СИСТЕМНЫЙ АНАЛИЗ

УДК 004.8 **DOI: 10.31799/2077-5687-2025-2-3-9**

ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИСТОРИЙ ПРИ ПРОЕКТИРОВАНИИ ПЕРСОНАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЗАДАЧАМИ

К. В. Быченок, А. В. Аграновский
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассмотрен подход к проектированию персональной системы управления задачами на основе сбора требований через пользовательские истории. Описана методика получения функциональных требований из пользовательских историй и их последующего применения для моделирования системы.

Показано, что применение пользовательских историй обеспечивает ориентацию на потребности пользователя на ранних этапах разработки и может служить основой для формирования формальной модели требований к системе.

В качестве примера проведен анализ трех пользовательских историй, включающих функции интеллектуального помощника, получения статистики выполнения задач и формирования персонализированных рекомендаций, с формулированием на их основе набора функциональных требований. Представлена UML-диаграмма вариантов использования, на которой отражены разработанные требования.

Ключевые слова: система управления задачами, пользовательская история, User Story, сбор требований, функциональные требования, диаграмма прецедентов.

Для цитирования:
Быченок, К. В. Практическое применение пользовательских историй при проектировании персональной информационной системы управления задачами / К. В. Быченок, А. В. Аграновский // Системный анализ и логистика. – 2025. – № 2(45). – с. 3-9. DOI: 10.31799/2077-5687-2025-2-3-9.

Введение

Персональные системы управления задачами (task management systems) стали неотъемлемой частью инструментария для повышения продуктивности и организации дел как в профессиональной деятельности, так и в повседневной жизни. Лучшие мировые практики управления задачами, например, Trello, Asana и их аналоги, стремятся адаптироваться под конкретного пользователя путем внедрения элементов искусственного интеллекта (ИИ). Так, систематический обзор по использованию ИИ в планировании проектов отмечает растущий интерес к применению интеллектуальных алгоритмов для повышения эффективности планирования и расписания задач [1].

При разработке систем управления задачами важно правильно выявить и сформулировать требования, ориентируясь на реальные потребности и ожидания пользователей.

В гибких методологиях разработки программного обеспечения (ПО) для описания требований к системе с точки зрения конечного пользователя широко применяются пользовательские истории [2]. Пользовательская история (User Story) представляет собой краткое повествовательное описание желаемой функциональности, включающее сведения о том, кому (какому типу пользователя) нужна та или иная возможность, что он хочет получить и зачем это ему необходимо. Такой формат позволяет установить взаимопонимание между разработчиками и заинтересованными лицами, как правило, не обладающими специальными знаниями в области разработки, на ранних этапах создания ПО и сфокусироваться на ценности системы для пользователя [1, 3].

Исследования показывают, что пользовательские истории могут служить эффективной отправной точкой для создания концептуальных моделей системы и иногда позволяют более детально выявить структуру предметной области [4]. Кроме того, пользовательские истории помогают составить детальный план проверки качества разработанной системы [5].

В то же время для сложных систем процесс преобразования наборов пользовательских историй в формальные требования может оказаться затруднительным по причине значительного количества, неполноты или неструктурированности стихийно собранных историй, неоднозначности естественного языка и значительных временных затрат на последующую систематизацию [6].

Цель данной работы – продемонстрировать методику перехода от пользовательских историй к функциональным требованиям на примере проектирования персональной системы управления задачами.

1 Сбор требований через пользовательские истории

Использование User Story подразумевает, что на начальном этапе потенциальные пользователи системы привлекаются к формированию видения продукта: посредством интервью и опросов выявляются основные проблемы и потребности, с которыми они сталкиваются при управлении личными задачами.

Каждая потребность оформляется в виде отдельной пользовательской истории на естественном языке, следующей общепринятому шаблону: *«Как [тип пользователя], я хочу [действие/функцию], чтобы [цель/польза]»*. Такой подход к фиксации требований фокусируется на том, что хочет получить пользователь и почему это для него важно, откладывая детальные технические подробности на последующие этапы разработки требований [3].

