

КАНБАН В NEMETSCHEK SCIA

Балансируя взаимодействие между
Кроссфункциональными Командами



Телефон зазвонил в офисе Жан-Пьера Раммана, генерального директора Nemetschek Scia. Он поднял его и услышал знакомый голос: «Иван и восемь других разработчиков уволились». Иван был старшим инженером-программистом, который вместе со всей остальной группой разработки находились в Чешской Республике. Он работал над самым важным продуктом компании: Scia Engineer - программным обеспечением для инженерами-строителями. Иван был автором кода для функционала отчетности¹, которое является настолько важным компонентом любого строительного программного обеспечения, что, если оно нестабильно работает или не точно считает, то инженеры вполне могут перейти на программное обеспечение от другого поставщика.

Сразу после телефонного звонка Жан-Пьер понял, что большая часть ключевого кода для отчетности просто ушла в мусор. Функционал отчетности Scia Engineer уже вызывал проблемы у инженеров по всей Европе. Приложение часто зависало, много раз теряя данные, которые уже были сгенерированы. Каждый сбой означал лишь одно - надо все начинать все заново. Отчет вел себя медленно и был помечен наивысшим приоритетом для восстановления. Но Иван только что ушел.

Это был 2009 год. В мае 2013 года Nemetschek Scia выпустила очередное глобальное обновление своего продукта с полным восстановлением функционала отчетности. Новая версия умела автоматически восстанавливать уже сгенерированные данные, данные, даже в случае сбоя. «Я не думаю, что у Scia Engineer когда-либо еще был такой важный релиз», - говорит Патрик Стюарт, тренер и консультант, который работал с компанией. Потребовалось почти пять человеко-лет, чтобы выполнить необходимое количество работы. Это достижение, которое раньше просто считалось невозможным. Это история о том, как Канбан превратил Scia Engineer из неблагополучного продукта в 2009 году в прогрессивное решение в 2013 году.

Предыстория

В компании было двадцать пять разработчиков и десять инженеров по разработке продуктов (PDE), которые отвечали за разработку Scia Engineer.

Почти все они были расположены в Чешской Республике и объединены в группы в соответствии с определенным доменом знаний - сталь, бетон, механизм принятия решения и т. д. Менеджеры по

продуктам и бизнес-аналитики находились в штаб-квартире Nemetschek Scia в Бельгии.

Вся в команда инженеров Scia имела опыт работы в области гражданского строительства, и каждый из инженеров получил специализированную роль в компании. Будь то разработка, управление продуктом, бизнес-анализ или что-то еще, каждый человек постепенно рос, чтобы занять ту должность, которую

он или она сейчас занимает.

Однако те, кто пришел на место разработчиков, ушедших в 2009 году, являлись исключительно ИТ-специалистами.

Патрик Стюарт занимался консультированием Nemetschek Scia с 2005 года. За эти годы он очень хорошо познакомился с чешскими разработчиками. Патрик говорит: «Я ездил в Прагу, чтобы встретиться с разработчиками много раз.

1. Отчетность является основной функциональностью в программных продуктах для проектирования и анализа крупных строительных проектов, таких как мосты и небоскребы. Отчет должен быть подробным и абсолютно точным. Он служит доказательным аргументом для разрешительных органов и страховых компаний, что строительство не провалится.

Каждый раз я замечал одно и то же: когда мы шли с ними ужинать, ужин затягивался до поздней ночи. Ребята долго беседовали о том, как и что мы должны делать в нашем продукте с точки зрения его разработки. Они просто не могли отпустить эти мысли и отдохнуть.”

Он говорил, что, поскольку они были инженерами-строителями до того, как стали разработчиками, их всегда волновал вопрос о том, как сделать продукт лучше.

Scia Engineer - это программное обеспечение для проектирования и анализа крупных строительных проектов, которое появилось в середине 90-х годов. Изначально он назывался Esa Prima Win. С тех пор было выпущено несколько версий, прежде чем он получил новое название в начале 2000-х годов. Код для этого продукта никогда не подвергался полной переработке, а это означает, что очень старый код все еще используется в более новых версиях. Каждая версия Scia Engineer выпускалась всего один раз в год.

