



Процесс выкладки вёрстки просто, быстро, безопасно

Евгения Фирсова, Яндекс.Деньги





I. Подготовка к выкладке





Что выкладываем?

Параллельные разработка и тестирование – «пакеты задач».
Последовательное обновление production – релиз.

Кодирование в названии:

- код «пакета»
 - ответ на вопрос «что?»
- номер релиза
 - ответ на вопрос «когда?»



Что выкладываем?

Из чего состоит релиз:

- Набор файлов и данных, реализующих функционал;
- Описание релиза:
 - список задач/требований;
 - список задействованных компонент, файлов, источников данных;
 - отчёт о тестировании;
 - список некритичных для выкладки багов;
 - инструкция по выкладке;
 - план выкладки.



Что выкладываем?

Из чего состоит релиз:

- Набор файлов и данных, реализующих функционал;
- Описание релиза:
 - список задач/требований;
 - список задействованных компонент, файлов, источников данных; → **эксплуатации**
 - отчёт о тестировании; → **заказчику, эксплуатации**
 - список некритичных для выкладки багов; → **заказчику**
 - инструкция по выкладке; → **эксплуатации**
 - план выкладки. → **эксплуатации**

Когда выкладываем?

Условия:

- Релиз успешно прошёл тестирование.
 - Найденные баги не являются критичными для выкладки.
- Заказчик принял результат и дал согласие на выкладку.
 - Да, мне это нравится.
 - Да, сейчас правильное время.
- День/время разрешены для выкладки.
 - Возможные ограничения: вечер, пятница, праздники...
- Есть технологическая возможность выполнить выкладку.



Предупреждён – значит вооружён

Служба поддержки должна знать:

- день и время выкладки (когда ждать жалобы?);
- список видимых пользователям изменений;
- список известных проблем;
- список возможных проблем:
 - что именно может произойти;
 - как скоро могут возникнуть проблемы.

Служба эксплуатации должна знать:

- день и время выкладки (не мешаем ли другим?);
- компоненты, реализующие функционал (где логи?);
- список узких мест.



Стакан наполовину пуст

Что может пойти «не так»:

- проблемы в процессе выкладки:
 - сетевые (пропала сеть, пропали доступы, ...);
 - технологические («умер» cvs, «глючит» cms, ...);
 - человеческий фактор (не то выложили, забыли обновить данные, ...);
- проблемы сразу после выкладки:
 - всё сломалось;
 - не сломалось, но и не работает или работает неправильно;
- отсроченные проблемы.



Стакан наполовину пуст

Что можно сделать заранее:

- подготовиться к экстренному откату:
 - технологически подготовиться к выполнению откатов;
 - подготовить пошаговую инструкцию по выполнению отката выкладки:
 - «подготовить» – значит записать;
 - «пошаговую» – значит подробную;
- научиться скрывать функционал от пользователей:
 - предусмотреть альтернативную версию функционала;
 - подготовить «рубильник».



«Рубильник»

«Рубильник» – инструмент для изменения нормального поведения портала:

- изменение внешнего вид страниц;
- настройка редиректа со страниц;
- ограничение доступа к страницам.

При выкладке «рубильник» устанавливается в состояние **off**:

- проверяем целостность портала;
- проверяем доступный только нам функционал.

При тестировании релиза «рубильник» тестируется отдельно.



II. Процесс выкладки





Способы выкладки

Простые способы:

- Пофайловое копирование:
 - + просто
 - возможна потеря целостности
 - медленно
- Копирование в отдельную папку + symlink:
 - + очевидное хранилище предыдущей версии
- Выкладка на временно недоступные пользователям хосты.

Способы выкладки

Правильные способы:

- Выкладка из CVS:
 - + гибкость
 - обязательное участие эксплуатации
 - ? контроль над сохранностью изменений между релизами
- Выкладка из CVS скриптами:
 - + формирование истории выкладок
 - + последняя проверка изменений – просмотра diff'ов
 - + фиксирование изменений – сохранение/рассылка выдачи CVS
- Выкладка пакетами:
 - + надёжно
 - сложный процесс подготовки пакета
 - обязательное участие эксплуатации



Выкладка на несколько хостов

Допустима ли рассинхронизация в процессе выкладки?

- Да:
 - никто не заметит;
 - кто-то может заметить, но зато мы протестируем;
- Нет:
 - подготовка версии в отдельных папках + переключение symlink'ов:
 - по расписанию;
 - по получению распоряжения.

Открытый вопрос – синхронизация времени.



Отслеживание

В процессе и сразу после выкладки желательно:

- разработчикам, тестерам:
 - проверять изменения в ручном режиме;
 - запускать автоматические тесты;
- эксплуатации:
 - анализировать логи;
 - следить за нагрузкой на сеть/железо;
- службе поддержки:
 - мониторить активность пользователей и их жалобы.



III. После выкладки





Проверка результата

Способы проверки:

- ручное тестирование;
- автоматические тесты;
- анализ логов;
- отслеживание изменений в поведении пользователей.

Контрольное время:

- можем 100% проверить самостоятельно ~ 2 часа;
- отложенный эффект:
 - ждём, сколько потребуется;
 - не ждём, но помним.



Всё пропало!

Если после выкладки обнаружались проблемы:

1. Есть возможность откатить изменения? **Откатываем!**
2. Есть возможность воспользоваться «рубильником»?
3. Есть возможность быстро выпустить хот-фикс?



Выводы

Выкладка вёрстки – это:

- **сложно:**
 - требует серьёзной подготовительной работы;
 - задействовано много людей;
- **долго:**
 - процесс может быть «размазан» по времени;
 - оценка результатов – с большой задержкой;
- **чревато:**
 - ошибки могут наслаиваться одна на другую;
 - на исправление уйдёт не меньше времени, чем на разработку.



Евгения Фирсова

<https://money.yandex.ru>

e-mail: saigo@yamoney.ru

