Разработка крупных и высоконагруженных проектов – рецепты успеха

Александр Сербул
Руководитель направления

1С·БИТРИКС

When I wrote this code,
only God & I understood what it did.

Now...
only God knows.

Проектирование

Успех веб-проекта

Успешный – далее только в техническом плане.

Может ли Клиент «завалить» успешный веб-проект?

Все условия созданы, а Партнер «заваливает» веб-проект, почему?

Люди или методологии?

Компетенция, сертификации, совесть: «зачем делать просто, если можно сложно»?

Когда процесса - нет

- Сделать к 1 ноября – конечно, деньги вперед!!!
- Ничего не проектируется - зачем, все «понятно»
- Разработчик делает «лишь бы работало до зарплаты»
- Тестировщик покликал – багов нет!
- Аврально вносятся изменения
- Документация – а что это?
- Этап сдан?

Делаем «на коленке»

Риски:

Систему все сложнее развивать (экспонента)

Новый программист пытается все переписать с нуля

Программист может и не разобраться в такой веб-системе

Веб-система - боится изменений

Никто не помнит, как все работает (даже Заказчик!)

Любое изменение рождает много ошибок

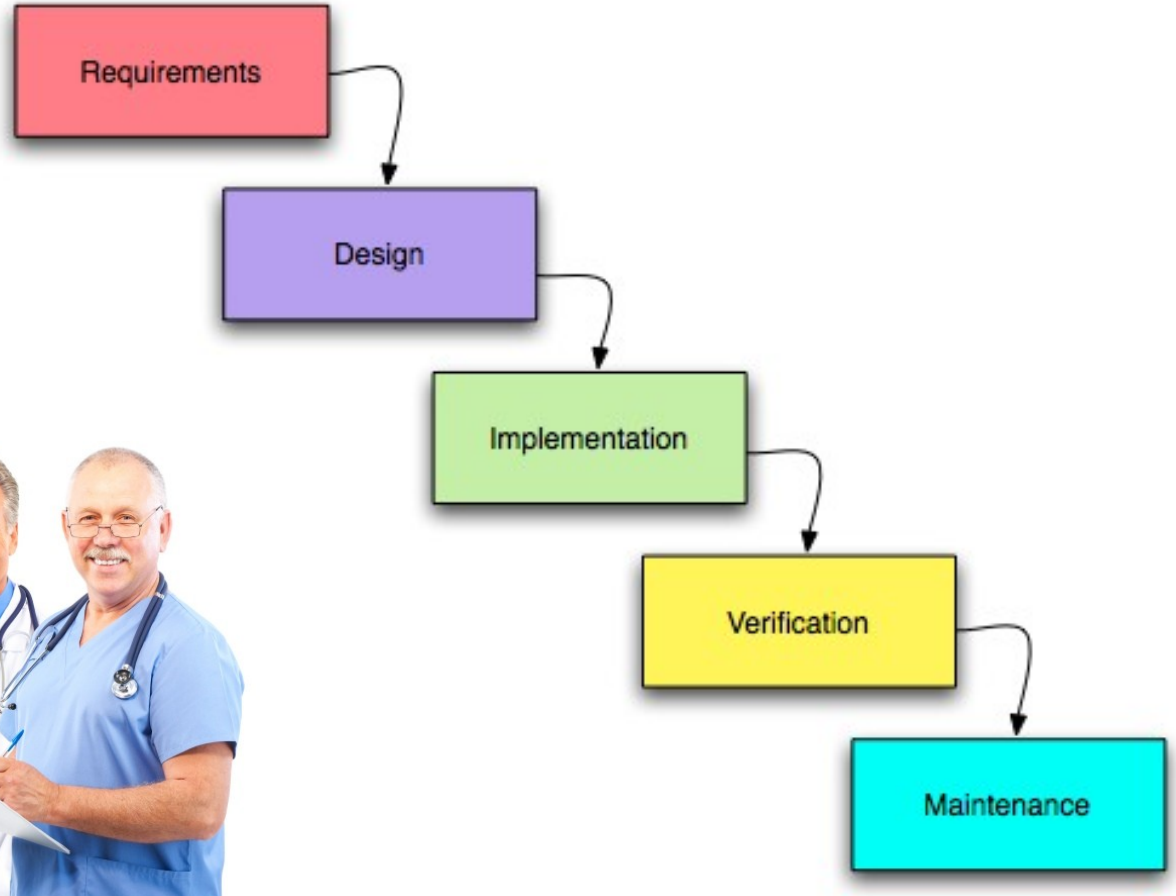
Тестировщик не знает, как все проверить

Давайте все пропишем в ТЗ!

Процесс – «Водопад»:
Подробно все проектируем, рисуем интерфейсы, описываем в ТЗ
Получаем ТЗ на 1000-20000 страниц
Кодируем
Проводим нагрузочные испытания
Тестируем
Сдаем проект/этап Заказчику

Работает на сложных/больших, специфических проектах.
Любое изменение требует больших затрат на пересогласование, перепроектирование…

Давайте все делать последовательно!



1С·БИТРИКС

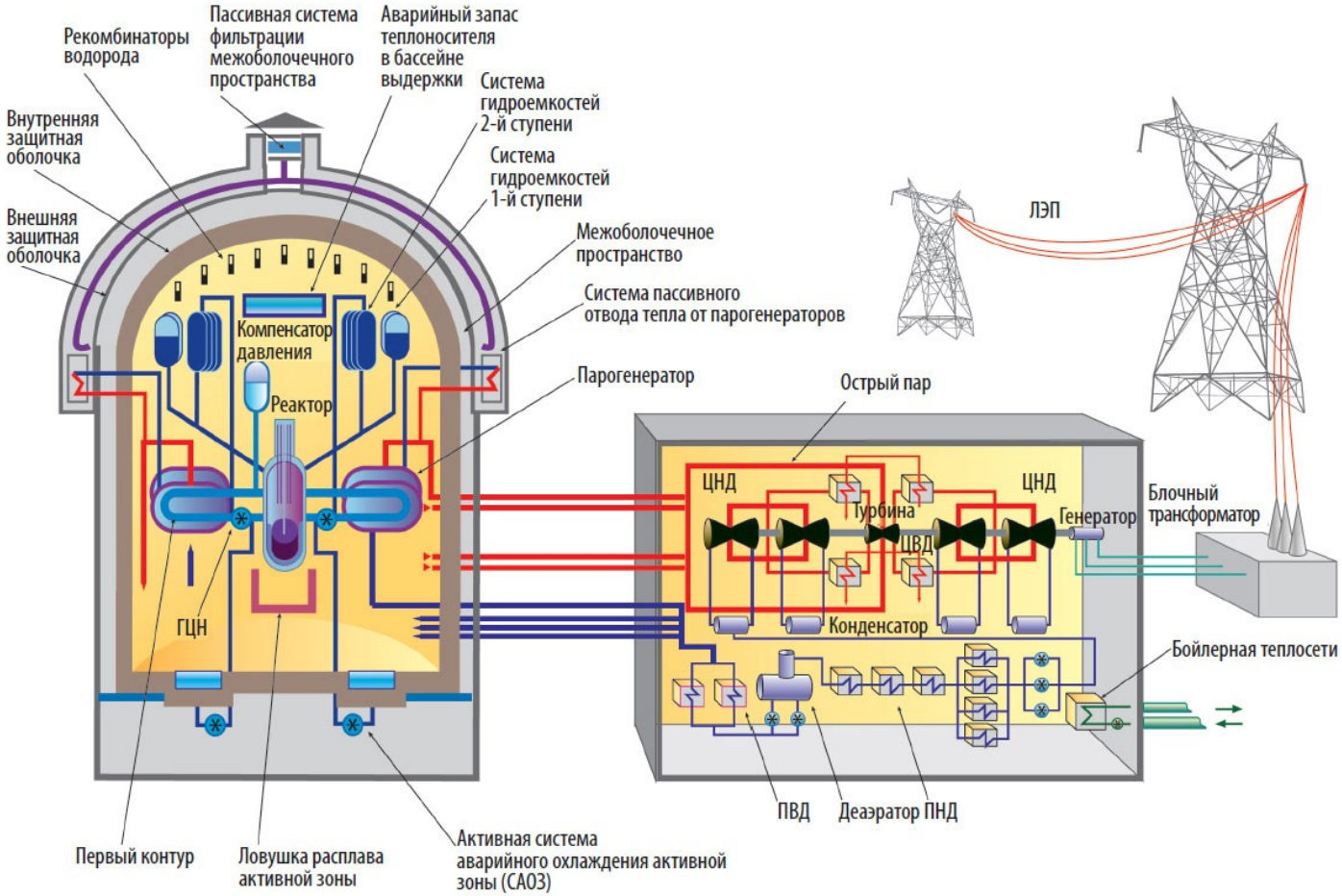
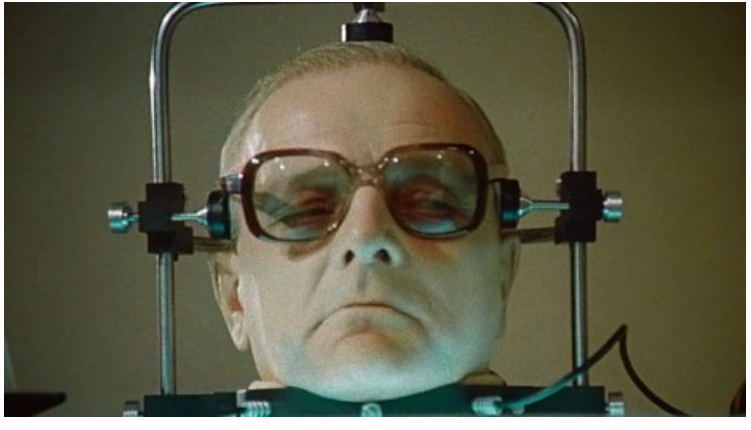


