Воркшоп «Карпаччо из слона»

Руководство фасилитатора

Авторы: Хенрик Книберг и Алистер Кокберн, 2013-07-11

Перевод на русский: Игорь Филипьев, 2023-06-28

Что это такое?

От авторов.

Вы читаете руководство к воркшопу по мелкой нарезке пользовательских историй.

Это отличный способ для разработчиков программного обеспечения попрактиковаться и научиться разбивать истории на действительно тонкие вертикальные ломтики. Он также приводит к интересным дискуссиям о качестве и технических практиках.

Упражнение было придумано Алистером Кокберном. Мы совместно применяли его несколько раз и призываем людей использовать его повсюду.

Это подробное руководство по фасилитации, в том виде, как его проводит Алистер, плюс некоторые незначительные адаптации от Хенрика.

Упражнение занимает 90-120 минут и хорошо масштабируется. Обычно мы делаем это с группой в 10-20 человек, но также у нас был опыт на группу в 30 человек.

Хронометраж

Лучше всего заложить 2 часа. Также можно уложиться и 1,5 часа, но тогда все проходит в небольшой спешке.

Тайминг

- 10 минут на прогрев группы: обсуждение текущего опыта использования историй и какие сейчас команда использует пользовательские истории
- 10 минут на теорию: обсуждение ценности пользовательских историй.
- 20-30 минут на подготовку беклога
- 40 минутный на кодинг
- 15-20 минутн на дебриф и рефлексию

Можно пропустить часть кодирования и просто попрактиковаться в создании бэклога. Но это не так весело, и вы также упускаете интересное обсуждение качества кода в конце.

Подготовка к воркшопу

- Убедитесь, что в каждой команде есть ноутбук для практической части.
- Убедитесь, что в помещении есть шнуры питания

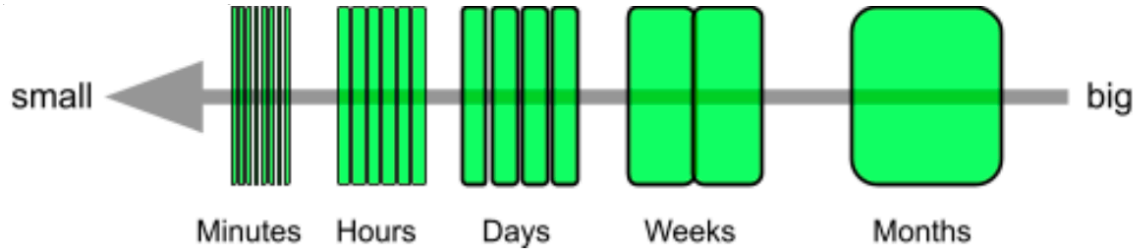
- Интернет на самом деле не нужен для проведения семинара , но полезно иметь возможность просматривать материалы во время программирования.

Фасилитационная карта воркшопа

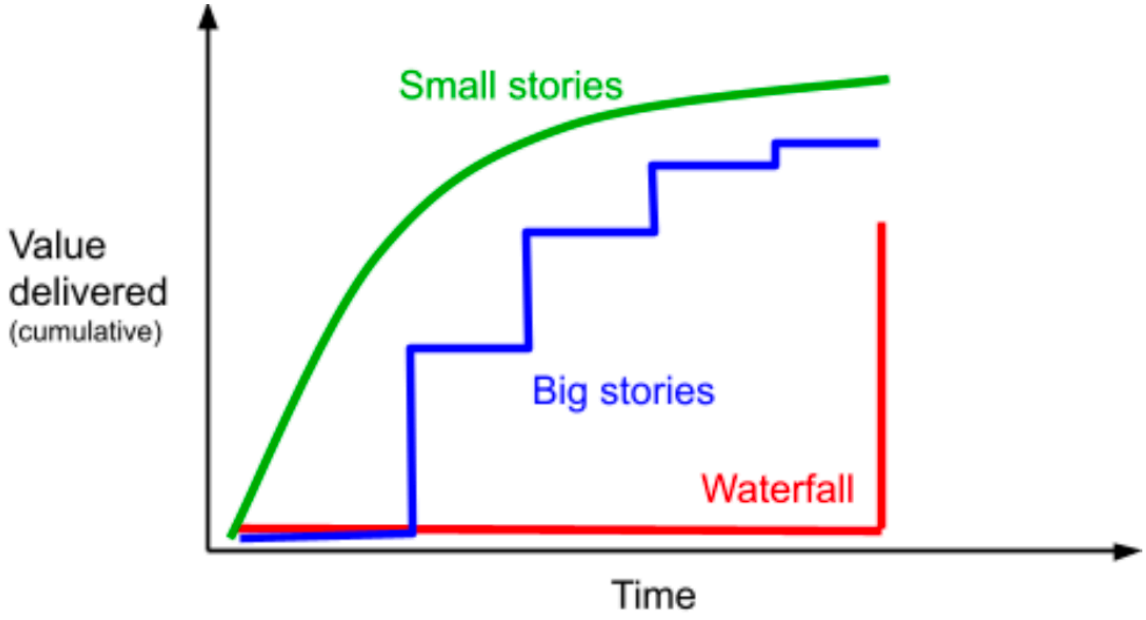
Тайминг 10 минут.

