



Архитектурное проектирование и анализ кода в

– Сергей Орлик
Microsoft

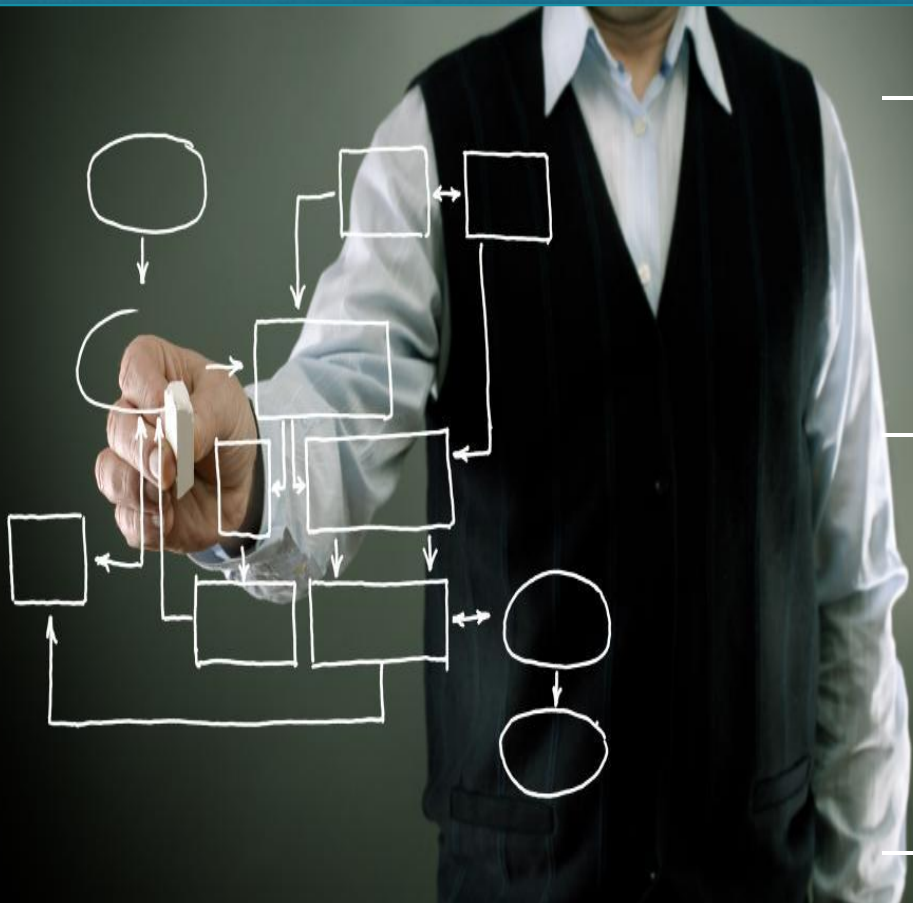
<http://blogs.msdn.com/sorlik>

Существующий код



- Созданный код может оказаться трудным для понимания спустя годы, месяцы, недели, и даже дни после его создания.
- Поддержка качества кода и архитектурный контроль – непрерывная деятельность.

НОВЫЙ КОД



- Вы уверены, что строите *правильную* систему?
- Вы понимаете *что* именно может быть нарушено когда вы добавляете новый код?
- Вы уверены, что новая реализация *соответствует* дизайну?

Движение в правильном направлении



- Понимание существующего кода
- Защита целостности архитектуры
- Облегчение коммуникаций

Программная “археология”

- Карты и исторические документы рассказывают только часть истории.
- Код является самой историей.
- Изучение “первоисточника” может оказаться затруднительным.