СХЕМА ЭНЕРГОБЛОКА И СИСТЕМЫ БЕЗОПАСНОСТИ

Вы – не эксперты!

Вы – не эксперты в предметной области!
Эксперт – Клиент (если повезет)
Нельзя «отрезать голову» у Клиента и заставить жить в офисе разработки
По мере формализации требований – будут появляться НОВЫЕ идеи
Это будет продолжаться – до запуска веб-системы

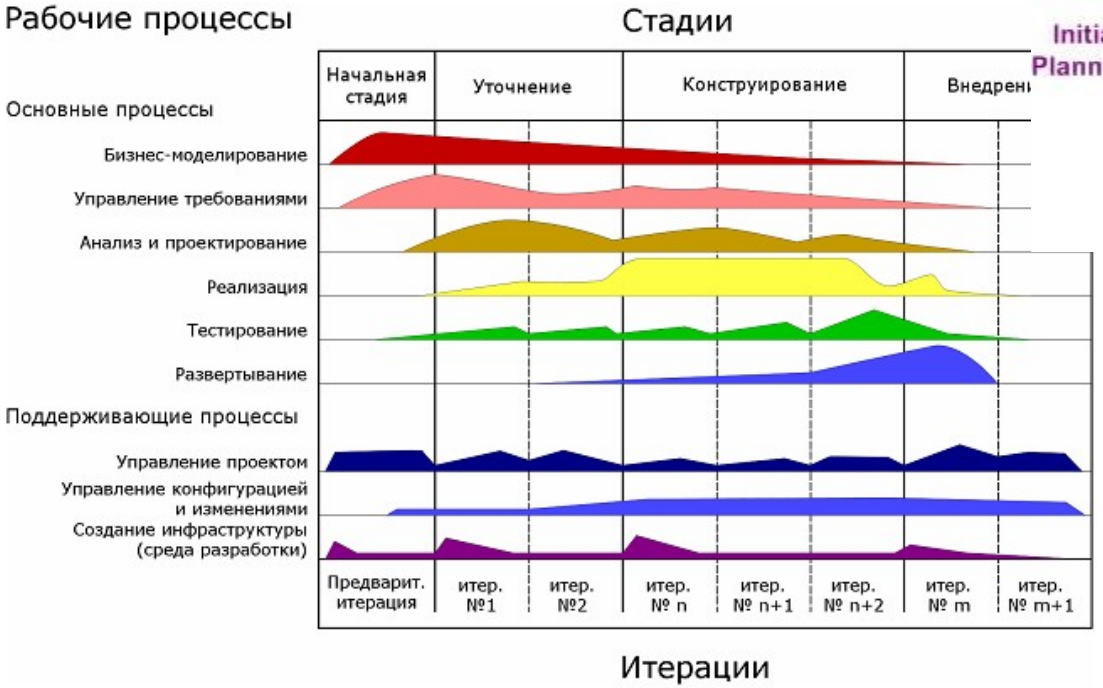


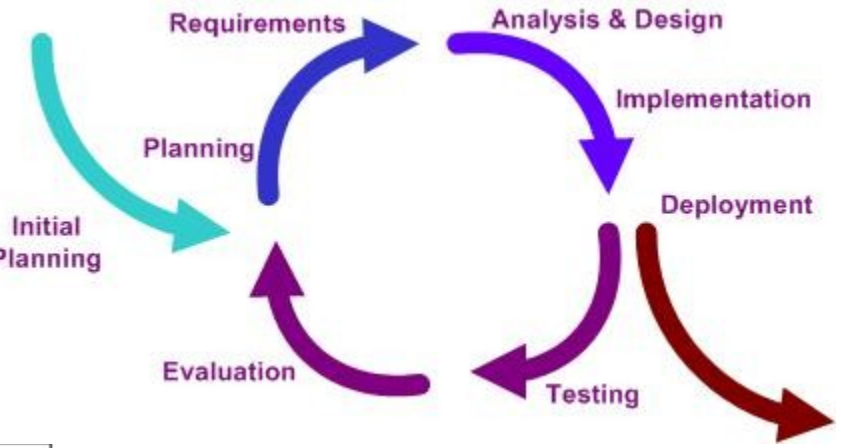
Подводные камни

Битрикс24

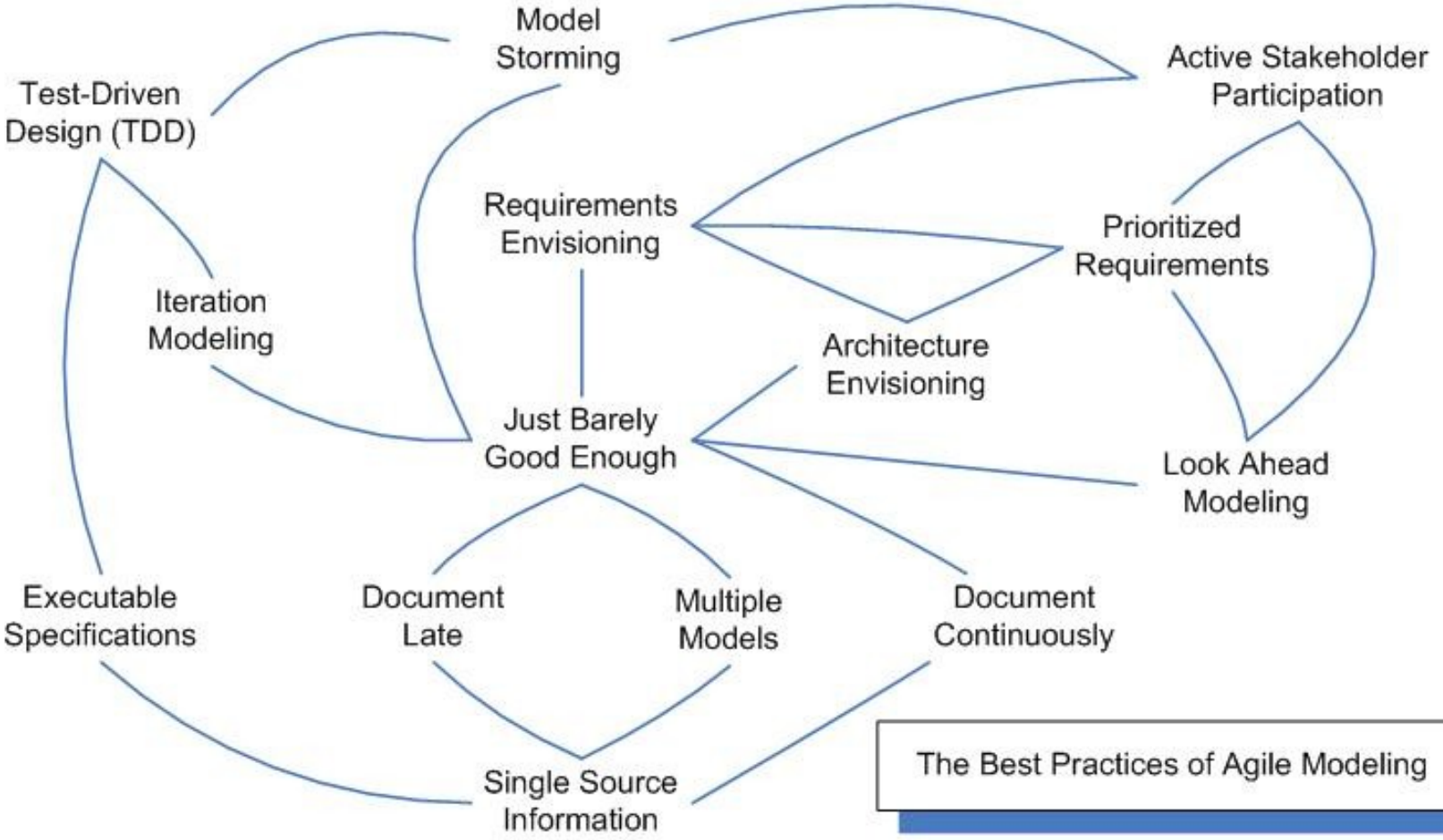
- «Проектирование» головного мозга:
  - что будет, если все все все описать в ТЗ
  - сколько нужно аналитиков и чем они занимаются
  - а правда, что бывают настоящие архитекторы и какие у них задачи в веб-проектах?
  - что делать со сложными, формальными требованиями?
  - как не утонуть в куче бумаги, которая завтра устареет к чертовой матери?