Спросите у группы:

1. Насколько велики ваши истории? задачи? совершает? (укажите в их размер на горизонтальной линии).



2. Зачем разделять истории? Пусть группа в парах обсудит ответ на этот вопрос. (попросите каждую пару сообщить, спросить, чего не хватает, добавить что-либо из приведенного ниже, чего все еще не хватает).
3. Посмотрите на кривую значений для водопада, больших историй и маленьких историй. Обсудите, почему кривая “маленьких историй” в конечном итоге имеет более высокую совокупную стоимость?

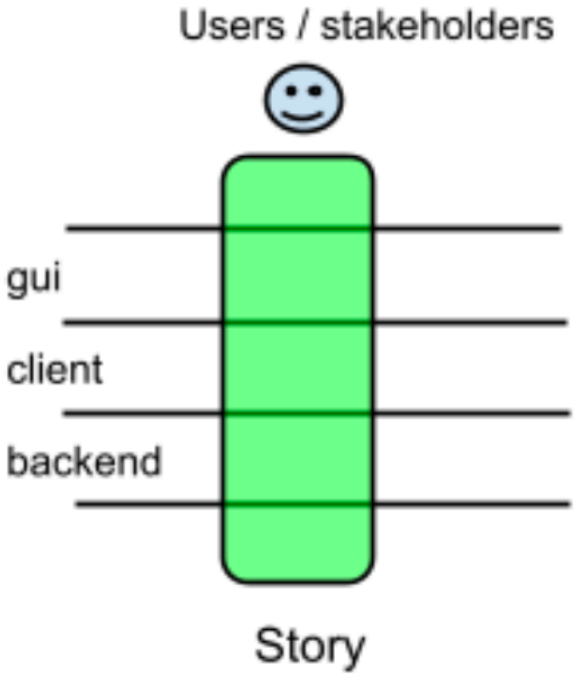


Теория

Тайминг 10 минут.

Цель этого семинара – научиться разбивать истории на действительно маленькие части.

- История – это вертикальная, тестируемая, удобная для пользователя часть выполненной работы. Охватывает несколько архитектурных уровней.



- Нарезка истории – создание более тонких историй (но по-прежнему вертикальных).

Цель: История – несколько дней. Задача – несколько часов. Коммит – несколько раз в час.

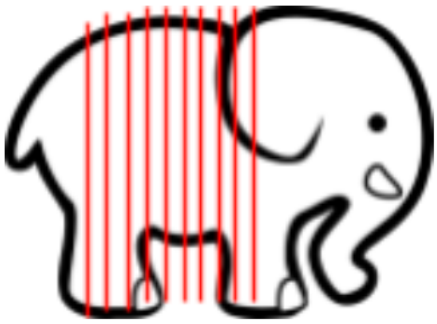
- Получайте обратную связь от интеграции кода быстрее. Поставляйте чаще. Получит более довольных бизнес-заказчиков.
- Больше синхронизации с другими людьми и командами. Лучшая расстановка приоритетов.
- Более качественный продукт раньше. Больше возможностей для бизнеса. Меньший риск.
- Чувствуйте ощущение скорости. Более легкое планирование.

Практика. Что мы будем делать

Тайминг 70 минут.

Решение о том, “насколько мало”, не должно ограничиваться фразой “мы не можем разделить эту историю”. На этом семинаре мы будем практиковаться в экстремальных значениях. Мы сделаем истории такими крошечными, что все, что вы делаете сегодня, по сравнению с ними покажется большим.

- Мы создадим простое приложение за 40 минут (можно в Excel), разделенных на 5 итераций по 8 минут.
- Большинство людей создали бы это приложение за 2-3 части, мы сделаем это за 15-20.
- Карпаччо из слона – это очень тонкие ломтики слона, где каждый ломтик по-прежнему в форме слоника. Вместе они образуют целого слона.



Часть 1. Обсуждение беглога продукта: 20-30 минут.

