

Методика тестирования WPF приложений

.NET Architecture Day
«Разработка пользовательского
интерфейса — современные подходы»

Баркан Георгий

gbarkan@microsoft.com

Эксперт по технологиям разработки ПО

Microsoft Россия

Зачем?

Тестирование — важно!

Тестирование UI — самое сложное

Целый арсенал методов — нужно
построить стратегию

Что это?

Оценка параметров

— соответствуют ли ожиданиям?

Параметры

— что тестировать

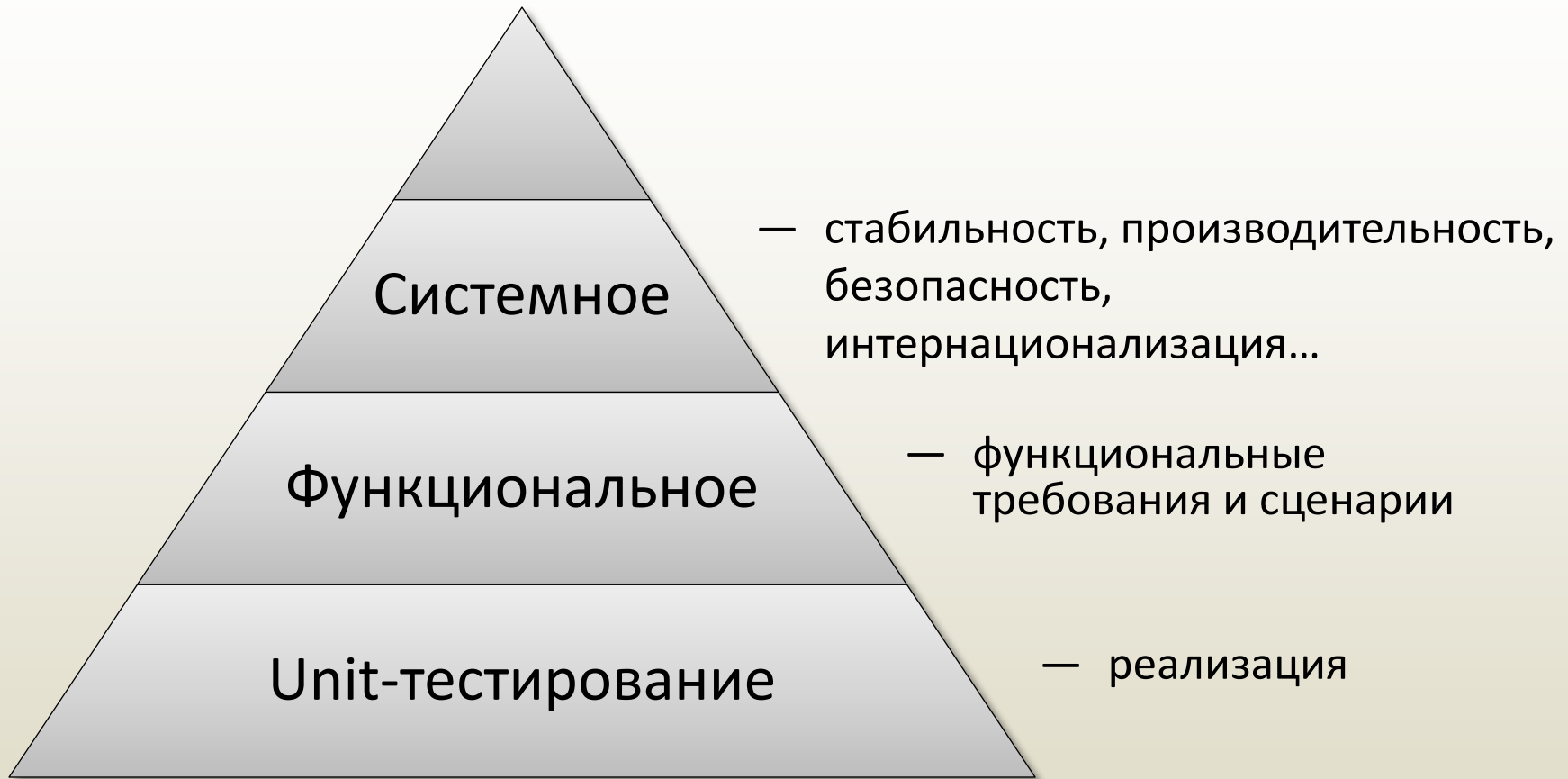
Методы оценки

— как тестировать

Ожидания

— требования

Что тестировать?



А также:

требования, сценарии, тесты, документация...