Итеративный процесс





1С·БИТРИКС



The Best Practices of Agile Modeling

Copyright 2005-2011 Scott W. Ambler

Артефакты проектирования

- Глоссарий
- Логическая модель данных
- Сценарии использования
- Спецификация потоков данных
- Поможет ли UML?

Техническое проектирование

- Компоненты
- Модули
- Классы
- Объекты или функции или скрипты
- Тесная связь автотестирования с техническим проектированием, DI, моки

Большой интернет-магазин/проект

Битрикс24

- Итерации, иногда упрощенный RUP, Agile Unified process
- Проекция требований на Битрикс, спецификация потоков данных, архитектура веб-кластера, план нагрузочного тестирования
- Дизайн/верстка в итерациях
- Интеграция в 20-50 итераций (сотни стандартных компонентов, 5-50 нестандартных и модулей, добавление контента и меню, поиска в Sphinx, REST API и др.)
- Выгрузка на хостинг, многоуровневые тестовые среды, настройка веб-кластера, настройка мониторинга, DevOps

Большой интернет-магазин/коммуникации

Битрикс24

- Model Storming, Story Mapping, облако идей сначала на стенах, затем в группах/вики/задачах (product/sprint backlogs)
- Прототипирование высоконагруженных узлов будущей системы, тестирование субкомпонентов – итерационность из RUP/AUP
- Параллельный Scrum команд разработчиков, дизайнеров/верстальщиков, аналитиков, DevOps. Scrum of Scrums
- Эффективные ретроспективы
- Обмен знаниями через базы знаний, бесплатные/платные курсы, сертификацию

Управление крупными проектами

Подводные камни

Битрикс24

- «Куры» и «свиньи»:
  - почему боятся прозрачности
  - как поощрять всплытие проблем и открытость на ретроспективах
  - как Scrum позволяет бороться с неэффективным менеджментом за разумное качество и честные сроки
  - как бороться с перекладыванием ответственности и футболом задач, ситуационное лидерство в Agile
  - как заставить людей работать в позитивном ключе, без принуждения к труду, в Agile

Подводные камни

Битрикс24

- ТЗ и ХЗ. Когда цели запустить проект … просто нет:
  - чем больше бумаги, чем п..па чище и User Stories
  - Agile и … «сколько будет стоить проект?»
  - Agile и … «мы должны были уже запуститься 1 июня 1666 года»
  - Agile и работа с крупными крупными компаниями
  - почему за баги вводят финансовую ответственность в договор
  - стоимость качества, бета-проекты в Интернет
  - «дурака работа любит», «садомазо: жертвы и герои», «мама спаси», «моя хата с краю» и другие отклонения он нормального поведения в Agile-командах

Популярные «косяки»

- Менеджер включает дурачка: «ТыЖ программист, ТыЖ умный!»
- Менеджер не хочет разбираться в деталях сценария, алгоритма, бизнес-логики – голова начинает болеть
- Заказчик забывает как все работает, как и проектный/продуктовый менеджмент. Программисты увольняются, автотестов нет. А деньги потрачены ☺ Наступает ЖОПА.
- Код разработчиков никто не проверяет, а это очень важно.

WINTER IS ~~COMING~~
Here

Битрикс24

Манифест, б..ть - 1

Что говорят менеджеры?	Что им нужно на самом деле?	Что мы делаем в итоге?
Продуктивная командная работа	*Сотни рабочих часов в неделю*	**Мы пишем код б..ть!**
Качественные продукты	*100% покрытие кода юнит-тестами*	**Мы пишем код б..ть!**
Учитывание пожеланий пользователей	*Выполнять любой каприз заказчика*	**Мы пишем код б..ть!**
Продукт должен быстро эволюционировать	*Постоянные изменения требований и пустопорожние обсуждения новых фич*	**Мы пишем код б..ть!**

Манифест, б..ть - 2

Битрикс24

- Мы — сообщество программистов, которые вдоволь намучились с разнообразными методологиями разработки.
- Нас уже тошнит и от экстремального программирования, и от Скрама, и от Канбана, и вообще от всего, что напрямую не связано с **написанием б..ть кода**.
- Нам очень неприятно, что нас считают асоциальными дебилами, которые ничего не могут сделать без указания менеджеров. Нас просто бесит, когда нас заставляют заниматься парным программированием, передвиганием листочков по доске, утренними standup-митингами и другими дурацкими приседаниями просто потому что из 10 менеджеров в нашей конторе **никто не умеет б..ть писать код** (а какую-то видимость работы создавать нужно).
- Мы выступаем за то, чтобы отказаться от всех существующих методологий разработки в пользу одной, самой простой и единственно верной. Эта методология называется **«пиши код б..ть»!**
- Мы уверены, что вся хрень, написанная в первых двух колонках, абсолютно никому не нужна. Есть только одно дело, которым действительно нужно заниматься: **писать код б..ть!**

Agile Manifesto

Битрикс24

«Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь разработкой непосредственно и помогая в этом другим. Благодаря проделанной работе мы смогли осознать, что:

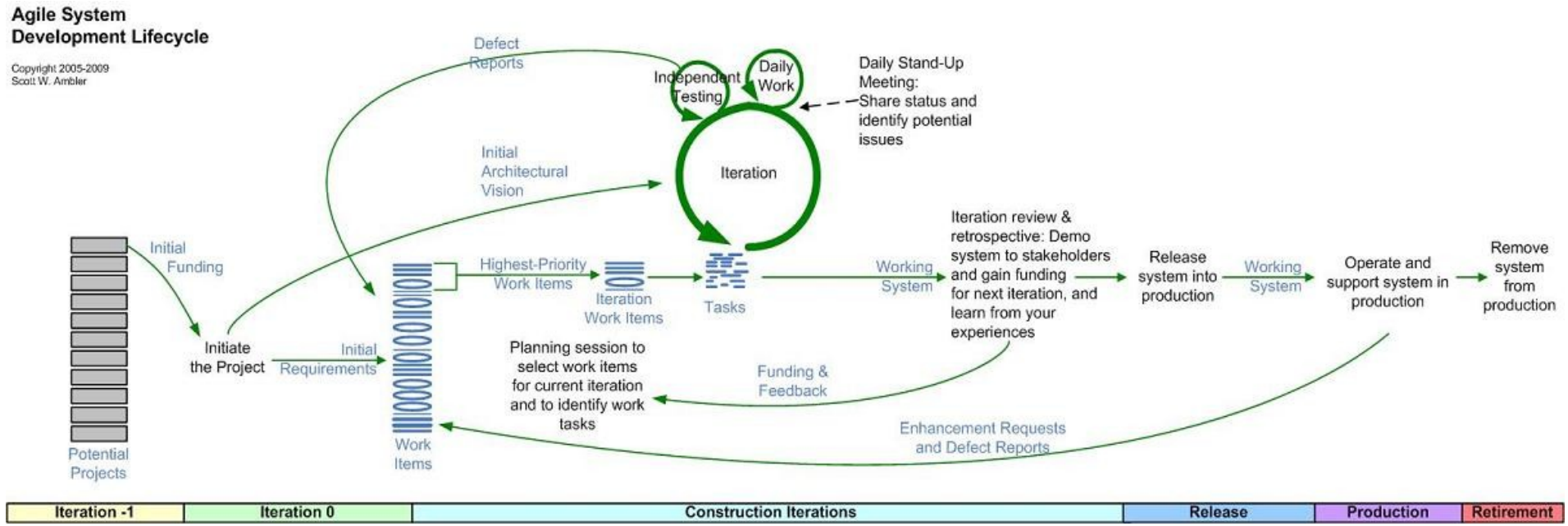
- Люди и взаимодействие важнее процессов и инструментов
- Работающий продукт важнее исчерпывающей документации
- Сотрудничество с заказчиком важнее согласования условий контракта
- Готовность к изменениям важнее следования первоначальному плану

То есть, не отрицая важности того, что справа, мы всё таки больше ценим то, что слева.»

