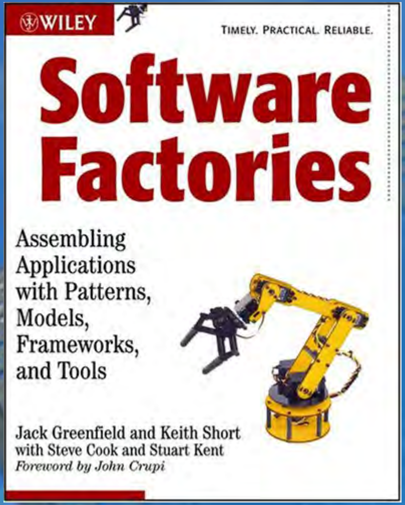
Software Factories



WILEY

TIMELY. PRACTICAL. RELIABLE.

Software Factories

Assembling Applications with Patterns, Models, Frameworks, and Tools

Jack Greenfield and Keith Short
with Steve Cook and Stuart Kent
Foreword by John Crupi

Steve Cook
Architect
Enterprise Tools, Visual Studio
Microsoft Corporation

The Software Crisis (ca. 2004)

- $250B/yr in US (average $430K to $2.3M per project)
  - 16% on time and budget but deliver less than planned (avg 42%)
  - 53% overrun (avg 189%)
  - 31% are canceled, losing $140B/yr

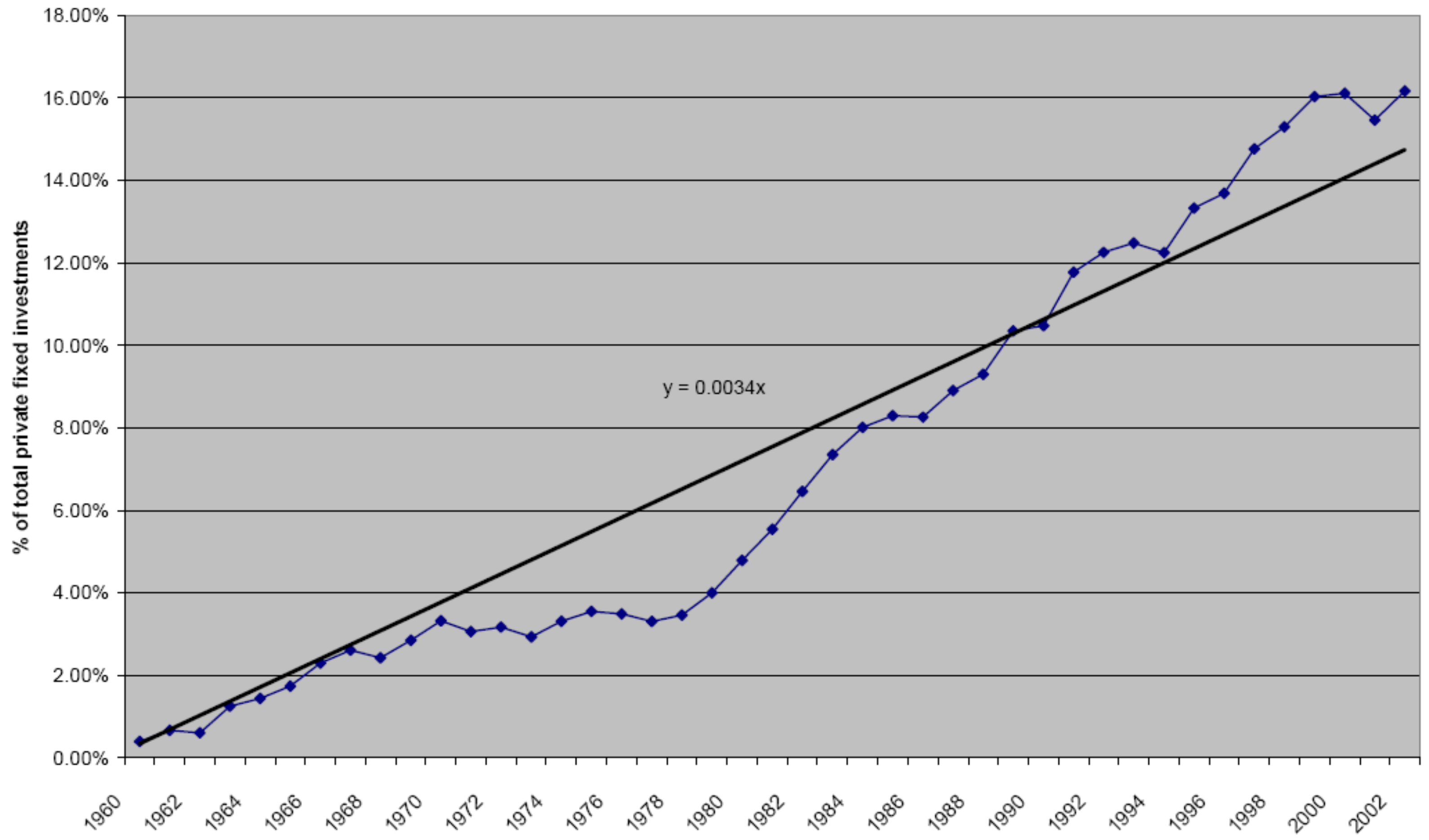


Figure 2. Computer and software investments in the US, 1960-2002 (calculated from US National Income and Product Accounts).[11]

From “Kurzweil, Moore, and Accelerating Change”, Ilkka Tuomi, http://www.jrc.es/~tuomiil/articles/Kurzweil.pdf

Agenda

- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

Agenda

- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

Questions We Hear…

- What types of systems can I build?
  - What’s the architecture of each type?
- How do I go from requirements to deployment?
  - What artifacts do I need to build?
  - How are they related?
  - What are the key decisions that need to be made?
- Why are methodologies so abstract?
  - Why can’t I get concrete guidance for my project?
- Why isn’t modeling more effective?
  - Why can’t tools generate production quality code?
  - Why don’t models stay synchronized with code?
  - Why don’t models fit my file-oriented environment?

Is Agility The Answer?

- Agile methods optimize for change
  - Collaborating instead of documenting
  - Building and running in small iterations
  - Continuously validating requirements
  - Continuously refactoring the software
  - Time boxing or cash boxing the project
- Where do they fall short?
  - Don’t scale up to large or complex projects
  - Lack of documentation creates integration issues
  - Lack of metadata limits automation opportunities
  - One-off development of generic systems

Is Methodology the Answer?

- Software methods optimize for complexity
  - Prescribing roles, artifact, activities
  - Emphasizing requirements, analysis, design
  - Using general-purpose models to document architecture
- Where do they fall short?
  - Don’t respond rapidly to change
  - Coding, testing, debugging, instrumentation, deployment, management, maintenance
  - Informal modeling limits automation opportunities
  - One-off development of generic systems

Agenda

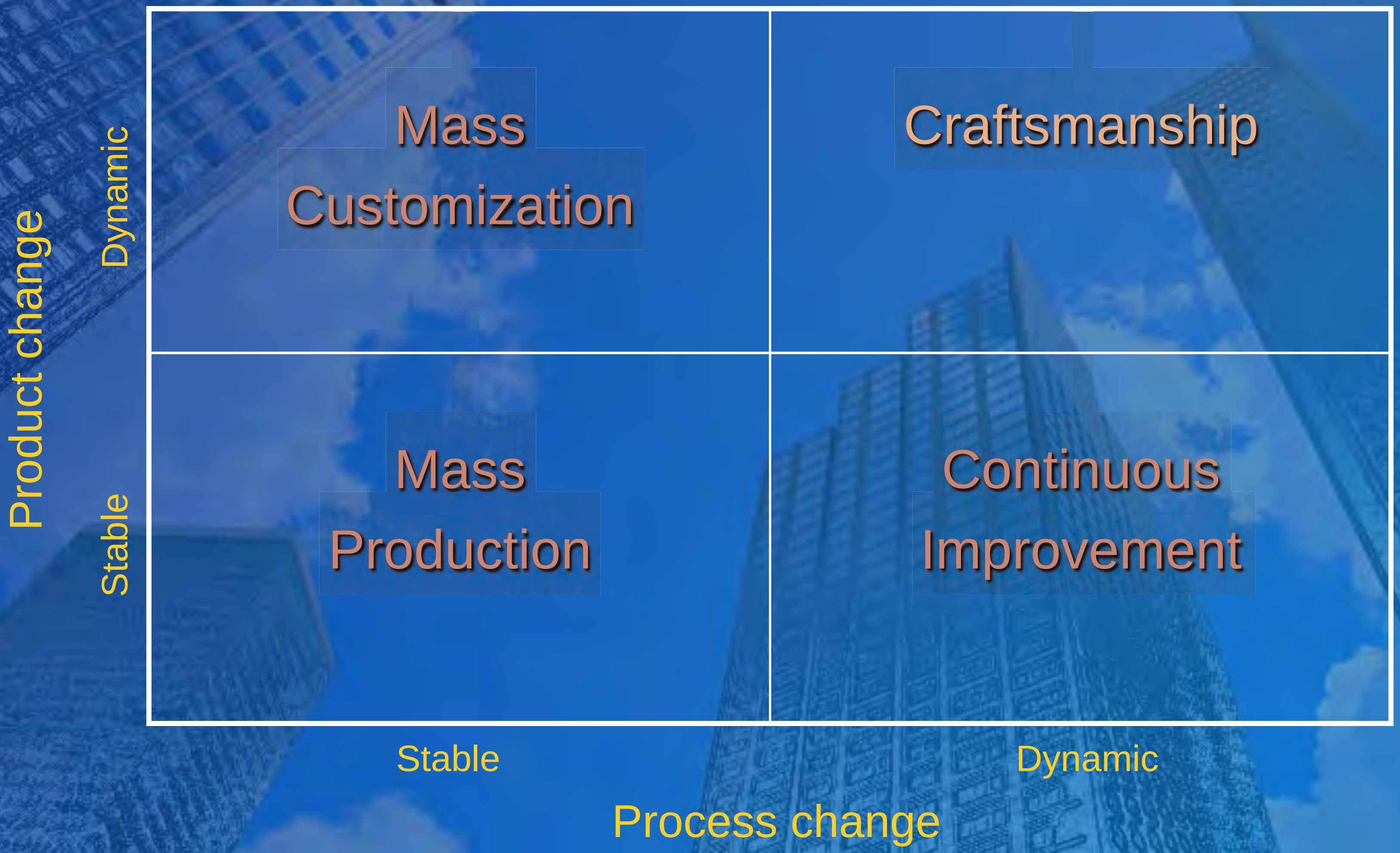
- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