Unit-тестирование

Проверка качества реализации блока

Выполняется вместе с разработкой блока

— обычно делает сам разработчик блока

Принцип
белого ящика

— статический анализ кода

— анализ покрытия кода тестами

Unit-тестирование

применительно к UI

Выделить логику
и модель данных

Model–View–Controller
для WPF — Model–View–ViewModel

Тестируется ViewModel (Controller)
и опосредовано Model

Разница в
требованиях — реализационные (design), а не функциональные

Функциональное тестирование

Функциональные требования — спецификация

Качество самих требований: полнота и
непротиворечивость

Принцип
черного ящика!

Сценарии использования

высокого качества!

— не должно быть слишком много

- История о пользователе и мотивации на использование продукта
- Ключевые требования
- Правдивая история
- Экспертный уровень для сложных продуктов, либо интуитивный для простых
- Результаты тестирования сценария должны проверяться легко и однозначно

Как тестировать?

Вручную

— все равно нужно!

Автоматически

- освобождает от рутины
- дает быструю оценку качества сборки
- инвестиции в разработку и поддержку

насколько критична функциональность?

насколько часто ломается?

насколько быстро нужно отреагировать на проблему?

легко поддается автоматизации?

Нужна стратегия

Пример:	<i>автоматически</i> — срабатывание точек останова в отладчике, корректные обработчики событий в UI
Visual Studio	<i>вручную</i> — проверка Find and Replace, выделение цветом в редакторе

О ручном тестировании

Глоссарий!

Конкретный способ важен —
описать процедуру как можно точнее

*«добавить кнопку в верхнюю
часть окна дважды щелкнув
по ней на панели
инструментов»*

Конкретный способ неважен —
описать процедуру максимально общо

*«перенести значение в конец
документа», вместо
«выделить значение, нажать
Ctrl+X, Ctrl+End, Ctrl+V»*

Последовательности шагов,
которые использовать повторно

Описывайте ожидаемый эффект как можно точнее

Одна фича в одном тесте

Выделяйте подготовительные шаги для групп тестов

Автоматические тесты

А — непосредственно на UI

Б — автоматизация UI

—Automation UI,
TestAPI

В — низкоуровневые API

—TestAPI

Г — драйвер

Д — механический робот

Automation UI



DockPattern, ExpandCollapsePattern,
GridPattern, GridItemPattern,
InvokePattern, MultipleViewPattern,
RangeValuePattern, ScrollPattern,
ScrollItemPattern, SelectionPattern...

По шагам:

1. Навигация по элементам

RootElement

далее поиск (FindFirst, FindAll)
обход дерева (TreeWalker)
по положению фокуса
по экранным координатам

AutomationId,
control name,
control type,
PropertyCondition...

2. Получаем шаблон управления

3. Оцениваем состояние элемента управления

4. Выполняем действия

все манипуляции из отдельного потока,
в MTA нужно создать отдельный STA поток

Демо

Автоматизация тестирования пользовательского интерфейса в приложениях WPF

Джеймс Мак-Кэффри

<http://msdn.microsoft.com/ru-ru/magazine/dd483216.aspx>

TestAPI

Библиотека

низкоуровневая работа с клавиатурой и мышью



Mouse



Keyboard

.NET код, но работает через Win32

В случае наличия *сложной графики*
может потребоваться

Визуальная верификация

Важно не переусердствовать!

- очень затратная реализация
- только если UI достаточно стабильный
- может использоваться для регрессионного тестирования

По шагам:

Поддерживается
библиотекой TestAPI

1. Снимается фактическое изображение окна
2. Строится образцовое изображение
3. Строится разностное изображение
4. Проверяется разностное изображение
 - точное попиксельное совпадение — мало реально
 - допуск
 - проверка по распределению цветов (гистограмма)
 - наложение масок

либо сохраненный образец,
либо генерируется программно

Системное тестирование

Стабильность

Стресс-тест

Не скрывать исключения

Минимизировать отладочную
и тестовую печать

Имеет смысл измерять
производительность

— маскирует возможные
«гонки»

Не нужно смешивать с
функциональным тестированием

— не проверять
корректность
выполнения

Стресс-тест

Комбинация различных сценариев

стрессовое действие — атомарное, но часто используемое

взаимодействие с элементами управления,
либо случайные пользовательский ввод в окно

бесконечная последовательность
случайно выбираемых действий

— случайный генератор
запускать в
повторяемом режиме

одновременно выполнять
несколько действий

собираемой информации должно быть достаточно
для расследования катастрофического сбоя