В рамках гибкой методологии agile считается полезным сначала собрать максимально полный набор историй, ранжировать их по приоритетам пользователя, а затем на их основе формулировать функциональные требования и создавать модели системы [7].

В данной работе мы придерживаемся следующей последовательности:

1. отбор ключевых пользовательских историй, наиболее критичных для ценности системы;
2. детальный анализ каждой истории с точки зрения того, какие функции системы потребуются для ее реализации;
3. формирование функциональных требований, удовлетворяющих соответствующей пользовательской истории;
4. построение UML-диаграммы вариантов использования, агрегирующей полученные требования на концептуальном уровне.

Этот процесс обеспечивает прослеживаемость от исходных потребностей пользователя (историй) до элементов проектного решения.

2 Анализ пользовательских историй

На основе проведенных интервью были сформулированы три обобщенные ключевые пользовательские истории, отражающие основные ожидаемые возможности персональной системы управления задачами, связанные с применением искусственного интеллекта.

2.1 Интеллектуальный помощник для планирования.

История 1: «Как пользователь, я хочу иметь в системе интеллектуального помощника, который умеет давать советы по управлению моими задачами и расписанием, чтобы повысить продуктивность и ничего не упустить». Данная пользовательская история указывает на необходимость встроенного ИИ-ассистента, способного взаимодействовать с пользователем и помогать ему в организации дел. Анализ этой потребности показывает, что в составе системы должна быть реализована подсистема, отвечающая за рекомендации и автоматизацию рутинных операций планирования.

Функционально это может включать обработку естественного языка (понимание запросов пользователя), генерацию рекомендаций (например, предложить оптимальное время

для выполнения задачи или дедлайна), напоминания, автоматическое переназначение просроченных задач и пр. Для реализации таких возможностей в требования следует заложить интеграцию методов искусственного интеллекта. Можно ожидать, что интеллектуальный помощник будет снижать когнитивную нагрузку на пользователя и повышать эффективность планирования, что подтверждается исследованиями в области интеллектуальных ассистентов [8].

Таким образом, из Истории 1 вытекает потребность в следующих функциях:

- интерактивное взаимодействие пользователя с помощником (через чат-интерфейс или голос),
- анализ списка задач и календаря,
- генерация советов и действий (например, автоматически перенести задачу или установить напоминание).

2.2 Статистика и прогресс по задачам.

История 2: «Как пользователь, я хочу просматривать статистику выполнения своих задач (например, сколько задач выполнено за неделю, процент завершенных дел и т.д.), чтобы анализировать свою продуктивность и улучшать процессы планирования». Анализ этой истории указывает, что системе необходимо накапливать данные о задачах (даты создания, сроки, факты завершения, категории и пр.), а затем визуализировать их в удобной форме.

Из Истории 2 непосредственно вытекают функциональные требования:

- хранение истории изменений задач,
- вычисление метрик (количество выполненных/невыполненных задач за период, среднее время выполнения задачи, соблюдение дедлайнов),
- представление данных в виде графиков или отчетов.

Пользователю должны быть доступны средства отслеживания личной продуктивности: например, диаграмма выполненных задач по дням недели, отчет о наиболее часто откладываемых делах. В требования необходимо включить функции генерации статистических отчетов и визуализации данных. Реализация данного набора функций позволит пользователю получать обратную связь и осознанно подходить к управлению временем.

2.3 Персонализированные рекомендации задач.

История 3 «Как пользователь, я хочу получать от системы персональные подсказки по выбору следующей задачи для выполнения, чтобы эффективно расставлять приоритеты и быстрее достигать своих целей». Эта история дополняет первую, делая акцент именно на рекомендациях относительно приоритизации задач. В отличие от интеллектуального помощника в форме диалога (История 1), здесь может подразумеваться автоматическое ранжирование или выделение задач. Анализ показывает, что для реализации такой возможности система должна собирать информацию о предпочтениях пользователя и его привычках (например, какие задачи он чаще откладывает, в какое время суток наиболее продуктивен и т.д.), а также учитывать текущий контекст (например, приближающиеся дедлайны).