Software Development as Craftsmanship

- Labor Intensive
- Generic Tools
- Generic Processes
- One off applications
- Hand stitched from scratch
- Minimal reuse

Overruns, defects, security holes, project failures

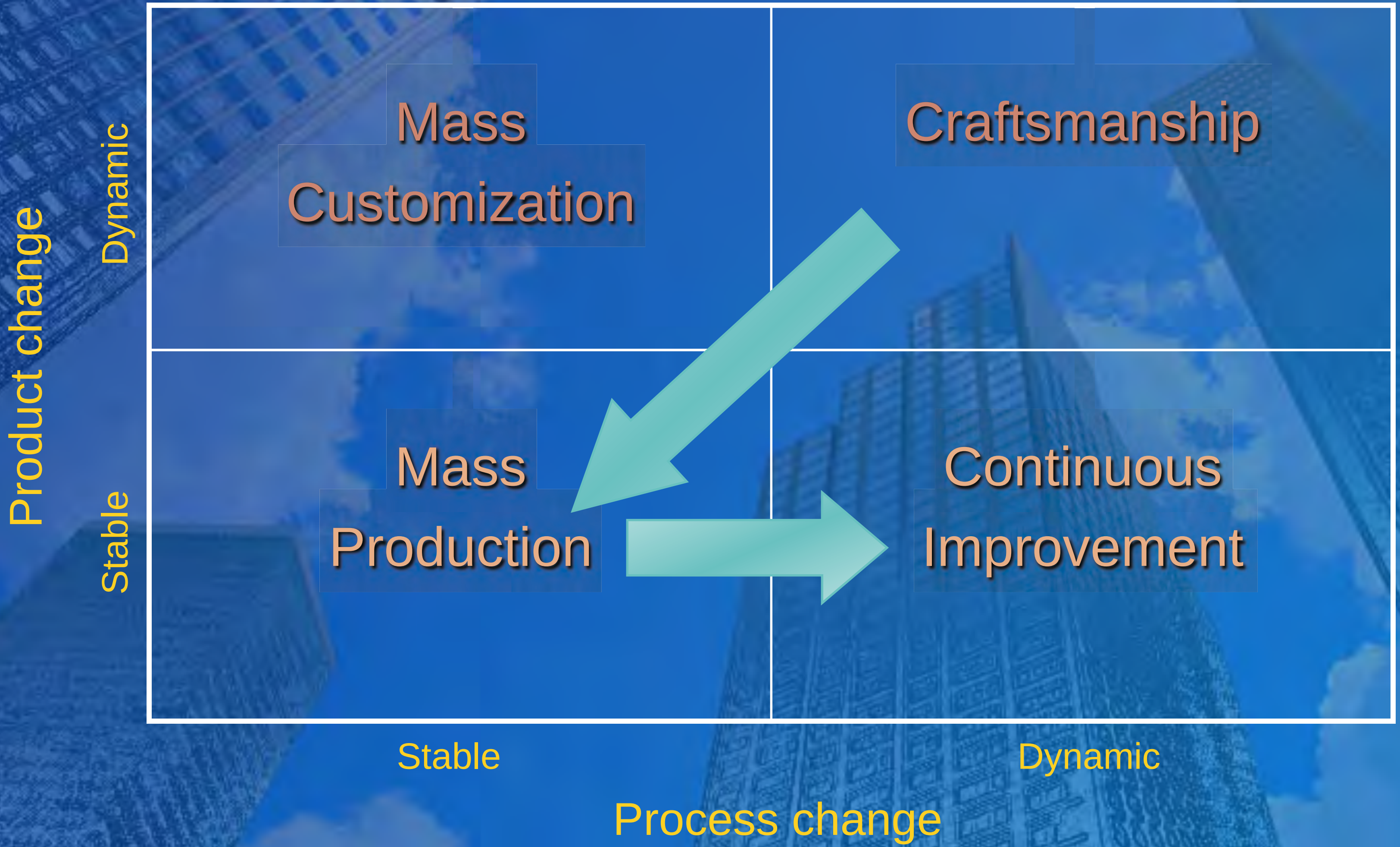
Industrialization



Exploiting Commonality

- We already exploit *economies of scale* to automate *production*
- Stamping out many identical *copies* of a *prototype*
- Used to produce CDs/ DVDs
- Does nothing to help *development*

Industrialization

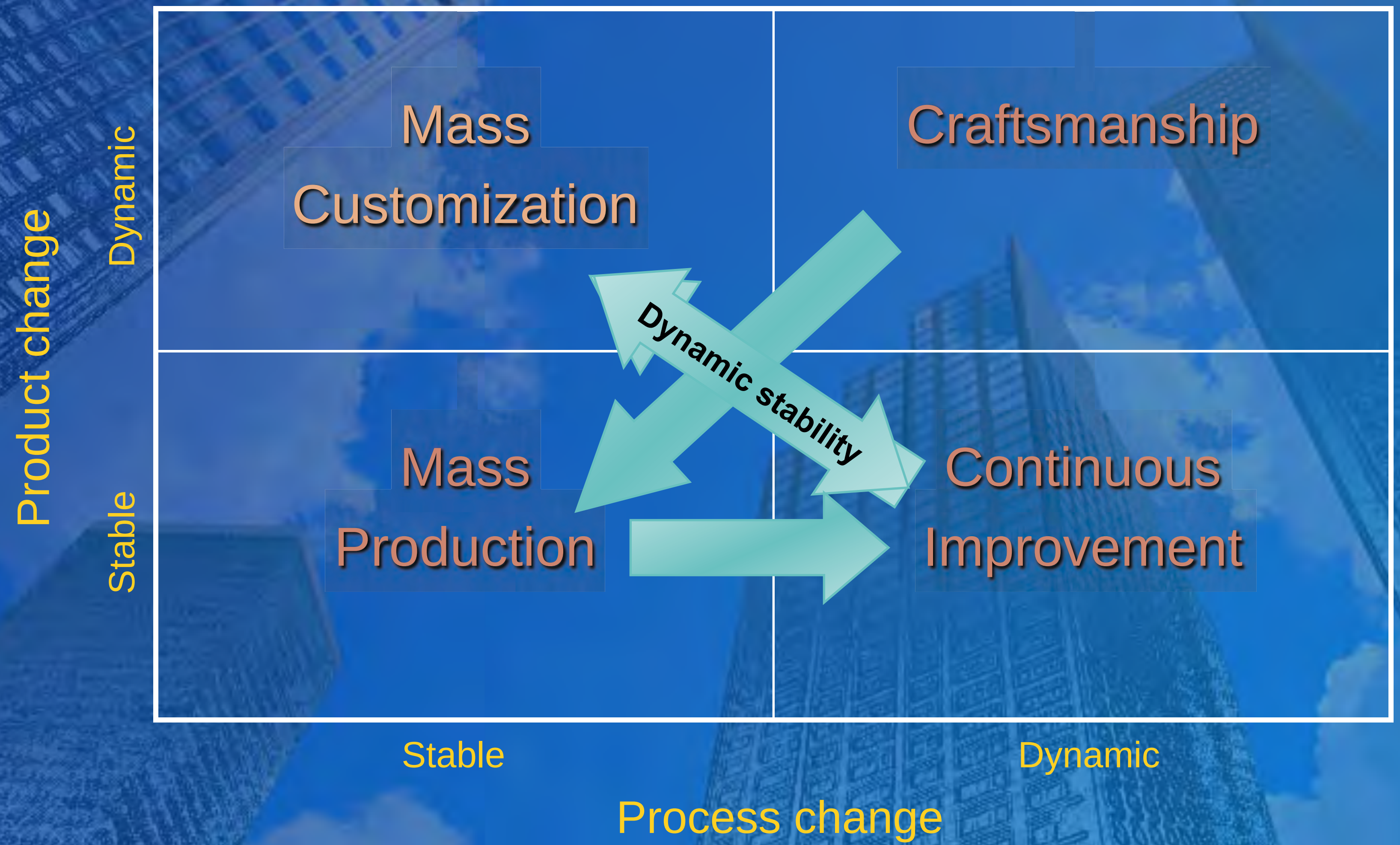


Exploiting Commonality

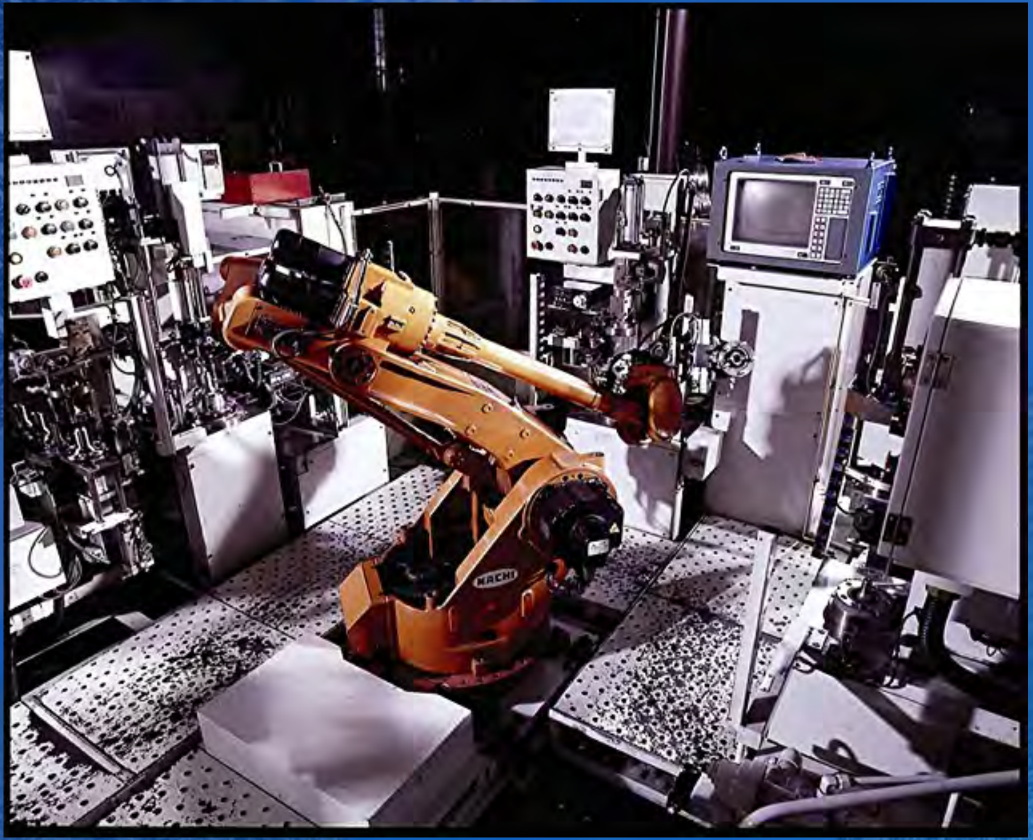
- We can also exploit *economies of scope*
- Reuse *designs* & *components*
- Build many similar but distinct *prototypes*
- Key is supporting *variability*

Define only the unique pieces of each system

Industrialization



Software Factories



- *Domain-specific process*
- *Domain-specific tools* & *languages*
- *Domain-specific content*
- *Automate* rote and menial tasks

General-purpose IDEs become domain-specific software factories

Agenda

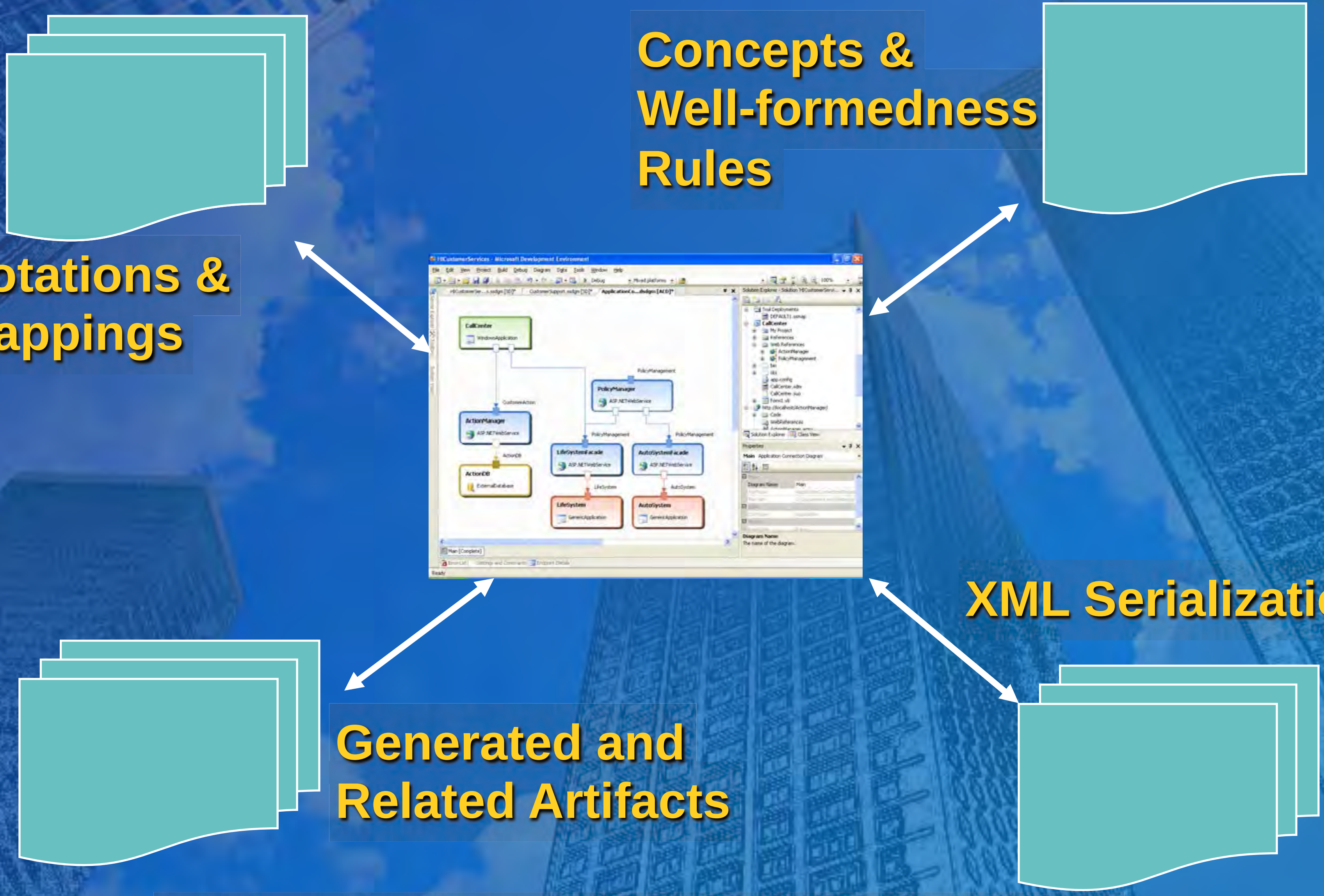
- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

Domain Specific Languages

- Focused on a single aspect of app building
  - Success in broad horizontal domains: SQL, Windows Form Designer
- Designed to support the concepts defined by an underlying framework
  - Automate rote tasks with effective code generation
- Increase agility by visualizing concepts, generating code and other artifacts, enabling rapid iteration
- Artifacts synchronized through integrated metadata

