

Т.Г. Григорян, Л.Ю. Шатковский

**МОДЕЛИ ПРОЦЕССОВ ПРИНЯТИЯ РЕШЕНИЙ ПРИ  
ЦЕННОСТНО-ОРИЕНТИРОВАННОМ УПРАВЛЕНИИ  
ТРЕБОВАНИЯМИ В ИТ-ПРОЕКТАХ**

Предложен подход к принятию решений при управлении требованиями, учитывающий турбулентную среду ИТ-проектов и необходимость постоянного вовлечения заинтересованных сторон при определении содержания и расписания проекта. Подход позволяет максимизировать ценность, поставляемую всем заинтересованным сторонам ИТ-проекта, благодаря организации взаимодействия участников проекта и реализации предложенных моделей принятия решений при ранжировании пользовательских историй. Рис. 7, ист. 31.

Ключевые слова: управление проектами, управление требованиями, ценность, принятие решений, ИТ-проект, Agile, вербальный анализ решений.

JEL O22, D46, D70, D81, C60

**Постановка проблемы в общем виде и ее связь с важными научными или практическими задачами.** В настоящее время значительная доля проектов по разработке ПО выполняется в соответствии с гибкой (Agile) методологией. Это обусловлено тем, что проекты, управляемые в соответствии с принципами и рекомендациями, изложенными в Agile манифесте по разработке программного обеспечения [1] (далее Agile-проекты), в три раза успешнее традиционных [2].

Так как терминология в командах разработки ПО в основном используется англоязычная, то и в дальнейшем по тексту термины будут использоваться соответствующие, и вводимые сокращения также будут на английском языке.

Согласно исследованию Standish Group только 29 % инициированных в 2015 году ИТ-проектов завершились успехом, 52 % проектов завершились не в сроки, с превышением бюджета или выпуском продуктов с меньшим функционалом, чем планировалось, а 19 % вовсе были провальными [2]. Основными проблемами ИТ-проектов являются: частые изменения спецификаций и искажение данных, полученных от заказчика, классифицируемых как процессы сбора и управления требованиями [2]. Это, в свою очередь, приводит к увеличению стоимости работ и /или получению продукта, который оказывается не актуальным или не нужным заказчику. Таким образом, тщательность сбора и управления требованиями к проекту и продукту напрямую влияет на успех проекта.

С другой стороны, в настоящее время методология управления проектами проходит в развитии определенную фазу качественного изменения, когда все большее распространение получает логика управления, ориентированная на создание ценности [3, 4]. Таким образом, принятие управленческих решений при управлении проектами все больше ориентируется на создание и передачу ценности заинтересованным сторонам в виде готового продукта проекта [5]. Создание ценности непосредственно связано с определением и управлением содержанием проекта. Под ценностью продукта Agile-проекта в данном случае мы будем понимать такую характеристику разрабатываемого ПО, которая описывает его личностную и социально-культурную значимость в субъективном видении заинтересованных сторон [6]. Следовательно, в ИТ-проектах и проектах по разработке программного обеспечения, в частности, которые в большинстве случаев управляются сегодня в соответствии с гибкой методологией, создание

ценности связано с процессами управления требованиями. Таким образом, исследование, направленное на повышение эффективности процессов принятия решений при управлении требованиями в ИТ-проектах с учетом создаваемой при этом ценности, является актуальным.

**Анализ исследований и публикаций и выделение нерешенных ранее частей общей проблемы.** В соответствии с PMBoK требования должны быть выявлены, проанализированы и зарегистрированы с достаточной степенью детализации на самых ранних этапах проекта – инициации и планирования [7]. Подробное описание содержания является основой для ключевых решений по проекту, и позволяет связать стратегические цели компании и результаты проекта с выгодами, которые должны быть получены в результате его успешного выполнения [3, 8].