- Закройте свои ноутбуки!
- Мы создадим розничный калькулятор. Это работоспособное приложение с пользовательским интерфейсом, тремя входами и одним выходом.
- Используйте любой язык программирования, даже формы в Excel.
- Интерфейсом может быть консоль, командная строка, веб, графический интерфейс, что угодно.

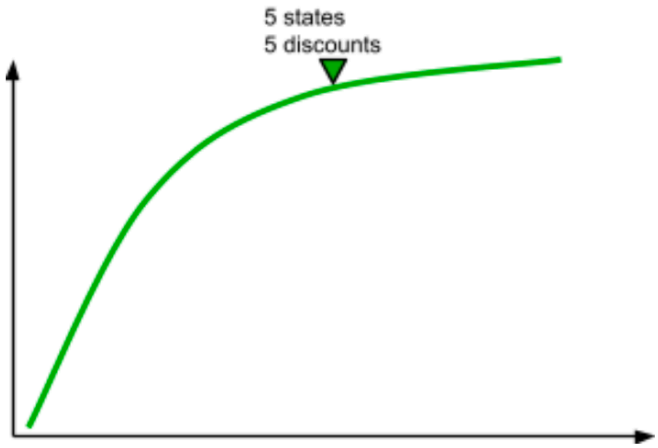
Принимаются 3 вида входных данных от пользователя:

1. Количество товаров.
2. Цена за штуку.
3. 2-х буквенный код штата.

Результат: калькулятор должен выводить общую стоимость всех товаров. Посчитайте скидку на основе общей цены, затем добавьте государственный налог на основе кода штата и цены со скидкой.

Я проиллюстрирую приоритет для нарезки историй на этой ценности кривой.

Наша цель: сделать калькулятор, который считает 5 скидок в 5 штатах.

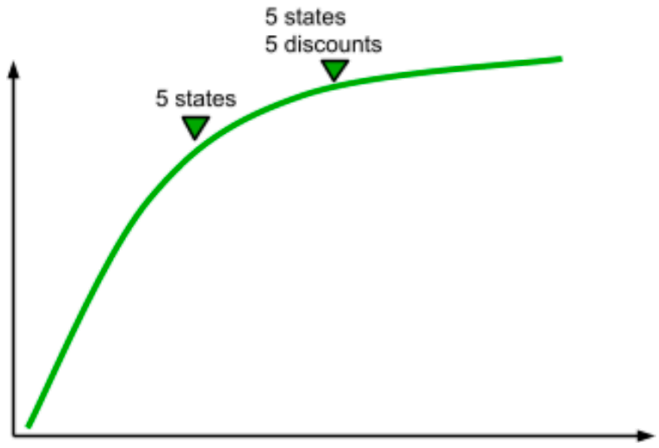


Разбейте эту финальную цель на 10-20 ломтиков-шагов, как вы к этому придете.

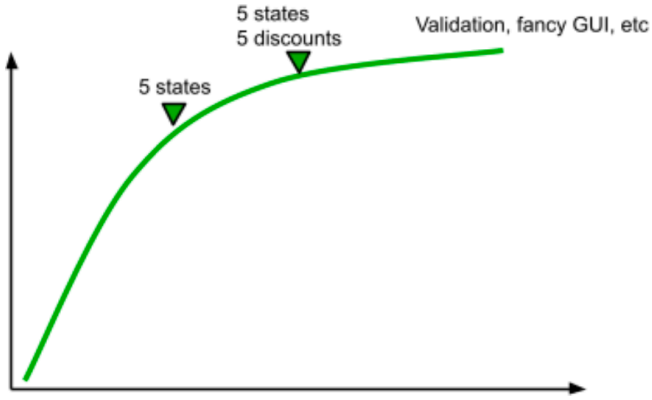
Ломтик принимается, только если он имеет «Пользовательский интерфейс», Поля

«ввода», вывод результата и заметно отличается от предыдущего фрагмента.

Я хочу сначала видеть налог по 5 штатам, прежде чем вы сделаете что-нибудь со скидками. Почему? Государственный налог является требованием закона, скидка - нет.



- Проверку и необычный графический интерфейс нужно делать после реализации налога в 5 штатах и расчета 5 скидок. Нетратьте на это время раньше!



Никаких приукрашиваний! Практикуйтесь всегда сначала создавать самое ценное

Например, до того, как у вас появится расчет налога в 5 штатах не тратьте ни одного нажатия клавиши на код, связанный со скидкой!

Меня не волнует, как далеко вы продвинетесь на этой кривой. Важно то, что вы практикуете микрорезку. Если вы получили результат всего до 3 ломтика, вы зря потратили время и упустили суть этого упражнения.