Building them must be fast, cheap and easy

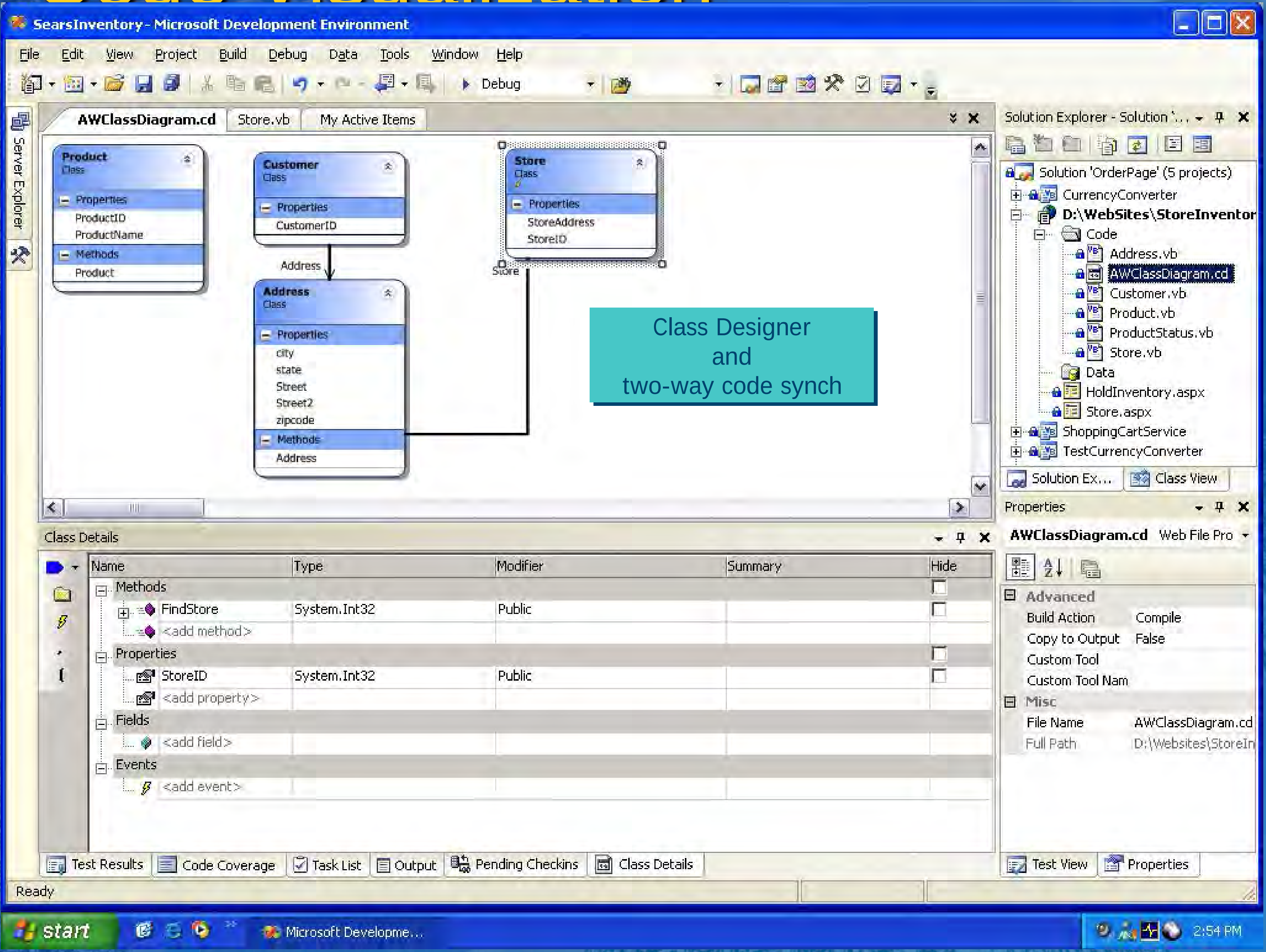
Building A DSL



Agenda

- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

Code Visualization



Class Designer
and
two-way code synch

Class Details

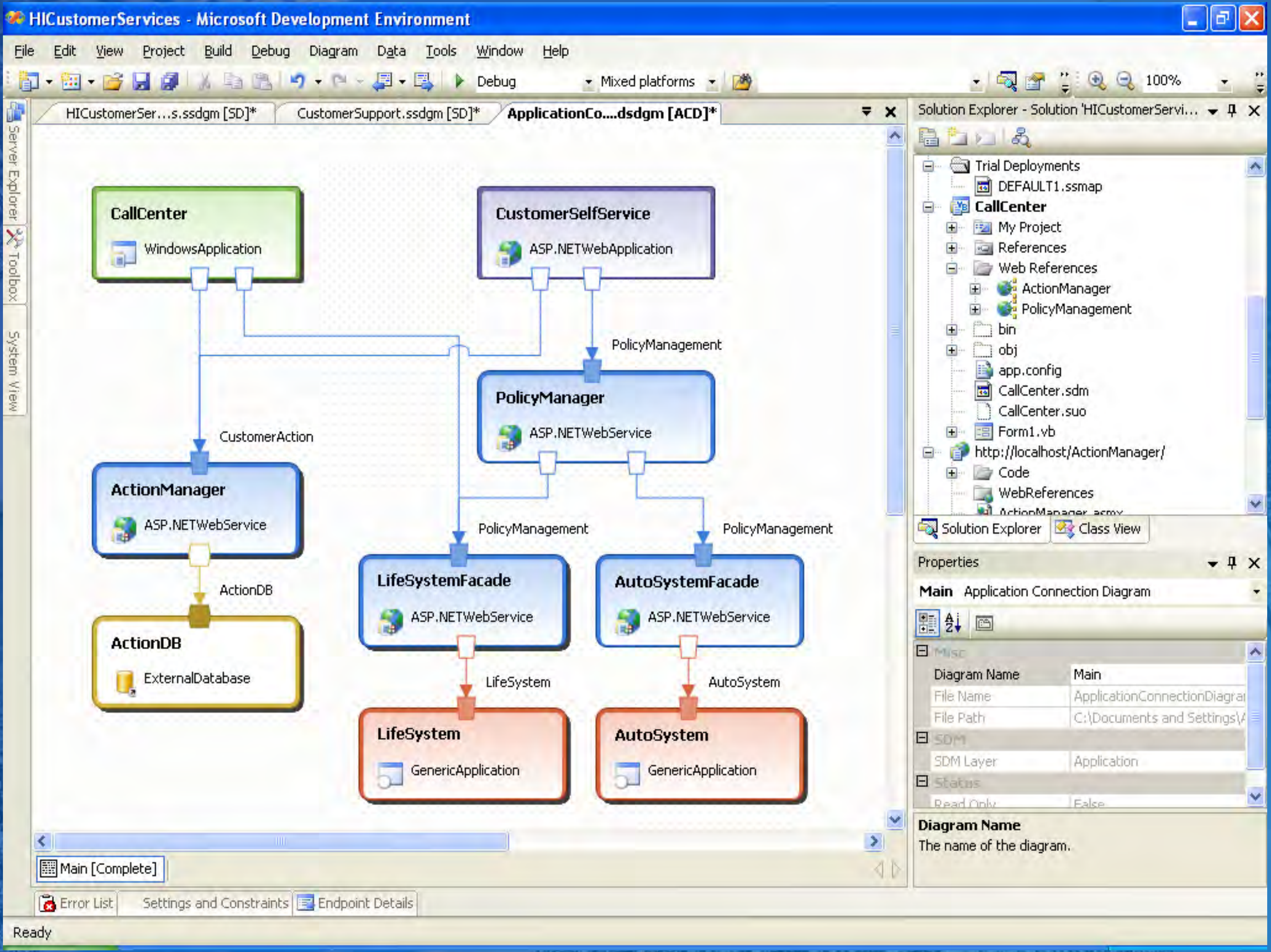
Name	Type	Modifier	Summary	Hide
Methods				☐
FindStore	System.Int32	Public		☐
<add method>				
Properties				☐
StoreID	System.Int32	Public		☐
<add property>				
Fields				
<add field>				
Events				
<add event>				

Properties

AWClassDiagram.cd Web File Pro

Advanced	
Build Action	Compile
Copy to Output	False
Custom Tool	
Custom Tool Nam	
Misc	
File Name	AWClassDiagram.cd
Full Path	D:\Websites\StoreIn

Vertical Mapping - System Design



Main Application Connection Diagram

Misc

Diagram Name	Main
File Name	ApplicationConnectionDiagra
File Path	C:\Documents and Settings\

SDM

SDM Layer	Application

Status

Read Only	False

Diagram Name
The name of the diagram.