«Вы просто добавляете новый функционал и смотрите, не разрушит ли он все здание», - объясняет Крис Ван Лук, менеджер по качеству в Nemetschek Scia. Это наукоемкий программный продукт, и он требует определенного процесса от анализа до тестирования.

Первая необходимость в изменениях

Еще в 2005 году у руководства Nemetschek Scia возникла разрастающаяся проблема. Релизы для Scia Engineer обычно выходили с огромными задержками (до года), если они вообще появились. Руководство считало, что основной причиной задержек могло быть

использования старого кода из предыдущих версий.

Патрика Стюарта привлекли для проверки качества кода. Он пообщался с такими людьми, как Крис Ван Лук и другими сотрудниками компании и стало очевидно, что качество старого кода не является основной причиной проблем.

Фактически причиной задержек было то, как управление продуктом и разработка взаимодействовали друг с другом: процесс принятия решений о приоритетах того, что будет разрабатываться (или не разрабатываться), замедлял процесс. Таким образом был смещен фокус с переработки кода на пересмотр процесса разработки и управления продуктами.

В 2009 году, когда Патрик услышал, что Иван и еще восемь человек уходят, он был просто сбит с толку. Что пошло не так и почему? Скрам, популярный процессный подход, был принят компанией еще в 2005 году. Он использовался для улучшения рабочих отношений между продуктовыми менеджерами и разработкой, а также для сокращения задержек. Скрам должен был предотвратить возникновение проблем, а не преувеличить их количество.

До внедрения Скрама, Scia Engineer работали в режиме одного релиза в год. Поскольку все в компании имели опыт работы в области гражданского строительства, все чувствовали себя одинаково квалифицированными, чтобы иметь право голоса при определении приоритета, какой функционал должен быть разработан и включен в ближайший выпуск продукта. Индивидуальные обязанности никогда не были определены очень четко, и без четко выделенных ролей продуктовыми менеджерами и разработчиками считали, что знают,

что лучше всего взять в работу.

Эти две команды, которые были не просто физически распределенными друг от друга, но и культурно разными, достаточно интенсивно придумывали различные идеи и часто меняли свое мнение о развитии продукта. Активно участвуя в принятии продуктовых решений, разработчики старались выделять достаточно времени для разработки. И все равно, курс продуктовой разработки и приоритеты менялись слишком часто. Многие идеи были начаты, но не доделаны. Задержки накапливались до такой степени, что функционал выпускался на год позже запланированной даты. Жан-Пьер и остальная часть руководства искали объяснения такой ситуации и решения.

Обзор процессов компании показал необходимость большей гибкости. Годовое планирование было неэффективным. Вместо того, чтобы добиваться улучшения, он отвлекал всех и создавал трения между отдельными членами команд. Люди вели себя очень похоже на инженеров-строителей, в сложном и дорогостоящем мире строительства, переделки и модификации непомерно дороги и требуют много времени, а безопасность имеет решающее значение. Разработка программного обеспечения страдала, во-первых, от такого мышления, а во-вторых, от недостатка внимания.

Оценка и разработка крупных проектов существующей функциональности, были кошмаром для продуктовых менеджеров и команд разработки. Подобных проектов на самом деле избегали, потому что люди были слишком не уверены, что смогут их сделать за вменяемый срок, хотя проекты уже были декомпозированы и оценены. По факту, запросы из разных

проектов были смешаны в один план разработки. Запросы шли из нескольких источников: руководство компании - стратегические запросы, служба поддержки - улучшения и исправления ошибок, рутинные задачи по обслуживанию продукта, обеспечению совместимости с обновлениями операционной системы.

...и пришел Скрам

В 2005 году, наше решение состояло в том, чтобы ввести два основных изменения: выстроить процесс, который бы улучшил и структурировал поток работы, а также установить здоровые ограничения, которые четко определяли бы обязанности людей. Это должно было сфокусировать внимание каждого на конечном результате. Мы надеялись, что в результате люди смогут составлять реалистичные планы и сделать разумный график выпусков продукта. Скрам с его итерациями, а также четким распределением ролей, казалось, хорошо приспособлен к такого рода проблемам.

Вероятно, никто не мог ожидать, что эта целостная стратегия разработки продукта скрывает зловещий потенциал, который может стоить компании некоторых из ее самых талантливых и активных разработчиков в 2009 году.

Благодаря Скраму удалось разделить обязанности разработчиков и продуктовых менеджеров. Расстановка приоритетов и принятие продуктовых решений были оставлены исключительно команде продуктовок в Бельгии. Это ограничение было неизбежно. Если бы слишком много людей привлекались в определение приоритетов и постановку целей, разработка продукта продолжала бы терять фокус. Существовала

серьезная опасность, что в конечном итоге мы потеряем рыночную позицию.

Разделение ролей было предпринято в попытке организовать более быструю доставку с более высоким качеством. Продуктовые менеджеры считались теми, кто мог определить, что лучше для продукта, и передать эту информацию разработчикам. Они должны устанавливать цели, направление развития и превращать это все в пользовательские истории и задачи для разработки.

Продуктовые менеджеры составляли беклог продукта, а уже из него команды разработки могли просто взять то, что было вверху и постараться реализовать этот функционал. Работа с приоритетами больше не будет частью обязанностей команд разработчики.

Обязанности между разработкой и продуктовым менеджментом мы разделили, следующим шагом было введение итераций с фиксированной длиной. Такие итерации необходимы для того, чтобы команды разработки могли реализовывать функционал по краткосрочным целям и могли часто что-либо показывать бизнесу.

Такие итерации в Скраме называются спринтами и учитывая нашу специфику работы мы начали с шестинедельных спринтов. Объем работы на ближайший спринт обсуждался и планировался на специальном собрании - Планировании Спринта, которое мы у себя назвали Собранием по Запуску Спринта.

В следующие шесть недель команды разработки должны были реализовывать запланированную работу. Первые четыре недели спринта они должны были делать функционал, последние

две недели стабилизировать его, исправлять ошибки и заниматься интеграцией. Каждое утро были конференс-колла между PDE и разработчиками. На этих ежедневных созвонах команды разработки рассказывали о том, что они сделали накануне, что они планировали сделать сегодня, и столкнулись ли они с какими-либо проблемами.

Таким образом, каждый отдельный член команды получал представление о том, как идет прогресс в реализации. Все понимали, как они приближаются к окончанию итерации и успеют ли сделать запланированное, и мы надеялись, что это поможет делать выпуски продукта чаще.

По своему определению, Скрам призван стимулировать совместную работу и объединить группу людей в команду, чтобы работать как единое целое для достижения общей цели. Внедрение Скрама в Scia Engineer было направлено на то, чтобы команда продуктовых менеджеров была вовлечена в процесс разработки, а не только в целеполагание и планирование. Считалось, что им необходимо наблюдение за сквозным рабочим процессом для достижения качественного результата.

В течение первых нескольких лет перехода к гибким процессам Патрик столкнулся с большим сопротивлением к Скраму внутри компании. Тем не менее, метрики показывали, что время выполнения запросов улучшилось. Или нам так казалось.

Дело в том, что в глубине души, разработчики в Чешской Республике становились все более несчастными. После отделения от продуктового менеджмента, разработчики были лишены возможности влиять на содержание продукта. Те же самые люди, которые так глубоко заботились о продукте, что обсуждали его до позднего вечера,

были лишены права голоса. Они потеряли контроль над контентом и вместо этого были заняты рутинной работой, такими как обзор кода или модульное тестирование². И они даже не понимали, почему так произошло. Там, где когда-то была их страсть к действиям, теперь расцветали разочарование и увядание.

По словам Криса Ван Лука, «У разработчиков всегда была фрустрация по поводу отсутствия кого либо из продуктовой команды во время обсуждений продуктовой стратегии. Что если принятые решения не полные, не была собрана и проанализирована какая-то информация и приняты нереалистичные решения или решения не нужные рынку? Чешские разработчики действительно всегда были достаточно энергичными, переживающими за свой продукт. Их энтузиазм увядал, будучи изолированным от продуктовых решений утомительной ежедневной работой. Не получая выплеска, их энергия накапливалась и могла привести к взрыву».

И Скрам стал искрой, которая эту пороховую бочку детонировала.

Опять нам нужны изменения

После того, как девять разработчиков ушли, чтобы создать свою собственную команду, стало ясно, что что-то в рабочем процессе необходимо изменить.

Задержки с поставкой ценности были решены с помощью Скрама, и разделение ответственности было осуществлено, но, очевидно, очень высокой ценой. Вся продуктовая команда и команды разработки сосредоточились

на своей работе и зоне своей ответственности, и это не давало им возможности научиться хорошо сотрудничать друг с другом.

Страсть, которую люди носили с себе, была задавлена занятостью.

Отсутствие сотрудничества привело не только к уходу девяти человек, но и к снижению темпа развития продукта, который перестал развиваться со скоростью, которую требовал рынок.

Однажды в 2010 году Патрик сказал Жан-Пьеру: «Я думаю, что мы должны попробовать Канбан».

Он объяснил: «Это поможет составить общую картину для всех, и это сподвигнет людей говорить друг с другом, не изменяя способ их работы каким-либо образом».

Он пришел к выводу, что сотрудникам нужен определенный механизм для выражения своего мнения. Канбан с досками для визуализации - где все прозрачно и подлежит обсуждению - может стать правильным решением.

Если что-то может вывести компанию из стоящей перед ней дилеммы, это будет ее собственный совокупный талант, реализующий весь свой потенциал. Это произойдет, только если люди почувствуют, что они эмоционально привязаны к продукту.

Эмоциональная привязанность начинается, когда каждый чувствует, что он является частью процесса принятия решений. Поэтому, если нужно было попробовать новый подход, все должны были с этим согласиться. Сами основы Канбана требовали такого постепенного принятия.

Патрик организовал стартовый семинар для выбранной пилотной команды. Основная цель семинара состояла в том, чтобы каждый человек сформулировал проблему,

прежде чем искать решение. Участникам задавали различные вопросы, чтобы выяснить, почему они чувствовали себя неудовлетворенными нынешним статусом-кво.

«Мы говорили вслух о всем том, что нас беспокоило», - вспоминает руководитель группы Питер. «Я помню как у меня в голове что-то перещелкнуло во время семинара. Патрик показал нам различные формы визуализации в Канбане и то, как организовывалась и распределялась работа. Это было так ясно. На этом примере досок я ясно увидел преимущества, которые они могли нам дать. Мы могли бы видеть, что происходит, появлялась бы возможность выявить узкие места или любые другие проблемы с потоком работ. Кроме того, помимо прозрачности, он показал простоту: Канбан не сложен; на самом деле это довольно простой и естественный способ понять процесс. Забавно, что мы никогда не задумывались об этом раньше».

“Канбан не сложен; на самом деле это довольно простой и естественный способ понять процесс. Забавно, что мы никогда не задумывались об этом раньше”

Наблюдая за прогрессом пилотной команды, все больше команд начинали Канбан-инициативны. Но самое большое испытание для Канбана было еще впереди. Жан-Пьер и Scia Engineer должны были переработать приложение для отчетности.

Проекту требовалась преданная команда с общим пониманием цели и того, как ее достичь. Возможно, Канбан-

2. Модульное Тестирование - это технический термин описывающий подход к низкоуровневому тестированию участков исходного кода с помощью написания различных проверочных функций.

доски были ключом к успеху этого крупного проекта.

В конце 2010 года команда из пяти разработчиков, двух продуктовых менеджеров и бизнес-аналитика собралась, чтобы составить план для проекта переработки функционала отчетности.

«Вам не стоит просто делать инкрементальное переписывание функционала. Вам необходимо четкое понимание и фокус на потребностях клиентов и тут все дело будет в дисциплине», - сказал Патрик.

Потребовалось около трех месяцев кропотливой работы, много встреч с клиентами и подготовки макетов, прежде чем реинжиниринг мог начаться. Продуктовые менеджеры подготовили специальный устав для проекта. В концептуальном документе дается приблизительный обзор того, как должны расставляться приоритеты, проводиться анализ, разработка, развертывание и интеграция в течение следующих двух лет, чтобы реинжиниринг

был успешным.

Конечно, был риск. Поток требований, которые необходимо было реализовать, был огромен. Но команда вовлеклась в проект и рассматривала риск как драйвер, а не как препятствие. Менеджер проекта следил за всем процессом.

В марте 2011 года начался реинжиниринг функционала отчетности. У команды был огромный поток запросов и требований, которые нужно оценить и приоритизировать. Все они были помещены в беклог продукта. Каждый, включая разработчиков, мог представить свои предложения.

Процесс реинжиниринга был организован в трех досках: Доска Исследований, Доска Системного Анализа и Доска Реализации.

Доска Исследований описывала процесс проработки запросов функционала к системе отчетности. Продуктовые менеджеры выбрали запрос из беклога (где каждый мог внести свой вклад) и помещали его на Доску Исследований. Каждый запрос становился отдельной

карточкой на доске, и по ходу работы он перемещался вправо.

«Доска Исследований» - это своего рода сортировка и отбраковка. Так и должно быть, потому что поток запросов всегда будет намного выше, чем возможности по реализации», - объясняет Герт Адриаенсенс, продуктовый менеджер Nemetschek Scia.

Большинство идей необходимо отбросить - это была жертва того факта, что возможности по реализации ограничены, - и только тот функционал, который были наиболее необходим для продукта и клиента, должен попадать в дальнейшую работу.

Доска Исследований и прозрачное прохождение этого процесса всеми запросами, помогли людям увидеть, какие идеи должны быть реализованы, а какие нет и почему. С такой информацией, доступной каждому, чувство утраченного влияния стало исчезать.

После того, как определенный запрос успешно проходил через каждый этап Доски Исследований,



Рис. 1 Это Доска Исследований. Каждый запрос попадая на эту доску проверяется на различных этапах на соответствие критериям необходимости для реализации. Каждый этап работы на доске разделяется на столбцы “В работе” и “Готово”. Система сигнализирует, если запрос залежался в столбце “Готово” и никто его оттуда не берет. На этой доске любой запрос может быть отклонен на полпути, потому что его могут посчитать непригодным к реализации.

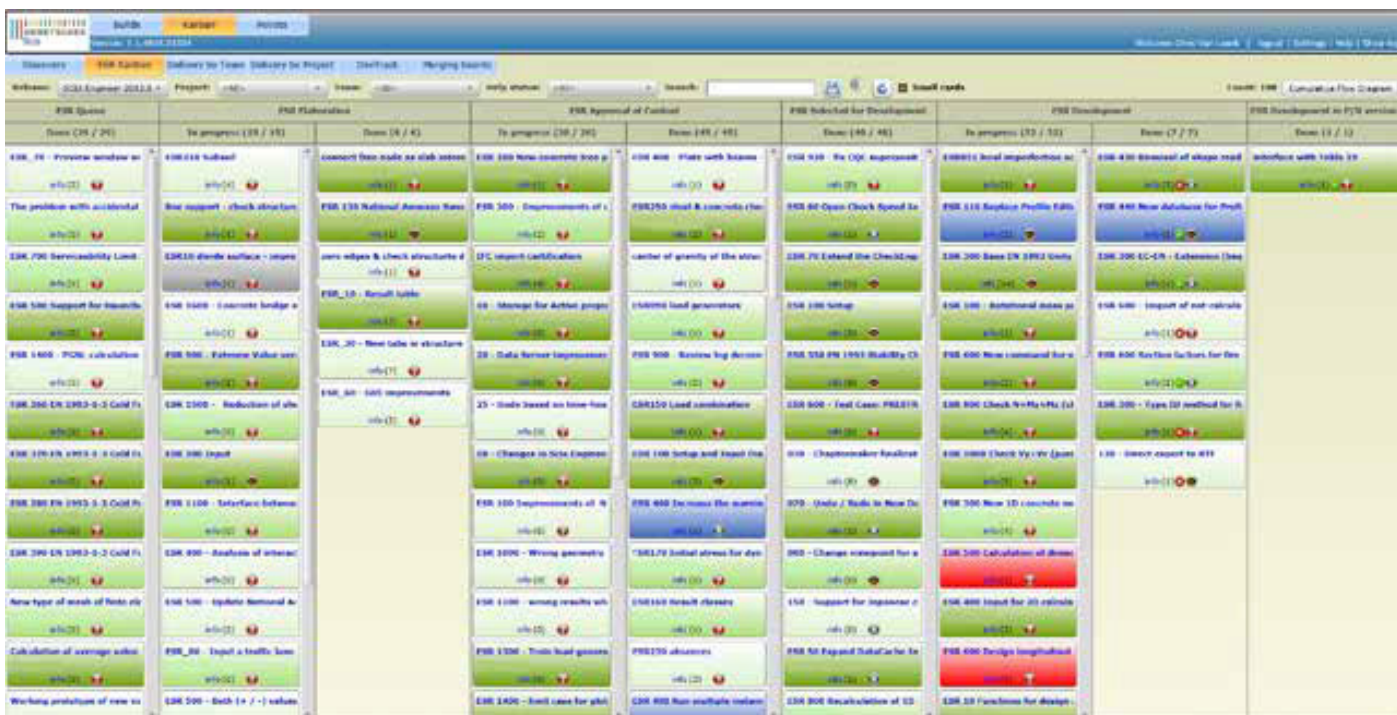


Рис. 2 Доска Системного Анализа. Менеджеры Продуктов и Команды Разработки прорабатывали запросы на этой доске.

он переходил на Доску Системного Анализа. К тому времени продуктовые менеджеры уже признали запрос жизненно важным для продукта.

Доска Системного Анализа давала возможность высказать свое мнение об этом требовании. Эта доска служила платформой для общения продуктовых менеджеров и разработчиков. На этой доске уточнялось, как

вписывается каждый запрос в общий продукт. Благодаря такому сотрудничеству каждая сторона соглашалась с необходимостью реализации запроса, чтобы у разработчиков не было отторжения и сопротивления к реализации на последующих этапах работы.

«Я помню, как я впервые увидел, что все были сосредоточены на том, что

находилось справа, а не слева», - говорит Патрик.

Наконец произошел тот самый момент, когда внимание к завершению запроса было важнее, чем запуск в работу нового запроса или потеря фокуса, в результате которой запросы оставались недоделанными. Поскольку это стало тенденцией, успешный реинжиниринг системы отчетности имел больше шансов к

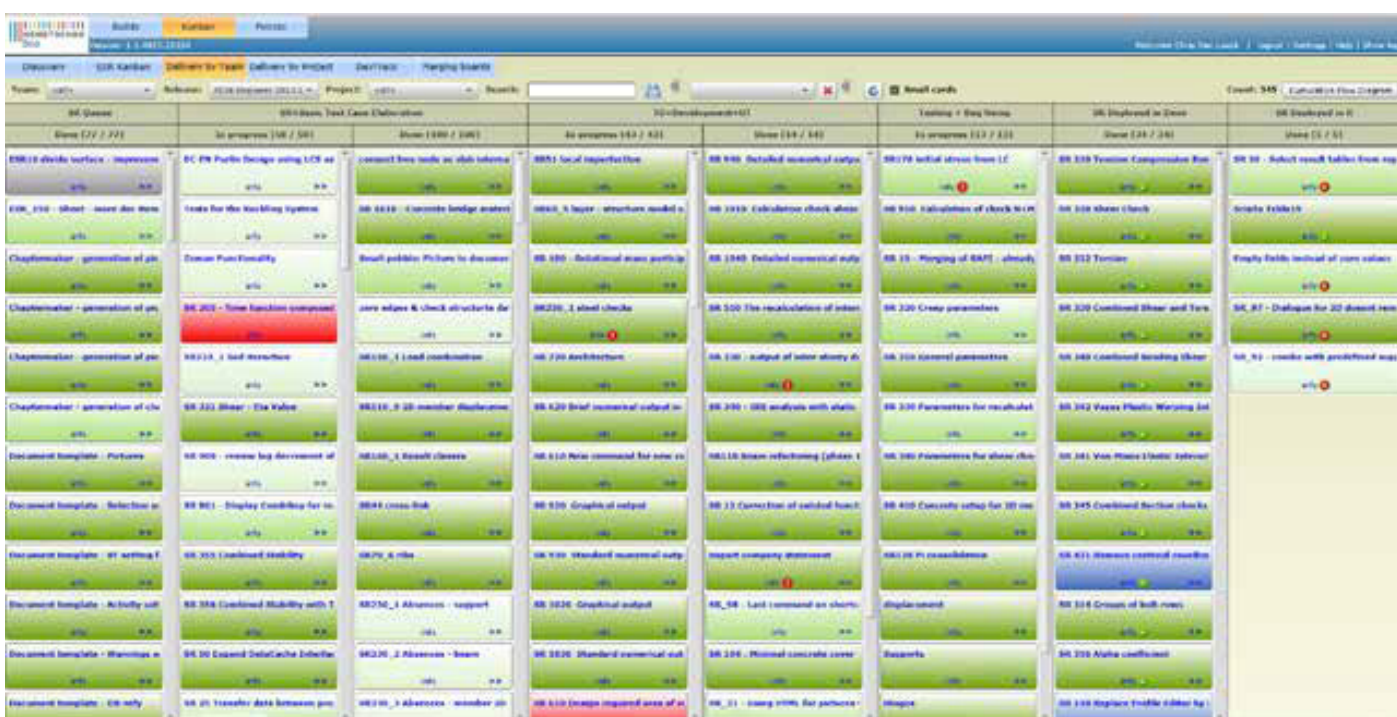


Рис. 3 Доска Реализации. Она показывает процесс прохождения запросов по разным этапам разработки.

завершению.

Как только запрос успешно проходил этот раунд проверки он автоматически переносился во входящую очереди Доски Реализации. На этой доске все запросы декомпозировались на более мелкие рабочие элементы. На этой доске было ограничение, на количество запросов одновременно находящихся в работе, поэтому, команда разработки могла взять в работу какой то запрос только тогда, когда какой-то из прежде взятых запросов был полностью реализован.

Руководитель разработки проводил Операционный Обзор, в котором участвовали все команды разработки, на каждой второй неделе спринта (напомним, что команды работали шестинедельными спринтами). Вместе они разбирали Накопительные Диаграммы Потока (см. Рис. 4), которые были сгенерированы на основе данных с каждой Канбан-доски, анализировали ситуацию и принимали решения об изменении приоритетов, если это было необходимо. Сборка общей картины того, как продвигается работа, позволила добиться существенных улучшений.

«Продуктовым менеджерам не приходилось тратить время на изучение того, все ли прошло гладко и в срок. Они просто смотрели на доску и им было все понятно. А освободившееся время они тратили на то, чтобы придумать как улучшить продукт. Информация из первых рук, от команд разработки, была бесценной для продуктовых менеджеров. И могла определить судьбу любого запроса», - объясняет Патрик.

Все эти простые дополнения к Скрам-процессу решали самые насущные проблемы Scia Engineer. Канбан-доски стали источником информации, с помощью которого



Рис. 4 Доска Реализации. Она показывает процесс прохождения запросов по разным этапам разработки.

можно принимать разумные и своевременные решения. Устав проекта действовал в качестве цита для дальнейшего совершенствования, используя данные досок. И то и другое давали всем ощущение комфорта, спокойствия и сосредоточенности, для того, чтобы продолжать реинжиниринг.находилось справа, а не слева», - говорит Патрик.

Наконец произошел тот самый момент, когда внимание к завершению запроса было важнее, чем запуск в работу нового запроса или потеря фокуса, в результате которой запросы оставались недоделанными. Поскольку это стало тенденцией, успешный реинжиниринг системы отчетности имел больше шансов к завершению.

Как только запрос успешно проходил этот раунд проверки он автоматически переносился во входящую очереди Доски Реализации. На этой доске все запросы декомпозировались на более мелкие рабочие элементы. На этой доске было ограничение, на количество запросов одновременно находящихся в работе, поэтому, командаработки могла взять в работу какой то запрос только тогда, когда какой-то из прежде взятых запросов был полностью

реализован.

Руководитель разработки проводил Операционный Обзор, в котором участвовали все команды разработки, на каждой второй неделе спринта (напомним, что команды работали шестинедельными спринтами). Вместе они разбирали Накопительные Диаграммы Потока (см. Рис. 4), которые были сгенерированы на основе данных с каждой Канбан-доски, анализировали ситуацию и принимали решения об изменении приоритетов, если это было необходимо. Сборка общей картины того, как продвигается работа, позволила добиться существенных улучшений.

«Продуктовым менеджерам не приходилось тратить время на изучение того, все ли прошло гладко и в срок. Они просто смотрели на доску и им было все понятно. А освободившееся время они тратили на то, чтобы придумать как улучшить продукт. Информация из первых рук, от команд разработки, была бесценной для продуктовых менеджеров. И могла определить судьбу любого запроса», - объясняет Патрик.

Все эти простые дополнения к Скрам-процессу решали самые насущные проблемы Scia Engineer.

Канбан-доски стали источником информации, с помощью которого можно принимать разумные и своевременные решения. Устав проекта действовал в качестве щита для дальнейшего совершенствования, используя данные досок. И то и другое

давали всем ощущение комфорта, спокойствия и сосредоточенности, для того, чтобы продолжать реинжиниринг.

Успех!

В мае 2013 года новый функционал системы отчетности был представлен клиентам Scia Engineer. Мы выполнили невероятный объем работы, связанный с этим серьезным обязательством перед рынком. Хотя обратная связь от клиентов еще не собрана в полном объеме, одно уже известно - Scia Engineer и его команды вернули себе право собственности на свой продукт. Они могут гордиться своими успехами. Они вернули себе уверенность в том, что у них есть все возможности, чтобы создать совершенно новый продукт Scia Engineer, который так необходим рынку. Это всего лишь вопрос времени и еще несколько Канбан-досок.

Если хотите узнать больше, свяжитесь Патриком Стюартом по адресу patrick.steyaert@okaloo.com и подпишитесь на его Twitter: [@PatrickSteyaert](https://twitter.com/PatrickSteyaert).

О Kanban University

Kanban University работает над обеспечением наивысшего качества обучения и применения Канбан Метода в сферах интеллектуальной и творческой работы. Наши Аккредитованные Тренеры (Accredited Kanban Trainers™) и Аккредитованные Консультанты (Accredited Kanban Consultants™) помогают компаниям по всему миру в проведении эволюционных изменений с использованием Канбан Метода.

Kanban University предлагает подготовку и аккредитацию Канбан Тренеров, Канбан Консультантов и Канбан Практиков по всему миру.



Kanban University

<https://kanban.university>