

Независимая
научно-практическая конференция
«Разработка ПО 2011»

31 октября - 3 ноября, Москва



Опыт автоматизации тестирования в Agile проекте

Зуева Татьяна



Содержание

- Особенности проекта
- Наш подход
 - Сравнение средств для автоматизации тестирования UI
 - Создание Automation framework
 - Процессные практики
- Итоги



Особенности проекта

- Клиент-серверное WPF приложение
- Интеграция с другими приложениями
- Свой GUI framework
- Scrum процесс



Тестирование в проекте

Автоматизированное
тестирование на UI
(тестировщики +
разработчики)

- Click to edit Master text styles

- Second level

- Third level

- Fourth level

- Fifth level

Ручное
тестирование
(тестировщики)

Модульные
тесты на код
(разработчики)

Проблемы

- Нет времени
- “Что-то не работает...” (помощь разработчиков)
- Тестируемое приложение слишком часто меняется
- Поддержка старых тестов



Выбор есть всегда...



	Quick Test Professional	Coded UI Tests	White
Поддерживаемые технологии	Win32, WPF, WinForm, SWT, Silverlight, Web etc.	Win32, WPF, WinForm, SWT, Silverlight, etc.	Win32, WPF, WinForm, SWT, Silverlight 2.0
Скорость работы скриптов	Средняя/ Медленная	Достаточно быстрая	Достаточно быстрая
Среда разработки	QTP среда	VS 2010	Любая (включая VS)
Запуск тестов	Из QTP, Из HP QC	Как юнит-тесты (nunit, reSharper, и др.) Из TFS	Как юнит-тесты (nunit, reSharper, и др.)
Язык программирования	Vbscript	C#	C#
Лицензия	Дорого	Дорого	Бесплатно, открытый код
Record&Play	+	+	-

Automation framework: цели

- Просто и удобно писать тестовые сценарии
- Расширяемость и поддерживаемость
- Регулярный запуск тестов и интеграция с CI
- Запуск тестов на произвольном ПК
- Читаемость кода тестов

