during the life of the design

System Definition Model

Domain Specific Languages

- Microsoft has learned from past industry experiences, and plans to avoid the pitfalls of CASE
  - A model should be a first-class artifact in a project—not just a piece of documentation
  - A model represents a set of abstractions that support a developer in a well-defined aspect of development
  - Models readily support consistency checks and other forms of analysis

Visual Studio Team System

- Change Management
- Work Item Tracking
- Reporting
- Integration Services
- Team Model
- Task Board
- Code Coverage
- Unit Testing
- Code Maturity
- Test Case Management
- Code Coverage
- Visual Studio Team Foundation Server CAL
- Visual Studio Professional Edition
Key Architect Themes

- **Executable Design - SDM**
  - Increase value of design
  - Improve collaboration between architects and developers
  - Synchronize design with code
- **Deployable Design**
  - Increase predictability of deployment
  - Improve collaboration across IT teams
  - Validate system designs against a model
  - Generate health models for MOM

Resources

- AVICode: [http://www.avicode.com](http://www.avicode.com)