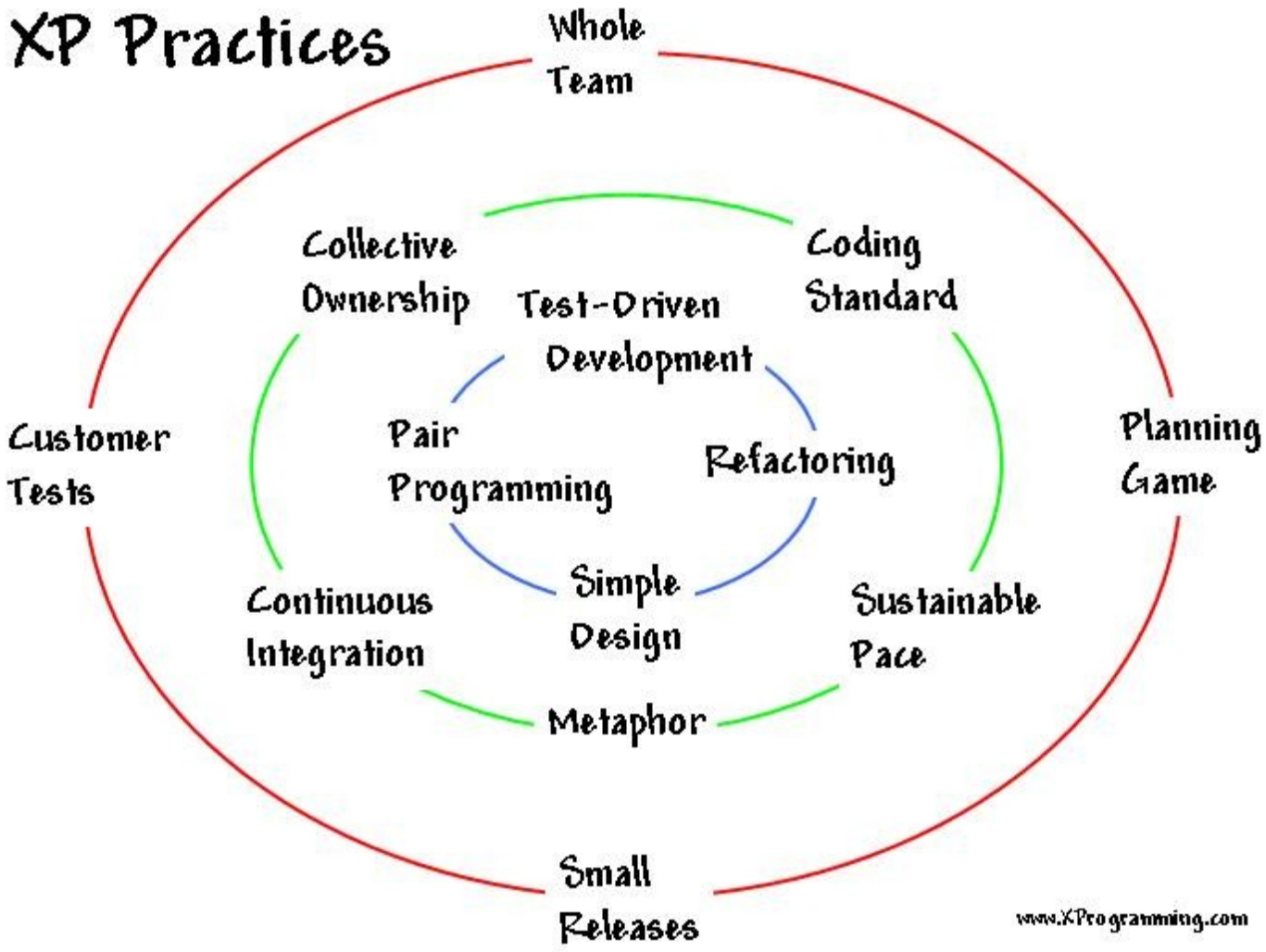
Agile Manifesto, 2001 год

Битрикс24

Процесс…



Битрикс24


XP Practices
Whole Team
Collective Ownership
Coding Standard
Test-Driven Development
Customer Tests
Pair Programming
Refactoring
Planning Game
Continuous Integration
Simple Design
Sustainable Pace
Metaphor
Small Releases
www.XProgramming.com

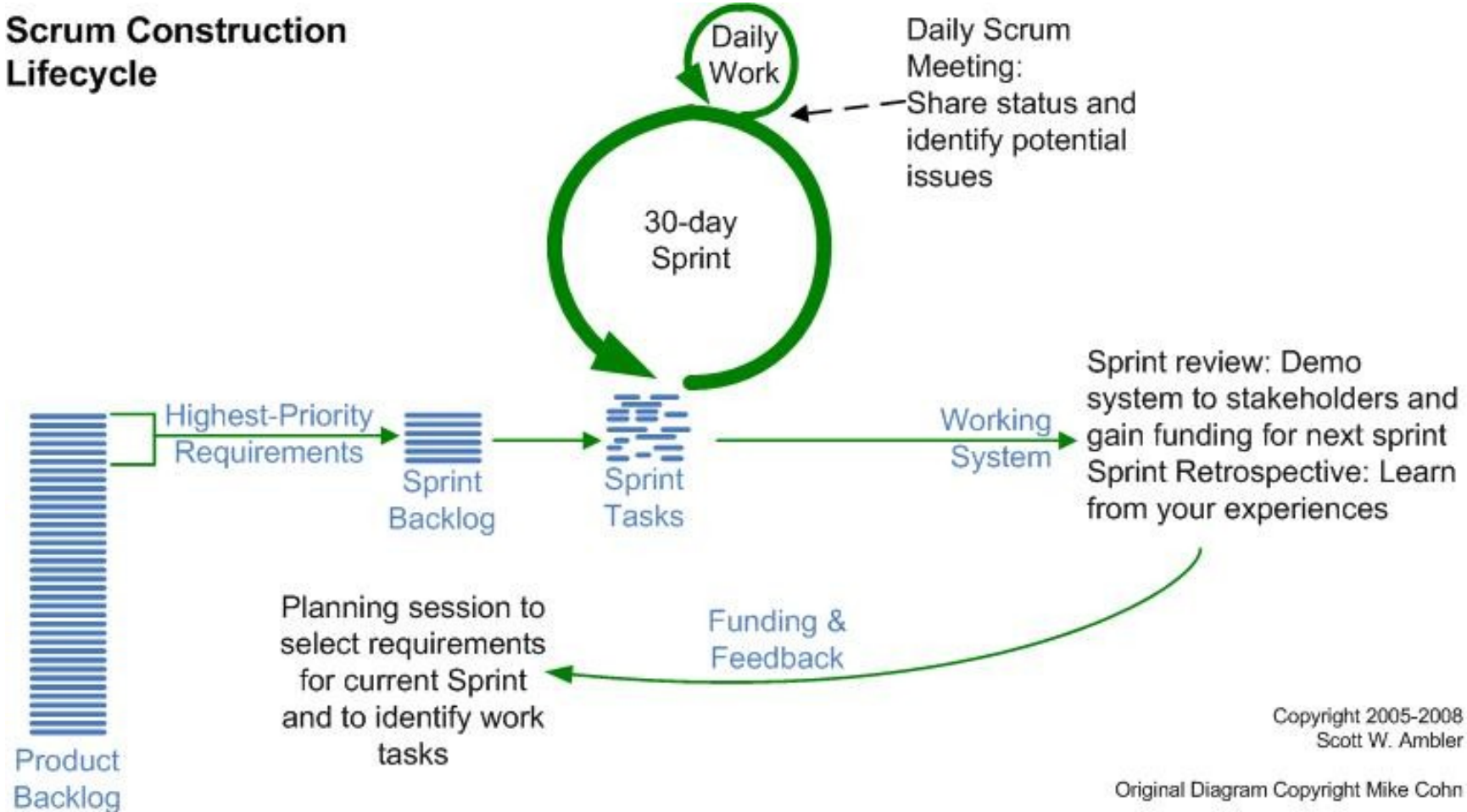
Битрикс24

Жизненно-важные для Agile инструменты

Визуализированные требования, Product/Sprint backlog	Wiki: Atlassian Confluence, Redmine, Битрикс24
Версионное хранилище артефактов	Wiki, SVN, OneDrive, Dropbox, Битрикс24.Диск
Трекер задач с приоритетами/правами	Atlassian Jira, Redmine, Битрикс24.Задачи/Канбан
Чаты/видеочаты для стендапов/митингов/ретроспектив	Skype, Hangouts, Битрикс24.Видеозвонки
Бронирование переговорных для митингов	Sharepoint, Битрикс24
Планирование митингов в календарях	Sharepoint, Битрикс24