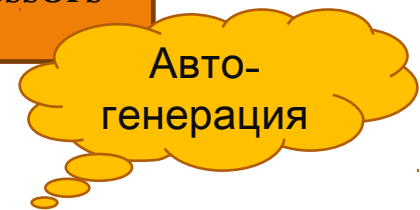
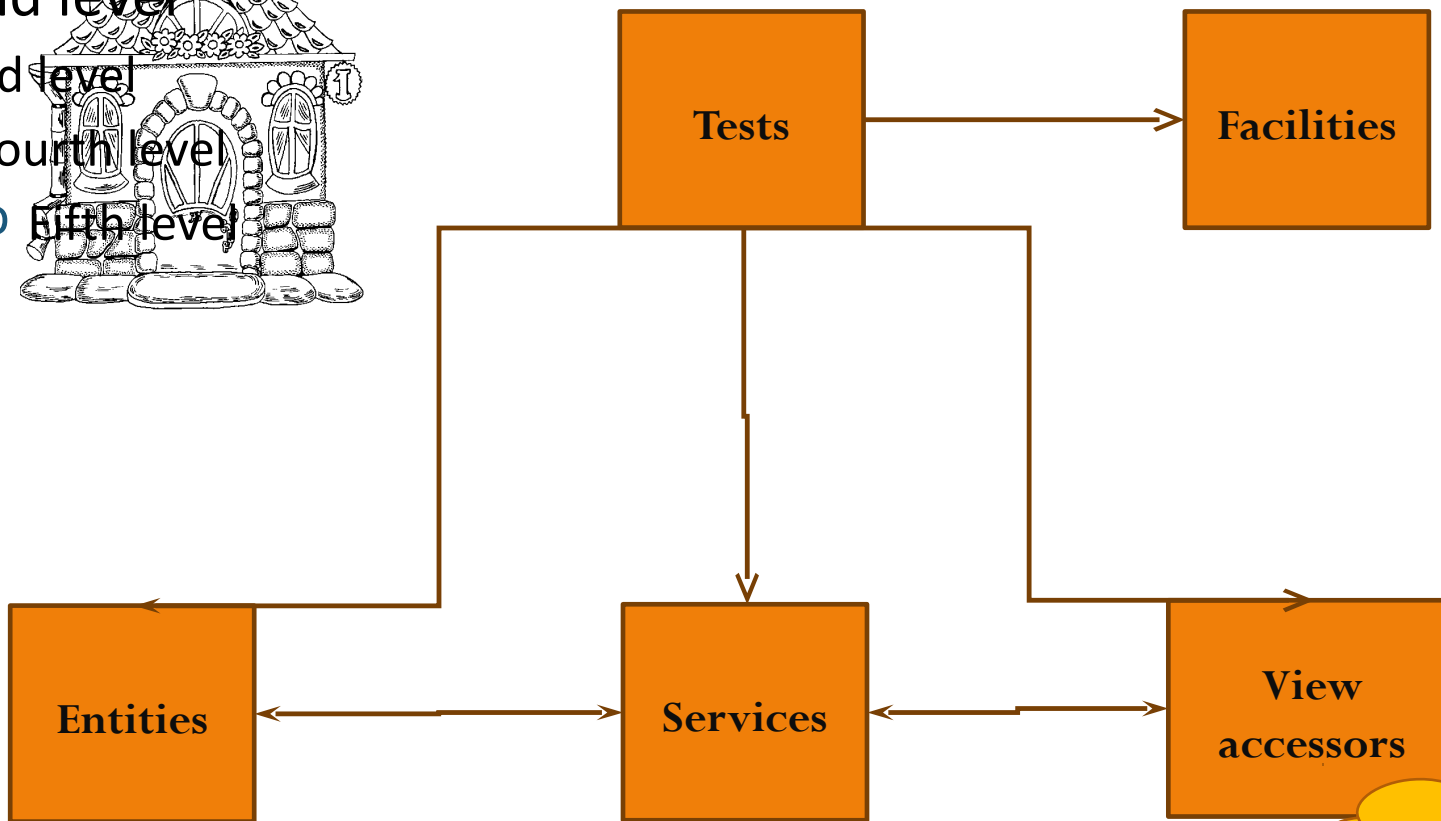
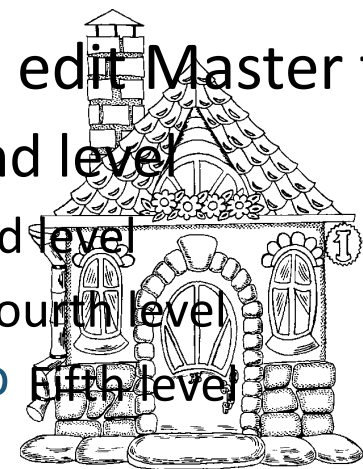


Automation framework: структура

click to edit Master text styles

Second level

- Third level
- Fourth level
- Fifth level



Распределение обязанностей

- Разработчики – доработка Automation Framework и написание Services
- Тестировщики – написание Services и построение тестовых сценариев



Мотивация

- Для разработчиков
 - В новом Automation Framework приятно и удобно работать
 - Быстрое получение результатов тестирования
- Для тестировщиков
 - Повышение квалификации
 - Меньше рутинной работы
- Для всех
 - Лучшее понимание процесса тестирования и разработки



Планирование

- Вместе с текущей итерацией
- Включение автоматизации в критерий готовности
- Регулярное отслеживание статуса
- Участие разработчиков и тестировщиков



How to?

- Процесс
 - Взаимодействие тестировщиков и разработчиков
 - Планирование итерации

Выбрать правильную технологию

Технологии

ЭНЙ Automation framework



И в результате...

