ИСПОЛЬЗОВАНИЕ МЕТОДОЛОГИИ KANBAN ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

USE OF THE KANBAN METHODOLOGY IN SOFTWARE DEVELOPMENT

Ершов Тимофей Андреевич, Студент, Хакасский государственный университет им. Н.Ф. Катанова, Россия, г. Абакан

Научный руководитель: Голубничий Артем Александрович, Старший преподаватель кафедры ПОВТиАС ХГУ им Н. Ф. Катанова, Россия, г. Абакан

Ershov T.A. Timofeyershovv@gmail.com

Scientific adviser: Golubnichy Artem Alexandrovich

Аннотация

В статье рассказывается о методологии программной разработки Kanban. Рассмотрена история создания методологии, а также описаны возможности kanban-досок. Приведен краткий перечень электронных сервисов, основанных по большей части на командной работе, которые используют данную методологию в разработке программного обеспечения.

Annotation

This article provides the kanban software development methodology. The history of the creation of the methodology is considered, as well as the capabilities of kanban boards are described. A short list of electronic services based mostly on teamwork that use this methodology in software development is given.

Ключевые слова: методология, канбан, разработка, система, программное обеспечение.

Keywords: methodology, kanban, develop, system, software.

Современная разработка программных продуктов – это командная и кросс-функциональная деятельность невероятно большой сложности. Для правильного и продуктивного взаимодействия участников работы над проектом используются разные модели, абсолютно различные инструменты и методологии, связанные с разработкой программных продуктов. Подходы к разработке определяют успех конечной программы, ведь без правильно подобранного метода разработки будет сложно достичь стабильности, безопасности и устойчивости функциональных особенностей в работе программы. Ведущие эксперты команды-разработчика стараются найти наиболее оптимальный вариант из множества подходов к разработке. В данной статье рассмотрена система Kanban – одна из методологий разработки ПО, достоинством которой является визуализация всех этапов проекта на специальной доске.

Kanban был разработан инженером Toyota из Японии – Taiichi Ohno (в дальнейшем “Ohno”), в конце 1940-х годов. Ohno понял, что может улучшить производственную систему Toyota, включив элементы бережливого производства: вместо создания новых продуктов на основе ожидаемого спроса, система Ohno – Kanban производила и повторно поставляла продукты в соответствии с потребительским спросом. Структура Kanban перевела производственный процесс Toyota из процесса «выталкивания» (продукция продвигается на рынок) в процесс «вытягивания» (продукция создается на основе рыночного спроса).

Основа бережливого производства, которую построил Ohno, опиралась на карты Kanban. Фактически, «Kanban» представляет собой комбинацию двух японских слов: 看 (Kàn), что означает «знак», и 板 (Bǎn), что означает «доска». В Toyota Kanban-карты представляли собой бумажные карточки, которые указывали на необходимость производства нового продукта, детали или

инвентаря и запускали соответственно производственный процесс для этого элемента.

Kanban – это способ помощи командам разработчиков программного обеспечения (ПО), или другим людям не связанным с разработкой программного обеспечения, сбалансировать работу, которую они должны выполнять, с доступными возможностями каждого члена команды. Принцип Kanban построен на философии непрерывного совершенствования, при которой рабочие элементы «вытягиваются» из невыполненных задач в постоянный поток работы. В Kanban приложениях работа отображается на досках проекта, организованных по столбцам. Традиционно каждый столбец доски представляет собой этап работы, к примеру стандартная доска для разработчиков ПО это наличие столбцов «Backlog», «Develop», «Testing», «Review», «Completed». Но доски не ограничиваются только этим набором столбцов, также можно добавлять и другие. Отдельные задачи представляются в виде наглядных карточек на доске, к ним добавляются приоритеты, подзадачи, дедлайны для карточек, и они перемещаются по столбцам, пока соответственно задача не будет выполнена. Почему стоит ли работу над проектом отображать в виде карточек на Kanban-доске? Это наглядно, показывает всем членам команд состояние готовности каждой части проекта или каких-то конкретных задач и удобнее наблюдать на разрабатываемый продукт в целом. В карточках представлена вся нужная информация по определенной задаче, и это всё доступно для любого члена команды: имя человека, который отвечает за данную часть проекта, краткое описание проделанной работы, оценка времени необходимого для решения и так далее. Если рассматривать электронные версии Kanban-досок, там можно прикреплять скриншоты или другие нужные технические нюансы. И конечно, если вся команда может в любой момент времени посмотреть статус решения каждой задачи проекта и по неё дополнительную информацию, то это всё в совокупности приводит к повышению собранности и концентрации на выполнение, и это обеспечивает полную прозрачность работы над проектом [1].

На данный момент, Kanban является одной из самых популярных методологий разработки программного обеспечения, используемых в командах, связанными с agile-разработкой. Несомненно, командам разработчиков, не зависимо от численности, Kanban предоставляет ряд преимуществ, которые касаются планирования работы и обеспечения эффективности выполнения.

– Гибкость планирования разработки. При использовании Kanban, происходит полное погружение команды только в текущий проект. И по выполнении определенной задачи, разработчики берет следующую задачу с верха «бэклога»;

– Визуализация работы – Довольно весомое преимущество методологии –внимание к повышению эффективности команды при работе над отдельной задачей проекта. Когда у команды есть доступ к задачам и их удобная визуализация, проще заметить узкие места в процессе.