Vertical Mapping - Data Center Design

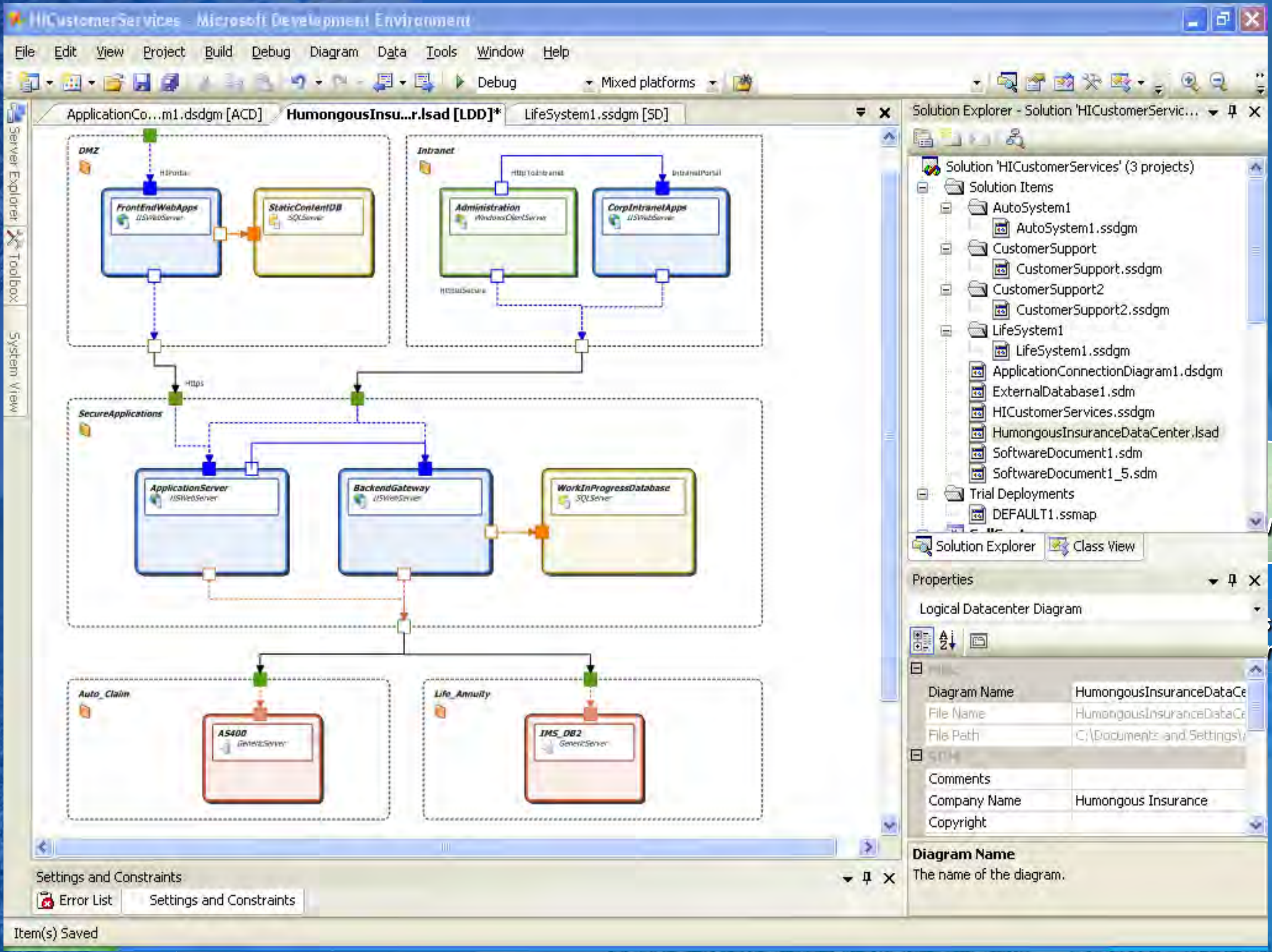


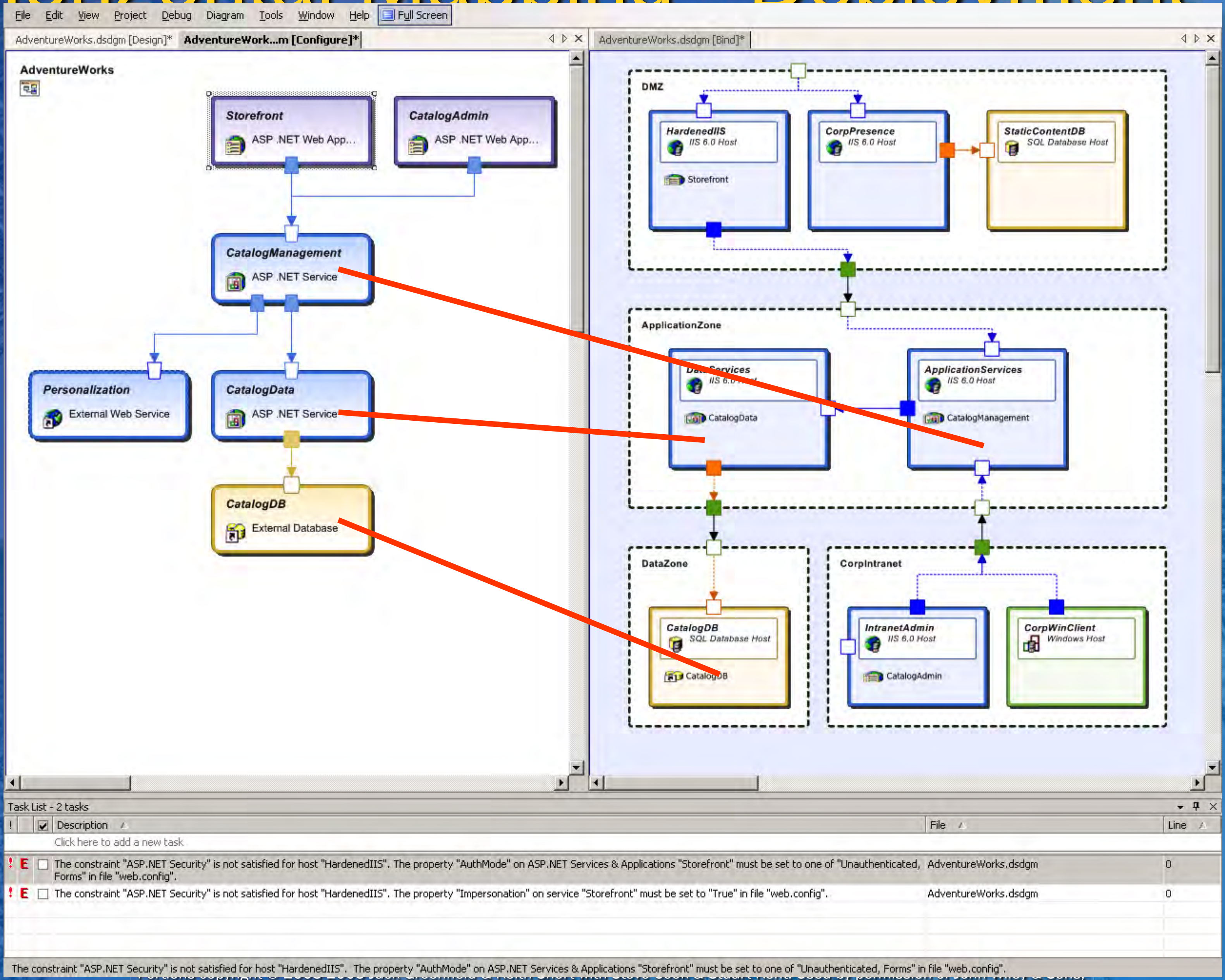
Diagram Name	HumongousInsuranceDataCe
File Name	HumongousInsuranceDataCe
File Path	C:\Documents and Settings\
Comments	
Company Name	Humongous Insurance
Copyright	

Diagram Name
The name of the diagram.

Center
ware

traction/
nement

Horizontal Mapping - Deployment

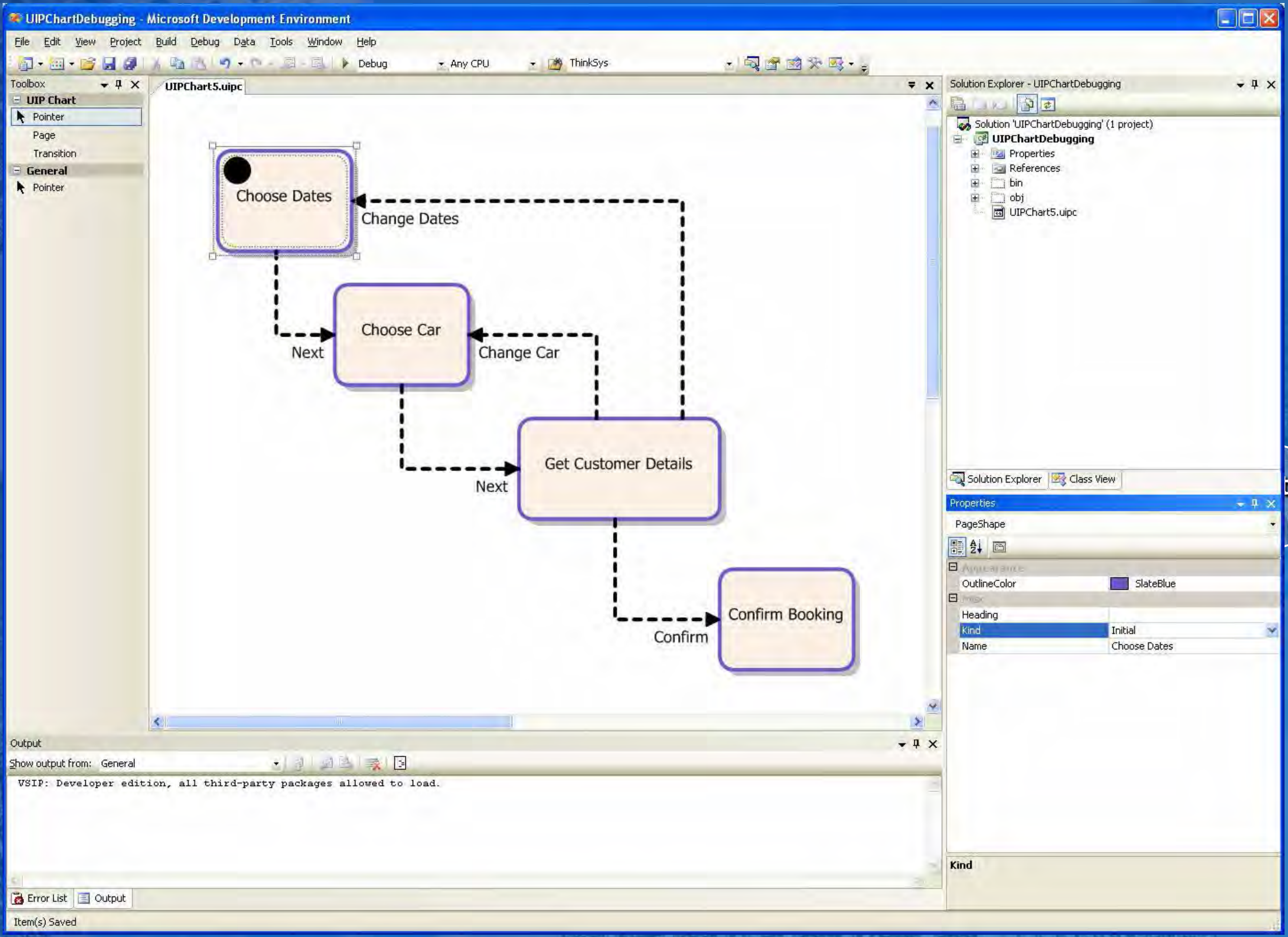


Task List - 2 tasks

!		✓	Description	File	Line
			Click here to add a new task		
!	E	☐	The constraint "ASP.NET Security" is not satisfied for host "HardenedIIS". The property "AuthMode" on ASP.NET Services & Applications "Storefront" must be set to one of "Unauthenticated, Forms" in file "web.config".	AdventureWorks.dsdgm	0
!	E	☐	The constraint "ASP.NET Security" is not satisfied for host "HardenedIIS". The property "Impersonation" on service "Storefront" must be set to "True" in file "web.config".	AdventureWorks.dsdgm	0

The constraint "ASP.NET Security" is not satisfied for host "HardenedIIS". The property "AuthMode" on ASP.NET Services & Applications "Storefront" must be set to one of "Unauthenticated, Forms" in file "web.config".