Разделитесь на группы по 2 или 3 (2 лучше). В каждой группе должен быть по крайней мере один программист с рабочей средой разработки. Возьмите раздаточный материал.

Задание на грумминг беклога

Тайминг 15 минут.

В течение 10 минут распишите весь беклог на бумаге (ноутбуки все еще закрыты).

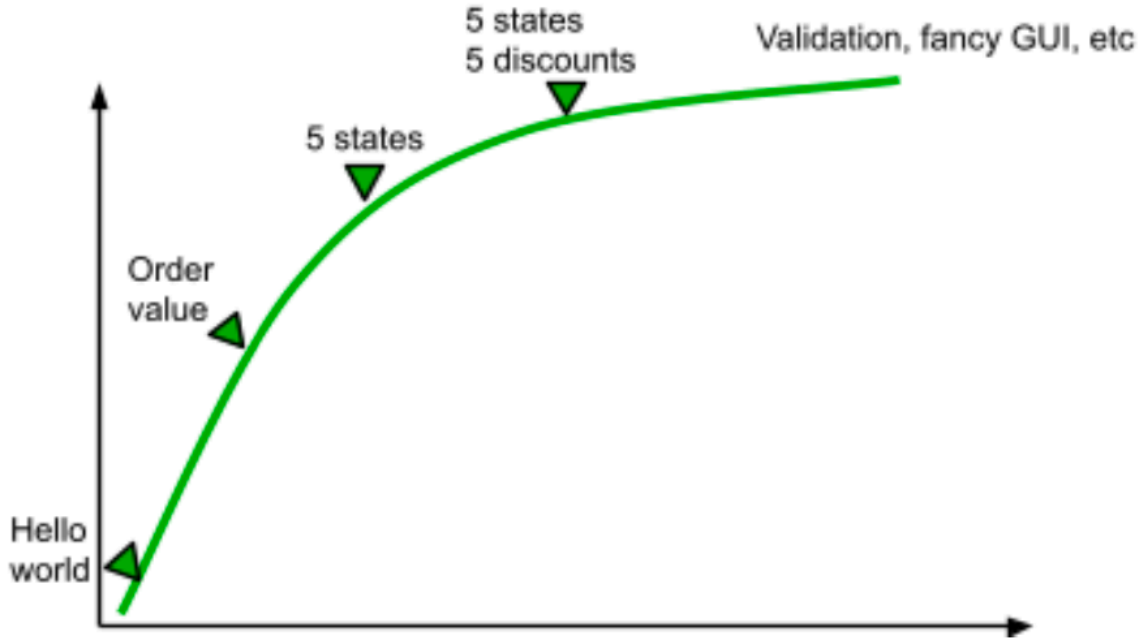
Сделайте 10-20 стикеров с настоящими пользовательскими историями (“ломтиками”), которые помогут вам с нуля создать калькулятор для 5 штатов и 5 видов скидок.

Рекомендации по нарезке пользовательских историй:

- Реализуемо (включая пользовательский интерфейс) за 2-6 минут.
- Заметно отличается от предыдущего фрагмента.
- Более ценный для покупателя, чем последний ломтик (исключение: для первых пары кусочков можно сосредоточиться на снижении риска).
- Никаких ломтиков в виде просто макета или пользовательского интерфейса, структуры данных или тестового примера.

Каким будет ваша первая пользовательская история?

- Все, что больше, чем «Пример, мир!» или «echo input to output» слишком велико).
- Обсудить значение снижения риска реализации и «доставки» «Привет, мир» в качестве первого фрагмента. Ценность = ценность для клиента + ценность знания.
- Обсудите ценность быстрого создания «ходячего скелета» версии приложения (все ключевые архитектурные компоненты на месте и подключены, но «мяса» нет, поэтому у нас есть ходячий скелет).
- Где-то на этой кривой вы должны достичь контрольного показателя "стоимость заказа", где у вас есть 2 входных параметра ("Количество товаров" и "цена за товар") и умножить их. Никаких скидок и налогов штатов еще нет.



- Какой у вас следующий ломтик после общей стоимости? (вероятно, он будет слишком большим, уменьшите его!).

Пример прогрумленного беклога:

- Значение заказа. 2 параметра на входе, 1 на выходе. Никаких налогов и скидок.
- Государственный налог прописан в коде. По-прежнему 2 входных параметра, 1 выходной. Налог штата Юта добавляется автоматически. Теперь мы можем продавать товары в Юте.
- 3 входных параметра (цена за товар, количество товаров, государственный налог %). На выходе стоимость заказа с добавлением государственного налога.
- Пользователь вводит фактический ставку налога, а не код штата. Это дает более простой код (нет внутренней структуры данных для сопоставления кода штата с налоговой ставкой), поэтому мы получаем более быструю доставку.
- 3 входных параметра (цена за товар, количество товара, налог штата), но поддерживаются только два штата. Любой другой штат выдает сообщение об ошибке.