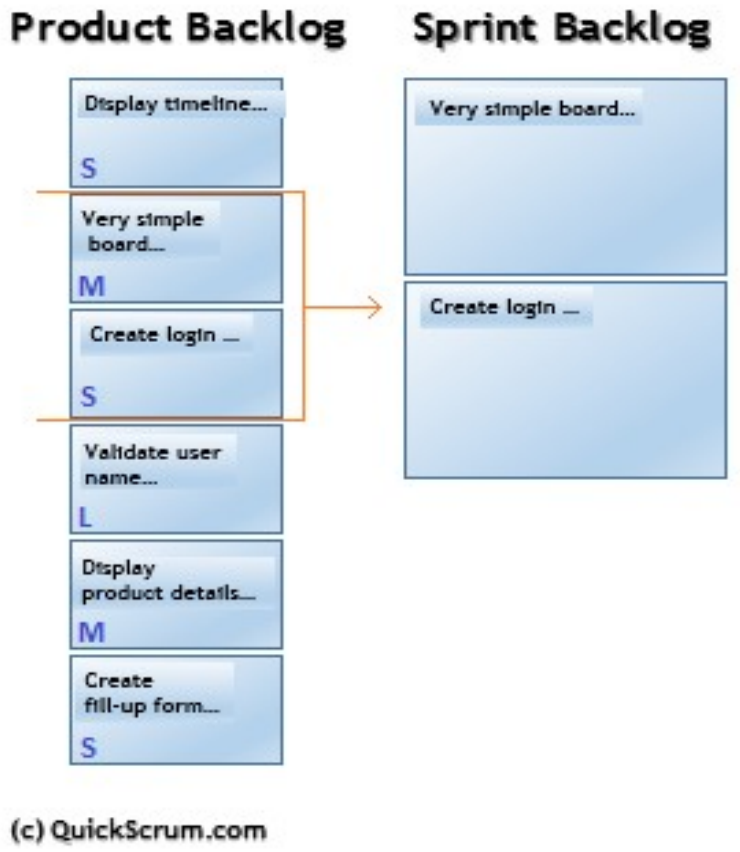
Процесс в Scrum

Битрикс24



Product Backlog

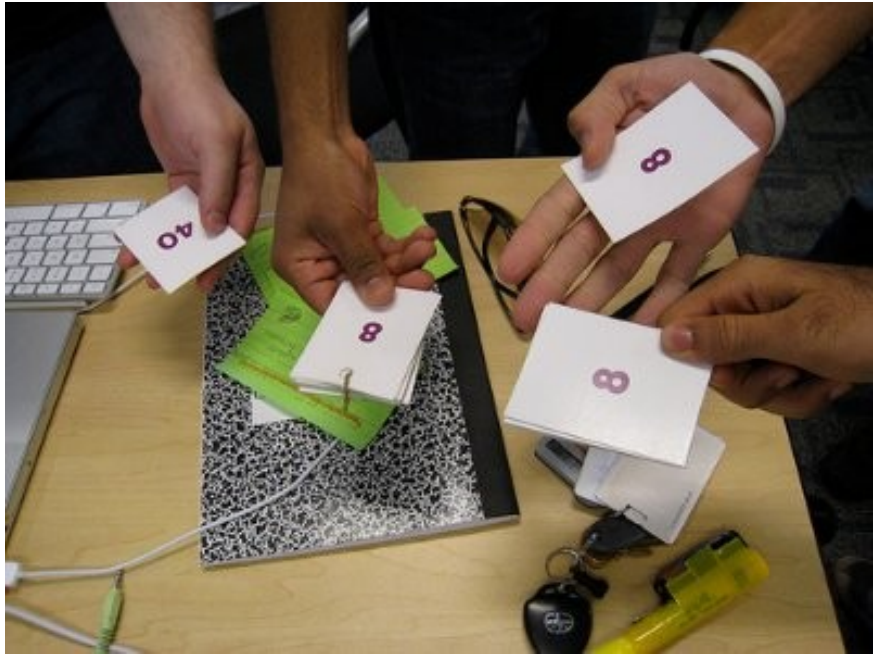
- Что нужно и как проверить
- Оценка в «условных» единицах
- Важное – вверху
- Story Mapping



Битрикс24

Planning Poker

- Коммуникация
- Снижение разброса оценки
- Командная ответственность

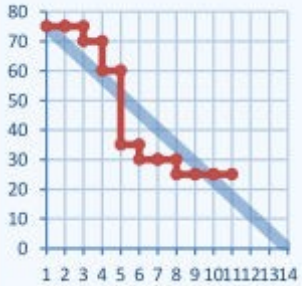


Битрикс24

Sprint

Sprint Goal	To Do	Doing	Done
The goal of this sprint is To make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.			Item #1 t.1.6 t.1.1 t.1.3 t.1.5 t.1.2
	Item #2 t.2.7	t.2.6 t.2.5	t.2.1 t.2.3 t.2.2 t.2.4
	Item #3 t.3.4 t.3.5 t.3.3 t.3.2	t.3.1	
	Item #4 t.4.4 t.4.2 t.4.5 t.4.1 t.4.3		
	Item #5 t.5 t.5.4 t.5.1 t.5.5 t.5.2		

