



2010
Разработка ПО

в мире SharePoint

УПРАВЛЕНИЕ ЖИЗНЕННЫМ ЦИКЛОМ АРХИТЕКТУРЫ

Автор



- SharePoint Engineer 
- Belarus SharePoint User Group Lead. 



архитектура


Сдвиг фокуса



«Лучше день потерять, а
потом за пять минут долететь»
© Гриф



План

- Разработка архитектуры
 - Коммуникация
 - Внедрение
 - Контроль
 - Повторное использование
- 



Разработка архитектуры

Да будет свет!



Разработка архитектуры

- Общие принципы
 - Separation of concerns
 - Single Responsibility principle
 - Principle of Least Knowledge
 - Don't repeat yourself (DRY)
 - Minimize upfront design

Разработка архитектуры

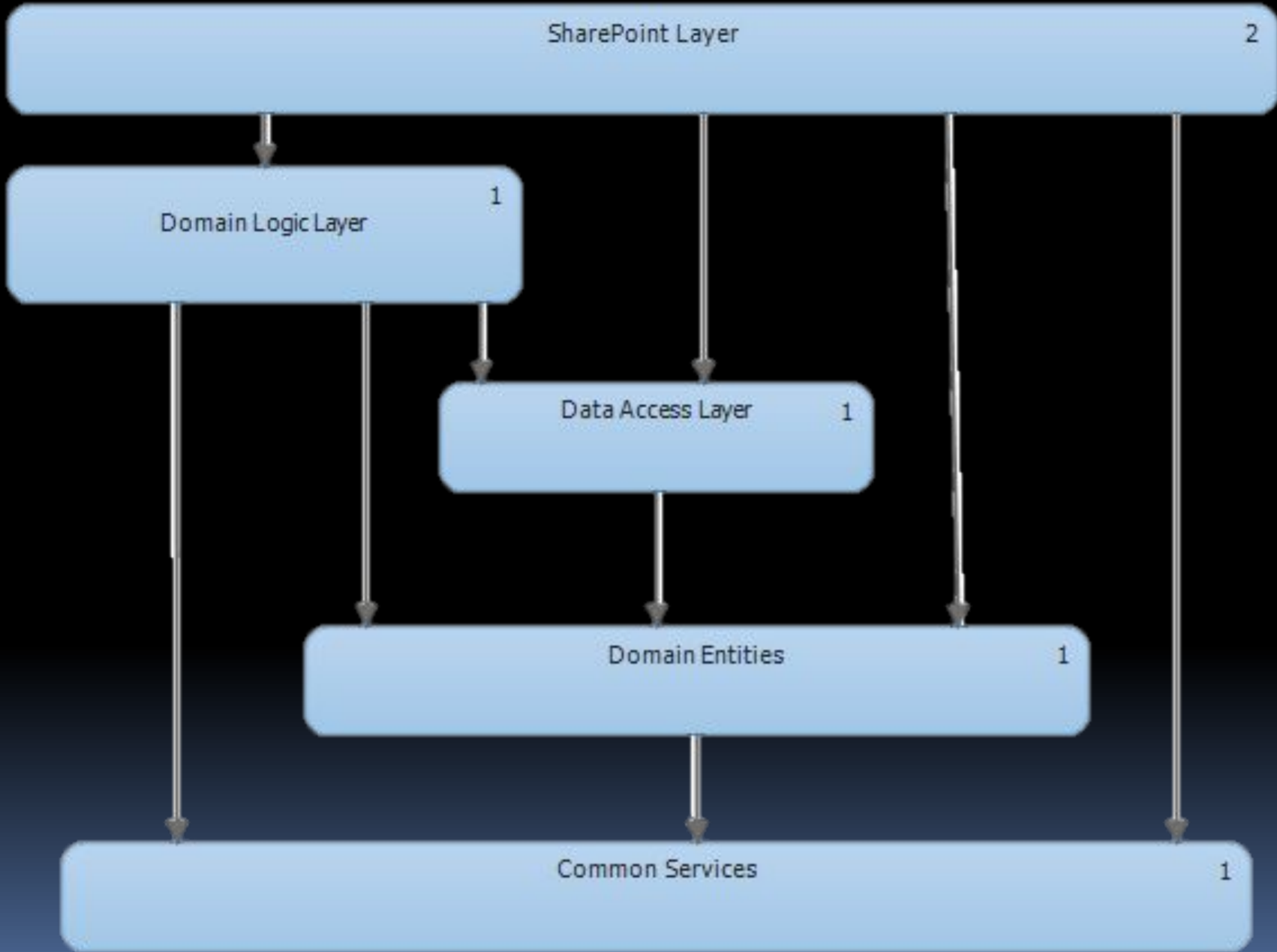
- Положительные особенности SharePoint
 - Authentication
 - Authorization
 - Communication
 - Deployment
 - Performance
 - Data Access API

Разработка архитектуры




Разработка архитектуры

- Разделение на слои
 - Минимум 1 слой (SharePoint Layer)
 - Максимум N слоев
 - В среднем – 5





1 SharePoint Layer

- Web Parts
 - Features
 - Receivers
 - Timer Jobs
 - Elements Definitions
 - Workflows*
- 