В пятой версии PMBoK Agile-проекты определяются как проекты, которые имеют адаптивный жизненный цикл, направлены на реагирование на высокие уровни изменений и требуют постоянной высокой степени вовлеченности заинтересованных сторон проекта [7]. В работе [9] говорится о том, что при планировании и реализации Agile-проекта его содержание разбивается на набор требований, а совокупность соответствующих работ, называемых *бэклогом* (от англ. *Backlog*), а в [10] – что бэклог формируется бизнес аналитиком в процессе сбора требований при коммуникации с заинтересованными сторонами проекта. Также в [Ошибка! Источник ссылки не найден.9] Бэклог определяется как множество пользовательских историй (от англ. *user stories*, далее - US), – основного инструмента документирования пожеланий заказчика в Agile-проектах. US – это краткие описания требований к разрабатываемой системе, сформулированных на деловом языке пользователя. US – быстрый способ документировать требования, без необходимости разрабатывать обширные формализованные документы и впоследствии тратить ресурсы на их поддержание. В работе [11] требования связываются с ценностью в том плане, что после документирования пользовательские истории ранжируются заказчиком или его представителем – *владельцем продукта* (от англ. *Product Owner*) на основе анализа ценности, которую получает заказчик при реализации данной функциональности продукта. Таким образом, важнейшим преимуществом гибких подходов является минимизация рисков при реализации проекта благодаря частому и регулярному прямому общению заказчика и исполнителей [12].

Согласно исследованиям IBM, 60 % затрат времени разработчики программного обеспечения несут вследствие неэффективной организации процесса управления требованиями [13]. Заказчики, в свою очередь, требуют от проектных команд согласовывать сроки и бюджет проекта сразу после определения концепции продукта, что практически невозможно при использовании гибких методик. Для решения этой проблемы используется комбинирование экспертных оценок и приоритизации требований [10].

С точки зрения методологии управления проектами, управление требованиями в ИТ-проектах направлено на решение комплекса задач, связанных с разработкой базового плана проекта. При этом важнейшей задачей принятия решений является определение очередности выполнения работ, которое должно быть направлено на обеспечение создания и передачи ценности заинтересованным сторонам проекта, и которое в Agile-проектах сводится к приоритизации требований.

Приоритизация требований является сложным процессом, для которого характерны постоянный анализ и ранжирование альтернатив при многих критериях, обязательно учитывающих ценность, создаваемую в проекте, что

существенно усложняет привлечение экспертов, необходимое при решении подобных слабоструктурированных задач. В качестве методов и инструментов, зарекомендовавших себя как эффективных, необходимо упомянуть метод ЗАПРОС, предложенный в работах О.И. Ларичева [14]. Эффективное применение метода ЗАПРОС в задачах управления проектами и портфелями проектов рассмотрены в работах [15, 16, 17]. Применение метода ЗАПРОС при решении задачи ранжирования пользовательских историй имеет следующие преимущества: возможность учета множества критериев, причем носящих качественный характер; возможность кардинального повышения эффективности процессов принятия решений за счет отчуждения знаний о приоритетах заинтересованных сторон на основе их системы ценностей; возможность учета субъективной информации и субъективных оценок, как работ, так и содержания продукта, учитываемых при ранжировании множества пользовательских историй. С другой стороны, выбор метода в значительной степени определяет структуру процессов принятия решений и содержания используемой и генерируемой при этом информации.

В [18, 19] также утверждается, что одним из важнейших элементов процесса управления жизненным циклом требований является их приоритезация. Т.к. пользовательские истории представляют собой формализацию требований заказчика и составляют бэклог продукта, элементы которого должны содержать приоритеты для определения очередности их реализации дабы максимизировать ценность для заказчика, то посредством ранжирования пользовательских историй в бэклог производится непосредственное влияние на процесс управления требованиями. Рациональное и своевременное ранжирование пользовательских историй, направленное на сокращение числа неточных, неполных или упущенных требований, позволяет снизить расходы по проекту до 20% [13, 19].

**Целью данного исследования** является разработка новых и совершенствование существующих моделей процессов принятия решений при управлении требованиями, использование которых позволит повысить эффективность управления ИТ-проектами по разработке ПО, благодаря максимизации ценности, доставляемой всем заинтересованным сторонам проекта.