User Interface Process Design



OutlineColor	SlateBlue
Heading	
Kind	Initial
Name	Choose Dates

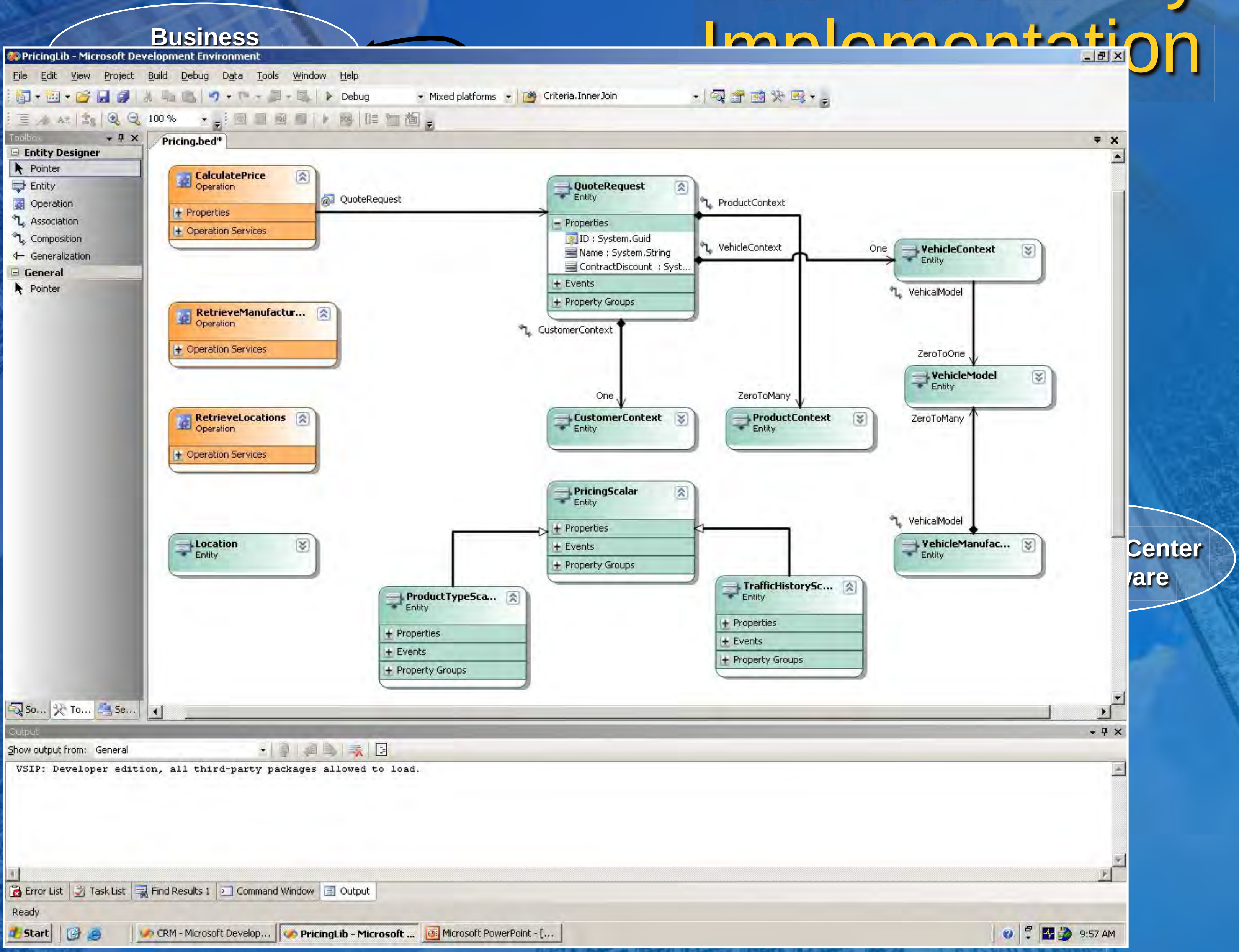
Kind

Show output from: General

```
VSIP: Developer edition, all third-party packages allowed to load.
```

ter

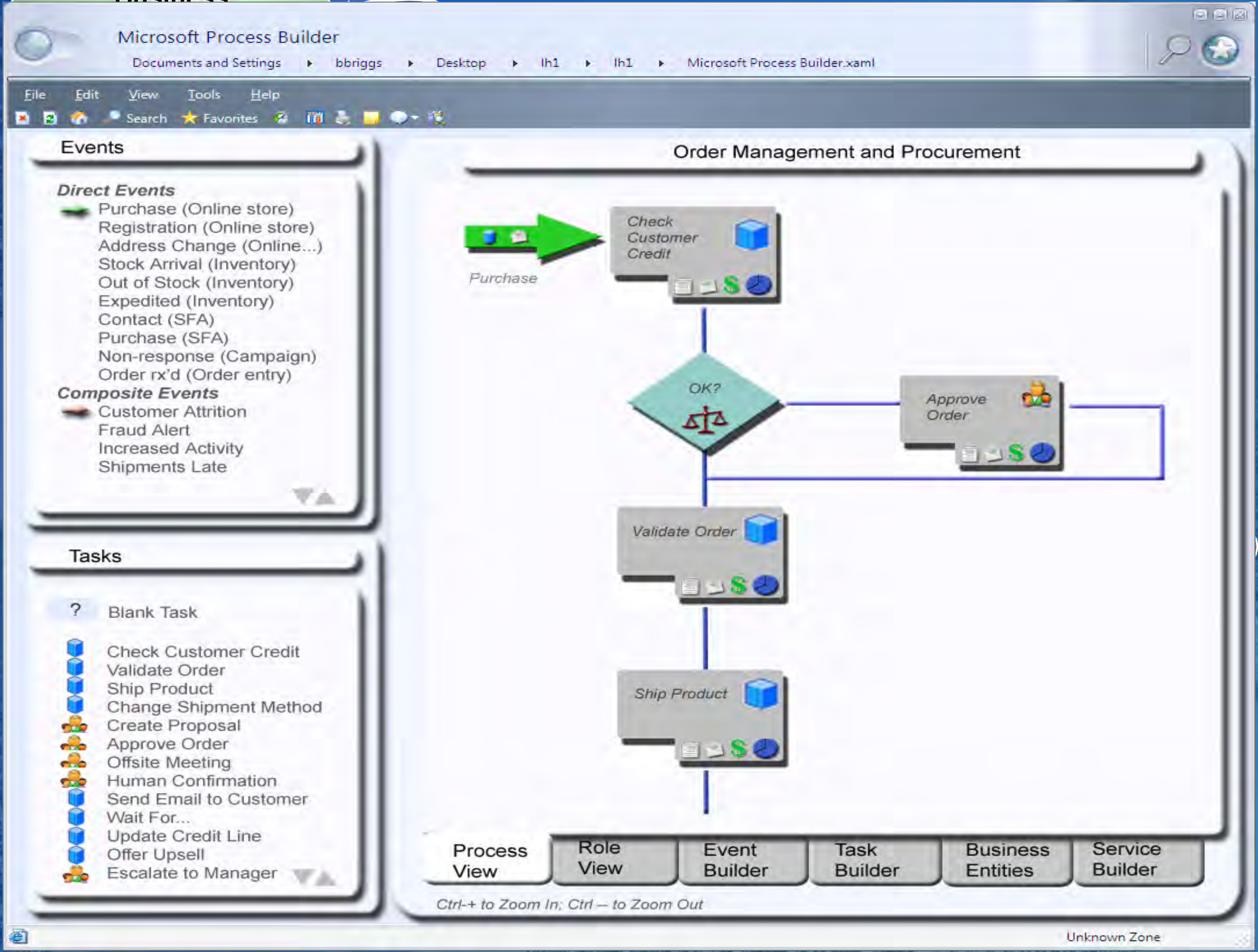
Business Entity Implementation



```
VSIP: Developer edition, all third-party packages allowed to load.
```

Business Process Modeling

Business



Microsoft Process Builder

Documents and Settings ▸ bbriggs ▸ Desktop ▸ lh1 ▸ lh1 ▸ Microsoft Process Builder.xaml

File Edit View Tools Help

Search Favorites

Events

Direct Events

- Purchase (Online store)
- Registration (Online store)
- Address Change (Online...)
- Stock Arrival (Inventory)
- Out of Stock (Inventory)
- Expedited (Inventory)
- Contact (SFA)
- Purchase (SFA)
- Non-response (Campaign)
- Order rx'd (Order entry)

Composite Events

- Customer Attrition
- Fraud Alert
- Increased Activity
- Shipments Late

Tasks

- ? Blank Task
- Check Customer Credit
- Validate Order
- Ship Product
- Change Shipment Method
- Create Proposal
- Approve Order
- Offsite Meeting
- Human Confirmation
- Send Email to Customer
- Wait For...
- Update Credit Line
- Offer Upsell
- Escalate to Manager