1 SharePoint Layer

- “Presentation Layer”
 - User Controls
 - Application Pages

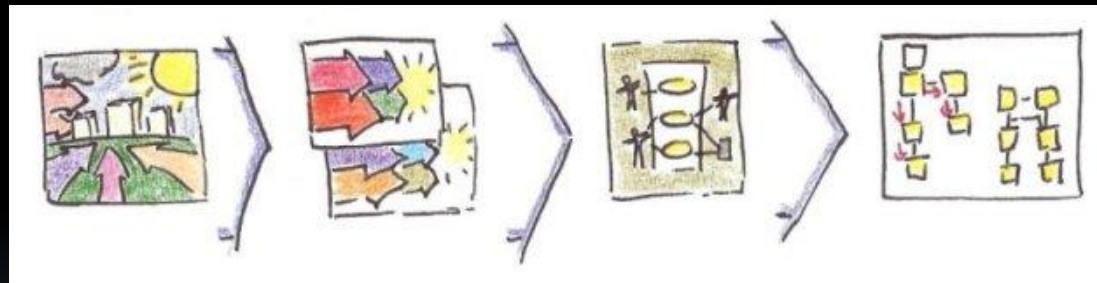
- Альтернатива – самостоятельный слой

1 SharePoint Layer

- Он есть всегда
- Может состоять из нескольких независимых проектов
- Создается с помощью VS templates
- Содержит единицы развертывания WSP


2 Domain Logic Layer

- Бизнес-требования, переведенные в код





2 Domain Logic Layer

- Создается по мере необходимости
 - Обычная .NET сборка
 - Главный субъект модульного тестирования
- 




3 Data Access Layer

- Repository
 - Row Data Gateway
 - Table Data Gateway
 - Active Record
 - Service Locator
- 



3 Data Access Layer

- Создается по мере необходимости
 - Обычная .NET сборка
 - Особенно полезна при работе с разнородными источниками данных
 - Может включать логику List Throttling
- 

4 Domain Entities

- SPMetal как основа
 - Создается сразу
 - Определения и свойства
 - Нет методов
- Расширена за счет Partial Class

4 Domain Entities

- Вынесена отдельно
 - Автогенерация
 - Нет логики

Но

- Может стать ядром Domain Logic Layer
 - Расширить Partial Class методами

5 Common Services Layer

- 2 типа общих сервисов
 - Хелперы
 - Shared Services Applications

5 Common Services Layer


- 2 типа общих сервисов

- Хелперы

- Shared Services Applications (**DAL**)



5 Common Services Layer

- Constants
 - GUIDs
 - Custom Exceptions
 - Extension Methods
- 




Коммуникация

Архитектуру – в массы





Коммуникация

- Заказчик (business people)
 - Руководитель проекта (ПМ)
 - Аналитик
 - Команда разработчиков
- 

Коммуникация

- Заказчик (business people)
 - «продажа» архитектурных решений
 - Совместно с Business Analyst
 - Обычно с высоты >10 км над уровнем моря
 - Диаграммы компонентов + слоев достаточно
 - Слайды

Коммуникация

- Аналитик (business analyst)
 - Объяснение узких мест (риски)
 - Четкая постановка альтернатив (или/или)
 - Диаграмма компонентов
 - Диаграмма вариантов использования
 - Диаграмма активностей



Коммуникация

- Команда разработчиков (+ ПМ)
 - Объяснение основных принципов
 - Reference Implementation
 - Architecture Guidance



Коммуникация

- Команда разработчиков (+ ПМ)
 - Убедиться, что идеи поняты правильно
 - Убедиться, что идеи приняты и будут исполняться
 - Учесть обратную связь в Architecture Guidance



Коммуникация

- Команда разработчиков (+ ПМ)
 - Убедиться, что идеи поняты правильно
 - Убедиться, что идеи приняты и будут исполняться
 - Учесть обратную связь в Architecture Guidance



Внедрение

Оживление франкенштейна.





Внедрение

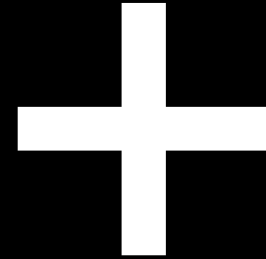
- Architecture Guidance Document
 - Допущения
 - Ограничения
 - Особенности платформы
 - Основные паттерны

Внедрение

- Deployment Guidance Document
 - Обычно – глава в Architecture Guidance
 - Развертывание
 - Резервное копирование
 - Восстановление
 - Upgrade

Внедрение

- Общие рекомендации
 - Review + update
 - Reference Implementations






Контроль

Кто не все – того накажем.





Контроль

- Peer Code Review
 - Architecture Review
- 

Контроль

- Сделать Reviews частью процесса
- Проводить их XP-style
- Используйте инструменты (MSVS2010U)
- Обновляйте проектную документацию
- Заводите «технические истории»
- Проводите разбор с командой



Повторное использование

Берегите природу.



Повторное использование

- Общий код
- Общий компонент
- Managed Metadata Service Application
- Workflow Activities



Повторное использование

- Reference Implementation
 - Guidance Documents
 - Patterns
- 



Заключение


А что у вас есть против оборотней?





Заключение


Не бывает одинаковых проектов, но бывают
очень похожие






Заключение

Архитектура – это здорово. Но лучше успешный проект с плохой архитектурой, чем проваленный проект с хорошей.





blog: <http://vspug.com/sharepointby>
twitter: [@sharepointby](#)
web: <http://sharepoint.by>
profile: <http://linkedin.com/in/ivanpadabed>



СПАСИБО!