**Изложение основного материала исследования.** Для определения содержания работ по принятию решений и используемой при этом информации рассмотрим модель процесса ценностно-ориентированного ранжирования пользовательских историй в Agile-проектах в BPMN-нотации (рис. 1). Ранжирование пользовательских историй, которое представляет собой пример одной из ключевых задач принятия решений, реализуется различными участниками процесса создания ПО:

- стейкхолдерами, которые определяют логику реализации проекта, его цели и задачи;
- бизнес-аналитиком, функции которого в Agile-проектах может выполнять владелец продукта, являющимся связующим звеном между заинтересованными сторонами и командой проекта;
- менеджером проекта, координирующим работы по проекту и взаимодействие команды с заказчиком, его представителями и другими заинтересованными сторонами.

Ключевыми задачами процесса ценностно-ориентированного ранжирования пользовательских историй в Agile-проектах являются:

- 1) идентификация ценностей для стейкхолдеров;
- 2) формирование структуры декомпозиции ценностей проекта;

- 3) сбор пожеланий стейкхолдеров;
- 4) написание пользовательских историй;
- 5) формирование критериев ранжирования пользовательских историй;
- 6) формирование шкалы для оценки историй;
- 7) непосредственное ранжирование списка.

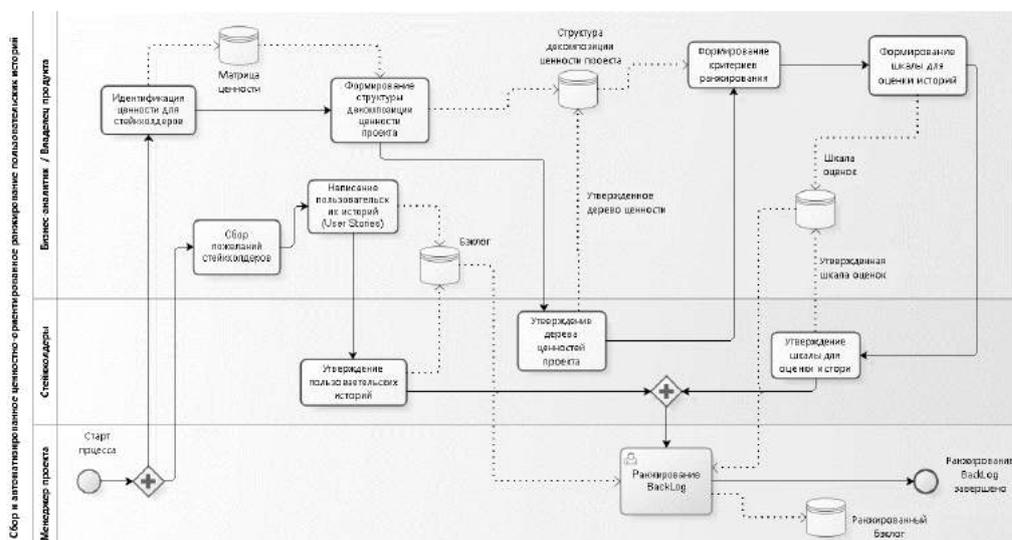


Рис. 1. Модель процесса (BPMN-нотация) ценностно-ориентированного управления пользовательскими историями в Agile-проектах

Содержание данного перечня задач определяется логикой используемого метода принятия решений – ЗАПРОС, а также особенностями ориентации на создание ценности при принятии решений.

В контексте ценностно-ориентированного управления перечисленные задачи логически объединяются в следующие группы:

- анализ и структурирование ценности – задачи 1 и 2;
- определение содержания продукта – задачи 3 и 4;
- разработка модели принятия решения – задачи 5 и 6;
- ранжирование (непосредственное) списка пользовательских историй – задача 7.

Важнейшими контрольными событиями данного процесса являются факты завершения формирования структуры декомпозиции ценностей, написания пользовательских историй и самого ранжирования историй. Их результаты, а также вся полученная информация заносятся в базу знаний проекта, и используется для ранжирования пользовательских историй в составе *бэклог*. Детальное описание функций процесса ценностно-ориентированного управления пользовательскими историями в Agile-проектах представлено ниже.

*Анализ и структурирование ценности* направлены на идентификацию ценности для стейкхолдеров и начинается сразу после старта проекта. Выполнение данной функции необходимо для выявления ценности, которую должен формировать продукт проекта для ключевых стейкхолдеров. Данная ценность является базой и основным критерием для приемки продукта проекта. Заключается функция в проведении владельцем продукта или бизнес-аналитиком сессий сбора требований со стейкхолдерами, на которых

выявляются явно и неявно выраженные требования к продукту, пожелания и опасения стейкхолдеров.

*Идентификация ценностей для стейкхолдеров.* Для документирования ценностей используется конфигурация продукта проекта – понятие, предложенное в работе [6]. Конфигурация  $C$  продукта проекта является отношением на множествах ценностей для стейкхолдеров и свойств продукта проекта, и описывается выражением

$$C \subset V \times P,$$

где  $V$  – множество ценностей для стейкхолдеров,  $P$  – множество свойств продукта проекта:

$$V = \{v_{ij}\}, P = \{p_{kj}\},$$

где  $i=1..l, j=1..m, k=1..n, l$  – общее количество выявленных ценностей,  $m$  – количество стейкхолдеров,  $n$  – общее количество выделенных свойств продукта проекта.

Особенностью отображения  $C$  является то, что нескольким ценностям для стейкхолдеров может соответствовать одно свойство продукта:

$$C = \{(v_{ij}, p_k) \mid \forall v_{ij} \in V \exists p_k \in P, p_k = C(v_{ij}) \& |\{p_k\}| \geq 1\},$$

где  $i = 1..l, j = 1..m, k = 1..n, l$  – общее количество выявленных ценностей,  $m$  – количество заинтересованных сторон,  $n$  – общее количество выделенных свойств продукта проекта.

Данное соответствие не является отображением, т.к. каждая ценность может создаваться несколькими свойствами продукта и, в то же время, каждое свойство может создавать несколько ценностей ( $C$  ни инъективно, ни сюръективно). В теории баз данных такое соответствие моделируется специальным видом отношений – "многие-ко-многим" и для его формализации вводится промежуточная абстрактная сущность [20].

В соответствии с определением, процесс управления конфигурацией продукта проекта предполагает обязательное выполнение следующих базовых операций:

- сбор информации о пожеланиях стейкхолдеров;
- формирование множества  $V$  ценностей для стейкхолдеров;
- сопоставление свойств продукта  $P$  множеству ценностей для стейкхолдеров;
- систематизация и агрегирование свойств продукта проекта;
- формирование модели продукта проекта.

Данный подход позволяет не только выявить и структурировать ценности для стейкхолдеров уже на ранних стадиях проекта, но и формализовать связь ожидаемых от проекта ценностей со свойствами продукта.

Для документирования и хранения конфигурации продукта проекта используется матрица ценности:

$$C = (c_{ij}), c_{ij} = \begin{cases} 1, & p_{ij} = C(v_{ij}) \\ 0, & p_{ij} \neq C(v_{ij}) \end{cases}.$$

*Формирование структуры декомпозиции ценности проекта* предполагает построение на основании данных, собранных у стейкхолдеров, структуры описания ценностей в привязке к базовому функционалу продукта проекта. К особенностям данной функции относится необходимость анализа владельцем продукта информации, полученной от стейкхолдеров, сопровождающаяся, при необходимости, консультациями с заинтересованными сторонами и экспертами предметной области.

Для дальнейшего анализа воспользуемся понятием структуры декомпозиции ценности (от англ. *Value Breakdown Structure, VBS*). Само понятие было предложено в работе S. Devaux [21]. Однако, в работах S. Devaux под структурой декомпозиции ценности, фактически, понимается структура, идентичная PBS (структура декомпозиции продукта – от англ. *Product Breakdown Structure*) или WBS (структура декомпозиции работ – от англ. *Work Breakdown Structure*), формируемая в соответствии с полученными ROI или NPV [22], что несколько не соответствует современному взгляду на ценность, создаваемую в проекте, которая представляет собой сложную многоуровневую и многомерную структуру [3, 23].

Воспользуемся определением, введенным в работе [24]: структура декомпозиции ценности – иерархия выгод, получаемых заинтересованными сторонами от продукта проекта, представленная в виде графа. В данном контексте под ценностью в структуре декомпозиции понимается именно способность продукта создавать ценность для заинтересованных сторон. При этом декомпозиция должна выполняться в разрезе предоставляемой ценности.