Southridge Realty

Southridge Realty | Client Profile for Henry Johnson

Search Criteria Search Results

Maximum Price Minimum Price

☒ Title ☒ Photo ☒ Rating
☒ Price ☒ Year ☒ Address
☒ Bedrooms ☒ Sqr. Footage ☒ Date Listed
☒ Bathrooms ☒ Appointment ☒ Date Updated

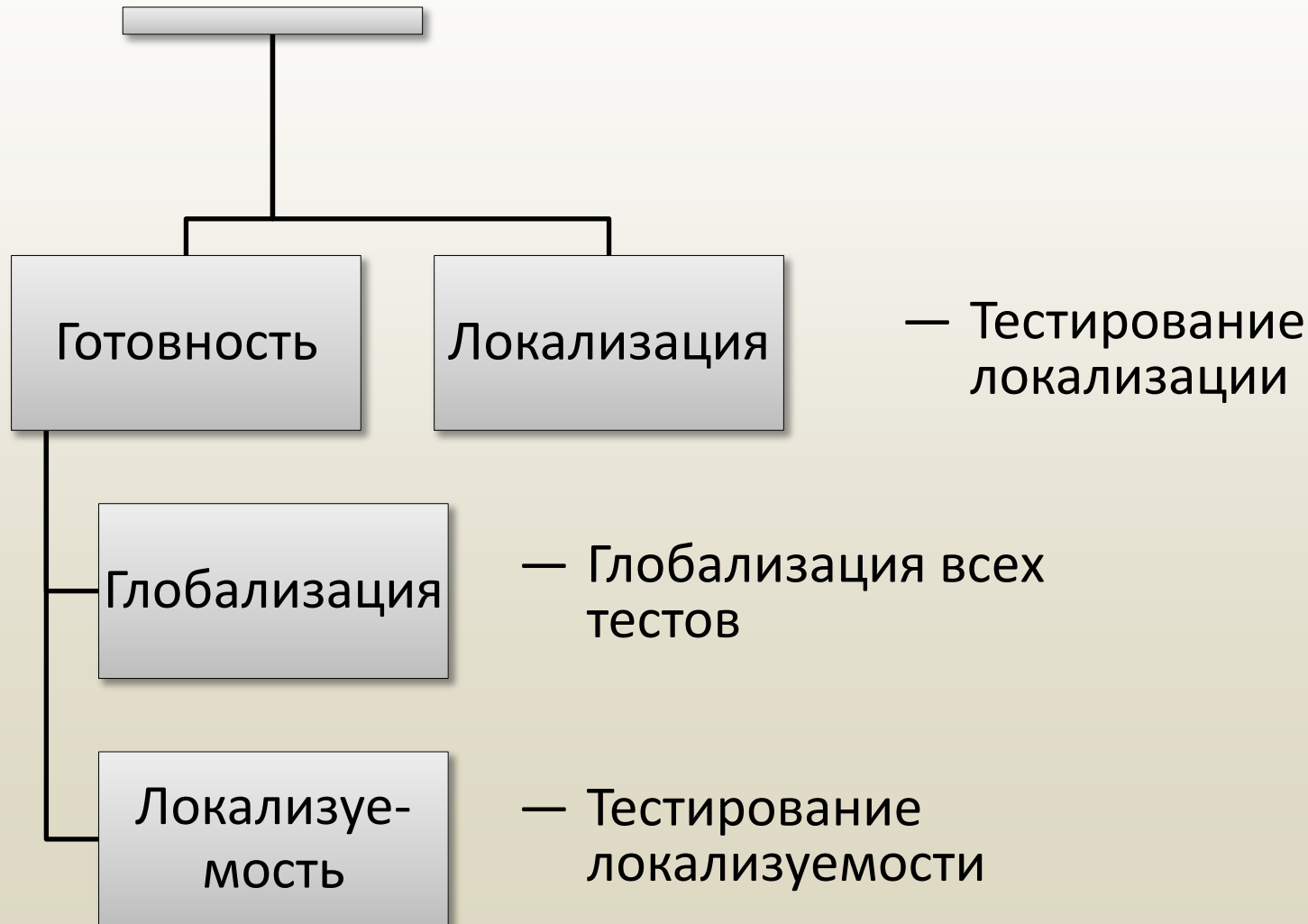
Bedrooms Bathrooms Year Built New Construction