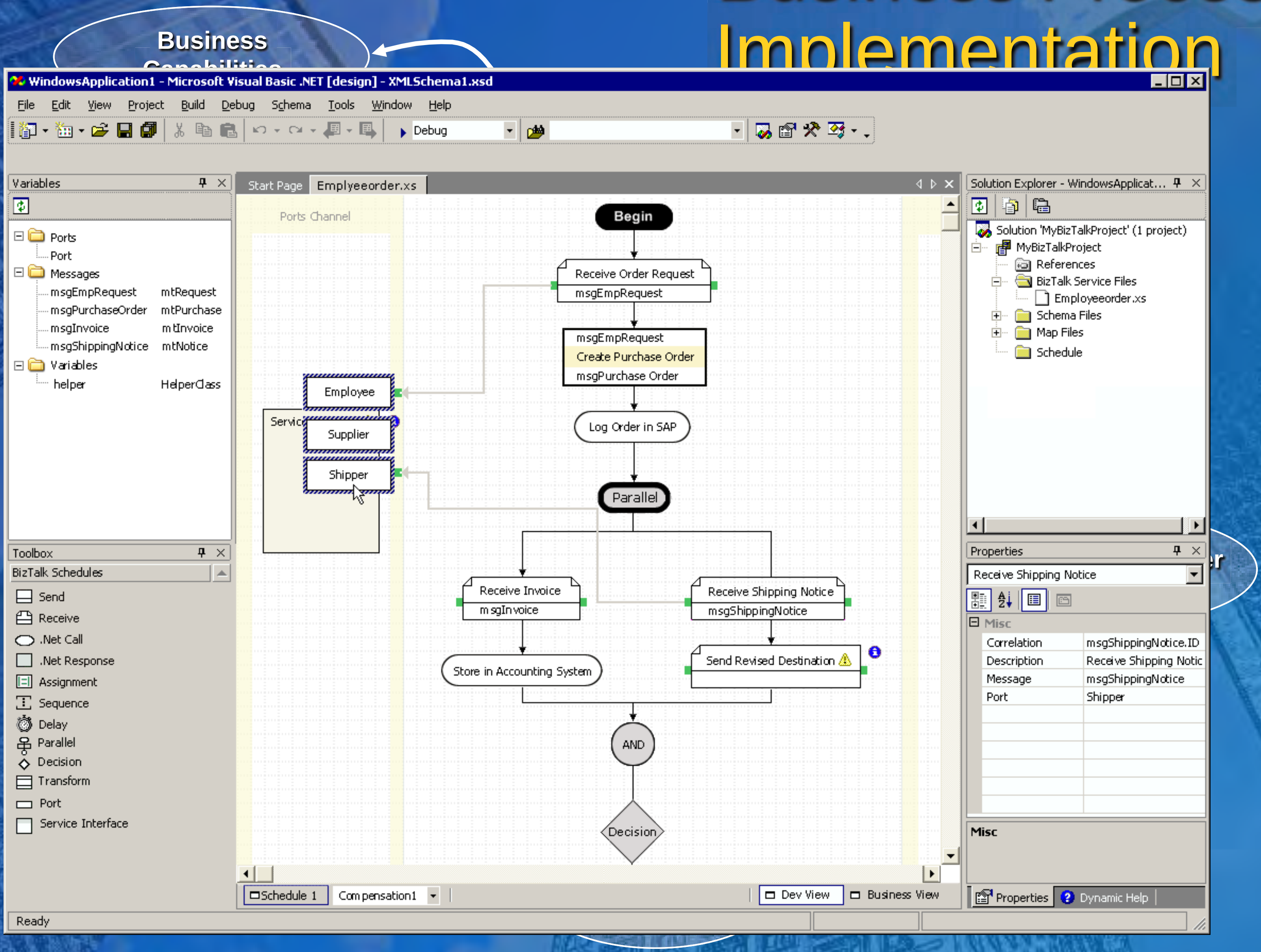
Order Management and Procurement

Process View | Role View | Event Builder | Task Builder | Business Entities | Service Builder

Ctrl-+ to Zoom In; Ctrl – to Zoom Out

Unknown Zone

Business Process Implementation

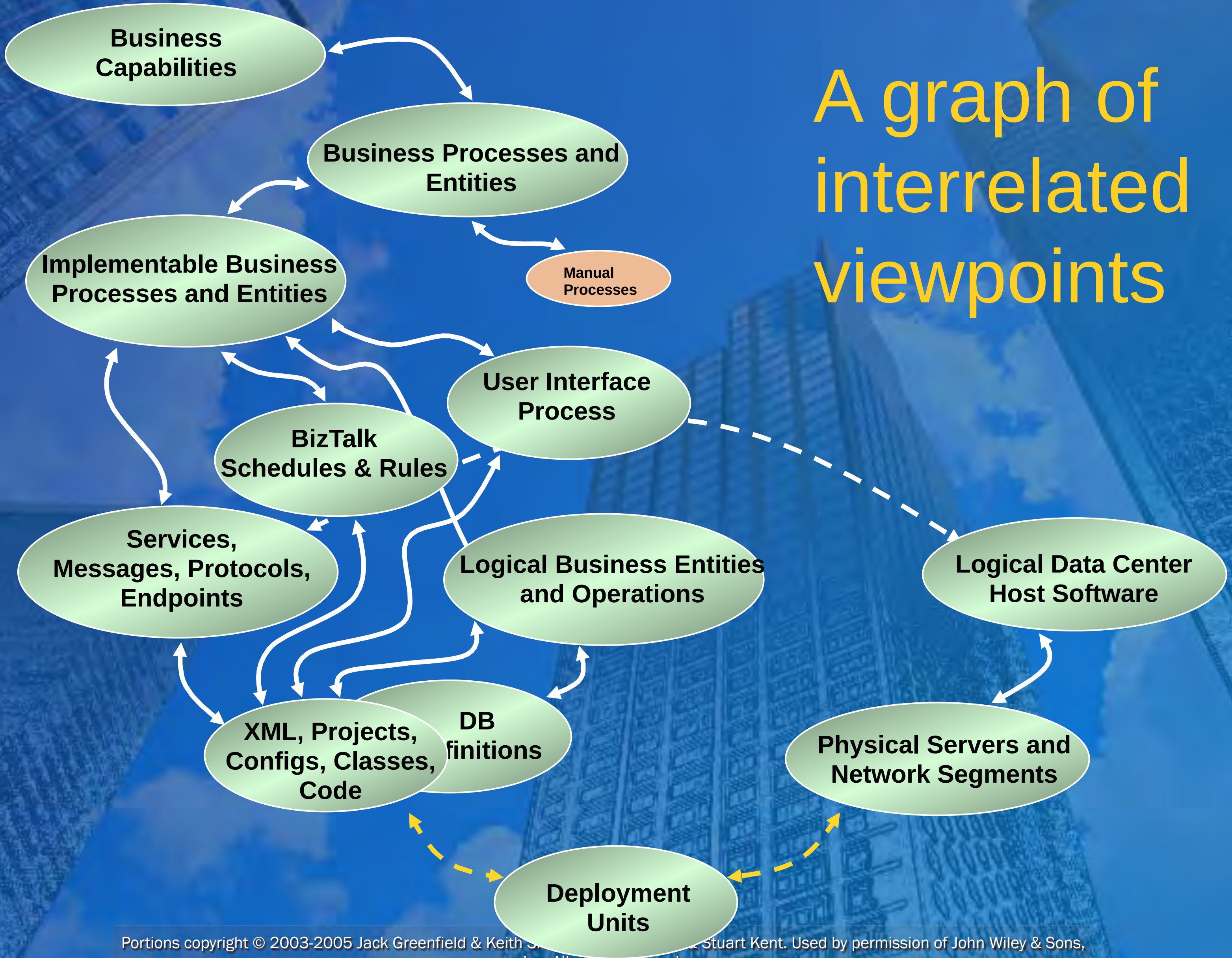


Correlation	msgShippingNotice.ID
Description	Receive Shipping Notic
Message	msgShippingNotice
Port	Shipper

Agenda

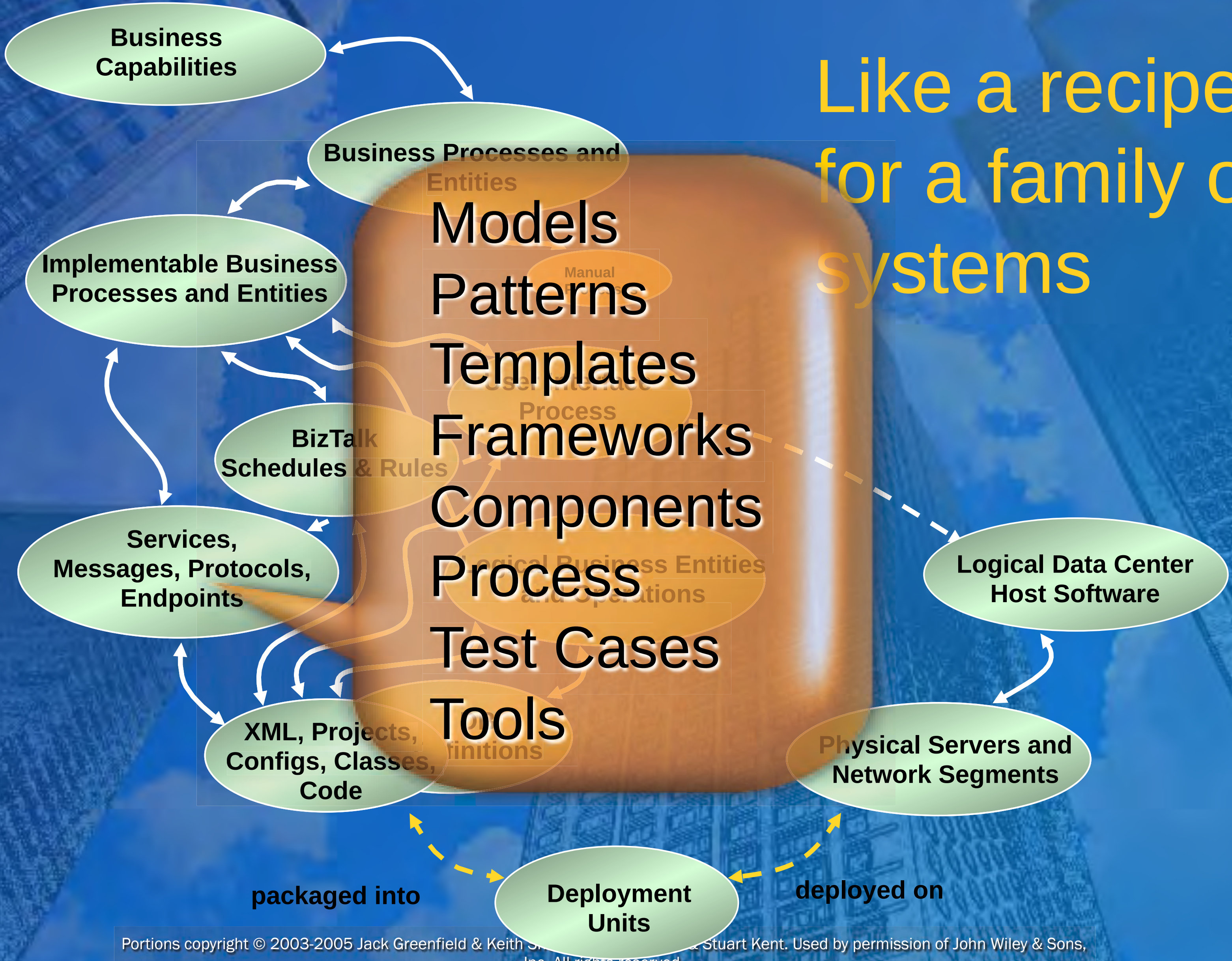
- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

