

# PRAIRIE DEV CON 2010

PRAIRIE DEVELOPER CONFERENCE

**Microsoft®**



## Developer Foundation Track

Sponsored By

*Jet* BRAINS

# Aspect Oriented Programming

Donald Belcham



# What is an Aspect?

Modular component that allows for good separation of concerns while still servicing cross cutting concerns

# Horizontal Layers

**Service Layer**

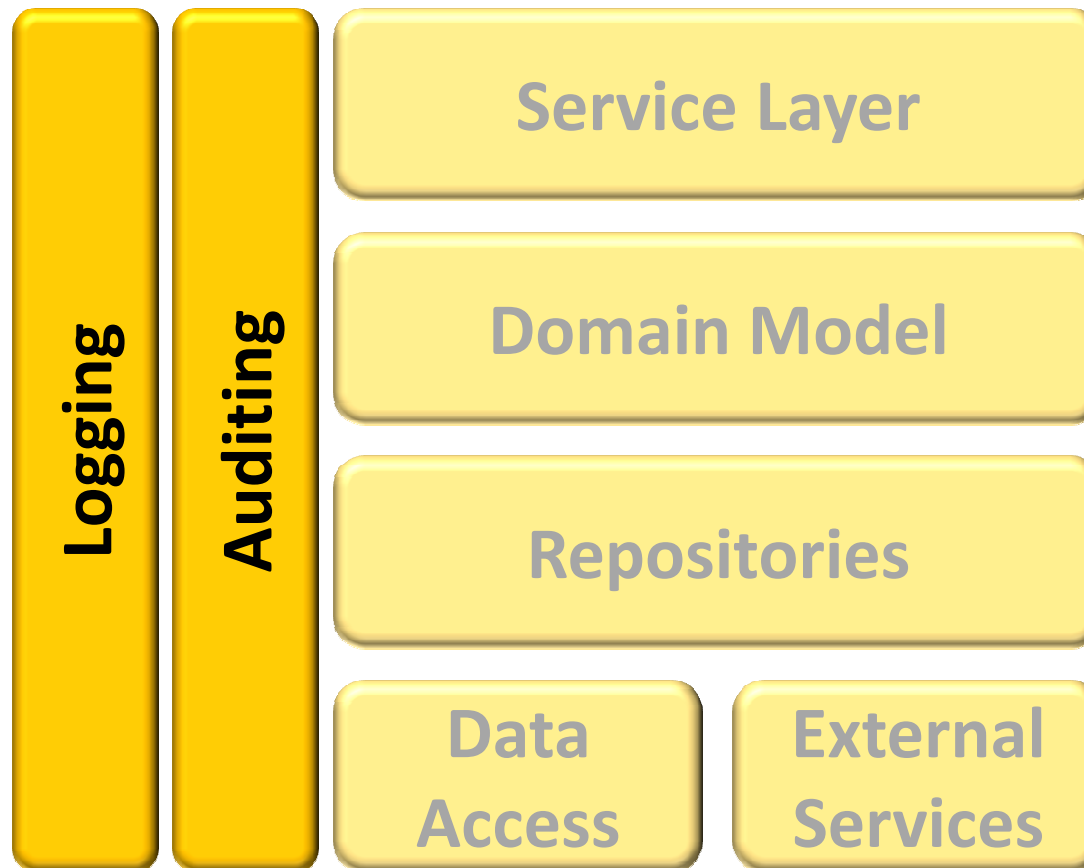
**Domain Model**

**Repositories**

**Data  
Access**

**External  
Services**

# Cross Cutting Concerns (Aspects)



# Logging As You See It Now

```
public void AddNew(Customer customerToAdd)
{
    Log.Usage("CustomerServices.AddNew call started");

    if (!_newCustomerValidator.For(customerToAdd))
    {
        Log.Error("Invalid Customer passed to AddNew");
        throw new ArgumentException("Customer is not valid");
    }

    _customerRepository.SaveOrUpdate(customerToAdd);

    Log.Usage("CustomerServices.AddNew call completed");
}
```

# Logging As We'd Like to See It

```
public void Delete(Customer customerToDelete)
{
    if (!_deleteCustomerValidator.For(customerToDelete))
    {
        throw new ArgumentException("Customer is not valid for deletion");
    }

    _customerRepository.Delete(customerToDelete);
}
```

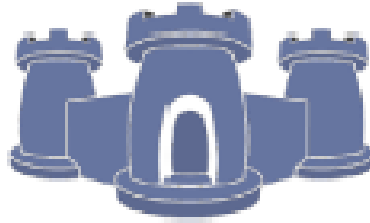
# How Though?







- Post process the compiled assembly
- Before, After, On Exception
- Attribute based



- Interface implementations
- Before, After
- Configuration based

Recap

donald.belcham@igloocoder.com  
@dbelcham  
www.igloocoder.com