Структура декомпозиции ценности должна обеспечивать решение следующих задач [24]:

- описывать структуру и содержание ценности продукта проекта через его свойства со степенью детализации, необходимой и достаточной для решения задач ценностно-ориентированного управления проектами, направленного на создание ценности в данном продукте проекта;
- представлять ценность продукта проекта в виде иерархической структуры (ценность должна декомпозироваться на составляющие до тех пор, пока не будет однозначно понятно, как обеспечить ее создание соответствующими свойствами продукта проекта);
- представляет свойства продукта таким образом, который характеризует создаваемую ими ценность;
- предоставлять возможность формализовать ценность, ожидаемую заинтересованными сторонами в объеме, необходимом и достаточном для обеспечения их удовлетворенности получаемой ценностью;
- предоставлять возможность не только описать, но оценить (измерить) результат проекта или его частей (*deliverables*), который рассматривается как результат управления проектом и выполнения работ, направленных на создание данной ценности для удовлетворения заинтересованных сторон.

На рис. 2 представлен пример структуры декомпозиции ценности для продукта ИТ-проекта – подсистемы автоматизированного планирования работы производственного цеха. Базовое структурирование ценностей выполнено на основе работы Н. Kerzner и F. Saladis, в соответствии с которой выделено три базовые группы ценностей: бизнес-ценности, стратегические и операционные ценности [3].

В соответствии с представленными выше свойствами *VBS* продукта в общем виде описывается ориентированным графом  $G(V,E)$ , где  $V$  – множество вершин, а  $E$  – множество дуг, с присущими ему базовыми свойствами: существует единственный узел  $u$ , называемый корнем –  $u \in V$ ; полустепень

захода корня равна 0, а полустепень захода всех остальных узлов равна 1; каждый узел достижим из корня; в структуре отсутствуют циклы –  $z(G) = 0$ .

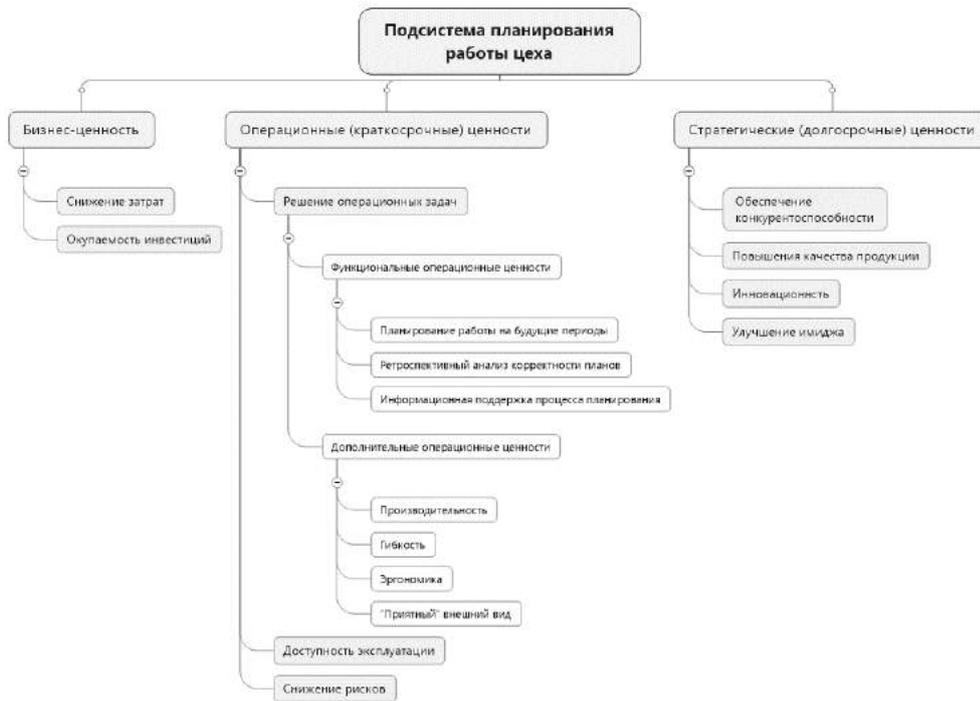


Рис. 2. Структура декомпозиции ценности подсистемы планирования работы производственного цеха

Концевые вершины графа VBS, именуемые листьями  $v_i^{term}$ , характеризуют конкретные ценности, предоставляемые продуктом проекта и ожидаемые заинтересованными сторонами. Для каждого листа VBS существует ветвь, ведущая в этот узел из корня –  $(u, v_i^{term})$ . Условимся называть данную ветвь *путем создания ценности*, а ценность, получаемую при этом заказчиком – *потоком создаваемой ценности* –  $c(u, v_i^{term})$ .

Важной особенностью VBS является наличие магистральных дуг, соединяющих корень графа VBS со смежными вершинами  $\{v_i^1\} \subset V$ , такими, что для них выполняется условие  $d^+ \langle u, v_i^1 \rangle = 1$ . Этим дугам соответствуют категории ценностей, выделяемые заинтересованными сторонами. В общем случае, к магистральным ценностям относятся следующие: бизнес-ценности, стратегические (долгосрочные) и операционные (тактические) ценности (см. рис. 2) [3].

Задача выявления и структурирования ценности является достаточно абстрактной. Для ее решения на практике может применяться широкий диапазон инструментов начиная от базовых офисных приложений и заканчивая специализированными средствами. При этом возможно применение различных

подходов и методологий [25]. Выбор каждой из них является результатом предпочтений команды проекта.

*Сбор пожеланий стейкхолдеров* направлен на документирование функциональных и нефункциональных требований стейкхолдеров к продукту проекта. Особенностью данной функции является то, что ее активность происходит параллельно с функцией идентификации ценности. Владелец продукта выявляет, структурирует и документирует высокоуровневые требования заинтересованных сторон к программному продукту.

$$P^0 = p_i^0,$$

где  $i = \overline{1..n}, n$  – количество пользовательских историй в бэклог.

Каждая пользовательская история представляется кортежем

$$p_i^0 = t_i^0, a_i^0, g_i^0, ci_i^0,$$

где  $t_i^0$  – название пользовательской истории,  $a_i^0$  – ее автор,  $g_i^0$  – цель, которую намерен достичь владелец продукта,  $ci_i^0$  – важность данной функциональности для потребителя.

*Написание пользовательских историй* предполагает формализацию высокоуровневых требований к продукту проекта в принятом формате, а также формулирование первоначальных критериев приемки каждой US. Особенностью данной функции является описание как требований так и критериев приемки в особом формате. Владелец продукта записывает требования, критерии по которым будут проверяться истории и приниматься решение об их успешной реализации. При необходимости проводит уточняющие сессии со стейкхолдерами и экспертами в предметной области.

Изначально все требования к продукту в виде пользовательских историй организованы во множество  $P^0$  бэклог. Каждая пользовательская история соответствует формату: Как <Автор> я хочу <Функция> для того чтобы <Цель владельца продукта>. Например: "Как начальник цеха я хочу иметь возможность планировать работу цеха по дням для того, чтобы ежедневно контролировать загрузку мощностей и персонала. Важность: высокая".

Для хранения пользовательских историй используются базовые офисные продукты (Microsoft Excel, Microsoft Word и пр.) или специальные (Atlassian Jira, Basecamp) средства ПО. Многие продукты предлагают средства поддержки управления Agile проектами. Так Atlassian предлагает Jira Agile – продукт в котором есть возможность управлять бэклог и канбан доской (рис. 3, 4).

Другой особенностью графа VBS является его комбинаторно-множественный характер, обусловленный свойствами продукта проекта и их возможностями по реализации ценности (1, 2). Например, такая ценность программного обеспечения для системы планирования, создаваемого в процессе реализации ИТ-проекта, как "Производительность" может быть обеспечена несколькими свойствами: 1) "Скорость реакции на воздействие" (скорость реакции системы на изменение исходных условий планирования); 2) "Скорость работы" (скорость непосредственно процесса планирования); 3) "Мощность аппаратного обеспечения"; 4) "Эргономичный интерфейс" и др.