- 1,5 спринта на Automation Framework
- 40% регрессионных тестов за 4 спринта
- Комфортная разработка
- Регулярное получение результатов
- Планирование и мониторинг

- Проблемы со временем
- Периодические проблемы со стабильностью



И в результате...

- Повышение качества продукта
- Личное развитие членов команды
- Улучшение атмосферы в команде



Независимая
научно-практическая конференция
«Разработка ПО 2011»

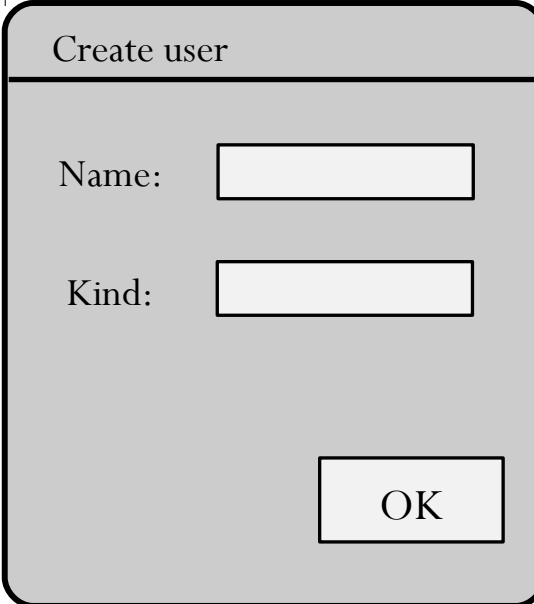
31 октября - 3 ноября, Москва



Спасибо за внимание

Дополнение: View Accessors

```
public class CreateUserAccessor {  
    string Title {  
        get { return "Create user"; }  
    }  
  
    Button OKButton {  
        get  
        {  
            return Window.TryGet<Button>("OK");  
        }  
    }  
  
    TextBox NameEditTextBox {  
        get  
        {  
            return Window.TryGet<TextBox>("NameEdit");  
        }  
    }  
    . . .  
}
```



Create user

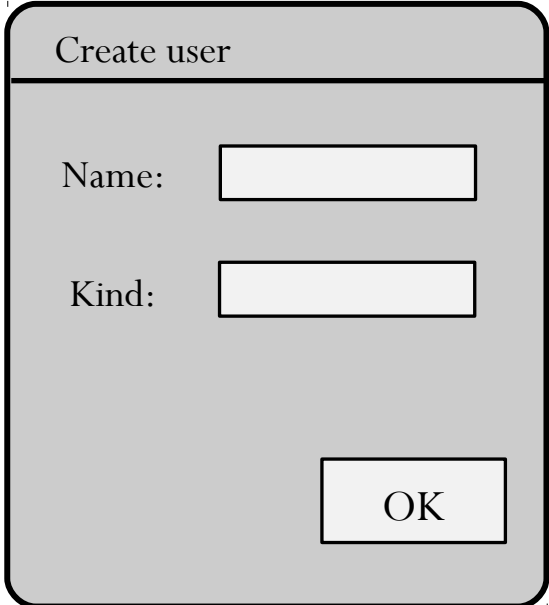
Name:

Kind:

OK

Дополнение: Services

```
public class UserManagement{  
  
    . . .  
  
    public void CreateUser(User user)  
    {  
  
        CreateUserAccessor accessor =  
  
            new CreateUserAccessor(application);  
  
        accessor.NameEdit.SetValue(user.Name);  
  
        accessor.OKButton.Click();  
    }  
  
    . . .  
}
```



Create user

Name:

Kind:

OK

Дополнение: Tests

```
[TestFixture]
public class TestExample{
    . . .
    public void GetTests ()
    {
        TestData input = GetTestData();
        User user = input.User;

        yield return new DynamicTest(
            ()=>CreateUser(user),
            "Create new user")
        . . .
    }

    private void CreateUser(User user)
    {
        Context.UserManagementService.CreateUser(user);
    }
    . . .
}
```

Create user

Name:

Kind: