

УДК 004.413

***О ПРИМЕНЕНИИ МЕТОДОЛОГИИ КАНБАН В РАЗРАБОТКЕ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ***

Гаязова Е.Э.

магистрант,

Тихоокеанский государственный университет,

Хабаровск, Россия

Стригунов В.В.

к.ф.-м.н., доцент,

Тихоокеанский государственный университет,

Хабаровск, Россия

Аннотация

В статье приводится обзор гибкого подхода к процессу разработки программного обеспечения Kanban. Уделяется внимание основным принципам методологии, отличительным чертам в сравнении с классическими принципами разработки программного обеспечения.

Ключевые слова: методология разработки, программное обеспечение, программная инженерия, Agile, Kanban

***ON THE APPLICATION OF KANBAN METHODOLOGY IN THE
DEVELOPMENT OF SOFTWARE***

Gayazova E.E.

undergraduate,

Pacific State University,

Khabarovsk, Russia

Strigunov V.V.

Ph.D., associate professor,

Pacific State University,

Khabarovsk, Russia

Annotation

The article provides an overview of the flexibility of the Kanban software development process. Attention is paid to the basic principles of the methodology, distinctive features in comparison with the classical principles of software development.

Keywords: development methodology, software, software engineering, Agile, Kanban

Разработка программного обеспечения – сложный технологический процесс, требующий грамотной организации. За всю историю программной инженерии сформировались различные методологии, используемые в разработке сложных программных продуктов. К ним относятся как классические модели, такие как каскадная модель («водопад»), инкрементная и спиральная модели, так и современные гибкие методологии, например экстремальное программирование XP, Agile, Scrum и Kanban.

В 1970 году американским ученым-компьютерщиком Уинстоном Ройсом был составлен документ, называвшийся «Управление развитием крупных программных систем» [6]. В данной работе Ройс представил классический жизненный цикл программного обеспечения – модель Waterfall (каскадная модель, «водопад»). Разработка ПО в данной модели является последовательностью этапов. Переход на следующий, иерархически нижний этап происходит только после полного завершения текущего этапа [3].

Долгое время большинство проектов управлялись в рамках данной каскадной модели. Однако выявленные в ходе использования существенные недостатки модели потребовали вначале ее видоизменения и развития в иные методологии, например, в инкрементную и спиральную модели, а затем и переход к новым гибким методологиям, позволяющим оперативно реагировать на меняющиеся требования, к программному обеспечению у заказчика. В 1993 году был предложен метод разработки Scrum [10], в 1996 году – методология экстремального программирования XP [7], а в 2001 году опубликован «Манифест о гибкой разработке программного обеспечения Agile» [8]. Особую популярность в последнее время получили методологии семейства Agile.

Kanban изначально не был связан с разработкой программного обеспечения. Эта методология берет свое начало в управлении процессом сборки автомобилей [9], предложенной компанией Toyota в 1959 г. В 2005 году Дэвид Андерсон, заново открыл методологию, но уже для разработки программного обеспечения. Он адаптировал систему, усовершенствовал процесс создания ПО. Таким образом Kanban является одной из самых молодых гибких методологий, применяемых в программной инженерии.

Сегодня Kanban используется многими мировыми «гигантами», такими как «Рено» (Франция), «Дженерал моторс» (США), «Microsoft» (США), «Siemens» (Германия) и другие [5]. Анализ опыта фирм Западной Европы, которые внедрили Kanban, показала, что его применение уменьшает производственные запасы на 50%, товарные запасы – на 8 % при значительном ускорении оборачиваемости оборотных средств и повышении качества продукции.

Kanban основан на четырёх основных принципах [1]:

- опора на существующие методы разработки. Kanban начинается с существующих методов разработки и стимулирует в них дополнительные изменения;

- предварительная договорённость о проведении важных изменений. Команда разработчиков учитывает постоянные изменения. Это способ улучшения существующего процесса разработки, однако реализация глобальных перемен несет большой риск. Kanban поощряет небольшие и эволюционные изменения;

- уважение к существующему порядку, ролям и обязанностям;

- поощрение инициативы. Приветствуются проявления инициативы каждого разработчика.

В методологии Kanban, как и в моделях общего семейства Agile предполагается, что при реализации проектов учитываются постоянно изменяющиеся факторы внешней и внутренней среды, присутствует обратная связь между заказчиком и разработчиком. Данный подход стимулирует инженеров и разработчиков экспериментировать, находясь в постоянном поиске новых решений [11].

Одна из целей Kanban в балансировке работы разнонаправленных специалистов внутри команды, чтобы избежать возникновения ситуации, при которой дизайнеры работают круглосуточно, а к разработчикам не поступают новые задачи.

Рассмотрим подробнее принципы работы в командах Kanban. Методология Kanban позволяет различным действиям от ранжирования новой работы, и до постановки текущей работы, реализовываться по принципу «точно во время (just in time)».

Вся команда едина, в системе Kanban отсутствует роль владельца продукта. Бизнес-процесс подразделяется в соответствии со стадиями выполнения определенных задач: планирование, разработка, тестирование, завершение и т. д.

К этой технике, предназначенной для управления разработкой, прибегают как к альтернативе Scrum. Ее особенность в возможностях [12]:

- исключения излишнего накопления задач для команд обеспечивающих качество продукта;

- перераспределения ресурсов (например, программисты после выполнения своего задания помогают с проведением тестов);

- повышения пропускной способности процесса разработки конвейерного потока с запросами на реализацию функций ПО [12].

Вначале работы по Kanban командой определяются этапы выполнения задачи, например: дизайн, разработка кода, тестирование и сдача проекта. Этапы записываются на доске, начиная с простых. Данные доски можно создавать как на настоящих досках, так и в приложениях вроде Jira, Trello и др. По мере заполнения доски, желательно для каждой колонки выставить ограничения по количеству активных задач в данной колонке. Можно выявлять недостатки и предлагать улучшение процесса, например, добавлять новые колонки, изменять формулировку задач или если, например, слишком много задач в разделе дизайна, можно нанять новых дизайнеров и расширить команду (рис. 1) [2].

Данная методология имеет много положительных отзывов от IT-компаний, применяющих её. Так основатель и руководитель Nygger Александр Сергеев выделяет следующие преимущества [4]:

- выявление большой очереди задач на тестирование и как решение – найм второго тестировщика;
- выставление приоритетов задач;
- возможность видеть весь процесс целиком, включая занятость сотрудников;

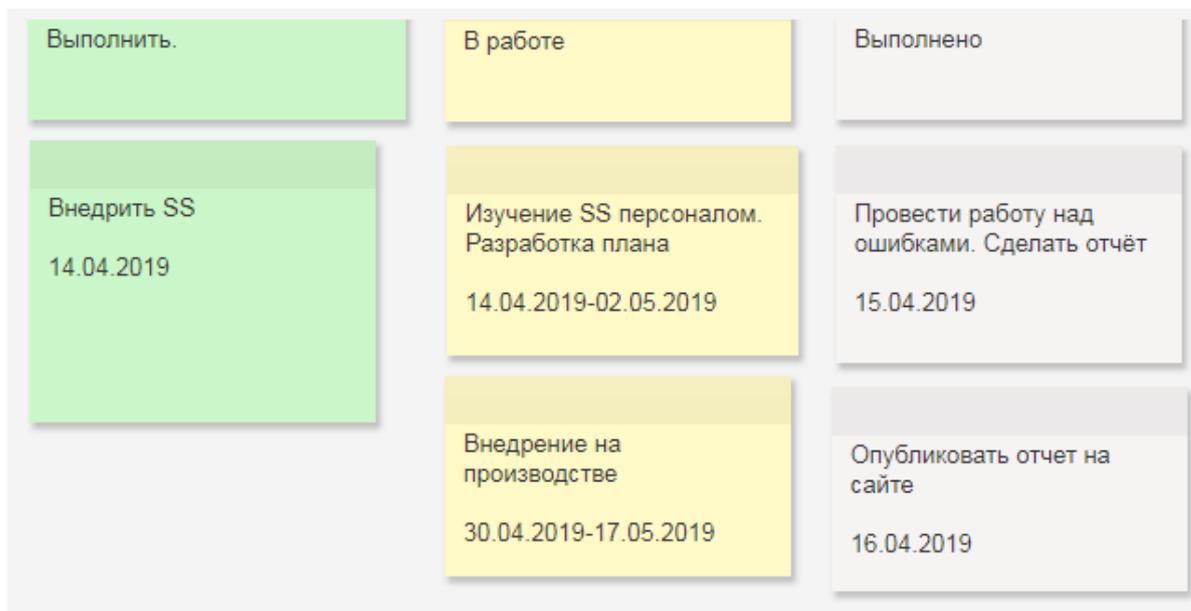


Рис. 1 – Пример самой простой Kanban доски

- гибкость, изменение требований “на лету”, причем без какого-либо сопротивления со стороны программистов;
- отказ от еженедельных отчетов и коммуникаций, издержки на которые забирают порядка 30% времени;
- слаженность команды;
- концентрация на одной задаче.

Несмотря на то, что Kanban является хорошим инструментом для управления потоком задач, имеются и ограничения в данной методологии. Так, количество незавершенных задач в Kanban ограничено. Поэтому если задача по какой-либо причине будет заблокирована, то это может привести к блокировке исполнителей. Если несколько задач будут заблокированы, работа над проектом может быть остановлена.

Каждый метод, используемый при разработке ПО, имеет свои преимущества и недостатки. Поэтому важной является задача выбора такого метода, который наилучшим образом соответствует требованиям и пожеланиям

компаний. Как видно из статьи, Kanban является популярным и успешно применяемым на практике гибким методом разработки ПО.

Библиографический список:

1. Жданов С.А. Экономические модели и методы в управлении. - М.: Дис, 2012 – 176 с.
2. Лайкер Дж. Дао Toyota: 14 принципов менеджмента ведущей компании мира: Пер. с англ. / Под ред. А. Баранова, Э. Башкордина, С. Турко. – М.: Альпина Бизнес Букс, 2005 – 27 с.
3. Орлов С. А. Технологии разработки программного обеспечения: Учебник. – СПб.: Питер, 2002. – 464 с.
4. Сергеев А.С. Канбан в IT: чертова дюжина преимуществ от внедрения Канбан досок для разработки ПО / Сергеев А.С. [Электронный ресурс]– Режим доступа: <https://vc.ru/flood/30834-kanban-v-it-chertova-dyuzhina-preimushchestv-ot-vnedreniya-kanban-dosok-dlya-razrabotki-po> (Дата обращения: 29.04.19)
5. Филиппев И. А. Что такое Канбан-метод – максимально коротко [Электронный ресурс] – Режим доступа: <https://scrumtrek.ru/blog/chto-takoe-kanban-metod-maksimalno-korotko/> (Дата обращения: 29.04.19)
6. Baleviciute, G. 2014. Whitepaper – Scrum vs Kanban vs. Scrumban. Retrieved September 10, 2016. Режим доступа: <http://goo.gl/dkrgGE> (Дата обращения: 29.04.19)
7. Bieliūnas, E. 2014. Scrum-ban for Project Management. Retrieved September 10, 2016. Режим доступа: <http://goo.gl/JgfaaA>. (Дата обращения: 29.04.19)
8. Bittner K., Lo Giudice D., DeMartine A., Mines C., Hammond J. S., Turrisi T., and Izzi M. 2016. Forrester Research – Boost Application Delivery Speed And Quality With Agile DevOps Practices.
9. Brechner, E. 2015. Agile Project Management with Kanban, Microsoft Press.

10. Jeffrey V. S. Краткий курс Scrum. [Электронный ресурс] – Режим доступа: <https://www.osp.ru/cw/2007/09/4038485> (Дата обращения: 29.04.19)
11. Klipp P. 2014. Getting Started with Kanban, Amazon Digital Services LLC.
12. Misevičiūtė, D. 2014. Scrumban: on demand vs. long-term planning. Retrieved September 10, 2016. [Электронный ресурс] Режим доступа: <http://www.eylean.com/blog/2014/11/scrumban-on-demand-vs-long-term-planning/> (Дата обращения: 29.04.19)

Оригинальность 94%