Динамические интерактивные инструменты визуализации + UML-

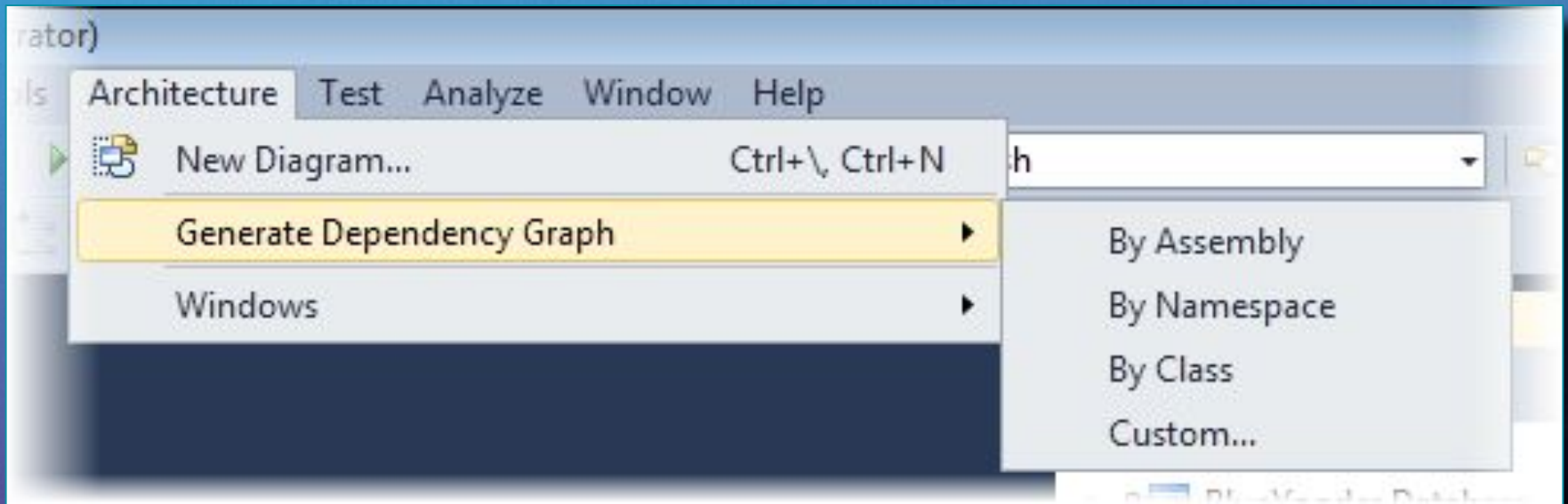
The image is a collage of several screenshots from Microsoft Visual Studio, illustrating dynamic and interactive visualization tools for UML diagrams. The screenshots are arranged in a layered, overlapping fashion.

- Top Left:** A screenshot of the 'ClassDependencies.dgml' file. It features a large yellow background with the text 'ExcelInterop' and 'ExportToExcel' in a rounded rectangle. Below it, a flow diagram shows 'ExportToExcel' leading to 'GetExcel'.
- Top Middle:** A screenshot of the 'FlightRebooking_btnStart_Click' sequence diagram. It shows a sequence of messages between 'this : FlightRebooking' and 'planner : ScheduleController'. The messages include 'SetControlState', 'SetControlState(false)', 'Create ScheduleController planner', and 'Create Lazy<ReadOnlyCollection> new Lazy<ReadOnlyCollection>'. The diagram is titled 'sd FlightRebooking_btnStart_Click'.
- Bottom Left:** A screenshot of a 'Layer Diagram' titled 'OurSystem.layerdiagram'. It shows a hierarchical structure with a purple box labeled 'Layer Interface' at the top, followed by 'Logic' (with a count of 2), 'Services' (with a count of 1), and 'Data' (with a count of 1). Arrows indicate dependencies between these layers.
- Bottom Right:** A screenshot of the 'Solution Explorer' showing a project named 'BlueYonder.MileTracker'. It lists various files and folders, including 'Layer References', 'ModelDefinition', 'AssemblyDependencies.dgml', 'ClassDependencies.dgml', 'FlightRebooking_btnStart_Click.sequencediagram', and 'OurSystem.layerdiagram'. The 'Properties' window is also visible, showing the 'Layer Model' properties.