- Добавьте остальные 3 штата. Обсудите, почему на данном этапе существует ограниченная ценность добавления 1 штата за раз:
    - - значения для снижения риска нет, и для всех 3 штатов за одну итерацию требуется работы буквально всего на несколько секунд больше).

Ключевой момент: минимальное количество нажатий клавиш на ломтик! Мы хотим получить максимальную отдачу при минимальных усилиях (Бережливость!)

Остановитесь, посмотрите на результат

В течение 5 минут нарежьте ломтики поменьше. Попробуйте приготовить не менее 15 ломтиков.

Как вы будете работать? (опциональная часть)

Этот семинар в первую очередь посвящен нарезке пользовательских историй, но добавление аспекта технической практики придает ему дополнительную пикантность.

- Как вы будете писать и тестировать свой код? Примите решение и придерживайтесь его. Привлеките себя к ответственности! Я, как фасилитатор, оставляю за собой право преследовать любую группу, которая этого не делает.
    - Вариант 1: TDD. По инструкции TDD. Красно-зеленый рефакторинг. Начните каждый фрагмент с написания теста, который выполняется, но завершается неудачей. Затем сделайте тест зеленым. Затем реорганизуйте код и сделайте его чистым.
    - Вариант 2: красно-зеленый. То же что и выше, но рефакторинг необязателен.
    - Вариант 3: Несколько тестов. Будет напишите несколько тестов, но не для всех фрагментов, и не обязательно сначала проверять.
    - Вариант 4: NFT (никаких чертовых тестов).
- Будете ли вы программировать в паре (часто переключаться) или у вас будет один бизнесмен и один программист (бизнес человек проводит тесты, дает отзывы и улучшает невыполненную работу)? И то, и другое прекрасно.

Часть 2. Практика: 40 минут, 5 итераций по 8 минут.

- В конце каждой итерации я буду говорить: “Время демонстрации!”. Это означает прекратить кодирование и продемонстрировать свое приложение
- Время между итерациями не останавливается! Так что не тратьте слишком много времени на демонстрацию.
- Кричите “пользовательская история” всякий раз, когда заканчиваете пользовательскую историю.
- Вперед! (запустите 8-минутный таймер. Не забывайте перезапускать его сразу после каждого спринта.

Также внимательно следите за тем, на какой итерации мы находимся прямо сейчас, это легко забыть).

Обзор результатов

Тайминг: 10-15 минут.

- Как много вы сделали? (отметьте приблизительное положение каждой команды на кривой ценности)
    - (типичный результат: некоторые команды получают скидки за последние 5 штатов и 5 скидок. Большинство команд, по крайней мере, получают жестко заданный государственный налог).
- Сколько историй у вас получилось?
- Приемочный тест:
    - Запустите приложение и введите эти значения: Я нахожусь в штате Юта, я покупаю

978 товаров, каждый товар стоит $ 270,99. (или выберите любые значения, которые вы хотите).
    - Расскажите мне о результатах! (запишите результаты каждой команды на доске).
- Не нужно возиться, просто запустите приложение, введите значения и посмотрите, что получится.
- Сравните результаты. Часто результаты отличаются, что забавно. Обсудите возможную причину этого. Некоторые команды не поддерживают десятичные числа для определения цены товара, обсуждают ложные предположения и способы их выявления).

Подведение итогов

- Не программисты: как это было?
- Программисты: на что это было похоже?
    - Как поживает качество кода, насколько вы им гордитесь? (каждый программист поднимает 1-5 пальцев).
    - У TDD программистов должно быть 4 или 5 пальцев, в противном случае они пропустили или неправильно поняли этап рефакторинга в TDD.
    - Оставшиеся программисты покажут 1-2 пальца. Обсудите, в чем здесь кроется потенциальная ловушка?
- Обсудите важность устойчивого темпа разработки и поставки новых пользовательских историй и как разработчики сами несут за это ответственность (один из принципов Agile – команда сама выбирает, сколько нужно выполнить работы).
- Обсудите воспринимаемое давление в сравнении с реальным.
- Есть другие вопросы или мысли?
- Вопрос каждому участнику: чему вы научились? Что будешь дальше применять? Назови один важный вывод из сегодняшнего дня и одну вещь, которую ты будешь делать по-другому в будущем?