



АВТОМАТИЗАЦИЯ СБОРКИ И ТЕСТИРОВАНИЯ В РАЗРЕЗЕ ЭФФЕКТИВНОГО ПРОИЗВОДСТВА

Олег Ладыгин. «Петер-Сервис»

О чем речь?

«Эффективное производство» – получение прибыли от внедрения технологий.

- Примеры расчета целесообразности
- Дополнительные преимущества автоматизации

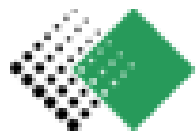
Это еще
один
спорный
взгляд



Кто я?

Олег Ладыгин

Ведущий инженер, менеджер группы
продуктов «Вспомогательные продукты»



PETER-SERVICE

Стоит ли мне верить?

- Создание системы СИ с нуля?
- 4 года автоматизации?
- Продумать, реализовать, внедрить, поддерживать...?

www.billing.ru

900 сотрудников

Сложный софт

ISO 9001:2008



Потому что это работает

Цель автоматизации - получение прибыли

Цели компании

- Ускорение циклов разработки и тестирования
- Соблюдение сроков
- Поднятие престижа компании / соблюдение стандартов
- Обеспечение заданного уровня качества продукта



Личные цели

- Увеличение инвестиций
- Личное развитие, изучение новых технологий
- Самореализация

Ускорение циклов разработки и тестирования

Где можно получить ускорение?

- Разработка быстрее получает фидбэк
- Чистая экономия времени тестирования
- Используем нерабочее время
- Нет переключения между задачами
- Быстрее получаем анализ



Ускорение циклов разработки и тестирования



Немного советов

- Система автоматизации эффективна для **единообразных приложений** – управляйте стандартами
- Минимизируйте **время на обучение** – упрощайте операции и формы
- Цена времени зависит от **даты сдачи** проекта

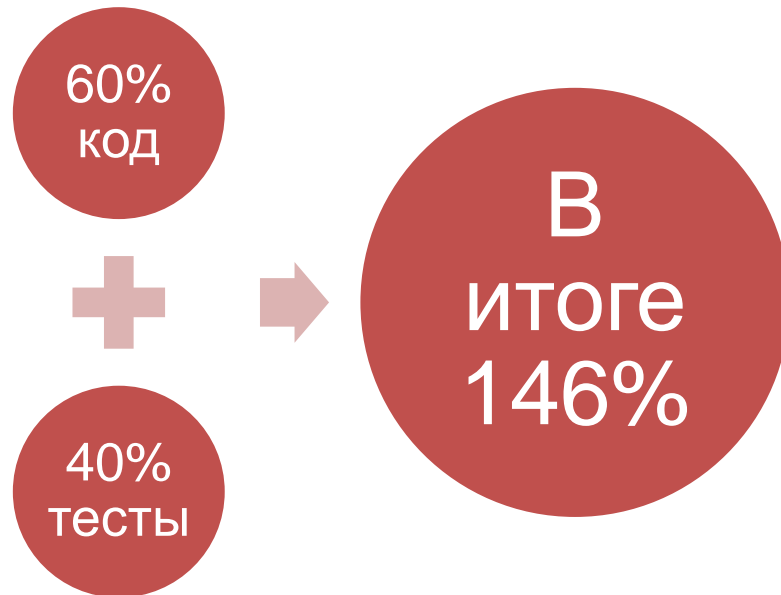
Ускорение циклов разработки и тестирования

Задача	Ручное	Авто	Итог
Сборка	14 мин	1 мин	13 мин
Передача в тест	3 мин	0 мин	3 мин
Установка	30 мин	5 мин	25 мин
Быстрая сборка	30 мин	10 мин	20 мин
Подготовка среды	20 мин	0 мин	20 мин

Итого на 1 продукт в день	81 мин
Итого на 10 продуктов в месяц	37 дней



Соблюдение сроков



**Почему увеличивается
время? Непонятно...**
«загадки во тьме»

**Сколько стоит продлить
срок проекта?**



Соблюдение сроков

Делать

- Единый прозрачный механизм оценки
- Регулярное измерение выполненной работы
- Контроль параметров задач

Ограничить

- Перенос дат в конце цикла
- Изменение приоритетов
- Добавление требований

Знать

- Начинайте с неизвестного
- Сразу исправляйте ошибки
- Вы тратите время на передачу работы

Соблюдение сроков

Идеология расчета

- Переносы сроков за последний год берутся за эталон
- Повышаем точность на 5% в квартал – это **наш выигрыш**
- Виноват – последний менеджер



Цена сроков

- Недополученная прибыль (просрочка/длительность*цена)
- Штрафы от заказчиков (возможно, 1% в день)

Обеспечение заданного уровня качества продукта

~~За обеспечение качества программного €~~
Я не буду рассказывать о качестве!
И за орфографию!