На основе этих данных потребуется алгоритм рекомендательной системы, который будет периодически выдавать пользователю советы: например, "В первую очередь сегодня рекомендуется заняться задачей X, так как до дедлайна остался 1 день". Персонализация может основываться на методах машинного обучения или экспертных правилах, однако в терминах требований достаточно указать сам функционал рекомендации.

Из Истории 3 следуют требования:

- наличие компонента, который вычисляет приоритеты списка дел для конкретного пользователя;

- отображение подсказок или маркировка задач, на которые стоит обратить внимание в первую очередь; возможность обучения системы на основе откликов пользователя (например, если рекомендации не подходят, пользователь может их игнорировать или отмечать неверными).

Подобные механизмы персонализированных рекомендаций задач рассматриваются в научной литературе как средство помочь пользователю находить наиболее подходящие задачи из имеющегося списка, что подтверждает актуальность указанной функции [9].

Результаты анализа сведены в таблицу 1, где для каждой пользовательской истории указаны вытекающие из нее функциональные требования к системе.

Таблица 1 – Соответствие пользовательских историй и функциональных требований

Пользовательская история	Соответствующие функциональные требования
История 1. Интеллектуальный помощник для планирования задач. «Как пользователь, я хочу иметь в системе интеллектуального помощника, который умеет давать советы по управлению моими задачами и расписанием, чтобы повысить продуктивность и ничего не упустить»	Система должна предоставлять интерфейс интеллектуального ассистента (чат-бот или аналог) для взаимодействия с пользователем. Система должна обрабатывать запросы на естественном языке и извлекать из них намерения пользователя. Система должна автоматически предлагать действия (добавить напоминание, перенести задачу, установить дедлайн) на основе контекста задачи. Система должна использовать алгоритмы ИИ для генерации рекомендаций по планированию (например, оптимальное расписание задач).
История 2. Статистика и прогресс. «Как пользователь, я хочу просматривать статистику выполнения своих задач, чтобы анализировать свою продуктивность и улучшать процессы планирования»	Система должна хранить историю выполнения задач (дата создания, завершения, статус и пр.). Система должна вычислять ключевые показатели (количество выполненных задач за период, процент выполненных задач, среднее время выполнения и т.д.). Система должна отображать статистическую информацию в удобной форме (графики, диаграммы, таблицы) в интерфейсе пользователя.
История 3. Персонализированные рекомендации. «Как пользователь, я хочу получать от системы персональные подсказки по выбору следующей задачи для выполнения, чтобы эффективно расставлять приоритеты и быстрее достигать своих целей»	Система должна анализировать параметры пользователя (историю действий, предпочтения, расписание) для определения приоритетов. Система должна генерировать персональные рекомендации по очередности выполнения задач. Система должна отображать рекомендацию (например, отмечать рекомендованную задачу как приоритетную либо выдавать уведомление с советом). Система должна адаптироваться на основе обратной связи (учитывать, следует пользователь рекомендации или нет, чтобы улучшать будущие советы).

Из таблицы 1 видно, что каждый сценарий использования системы, описанный языком пользователя, трансформируется в несколько конкретных требований к функциональности. Эти требования в совокупности определяют поведение и возможности будущей системы. На следующем этапе их необходимо отразить в формальной модели.

3 Моделирование функциональности

Для визуального представления полученных требований и проверки полноты охвата потребностей пользователя на рис.1 построен фрагмент UML-диаграммы прецедентов (Use Case diagram). Диаграмма прецедентов или вариантов использования наглядно демонстрирует, какие услуги система предоставляет актору (пользователю), оставаясь при этом высокоуровневым представлением требований.

На ней показано, что единственный актор взаимодействует с тремя основными прецедентами: «Вызвать интеллектуального помощника», «Просмотреть статистику задач» и «Получить персональные рекомендации». Каждый из этих прецедентов соответствует одной из рассматриваемых пользовательских историй и агрегирует связанные с ней функции системы. Границы системы обозначены прямоугольником; все прецеденты, представленные внутри него, относятся к функциональности разрабатываемого приложения «Персональная система управления задачами».

Прочие прецеденты (такие как добавление новой задачи, отметка выполнения, редактирование и удаление задач, интеграция с календарями и т.д.) не отражены на данной диаграмме с целью фокусирования внимания на функционале, вытекающем из рассмотренных пользовательских историй.

Дальнейшая декомпозиция рассмотренных прецедентов может производиться с помощью их описания по шаблону, представленному в [10]. Таким образом, UML-модель связывает воедино требования, полученные из пользовательских историй, и служит основой для дальнейшего детального проектирования системы.

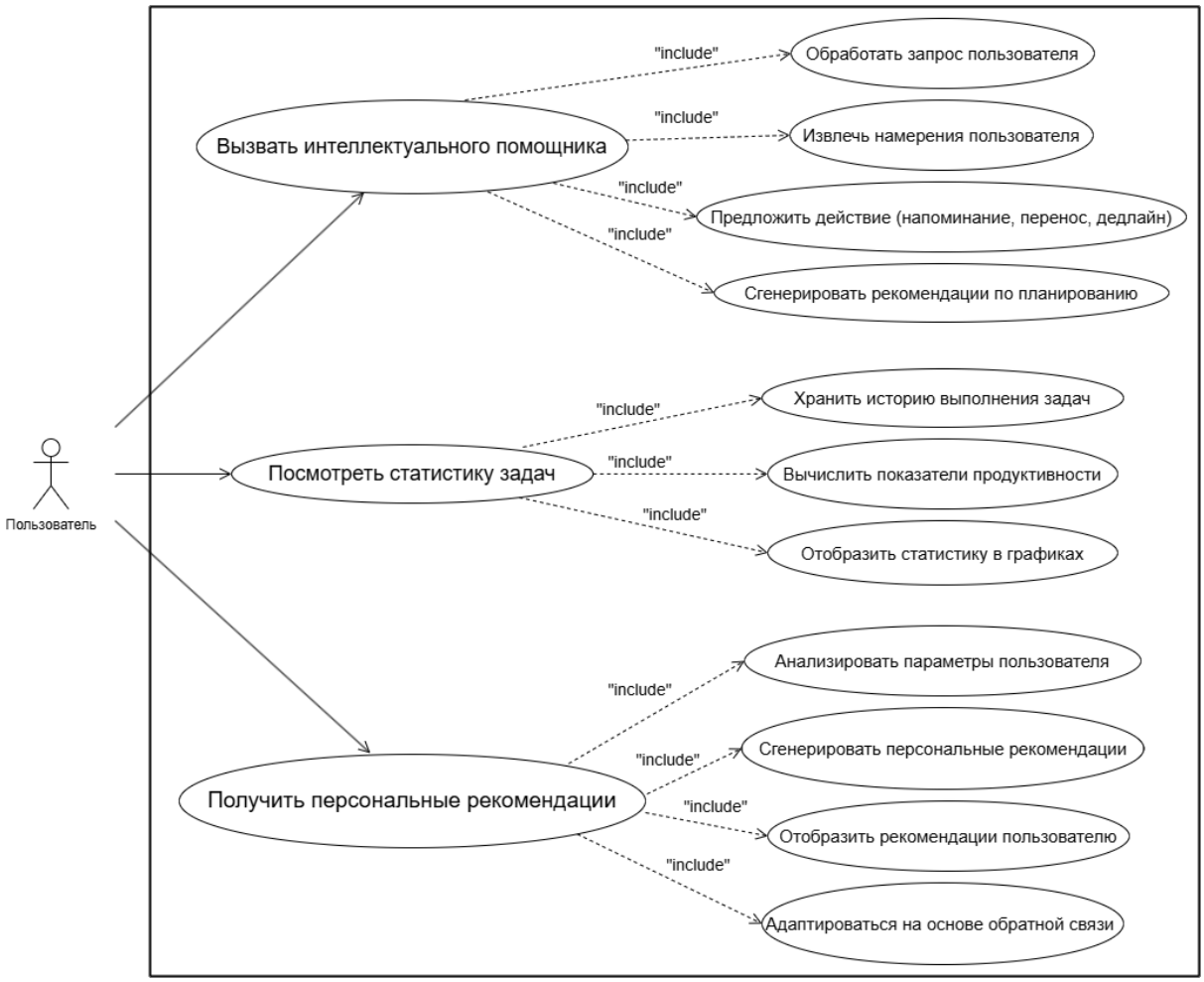

Рис. 1. Диаграмма вариантов использования персональной системы управления задачами