High Growth Near City Move Details

	Title	Price	Bedrooms	Bathrooms	Year	Sqr. Footage	Appointment	Rating	Address	Date L
	Center City Seattle Apartment	\$425,000.00	1	1	1995	1400	9/20/2008	★ ★ ★ ★ ★	678 Spring Street	09/01/
	Seattle Area Condo	\$475,000.00	2	2	1984	2780	Show Calendar	★ ★ ★ ★ ★	2345 California Ave.	09/25/
	Top floor view condo w/wonderful floor plan in the heart of the desirable Admiral District. Stroll to Admiral Theatre, Metropolitan Mkt. and trendy... More...									
	Downtown Bellevue Condo	\$515,000.00	2	2	2005	1600	Show Calendar	★ ★ ★ ★ ★	321 Bellevue Way	10/01/
	Unique Redmond Home	\$550,000.00	4	2.5	1976	2850	11/8/2008	★ ★ ★ ★ ★	567 Redmond Way	09/13/
	Classic Ballard Home	\$465,000.00	3	2	1945	2560	Show Calendar	★ ★ ★ ★ ★	1234 NW 57th St	08/03/
	Upscale Queen Anne Apartment	\$500,000.00	3	2	1991	2100	10/31/2008	★ ★ ★ ★ ★	765 Denny Way	02/02/
	Waterfront Kirkland Condo	\$535,000.00	2	1.5	2002	1850	Show Calendar	★ ★ ★ ★ ★	234 Lake Ave	01/10/
	Center City Seattle Apartment	\$425,000.00	1	1	1995	1400	Show Calendar	★ ★ ★ ★ ★	678 Spring Street	09/01/
	Seattle Area Condo	\$475,000.00	2	2	1984	2780	10/17/2008	★ ★ ★ ★ ★	2345 California Ave.	09/25/
	Downtown Bellevue Condo	\$515,000.00	2	2	2005	1600	Show Calendar	★ ★ ★ ★ ★	321 Bellevue Way	10/01/
	Unique Redmond Home	\$550,000.00	4	2.5	1976	2850	11/15/2008	★ ★ ★ ★ ★	567 Redmond Way	09/13/
	Classic Ballard Home	\$465,000.00	3	2	1945	2560	Show Calendar	★ ★ ★ ★ ★	1234 NW 57th St	08/03/
	Upscale Queen Anne Apartment	\$500,000.00	3	2	1991	2100	Show Calendar	★ ★ ★ ★ ★	765 Denny Way	02/02/
	Waterfront Kirkland Condo	\$535,000.00	2	1.5	2002	1850	9/20/2008	★ ★ ★ ★ ★	234 Lake Ave	01/10/
	Center City Seattle Apartment	\$425,000.00	1	1	1995	1400	Show Calendar	★ ★ ★ ★ ★	678 Spring Street	09/01/

Системное тестирование

Интернационализация



Интернационализация

Глобализация:

UltimateResourceFallback
в AssemblyInfo
InvariantCulture для
внутренних операция
xml:lang="..."

Тестируем *локализуемость*:

псевдолокализация
FlowDirection на
корневом элементе UI

Тестовые параметры для *локализации*:

Windows XP / Windows Vista / Windows 7

Локализованная версия ОС

Английская версия ОС, но с установленным MUI

Региональные настройки для пользователя (CurrentCulture)

Инструменты

UI Automation

System.Windows.Automation (.NET 3.x)

<http://msdn.microsoft.com/ru-ru/library/ms753388.aspx>

TestAPI

Делает команда WPF

<http://codeplex.com/testapi>

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

UI Spy

Показывает информацию по автоматизации UI,
идет с последним SDK

<http://msdn.microsoft.com/en-us/library/ms727247.aspx>

UIAutoCmd

Консольный аналог UI Spy

WPF Application Quality Guide

Методика тестирования + оптимизация + отладка

Десятки полезнейших ссылок

<http://windowsclient.net/wpf/white-papers/wpf-app-quality-guide.aspx>

http://blogs.msdn.com/ivo_manolov

Методика

Model–View–ViewModel

WPF Apps With The Model–View–ViewModel Design Pattern:

<http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>

Exploring a Model–View–ViewModel Application:

<http://www.codeproject.com/KB/WPF/ExploringWPFMVVM.aspx>

Model–View–ViewModel in WPF:

<http://www.codeproject.com/KB/WPF/ModelViewViewModelWithWPF.aspx>

Самое главное

Что тестировать?

1. Артефакты
2. Блочное
3. Функциональное
4. Системное

Как тестировать?

1. Стратегия (вручную, автоматически)
2. VSTS — блочное тестирование
3. Автоматизация UI (UI Automation + TestAPI)
4. Если нужно —
визуальная верификация (TestApi)
5. WPF Application Quality Guide

Спасибо!

isvrus@microsoft.com

http://blogs.technet.com/isv_team