– Снижение времени цикла разработки. При использовании данной методологии, можно заметить, что прохождения жизненного цикла задачи, от самого старта работы над ней до ее конечной реализации, сокращается в разы.

– Уменьшение количества узких мест в работе. Чем большее количество задач, требующих завершения стоит перед разработчиками, тем чаще команде приходится переключаться между ними, это же сказывается на сроках завершения проекта соответственно. Поэтому еще один главный принцип методологии состоит в минимальном количестве незаконченных задач в проекте. И если судить по количеству, можно быстро найти в работе проблемные моменты, вызванные недостаточным вниманием людей или нехваткой определенных навыков.

– Непрерывная публикация контента. Непрерывное развертывание (CD) предполагает частую публикацию выпусков продуктов для клиентов. Непрерывная интеграция (CI) - это практика постепенного автоматизированного построения и тестирования кода в течение дня. Вместе они образуют CI / CD, без которого команды разработчиков не могут обойтись, если они хотят быстрее

предоставлять качественное программное обеспечение. Канбан и CI / CD идеально дополняют друг друга, поскольку оба метода основаны на своевременной и последовательной разработке и публикации контента. Чем раньше команда сможет вывести на рынок новейшее решение, тем более конкурентоспособным будет ее продукт. Kanban-команды уделяют особое внимание оптимизации процесса доставки программных продуктов клиентам [2].

Соответственно, Kanban-доски могут быть представлены в физическом и электронном видах. Если рассматривать первый случай, то это обычная доска на стене, на которой начерчены столбцы, а задания прикрепляются на стикерах, которые можно перемещать из одного раздела, в другой. В электронных досках функции аналогичны, но в отличие от физических, ими можно пользоваться удаленно, когда необходимо. И другие сотрудники могут принимать участие в рабочем процессе. Примеры пяти самых популярных электронных Kanban – досок:

– Trello – Инструмент для командной работы, который объединяет разрабатываемые проекты в доски с колонками и карточками. Trello позволяет визуально управлять работой, а командам-разработчиков – продумывать, планировать, управлять своими проектами в совместной и организованной форме [3].

– Asana – Настраиваемый сервис управления контентом рабочего места, которая предназначена для помощи большой команде-разработчиков в достижении необходимых потребностей. Проще говоря, это мощный инструмент помогающий оптимизировать разработку программного обеспечения [4].

– YouTrack – Это веб-сервис для отслеживания проблем и управления проектами. Платформа специально оптимизирована для разработчиков программного обеспечения и agile-команд. Однако абсолютно любая команда

может настроить функционал сервиса под свои нужды, например для бизнес-логики [5].

– MeisterTask – Гибкое приложение для управления проектами, позволяющее управлять проектами большой командой. Основан сервис на системе управления Kanban, но может быть адаптирован и настрое в соответствии с потребностями разрабатываемого проекта или компании [6].

– Devprom ALM – Это модульная платформа для организации работы команды-разработчиков. Она делает более эффективными процессы разработки приложений различной сложности [7].

Система Kanban – это можно определить как просто стикеры приклеенные к доске, где люди подходят берут задачи, выполняют их, и перевешивают стикер в другое место по мере прохождения этапов разработки. Самый простой способ понять, как работать с этой методологией это просто начать применять в своей работе и прогресс будет виден на лицо. Визуализация рабочего процесса над проектом или установка ограничений незавершенных задач, управление потоком разработки, совместная работа вознесут процесс разработки на новый уровень, о чем можно было только думать. Kanban не ограничивает разработчиков – абсолютно любой проект, любой сложности в том числе не связанный с программированием, можно разделить на задачи, определить статусы и этапы, а потом работать с помощью визуализированных выполняемых задач. А выбор должна быть физическая или электронная доска, и какой лучше сервис выбрать для введения электронной доски зависит от команды разработчиков ПО, которые выбрали данную методологию в разработке.

Литература

1. UNISENDER. [Электронный ресурс] // URL: https://www.unisender.com/ru/
2. Kanbanize. [Электронный ресурс] // URL: https://kanbanize.com/
3. Trello. [Электронный ресурс] // URL: https://trello.com/
4. Asana. [Электронный ресурс] // URL: https://asana.com/ru/product
5. JetBrains. [Электронный ресурс] // URL: https://www.jetbrains.com/youtrack/

6. Zapier. [Электронный ресурс] // URL: https://zapier.com/blog/how-to-use-meistertask/
7. Devprom ALM. [Электронный ресурс] // URL: https://devprom.ru/

Literature

1. UNISENDER. [Электронный ресурс] // URL: https://www.unisender.com/ru/
2. Kanbanize. [Электронный ресурс] // URL: https://kanbanize.com/
3. Trello. [Электронный ресурс] // URL: https://trello.com/
4. Asana. [Электронный ресурс] // URL: https://asana.com/ru/product
5. JetBrains. [Электронный ресурс] // URL: https://www.jetbrains.com/youtrack/
6. Zapier. [Электронный ресурс] // URL: https://zapier.com/blog/how-to-use-meistertask/
7. Devprom ALM. [Электронный ресурс] // URL: https://devprom.ru/