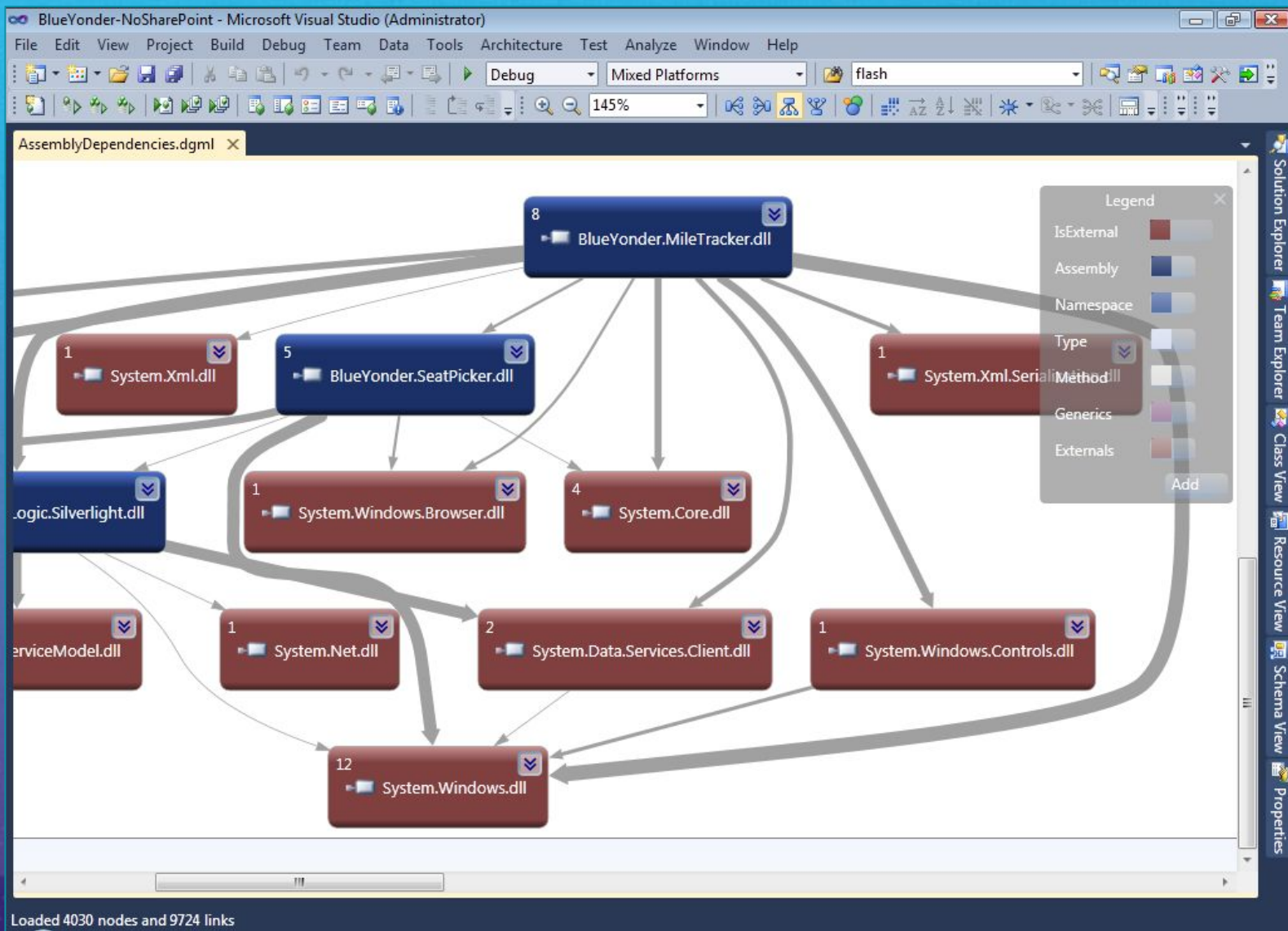
Other visible elements include the Visual Studio interface, menu bars, toolbars, and the 'Solution Explorer' showing the project structure. The overall theme is the integration of UML diagrams with development tools for dynamic visualization.