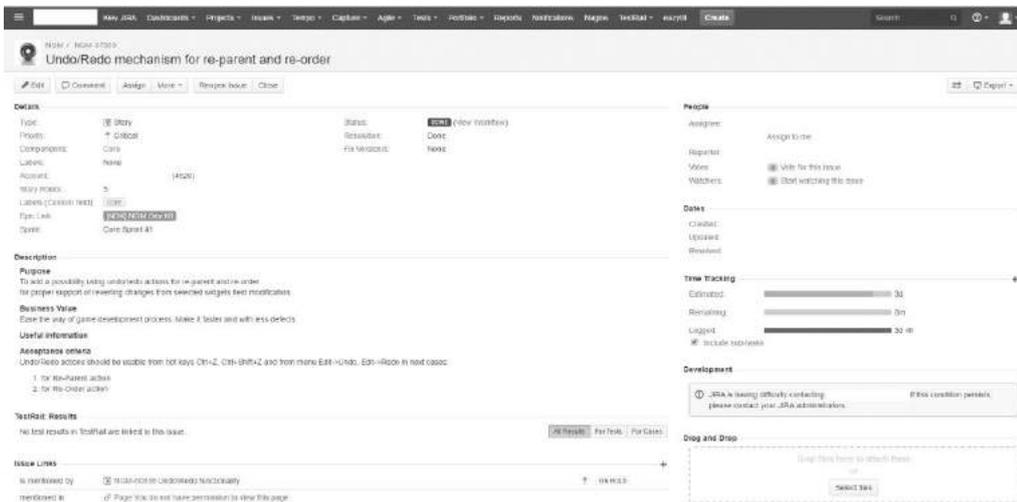


Рис. 3. Пример оформления пользовательской истории в программном продукте Atlassian Jira

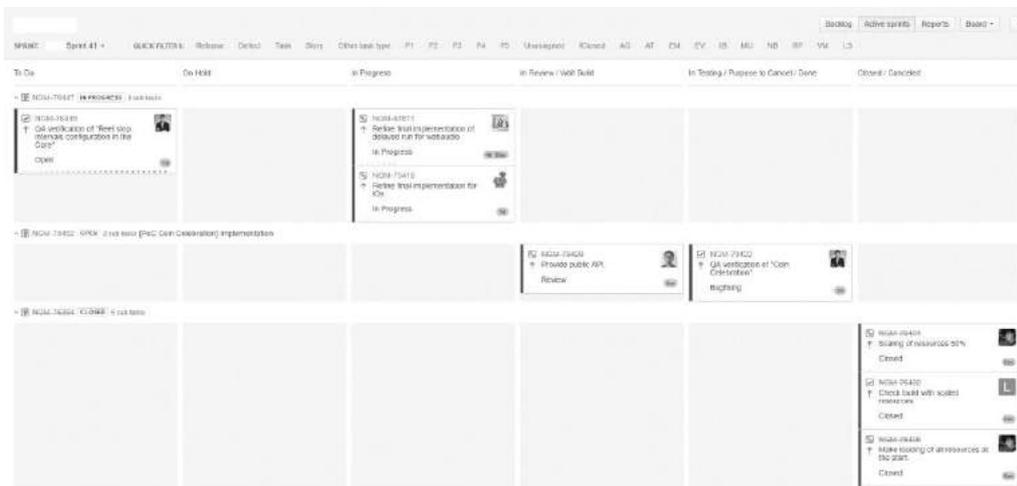


Рис. 4. Внешний вид канбан доски в программном продукте Atlassian Jira

Перечисленные выше свойства предполагают возможность совместной реализации, т.е. могут быть имплементированы в готовом продукте одновременно. В отличие от этого, такая ценность, как "Планирование работы на будущие периоды" обеспечивается возможностью продукта определять планы производства и реализуется в двух принципиально разных режимах работы – "Определение прогнозного плана" и "Определение уточненного плана работы". При этом данные свойства являются взаимоисключающими, т.е. подсистема может работать либо в одном режиме, либо в другом.

Если  $P_i$  – множество свойств продукта проекта, обеспечивающих создание  $i$ -ой ценности, то  $P_i \subset P$ . Условимся называть свойства продукта