Sprint Burn-Down Chart



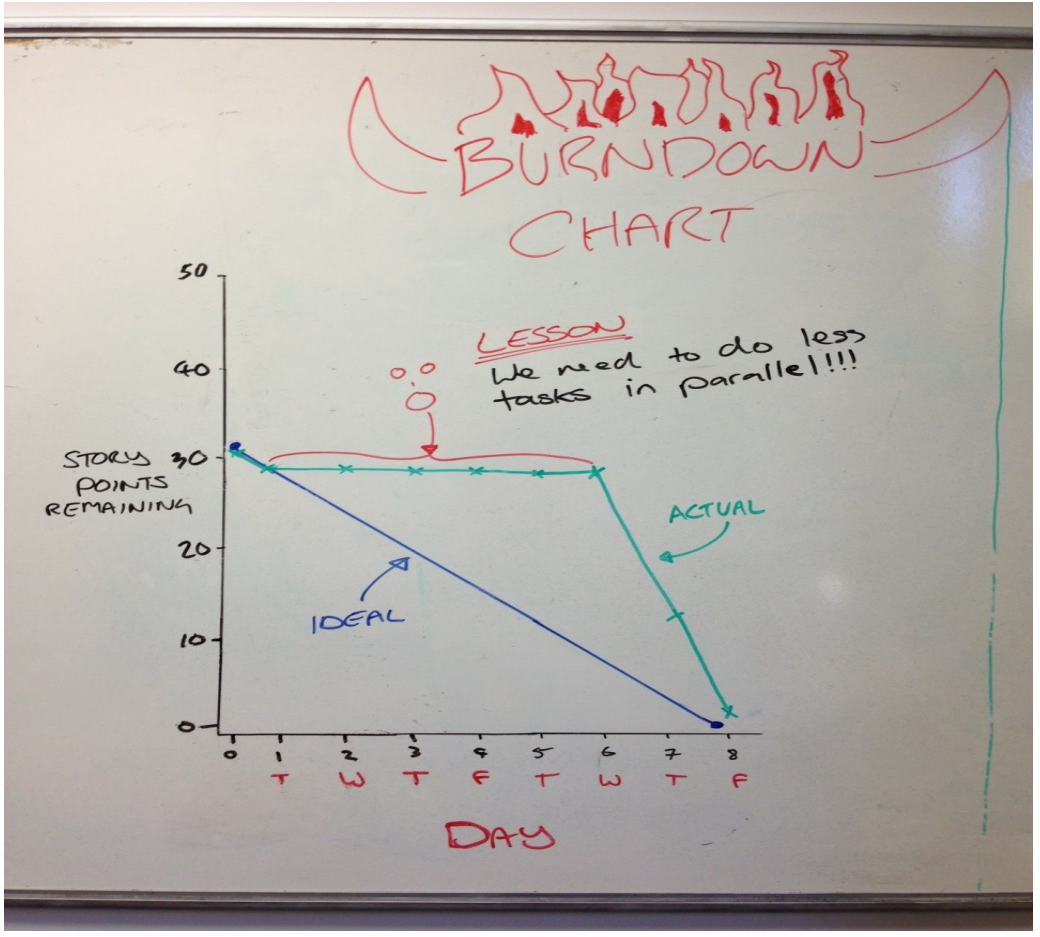
Daily meeting

Битрикс24

Burndown chart

Битрикс24

- Визуализация прогресса
- Отстающие
- Scrum of Scrums



Роль – Product Owner

Битрикс24

- Знает ответы на вопросы
- Принимает решения
- Один!!!

Voice of the customer

Owns value

Gathers feedback

Makes decisions

Роль – Scrum Master

Битрикс24

- Координация процесса
- Устранение препятствий
- Можно меняться
- «Добрый» пастырь

Sprint review meeting

Битрикс24

Итерация 1

Итерация 2

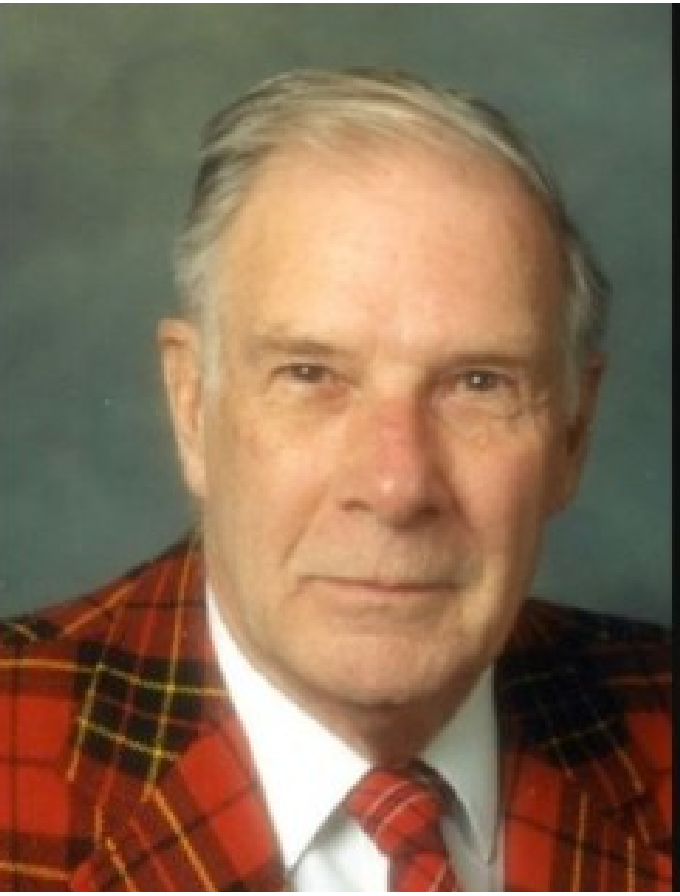
Итерация 3, да что же творится то?

Разработка крупных проектов

Войны технологий – усугубляют сложность

- Вместо сотрудничества – конкуренция за деньги на рынке
- Вместо «развития» языков (Scala), отмечается «межвидовая мутация» (Node.JS) и параллельное изобретение велосипедов (ES6, Go)
- Масштабные религиозные войны: Java vs C#, PHP vs Python vs Ruby
- Войны фреймворков: Angular vs React JS
- “Зачем делать просто, если можно сложно?”

1С·БИТРИКС



Mathematicians stand on each others' shoulders and computer scientists stand on each others' toes.

— Richard Hamming —

AZ QUOTES

«Шовинизм» в программировании

- Вы писали 5 лет на PHP… Но почему, за деньги чтоль? – на работу не беру
- Вы только байтики умеете на C гонять? – в нашу компанию вы не подходите
- Только Java/Spring, только своих… - и пофиг на скорость, качество, результат
- C#/Windows forever, Java/PHP - го…но
- Гошечка…

Битрикс24

“Иерархия” языков программирования

PHP

Python

Ruby

Lua

Java Script

Haskell

Akka/JVM

Erlang

Groovy

Java

C#

Scala

F#

Kotlin

Ассемблер

C – переносимый ассемблер

C++ - много мелких тонкостей и «дыр», большое и противоречивое наследство

Выразительная мощность

- Более «новые» (скриптовые) ЯП решают задачи часто более эффективно и лаконично
- В web законно царствует PHP – краткость, мощь, простота
- Компоненты и узкие сервисы за более «медленными в разработке и низкоуровневыми» языками
- Многопоточное программирование – сложно. Actors, абстракции

Битрикс24

Иерархия стандартов, языков программирования и технологий для web (фронтэнд и бэкэнд)

Браузер, как «операционная система»

Java Script, jQuery
(Type Script, Angular/React JS)

HTML, CSS

PHP – ничего лишнего и много полезного для web

TCP/IP, почта, сокеты

Файловая система

Регулярные выражения

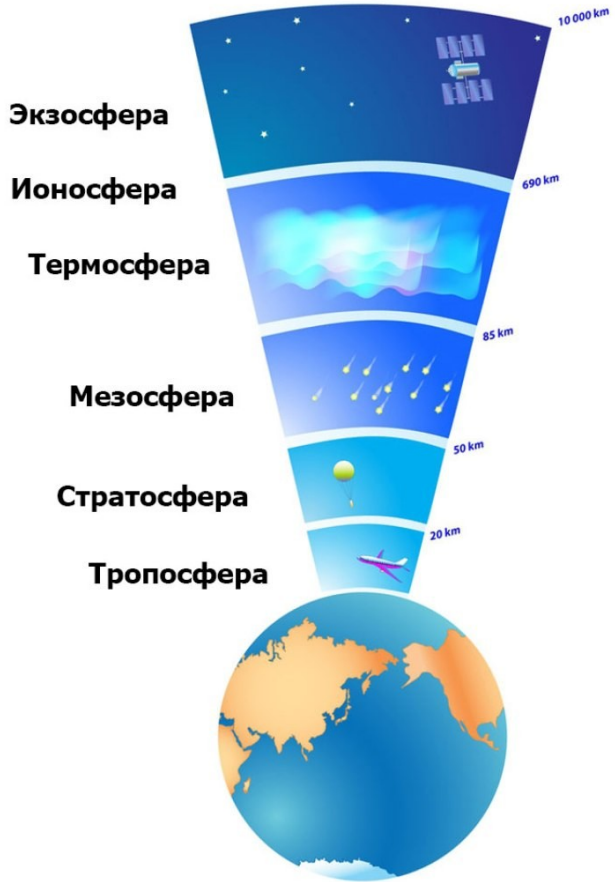
Базовые алгоритмы

База данных

Операционная система

Сильное, выраженное расслоение

- Фронтэнд
- Бэкэнд
- Мобильная разработка
- Веб
- Базы данных
- Бигдата
- Машинное обучение и AI
- DevOps



«Битрикс: Управление сайтом» - изнутри

- Управление сложностью:
  - слой реляционной алгебры и базы данных
  - слой низкоуровневых задач и инструментов
  - слой модулей, сущности и целостность
  - слой классов, логика и валидация
  - **слой компонентов и шаблонов**

Слои, упрощающие задачи для интеграторов

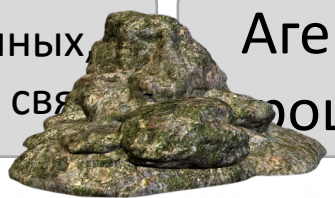
Шаблоны, страницы, меню, контент

Компоненты (высокоуровневая логика)

Модули с человеческим API, “D7” (низкоуровневая логика)