Визуализация кода

- Глубокий анализ кода с использованием графов зависимостей (dependency graphs)



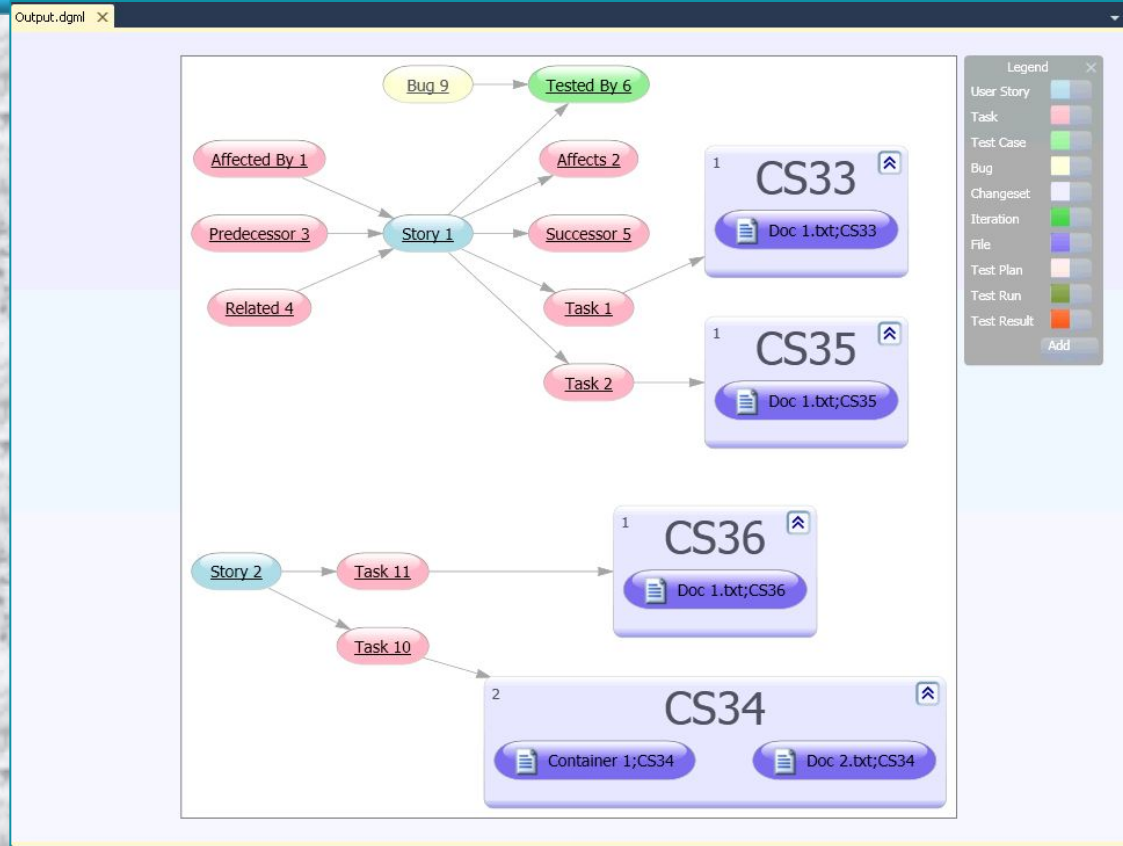
Интерактивные графы



Интерактивные графы: DGML

Directed Graph Markup Language

- Текстовое (XML) описание графа зависимостей
- Автоматическая визуализация средой
- Широкий спектр возможных применений



