Kanban

В основе Kanban-а лежит очень простая идея. Количество незавершѐнной работы (НЗР) должно быть ограничено, и что-либо новое может начинаться только тогда, когда какой-то существующий кусок работы поставляется или – в терминах lean – вытягивается следующим элементом конвейера.

Kanban – это подход к управлению изменениями.

Это не методология разработки программного обеспечения, не жизненный цикл или процесс управления проектом. Kanban – это подход, который позволит привнести изменения в существующий жизненный цикл разработки программного обеспечения или методологию управления проектом

Визуализация потока создания ценностей помогает понять текущий процесс, после чего вы устанавливаете ограничения незавершѐнной работы (НЗР) на каждом этапе этого процесса. А затем вы регулируете поток работ, вытягивая (pull) новую работу, только по соответствующему kanban-сигналу.

Kanban использует механизм визуального контроля для отслеживания того, как работа проходит через различные этапы потока создания ценностей. Как правило, используется или доска со стикерами, или еѐ электронный вариант. В идеале, наверное, лучше иметь и то и другое. Вытекающая из этого прозрачность процесса также вносит свой вклад в изменение организационной культуры. Agile методы хороши для обеспечения прозрачности состояния незавершѐнной и выполненной работы, а также для

генерации всевозможных метрик, таких, например, как производительность (количество работы завершѐнной за одну итерацию.)

Kanban наглядно демонстрирует членам команды и другим заинтересованным сторонам результаты их действий (или бездействия). Таким образом, ранние тематические исследования показали, что Kanban изменяет поведение организации и способствует более активному сотрудничеству на рабочих местах. Кроме того, визуализация узких мест, потерь, отклонений и их влияния способствует обсуждению улучшений, благодаря чему команды очень быстро начинают внедрять улучшения своих процессов.

Kanban, как система вытягивания (pull), способствует тому, чтобы действия как по приоритезации новой работы, так и по поставке текущей работы, выполнялись как можно позже (в соответствии с принципом "точно во время (just in time)"). Обычно команды согласовывают некий распорядок встреч по приоритезации. Эти встречи могут проводиться часто, потому что они, как правило, очень короткие.

Они должны дать ответ на очень простой вопрос – что-то вроде такого: "С последней встречи освободилось два слота. На сейчас оперативное время составляет 6 недель до поставки. Выберите 2 задачи, которые вы больше всего хотели бы увидеть реализованными через 6 недель?"

Эффект получается двойной. Простой вопрос, как правило, позволяет быстро получить качественный ответ, и встречи не затягиваются. А характер вопроса приводит к тому, что активности по выбору следующей работы откладываются до последнего момента, когда ещѐ не поздно принять решение.

Это повышает нашу гибкость за счет управления ожиданиями, сокращения оперативного времени (от принятия задачи в работу до поставки), и отсутствия переделок, так как минимизируется вероятность того, что приоритеты изменятся.

Последнее, что хочется добавить про Kanban – эффект ограничения незавершѐнной работы (НЗР) обеспечивает предсказуемость оперативного времени и делает сроки поставки более надежными. Метод "остановки конвейера" при обнаружении препятствий и дефектов, как, оказалось, способствует достижению очень высокого уровня качества и быстрому уменьшению количества переделок.

Scrum и Kanban – инструменты процесса

Инструмент – это то, что используется в качестве средства для выполнения задачи или достижения

цели. Процесс – это то, как вы работаете

Как и любые инструменты, Scrum и Kanban не совершенны и не универсальны. Они не расскажут вам все. Вместо этого они дают определенные правила, устанавливают ограничения и предписания. Например, Scrum предписывает фиксированные по времени итерации и кросс -функциональные команды, а Kanban – использование визуализации процесса на доске задач и ограничение НЗР.

Scrum более директивный, чем Kanban

Можно сравнить инструменты по количеству предписываемых правил, которое в них содержится. Директивный означает "содержащий больше правил, которым надо следовать", a адаптивный – "содержащий меньше правил".

RUP – очень даже директивный метод, – насчитывает более 30 ролей, более 20 мероприятий и более 70 артефактов;

RUP-реализации часто оказываются довольно тяжеловесными по сравнению с Agile-методами, такими как Scrum и XP.

XP (eXtreme Programming) – намного более директивен по сравнению со Scrum-ом. Эта методология включает в себя большую часть Scrum-а + кучу весьма конкретных инженерных практик, таких как TDD (разработка через тестирование) и парное программирование.

Scrum менее директивный, чем XP, так как он не навязывает использование никакой конкретной инженерной практики. Scrum более директивный, чем Kanban, поскольку он предусматривает такие вещи, как итерации и кросс-функциональные команды.

Одно из основных различий между Scrum и RUP в том, что в RUP вы получаете сразу много всего, но надо избавиться от лишнего. А в Scrum – слишком мало, и надо добавить недостающее.

Kanban оставляет почти все на ваше усмотрение. Единственные ограничения – визуализировать ваш ход работы и ограничить незавершенную работу (НЗР).

Kanban ограничивает НЗР по статусу задачи, Scrum – по итерациям

Таким образом, в Scrum-е незавершенная работа ограничена за единицу времени. В Kanban-е незавершѐнная работа ограничена по каждому из статусов

В Kanban, не более n задач могут быть в состоянии «Текущие» одновременно, независимо от длины итерации (спринта).

Вам надо определиться с ограничениями, которые будут применены к каждому из состояний рабочего процесса, то есть основная идея

заключается в том, чтобы ограничить незавершенную работу для всех состояний рабочего процесса, начиная с как можно более ранних этапов потока создания ценностей (value stream), и заканчивая как можно более поздними его этапами.

Scrum и Kanban являются эмпирическими в том смысле, что ожидается, что вы будете экспериментировать с процессом и заточите его под свои условия. На самом деле, вам даже придѐтся экспериментировать. Ни Scrum, ни Kanban не содержат ответов на все вопросы – они просто дают вам базовый набор ограничений, чтобы было от чего отталкиваться при улучшении вашего процесса.

Процесс эксперимента

Допустим, мы уменьшаем ограничение НЗР, исходя из предположения, что это улучшит наш процесс.

После этого мы наблюдаем, как изменяются такие параметры процесса как ПРОИЗВОДИТЕЛЬНОСТЬ, ВРЕМЯ ВЫПОЛНЕНИЯ ЗАДАЧИ, КАЧЕСТВОБ ПРЕДСКАЗУЕМОСТЬ

Потом делаем выводы и меняем что-то ещѐ, таким образом, непрерывно улучшая наш процесс.

Для этой деятельности придумано много названий. Кайдзен (непрерывное улучшение, в терминах Lean-а), "изучай и улучшай" (в терминах Scrum-а), адаптивное управление процессом

Самое важное тут – цепь обратной связи. Измените что-нибудь => Посмотрите что получилось => Сделайте выводы => Измените что-нибудь снова. Вообще, чтобы адаптировать процесс быстро, цепь обратной связи должна быть как можно короче.

Различия между скрамом и канбаном:

Добавление новых стори в спринт:

скрам - в текущий спринт не добавляются, может разве что добавиться в следующую

канбан - можно добавить/заменить стори из колонки "в планах", но числовые ограничения, накладываемые на нзп должны соблюдаться

Доска

Scrum-доска очищается каждую итерацию

В Kanban-е доска, как правило, штука постоянная – вам не нужно ее обновлять, чтобы начать все сначала.

Доска и временные ограничения

Scrum и Kanban базируются на инкрементальной разработке, т.е. работа разбивается на меньшиечасти.

Scrum-команда возьмется только за те задачи, которые возможно закончить за одну итерацию (согласно критерию готовности). Если задача слишком велика, команда и Product Owner будут пытаться найти способ разбить ее на более мелкие куски до тех пор, пока те не впишутся в спринт.

Kanban-команды пытаются минимизировать время выполнения и обеспечить равномерную загрузку, что неявно стимулирует разбивать задачи на относительно небольшие части. Но при этом не существует никакого явного правила, что элементы должны быть такого размера, чтобы вписаться в определенный временной интервал. На одной и той же доске у нас может быть один элемент, на выполнение которого уйдет один месяц, и другой, на который у

Оба позволяют работать над несколькими продуктами одновременно