Заключение

Проведенное исследование подтверждает, что пользовательские истории могут служить удобным связующим звеном между этапом выявления потребностей и этапом формализации требований. Описывая систему на языке пользователя, разработчики получают ясное понимание ожидаемого поведения системы, а затем последовательно превращают эти ожидания в спецификацию. Преимущество подхода состоит в сохранении фокуса на ценности для пользователя: каждая функция системы прямо следует из явно сформулированной потребности. Использование стандартизированных нотаций для моделирования, таких как UML, упрощает коммуникацию в команде разработки и с экспертами предметной области. В дальнейшем пользовательские истории позволяют организовать оценку качества разработанного программного продукта.

Таким образом, переход от пользовательских историй к модели требований зарекомендовал себя как эффективный подход, обеспечивающий человеко-ориентированное проектирование программных систем.

СПИСОК ЛИТЕРАТУРЫ

1. *Bahroun Z.* Artificial intelligence applications in project scheduling: a systematic review, bibliometric analysis, and prospects for future research / Z.Bahroun, M.Tanash, R.As'ad, M.Alnajar // Management Systems in Production Engineering. 2023. – Vol. 31, № 2. – P. 144–161.

2. *Mornie M. N.* Visualisation of User Stories in UML Models: A Systematic Literature Review/ M. N. Mornie, N. Jali, S. N. Junaini, E. Mit, C. W. Shiang, S. Saee // Acta Informatica Pragensia. – 2023. – Vol. 12, № 2. – P. 419–438.
3. *Dalpiaz F., Brinkkemper S.* Agile Requirements Engineering with User Stories [Электронный ресурс] // 26th International Requirements Engineering Conference (RE) – Utrecht, 2018. – 3 p.
4. *Dalpiaz F.* On deriving conceptual models from user requirements: An empirical study / F. Dalpiaz, P. Gieske, A. Sturm // Information and Software Technology. – 2021. – Vol. 131. – P. 1–13.
5. *Турнецкая Е. Л.* Программная инженерия. Тестирование и контроль качества программного обеспечения / Е. Л. Турнецкая, А. В. Аграновский. – Санкт-Петербург: Лань, 2025. – 172 с.]
6. *Паттон Д.* Пользовательские истории: искусство гибкой разработки ПО / Д. Паттон. – Санкт-Петербург: Питер, 2025. – 288 с.
7. *Alsaadi B.* Data-driven effort estimation techniques of agile user stories: a systematic literature review/ B. Alsaadi, K. Saeedi //Artificial Intelligence Review. – 2022. – Vol. 55, № 7. – P. 5485-5516.
8. Adamantiadou D. S. Leveraging Artificial Intelligence in Project Management: A Systematic Review of Applications, Challenges, and Future Directions / D. S. Adamantiadou, L. Tsironis // Computers. – 2025. – Vol. 14 (2). – P. 66. doi:10.3390/computers14020066
9. *Geiger D.* Current State of Personalized Task Recommendation/ D. Geiger // Personalized Task Recommendation in Crowdsourcing Systems. – Springer, 2016. – C. 15–29.
10. Сообщество аналитиков. Рекомендации по написанию спецификаций вариантов использования [Электронный ресурс]. – URL: https://www.uml2.ru/faq/faq-uc/73/ (дата обращения: 25.04.2025).

ИНФОРМАЦИЯ ОБ АВТОРАХ

Быченок Кирилл Владимирович
Магистрант
Санкт-Петербургский государственный университет аэрокосмического приборостроения
Россия, 190000, Санкт-Петербург, ул. Большая Морская, д.67, лит. А
E-mail: kirillbosi@mail.ru

Аграновский Андрей Владимирович
Доцент, канд. техн. наук
Санкт-Петербургский государственный университет аэрокосмического приборостроения
Россия, 190000, Санкт-Петербург, ул. Большая Морская, д.67, лит. А
E-mail: a_agranovskii@mail.ru

Дата поступления: 03.02.2025
Дата принятия: 01.07.2025