эния!



И промолчу о том, когда надо
останавливать тестирование.



Я уточню только 1 момент –
сколько стоит ошибка.



Обеспечение заданного уровня качества продукта

Измерьте хоть что-нибудь! К примеру...

- Отношение ошибок, найденных на тестировании и после.
- Если SLA на число ошибок – умножаете штраф на вероятность.
- Если выпускаете обновления – считайте затраты их выпуска.
- Сократите затраты на тестирование при достижении заданного порога.

Обеспечение заданного уровня качества продукта

Расчет для SLA

- Ошибка в релизе (квартал) стоит 50 000\$, обычно 0.72 ошибки в релиз
- Затраты на тестирование – 4 месяца (32 000\$)
- Тестируем 2 месяца, вероятность 0.91, выигрыш 4 500\$

Выпуск обновлений

- Статистика – 4 патча в год, всего 3 месяца, 6 000\$ в квартал
- Ускоряем выпуск на 1 месяц, выигрыш 2 000\$ в квартал

Минимизация простоя

Поиск простоев в работе

- Тестировщик ждет: разработку, билд, прогон регрессов, машинные ресурсы
- Разработчик ждет сборку и юнит-тесты



Все ждут

- Если ночной CI сломается, потеря 4 часов
- Ожидание восстановления оборудования и перезапуск



Минимизация простоя

Ищем альтернативы

- Автоматическое развертывание билда
- Диспетчеризация ресурсов
- Автовосстановление при сбоях



Задача	Ручное	Авто	Р	Итог
Поиск билда	2 мин	0 мин	100%	2 мин
Сбой сборки	4 часа	0 мин	5%	12 мин
Прогон тестов	30 мин	0 мин	80%	24 мин
Свободный сервер	2 часа	0 мин	15%	18 мин

Итого на 1 продукт в день	56 мин
Итого на 10 продуктов в месяц	29 дней

Рутинные действия

Рутинная работа – удар по качеству. Либо ее не делают, либо делают плохо.



ов под контроль VCS

и внесение в него

ярных отчетов

тестирования

ий в эталонные БД

заказчика

Рутинные действия



Рутинные действия

Считаем затраты

Задача	Ручное	Авто	P	Итог
Файлы с VCS	10 мин	0 мин	100%	10 мин
Ведение повторяющихся дефектов	1 час	0 мин	50%	30 мин
Прогон тестов	15 мин	0 мин	100%	15 мин
Подготовка чистой среды	1 час	0 мин	30%	20 мин
Ведение эталонных данных	30 мин	5 мин	20%	5 мин
Повторение среды	4 часа	10 мин	5%	11 мин

Итого на 1 продукт в день	91 мин
Итого на 10 продуктов в месяц	41 день

Обучение!

Обучение сотрудников – одна из самых трудоемких операций. В случае выбора между продуктами считайте стоимость:

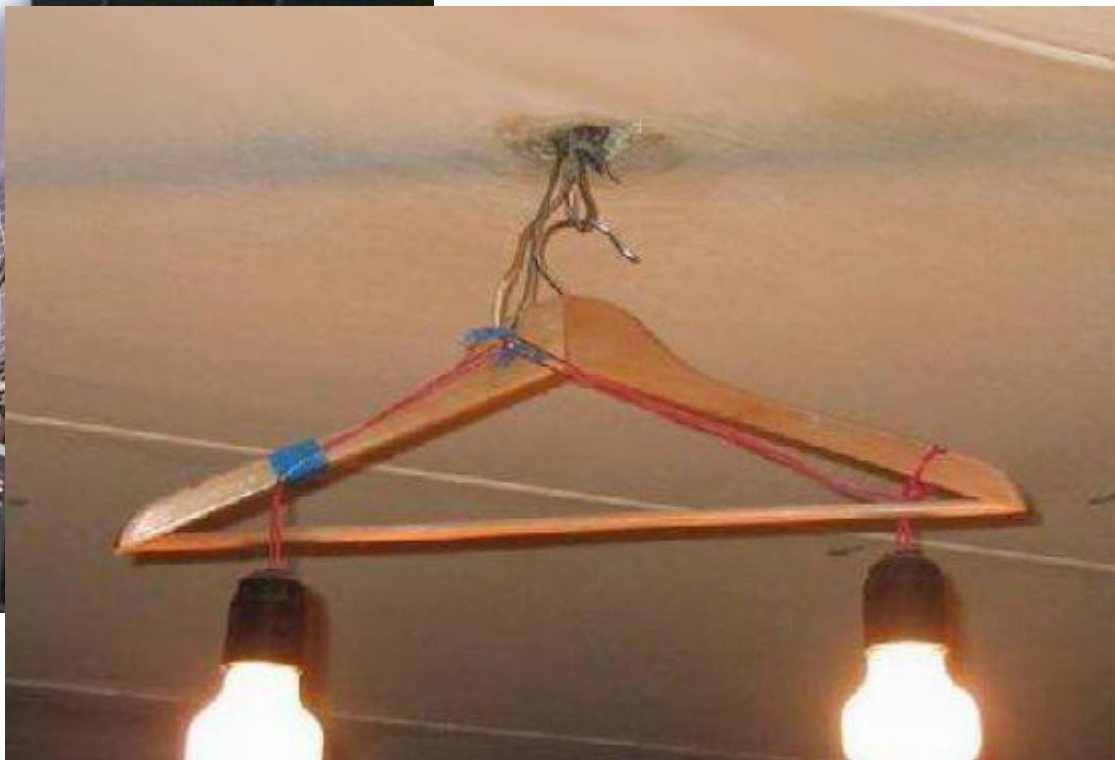
- Обучения каждого сотрудника
- Внедрения системы
- Внесения в нее сборок
- Регистрация всех тестов

А может
быть,
забабахать
свое?



Есть отличные идеи!

Написать что-то свое – **отличная** мысль! Благо есть много успешных примеров...



Итого по времени

Выигрыш в чистом времени

- Всего 81+56+91 минут в день на двух сотрудников
- Итого 1.9 часа на человека в день (1/4 времени)

Дополнительно

- Что-то про сроки?
- Управление рисками и качеством?

Сервисные процессы

- Накопление Знания о продукте
- Взаимозаменяемость (минимум 12% времени)



К борщу еще пампушки

Сначала добавьте дешевые операции

- Проверка орфографии
- Контроль ненормативной лексики
- Проверка стандарта оформления документации
- Проверка корректности html



И что-то посерьезнее



Особо интересные операции

- Генерация справочников процедур кода
- Формирование документации doxygen
- Анализ SLOC (source line of code)
- Анализ измененных процедур

Что они дают?

- Экономят время
- Внедряют качество в код
- Контролируют процесс изменения кода



Задача	Ручное	Авто	Р	Итог
Сравнение результатов между версиями	30 мин	0 мин	30%	10 мин
Поиск ошибочной функции	Нет	0 мин	100%	Восторг
Поиск кода для ревью и актуализации тестов	30 мин	5 мин	100%	25 мин

Не забудем про запасы!

Незавершенное производство – это:

- Непротестированный код - **уныние**
- Излишняя или не сданная функциональность - **обида**
- Тесты, которые мы не запускаем - **бессмысленность**
- Тесты, которые не ищут ошибки - **жадность**
- Дублирующие (лишние) тесты - **страх**



Запасы в тестировании

- Источник ошибок и косвенная потеря денег
- Понижатель настроения



Запасы в тестировании

Уменьшение запасов напрямую не приносит прибыли, но делает производство более гибким.

Запас	Потеря
Непротестированный билд	Время сборки + тестирования
Излишняя функциональность	% дохода соразмерно объему
Недоделанная функциональность	Все время работы
Незапущенные тесты	% найденных этими тестами ошибок / общее число ошибок
Тесты, которые не ищут ошибки	Время создания и поддержки теста
Дублирующие тесты	Время создания и поддержки теста

Вывод

«Нельзя управлять тем, что вы не можете посчитать»
(с)...

- Упрощайте
- Считайте
- Ищите тонкости
в нюансах
- Начинайте
с малого



Удачи!

Viam supervadet vadens / дорогу осилит идущий



Олег Ладыгин oladygin@gmail.com