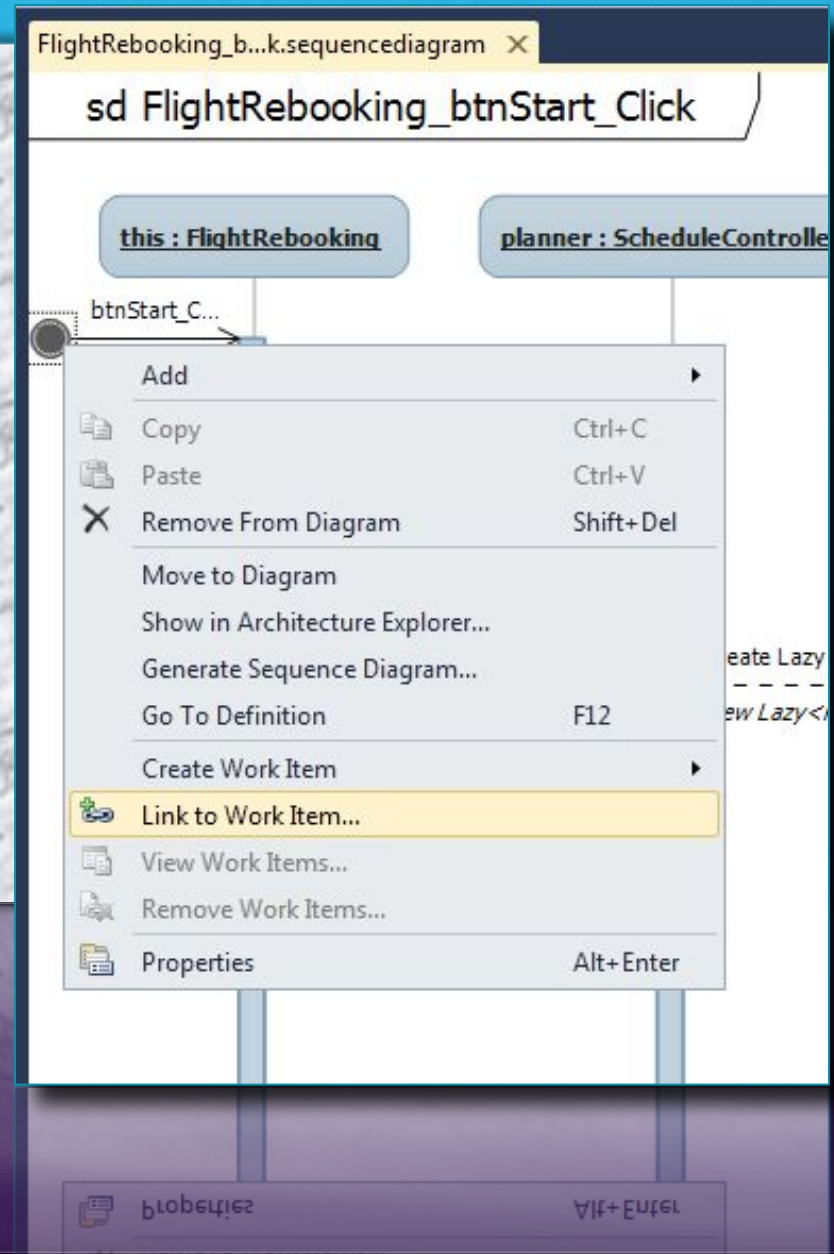
<http://visualization.codeplex.com/>

дем

Анализ кода и поддержка
UML

Unified Modeling Language (UML)

- UML-дизайнеры
 - Activity
 - Class
 - Component
 - Sequence
 - Use Case
- Modeling Project
- Интеграция с Work Items





Выразите – покажите

- Что вы хотите
- Что вы подразумеваете
- Что вам необходимо

Облегчение коммуникаций

- Повышение вероятности успеха
 - Architecture Layer диаграммы
 - поддержка UML
- “начинать с модели” или “начинать с кода”