A Software Factory Schema


Business Capabilities
Business Processes and Entities
Implementable Business Processes and Entities
Manual Processes
User Interface Process
BizTalk Schedules & Rules
Services, Messages, Protocols, Endpoints
Logical Business Entities and Operations
Logical Data Center Host Software
DB Definitions
XML, Projects, Configs, Classes, Code
Physical Servers and Network Segments
Deployment Units

A graph of interrelated viewpoints

A Software Factory Schema


Business Capabilities
Business Processes and Entities
Implementable Business Processes and Entities
BizTalk Schedules & Rules
Services, Messages, Protocols, Endpoints
XML, Projects, Configs, Classes, Code
Logical Data Center Host Software
Physical Servers and Network Segments
Deployment Units
packaged into
deployed on
Manual
User Interface Process
Logical Business Entities and Operations
finitions

Models
Patterns
Templates
Frameworks
Components
Process
Test Cases
Tools

Like a recipe for a family of systems

A Software Factory Template

- Implements software factory schema
  - Configures Visual Studio to build members of the class
  - Provides the necessary ingredients and tools
  - Solution template, project templates, file templates, patterns, dynamic help, work item types, workflow, check in policy, reports, groups & permissions, phase exit criteria
- Creates domain specific development environment
  - Integrates tools, process and content for the class of systems
  - Domain specific editing, compilation, debugging, refactoring, building, testing, deployment, configuration management, defect tracking, reporting

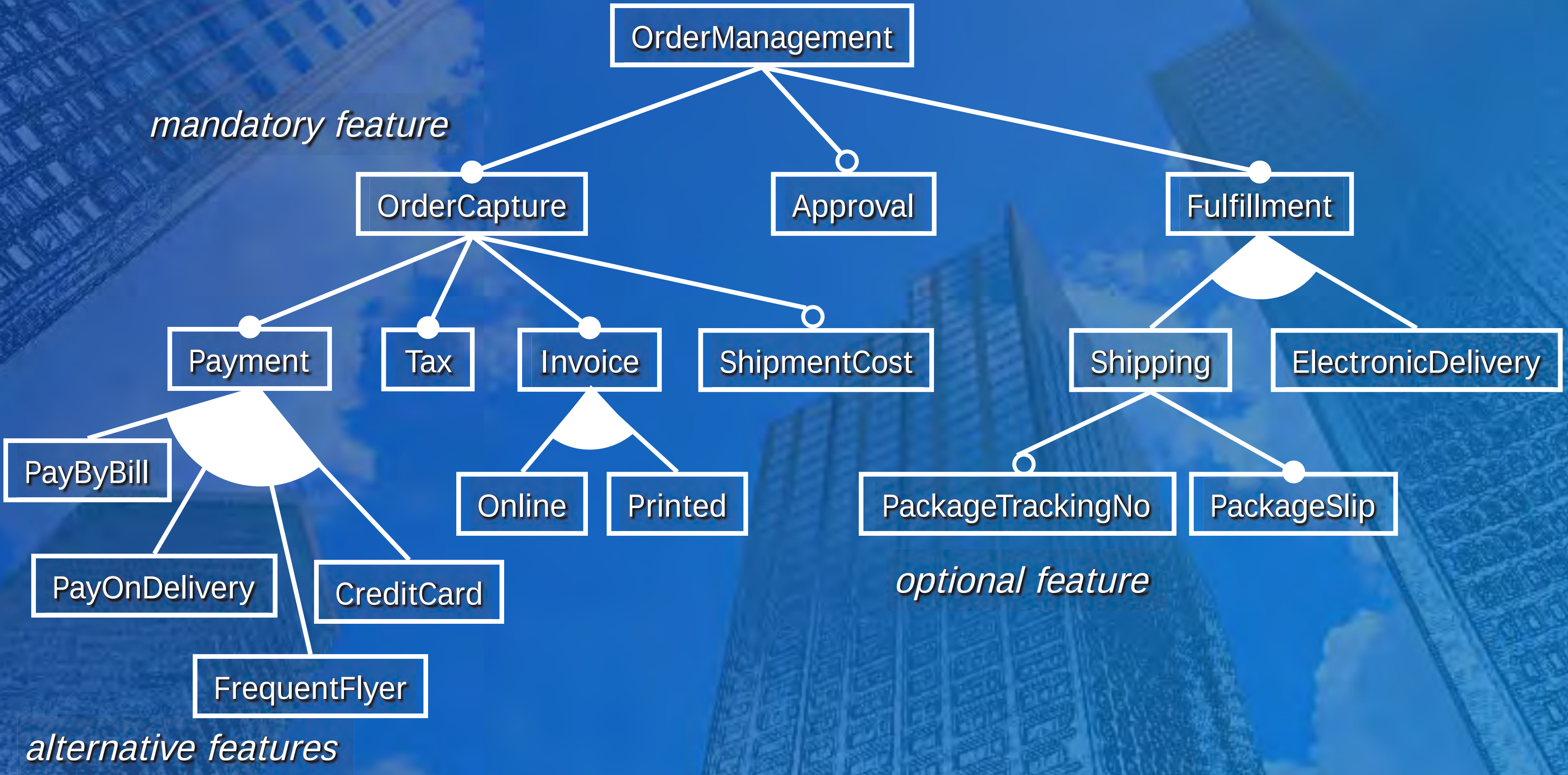
Using A Software Factory

- Progressively define system under development
  - Only unique features – common features are assumed
  - Defining system customizes schema and template
  - Adds, removes or changes viewpoints – configures tools, process and content
- Use customized factory to build system
  - Custom develop features outside factory scope
- Refactor system continually during development
  - Capture system definition in factory configuration
- Factory configuration defines delivered system
  - Provides basis for backtracking and refactoring
  - Simplifies maintenance and enhancement
  - Makes impact of changes easier to trace and understand
  - Changes propagate through factory
- Feedback to factory builders
  - Builders may be the same people as users

Building A Software Factory

- Define target class of systems
  - Use feature models to capture commonality/variability
- Build software factory schema
  - Define viewpoints and relationships for major life cycle phases
  - Requirements, Architecture, Implementation, Deployment, Testing, Operations, Maintenance
  - Define life cycle process and identify automation opportunities
- Build software factory template from schema
  - Build tools, VS templates, patterns, help, methodology template
  - Package as nesting parameterized install packages
- Refactor software factory as systems are developed
  - Based on new requirements and user feedback

Feature Modeling



Agenda

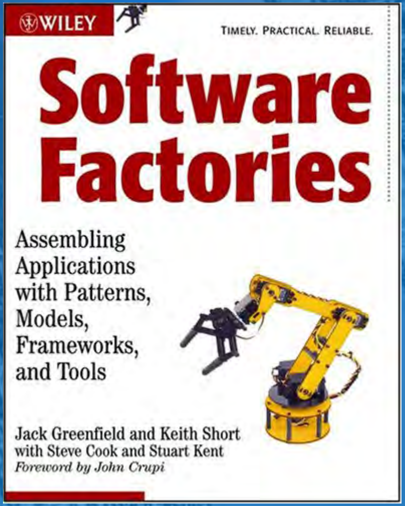
- Modeling and Methods
- Industrializing Software
- Domain Specific Languages
- Separating Concerns
- A Software Factory Schema
- Wrap Up

Why Software Factories

- Consolidate implicit business and system development knowledge into specialized tools, process, and content
- Increase productivity and predictability by better organizing and automating the development process
- Reduce cost and risk by distributing the software life cycle across networks of interdependent groups and partners

Resources

- Book
  - Software Factories by Jack Greenfield and Keith Short with Steve Cook and Stuart Kent
- Websites
  - http://msdn.microsoft.com/architecture/softwarefactories
  - http://msdn.microsoft.com/vstudio/teamsystem
  - http://lab.msdn.microsoft.com/vs2005/teamsystem/workshop
- Newsgroups
  - http://communities.microsoft.com/newsgroups/default.asp?icp=whidbey&slcid=us
- Email
  - stcook@microsoft.com
- Blog
  - http://blogs.msdn.com/stevecook


WILEY
TIMELY. PRACTICAL. RELIABLE.
Software Factories
Assembling Applications with Patterns, Models, Frameworks, and Tools
Jack Greenfield and Keith Short
with Steve Cook and Stuart Kent
Foreword by John Crupi

Microsoft®

Your potential. Our passion.™