Н

База данных, таблицы, свя



Агенты, процессы

ядра

Композитный кэш

Админки

Подводные камни динамических языков программирования: PHP, Python, Ruby, JavaScript…

В PHP – нестрогая динамическая типизация

1С·БИТРИКС

- А в Python – строгая ☺
- Компилятора – нет, поэтому тестировать может сразу «клиент»
- Рефакторинг – сильно затруднен, нужно сразу писать правильно
- Может упасть внезапно в runtime

- Обязательная проверка на буль и типы в публичных методах
- Обязательное наличие автотестов для нетривиальных функций и алгоритмов (а лучше – для всех)

Утиная типизация

- Удобно… в небольших проектах и экспериментах
- Меньше визуальный шум
- Увеличивается нагрузка на создателя функции, класса – требуется проверка на все возможные типы и выброс исключений
- Обязательное документирование с phpdoc

ООП

- «Дырявые» объекты
- Текущие абстракции
- Внезапные исключения. Как работать с исключениями. Цепочки вызовов.
- Билдеры, фабрики
- Наследование ломает инкапсуляцию, композиция
- Инкапсуляция и автотестирование
- Иммутабельность
- ООП головного мозга

ФП

- Можно неплохо жить без State в объектах
- Цепочки преобразования коллекций, возврат коллекций из функций/методов, Apache Spark…
- Генераторы
- Value-объекты или хэш-массивы
- Haskell – как прокачаться

Автотесты

- Модульные тесты (unit tests)
- Интеграционные тесты
- Как тестировать верстку, компоненты…
- Почему тесты нужно писать до кода
- Почему пишут после кода и к чему это приводит
- Почему появляется монолитно-нетестируемый код
- Анализ покрытия тестами

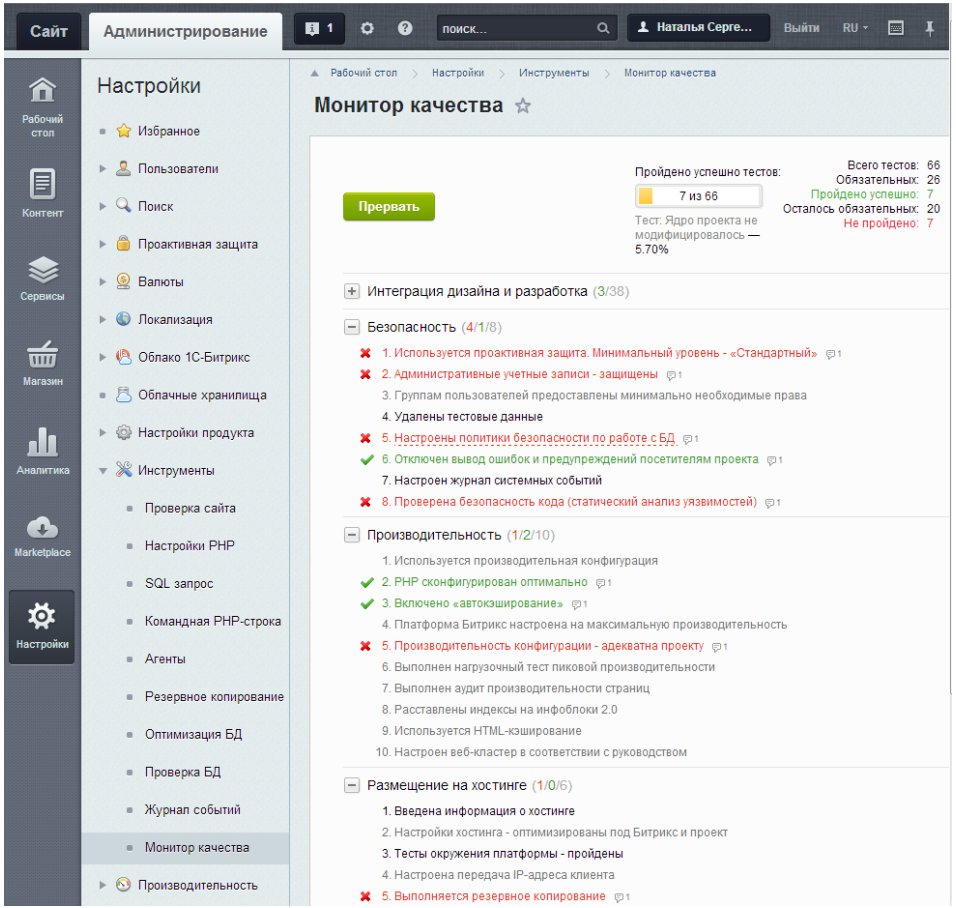
Шаблоны проектирования

1С·БИТРИКС

C	Абстрактная фабрика
S	Адаптер
S	Мост
C	Строитель
B	Цепочка обязанностей
B	Команда
S	Компоновщик
S	Декоратор
S	Фасад
C	Фабричный метод
S	Приспособленец
B	Интерпретатор
B	Итератор
B	Посредник
B	Хранитель
C	Прототип
S	Прокси
B	Наблюдатель
C	Одиночка
B	Состояние
B	Стратегия
B	Шаблонный метод
B	Посетитель

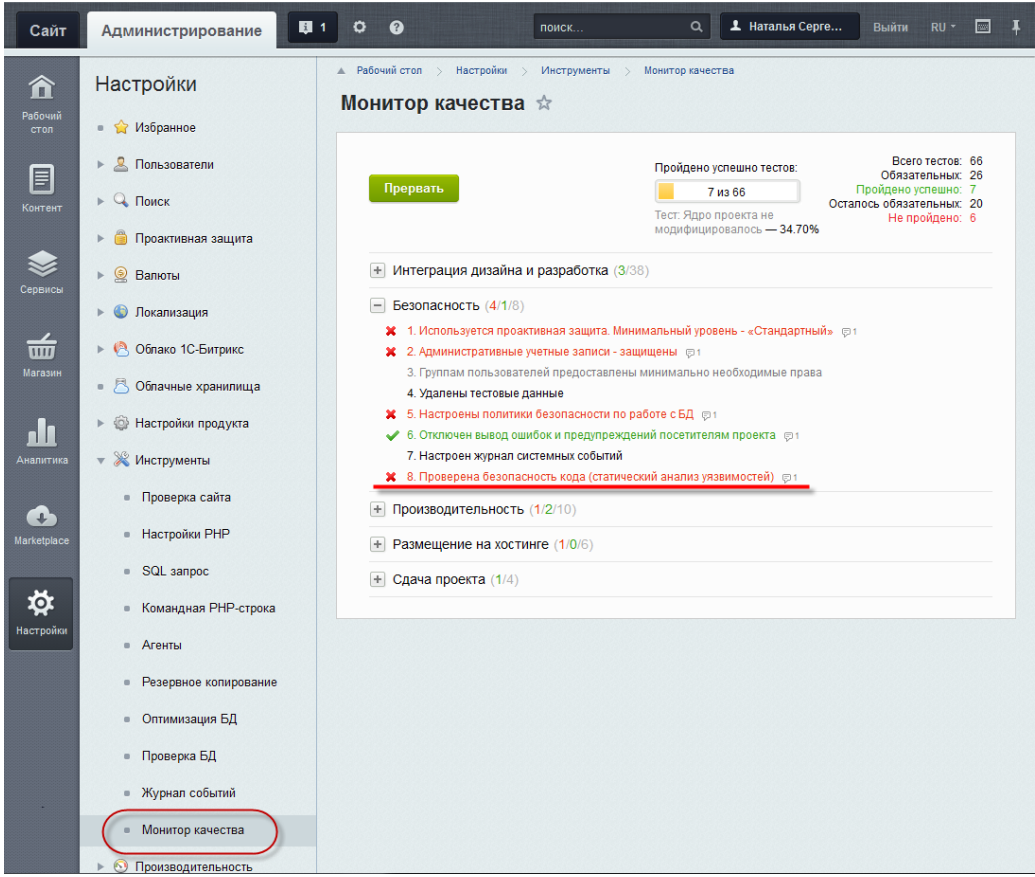
Монитор качества

- Десятки проверок
- Рекомендации экспертов
- Формирование процесса контроля качества



Информационная безопасность

- Code-style
- Аудит кода ведущим разработчиком
- Инструменты защиты Битрикс: анализ кода, веб-файервол, веб-антивирус



Как прокачиваться

- Изучаем другие фреймворки
- Изучаем другие, взрослые языки программирования
- Постоянно читаем на тему разработки, рефакторинга, автотестирования, неважно на каком ЯП
- Изучаем сетевые стандарты и протоколы
- Изучаем мир unix

Модные тренды – возможности и риски

- Docker
- Kubernetes
- Composer
- PSR
- Node.JS
- AI
- ЗОЖ
- Гомосексуализм
- Глобальное потепление

Подготовка к высоким нагрузкам

Ценности ПО

- Программа должна работать
- … работать правильно
- … работать быстро

Только в таком порядке!