Управление “дрейфом” кода



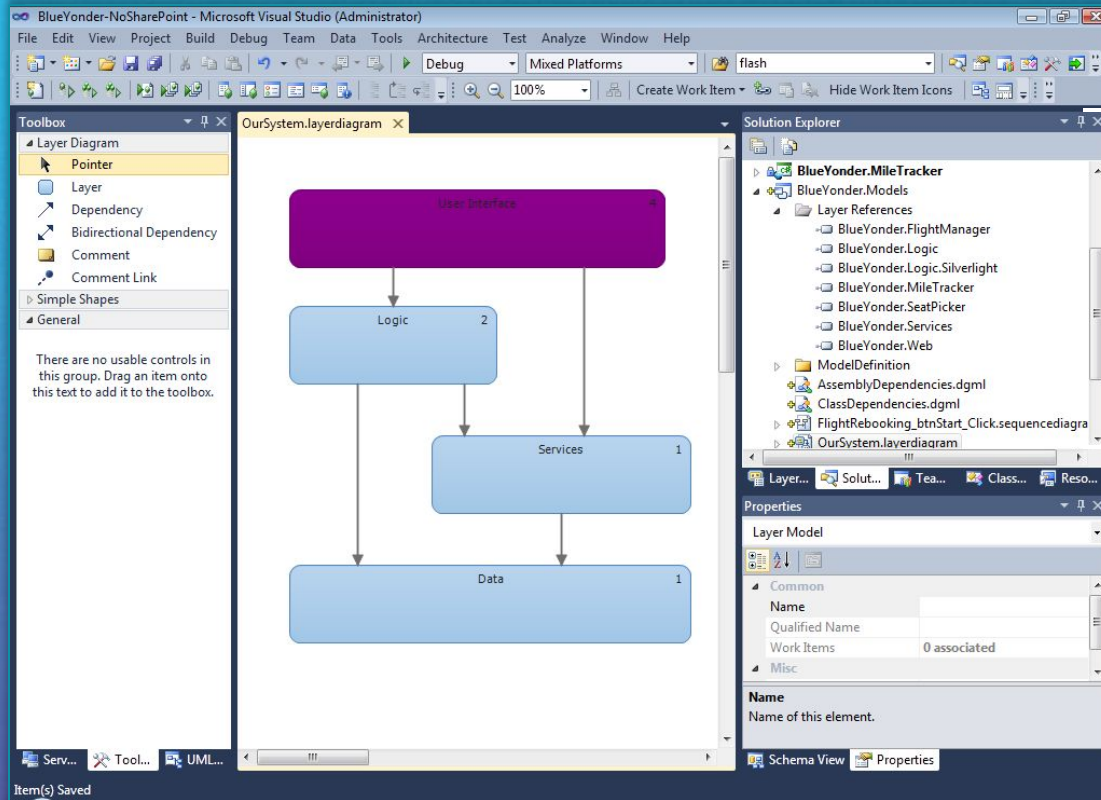
- Код может “дрейфовать” в сторону от первоначального дизайна и приводить к
 - Незапланированным зависимостям
 - Отклонениям в эволюции кода

Результат == план?



- Планирование часто проводится с использованием диаграмм и спецификаций
- Прошло ли ваше решение проверку на соответствие архитектуре ?
- Хотите ли вы чтобы ваши диаграммы были бОльшим, чем просто “картинками”?

Диаграммы уровней (Layer diagram)



Визуализация архитектурной концепции.

Проверка реализации на соответствие архитектуре.

Диаграммы уровней (Layer diagram)

– проверка кода на соответствие

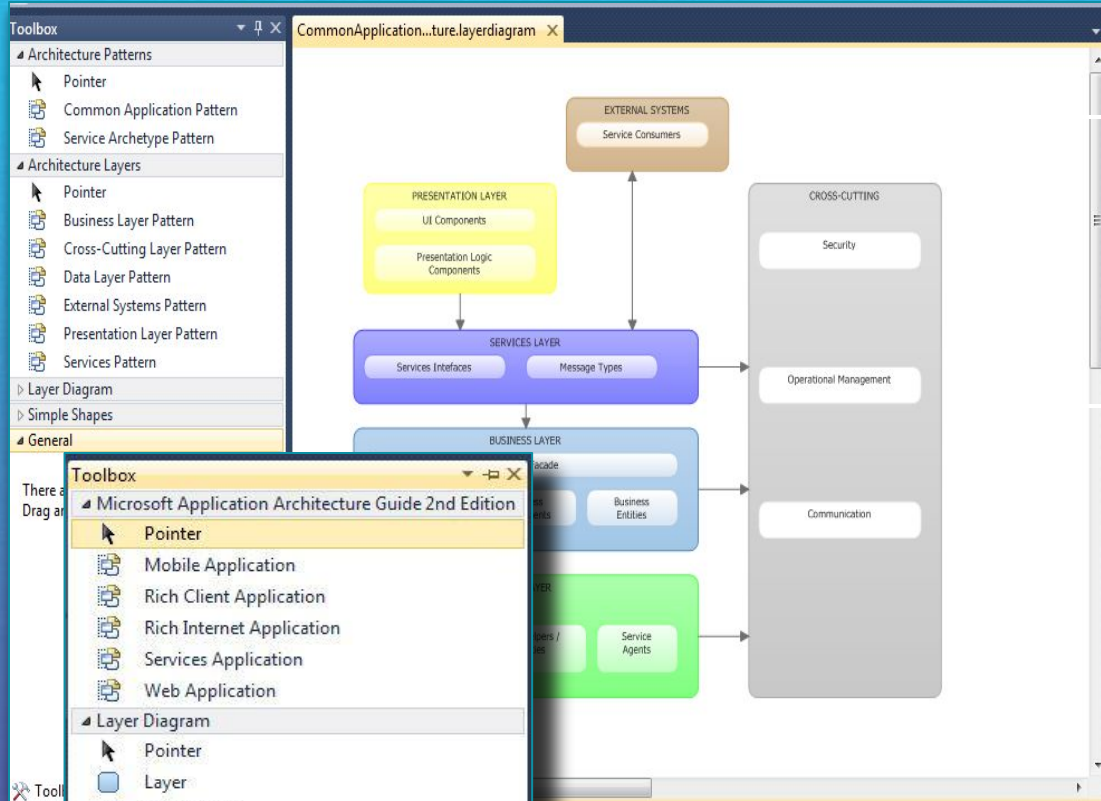
- Интеграция с подсистемой автоматической сборки Team Foundation Build

