



Whale Rider

Планирование веб-релизов
в условиях многопоточности задач
со скачущими приоритетами

Евгения Фирсова, Яндекс.Деньги



Манифест

Мы пытаемся:

- Радовать пользователей – запусками крупных проектов на фоне стабильного потока мелких улучшений;
- Выполнять требуемое заказчиками – делать то, что просят, и тогда, когда просят.

Мы стараемся:

- трезво оценивать свои возможности;
- не давать ложных обещаний;
- работать наиболее оптимальным и удобным **нам** способом.



Входящий поток – количественные оценки

Пример: 46 релизов / 177 задач за квартал – много?

+ проекты

+ «мелочи»

– багофикс

сколько успеваем за 91 день?

Апрель						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Май						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Июнь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					



Входящий поток – количественные оценки

Пример: 46 релизов / 177 задач за квартал – много?

+ проекты

+ «мелочи»

– багофикс

сколько успеваем за 65 дней?

Апрель						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	X	X
6	7	8	9	10	X	X
13	14	15	16	17	X	X
20	21	22	23	24	X	X
27	28	29	30			

Май						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	X	X
4	5	6	7	8	X	X
11	12	13	14	15	X	X
18	19	20	21	22	X	X
25	26	27	28	29	X	X

Июнь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	X	X
8	9	10	11	12	X	X
15	16	17	18	19	X	X
22	23	24	25	26	X	X
29	30					



Входящий поток – количественные оценки

Пример: 46 релизов / 177 задач за квартал – много?

+ проекты

+ «мелочи»

– багофикс

сколько успеваем за **36** дней?

Апрель						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
		X	X	X	X	X
6	7	8	9	X	X	X
13	14	15	16	X	X	X
20	21	22	23	X	X	X
X	X	X	X			

Май						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				X	X	X
4	5	6	7	X	X	X
11	12	13	14	X	X	X
18	19	20	21	X	X	X
X	X	X	X	X	X	X

Июнь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
X	X	X	X	X	X	X
8	9	10	11	X	X	X
15	16	17	18	X	X	X
22	23	24	25	X	X	X
X	X					



Изменения во входящем потоке

Изменения требований:

- фиксируем;
- оцениваем сложность и примерные трудозатраты
 - «может, на второй этап?»

Изменения внешних приоритетов:

- фиксируем;
- информируем о конфликтующих задачах, каскадно меняем внутренние приоритеты;
- информируем о возможных нарушениях в нормальном процессе выкладок.



Whale Rider

Профессиональная конференция
по управлению интернет-проектами

Как это работает



Требования к людям

Тимлид:

- держит в голове (почти) весь код;
- помнит про все текущие задачи, сравнивает их с планами на будущее;
- мгновенно начинает реагировать на изменяющуюся ситуацию.

Разработчик:

- умеет быстро переключаться между задачами;
- понимает чужой код;
- не стесняется задавать вопросы и просить о помощи.



Входящий поток – качественные оценки

Первичная оценка каждой задачи:

- полнота постановки (неполная ≠ не берём);
- непротиворечивость с другими задачами;
- приоритет по сравнению с имеющимися/ожидаемыми задачами;
- deadline'ы;
- наличие (оптимального) ресурса разработки;
- зависимость от других команд/компонент;
- возможность и тип необходимого тестирования;
- наличие «окна» для выкладки;
- варианты реализации;
- примерные трудозатраты.



Входящий поток – качественные оценки

Первичная «неофициальная» оценка каждой задачи:

- вероятность отмены задачи;
- вероятность значительного изменения требований;
- внутренняя потребность в результате;
- зависимости от текущего/планируемого рефакторинга;
- вероятность ошибки в реализации/тестировании;
- допустимость отката выкладки.



Пакеты и релизы

Параллельные разработка и тестирование – «пакеты задач».