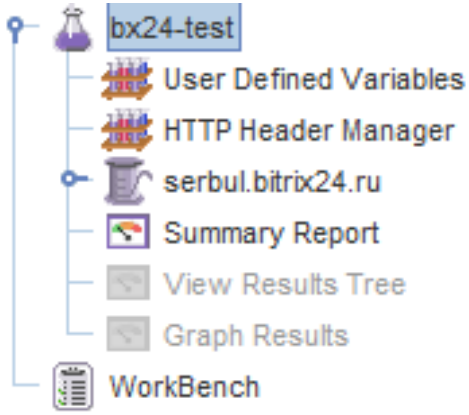
Кэширование и оптимизация SQL-запросов

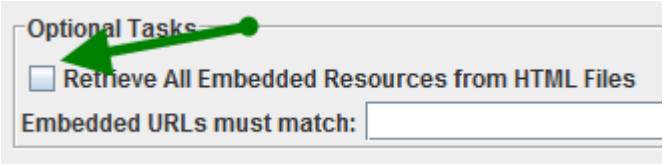
Кластеризация в Битрикс

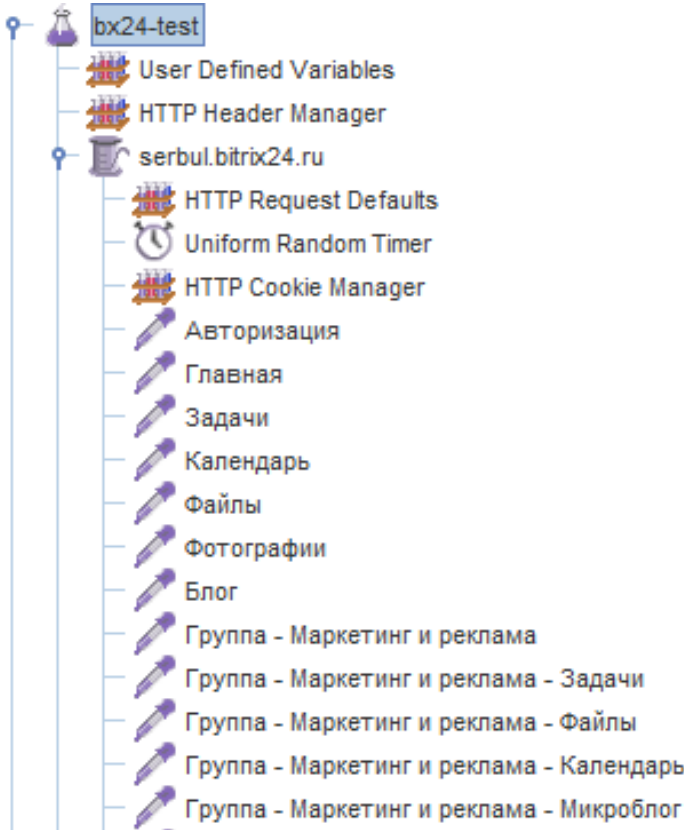
- Кластеризация серверов приложений
- Кластеризация MySQL
- Кластеризация кэша

Отладка веб-кластера

Подготовка и проведение нагрузочного тестирования







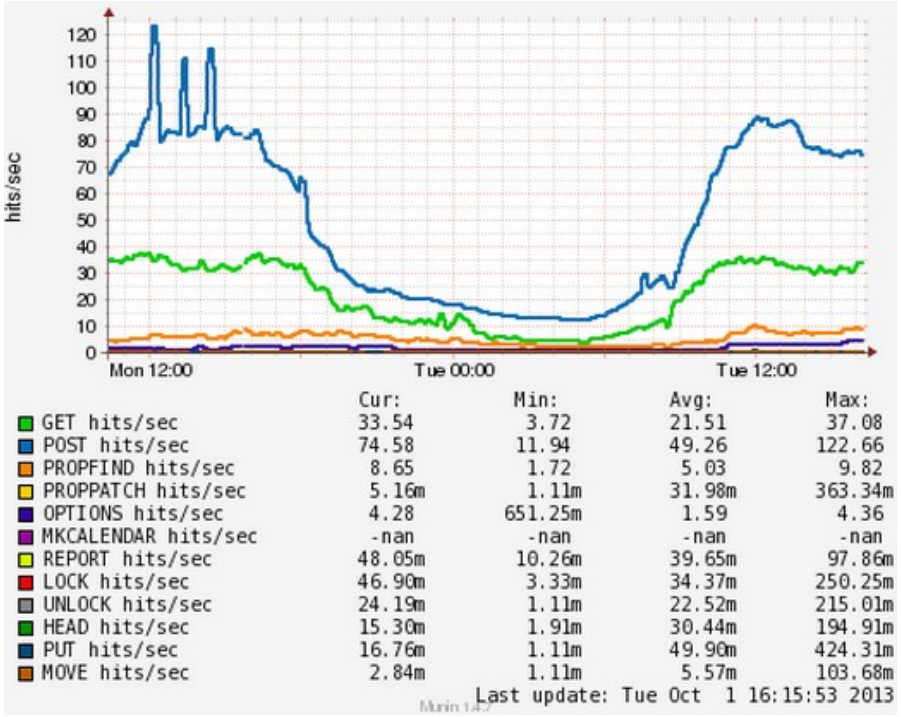
Инструменты отладки в Битрикс

- Отладка метрик компонентов
- Отладка SQL-запросов
- Xdebug
- XHProf
- strace
- gdb

Эксплуатация крупных проектов

Аналитика

- Munin
- Cacti
- Тренды
- Анализ рисков, планирование



Мониторинг

- Nagios
- Zabbix
- «Дымные тесты»
- 24/7
- Xhprof и анализ что случилось

mysql_bp_hitrate	OK	04-03-2013 16:55:21	3d 17h 10m 36s	1/3	OK - innodb buffer pool hitrate at 99.99%
mysql_clients_aborted	OK	04-03-2013 16:56:04	3d 17h 9m 37s	1/3	OK - 0.00 aborted (client died) connections/sec
mysql_conn_time	OK	04-03-2013 16:55:49	0d 3h 44m 52s	1/3	OK - 0.07 seconds to connect as nagios_monitor
mysql_cons_aborted	OK	04-03-2013 16:56:04	1d 17h 18m 3s	1/3	OK - 0.02 aborted connections/sec
mysql_health	OK	04-03-2013 16:55:55	5d 17h 3m 36s	1/3	OK - 3 long running processes
mysql_port	OK	04-03-2013 16:56:02	5d 17h 1m 54s	1/3	TCP OK - 0.003 second response time on port 3306
mysql_replica	OK	04-03-2013 16:55:43	5d 17h 0m 13s	1/3	Slave_IO_Running: Yes: Slave_SQL_Running: Yes: 0
mysql_slow_queries	OK	04-03-2013 16:56:04	0d 15h 35m 37s	1/3	OK - 0 slow queries in 59 seconds (0.00/sec)
mysql_threadcache	OK	04-03-2013 16:55:19	5d 17h 3m 35s	1/3	OK - thread cache hitrate 99.98%
mysql_threads_conn	OK	04-03-2013 16:56:10	0d 1h 20m 30s	1/3	OK - 5 client connection threads
mysql_threads_run	OK	04-03-2013 16:55:55	0d 15h 36m 29s	1/3	OK - 3 running threads

thewatt
СКОРО ДОМОЙ!

«Экзорцизм» в системном администрировании и высоких нагрузках

1. Мы тут еще дружно тусим: https://enterprise.1c-bitrix.ru
2. Размазывание сурового технологического опыта: продакшн Битрикс24 –> «1С-Битрикс: Крупные внедрения»
3. Полезный технический «справочник» по эксплуатации нестандартных кластерных проектов на PHP/MySQL: https://cp.bitrix.ru/~dfSCV (я написал часть глав, наслаждайтесь)

Спасибо за внимание!
Вопросы?

Александр Сербул
@AlexSerbul
Alexandr Serbul
serbul@1c-bitrix.ru