The screenshot displays the Visual Studio interface for configuring Layer Diagrams. The 'Process' tab is selected, showing build process parameters. A red circle highlights the 'MSBuild Arguments' parameter, which is set to '/p:ValidateArchitecture=true'.

Build process parameters:

Automated Tests	Run tests in assemblies matching <code>***test*.dll</code> using settings
Build Number Format	<code>\$(BuildDefinitionName)_\$(Date:yyyyMMdd)\$(Rev..r)</code>
Clean Workspace	All
Logging Verbosity	Normal
Perform Code Analysis	AsConfigured
Source And Symbol Server Settings	Index Sources
3. Advanced	
Agent Settings	Use agent where Name=* and Tags is empty; Max Wait Time:
Analyze Test Impact	True
Associate Changesets and Work Items	True
Copy Outputs to Drop Folder	True
Create Work Item on Failure	True
Disable Tests	False
Get Version	
Label Sources	True
MSBuild Arguments	/p:ValidateArchitecture=true
MSBuild Platform	Auto
Private Drop Location	

Применение архитектурных шаблонов



Применение готовых шаблонов в Layer диаграммах.

Создание и повторное использование собственных шаблонов всей командой.



<http://archlayerspatterns.codeplex.com/>
<http://visualstudiogallery.msdn.microsoft.com>






ДЕМ

Построение архитектуры
и обеспечение ее
целостности

VS 2010 Architecture Tools: партнёры*



Партнёр	Доступность* *	Решение
	★	Micro Focus Visual Cobol 2010 использует VS 2010 Architecture Tools для графического представления связей в программах на COBOL
	★	Pre-Emptive использует VS 2010 Architecture Tools для графического отображения связей между покрытием тестами и отслеживаемым использованием кода пользователями.
	★	NDepend –инструмент для VS 2010, облегчающий управление сложным .NET кодом с помощью широкого спектра стандартов и правил, улучшающих дизайн, расширяющих возможности рефакторинга и анализа кода.

* Полный список доступен в Microsoft Visual Studio Gallery

** Планируется одновременно с поставкой VS 2010 до конца мая 2010



VS 2010 Architecture Tools:

резюме

- Помогают понять существующий код
- Облегчают управление кодом сложных проектов
- Уменьшают риски при внесении изменений в код
- Помогают фокусироваться на том коде, который несёт полезный функционал
- Четко и однозначно отражают архитектурные решения

Клуб Архитекторов

- Моделирование и проектирование
- Корпоративная архитектура
- Прикладная инфраструктура
- Облака "публичные" и "приватные"
- Enterprise 2.0
- Перспективные технологии и фреймворки
- Стандарты и практики

Если вы архитектор,
принимающий
технологические решения

<http://blogs.msdn.com/sorlik> | ru-arc@microsoft.com

Microsoft®

Your potential. Our passion.™

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