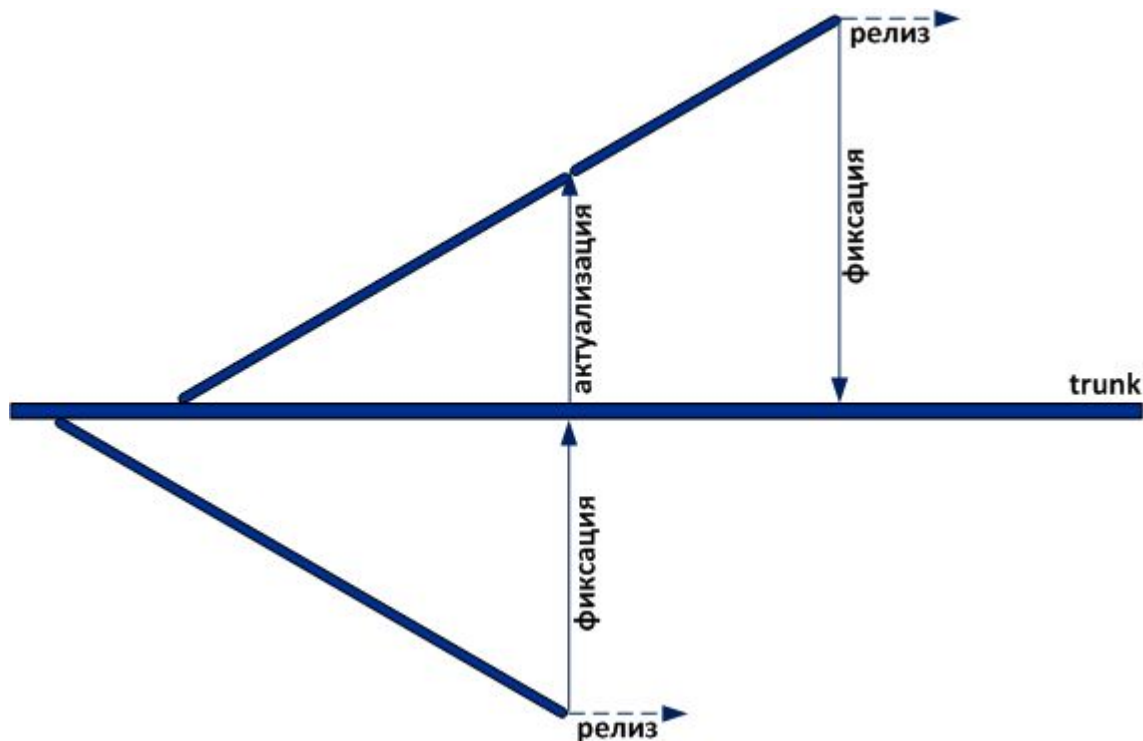
Последовательное обновление production – релиз.

Кодирование в названии:

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

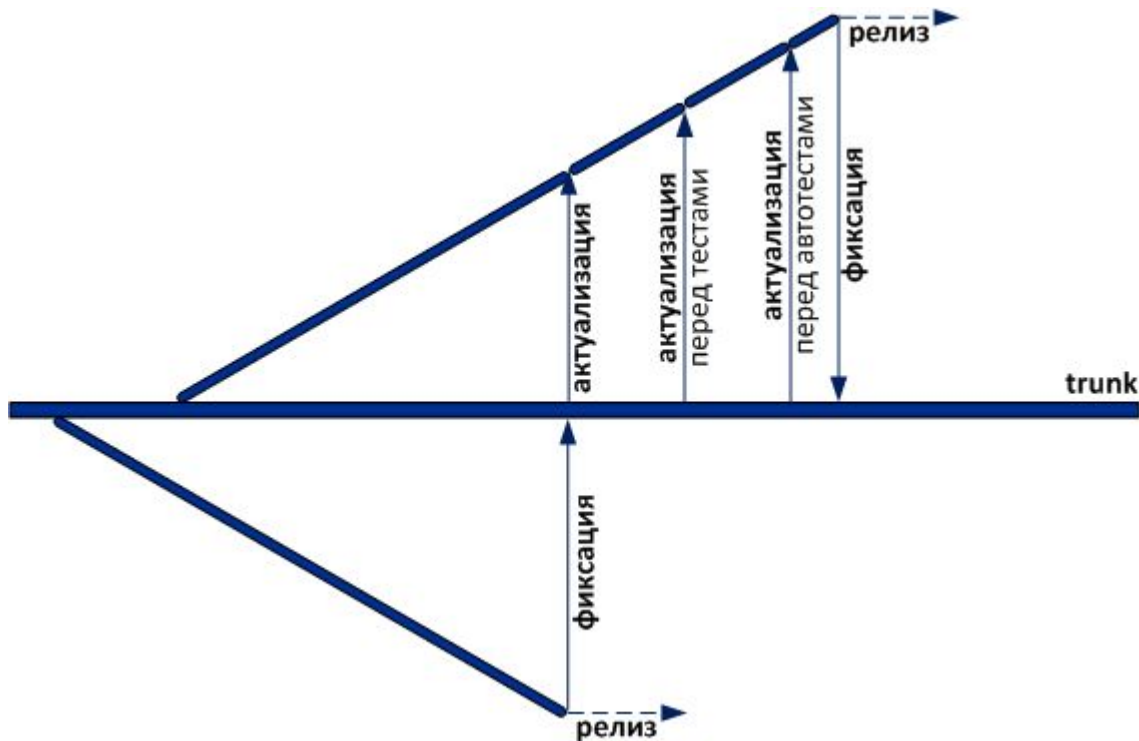


Разработка – пакеты в CVS





Разработка – пакеты в CVS





Узкие места

Потенциально проблемные моменты:

- логические ошибки при актуализации веток;
- повторное ручное тестирование;
- долгий рефакторинг;
- «реанимация» веток.



Базовые принципы

Условия работы конвейера:

- не «мариновать» собранные пакеты;
- при планировании компенсировать неравномерность выходного потока (календаря выкладок) ;
- оставлять время на исправление ошибок;
- знать о ещё непоставленных задачах;
- соотносить свой темп с командой тестеров.



Когда ~~начинать~~ заканчивать разработку

Откладываем начало работ, если:

- пакет не может сразу уйти в тестирование;
- после выкладки другого релиза заведомо предстоит переделка;
- параллельно идёт долгий рефакторинг;
- выгоднее тестировать вместе с задачами, постановка которых ещё не готова;
- известно, что разработку придётся прервать ради других задач.

Не откладываем:

- хотфиксы;
- «дешёвые» мелочи по упрощённой схеме тестирования.



Взаимодействие с тестерами

Помогаем друг другу:

- постоянно держим тестеров в курсе наших планов;
- совместно оцениваем длительность тестирования:
 - знаем, когда текущий пакет будет готов к выкладке;
 - знаем, сколько времени надо отводить на тестирование аналогичного пакета;
 - знаем среднее время проведения автотестов;
- «виртуально» планируем ресурсы тестеров:
 - знаем о специализации каждого тестера;
 - знаем примерную скорость работы каждого тестера.



Whale Rider

Профессиональная конференция
по управлению интернет-проектами

Взаимодействие с эксплуатацией

Помогаем друг другу:

- по возможности заранее сообщаем о планируемых релизах;
- планируем совместные действия на случай экстренных релизов;
- интересуемся планами и загруженностью админов.



Whale Rider

Профессиональная конференция
по управлению интернет-проектами

Результаты



Пытаемся планировать

- В рамках квартала – считаем будущие задачи по головам;
- В рамках месяца:
 - гарантируем работу над задачей (до момента каскадной смены приоритетов);
 - совместно с тестерами расписываем опорные точки поэтапного тестирования (считая, что баги правятся без задержки тестирования).
- В рамках недели:
 - «выкладываем завтра».



Критерии оценки

- Время подготовки релиза: от 5 минут;
- Минимальный временной диапазон между двумя последовательными релизами: 10 минут;
- Оценки входящей задачи: от 5 минут до трёх часов.