DOI: 10.25728/mlsd.2023.1191

# МОДЕЛЬ ИЗМЕРЕНИЯ КАЧЕСТВА ПОЛЬЗОВАТЕЛЬСКИХ ИСТОРИЙ ДЛЯ СНИЖЕНИЯ РИСКОВ ГИБКОЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИСТЕМ КРИТИЧЕСКОЙ ИНФРАСТРУКТУРЫ

**Жарко Е.Ф.**
*Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия*
zharko@ipu.ru

*Аннотация. В докладе анализируются и собираются ключевые факторы качества пользовательских историй и предлагается модель измерения качества пользовательских историй (МИКПИ). Применяя модель МИКПИ, можно улучшить качество требований в процессе гибкой разработки программного обеспечения и снизить риск изменения требований.*

*Ключевые слова: гибкая разработка, история пользователя, программное обеспечение, измерение качества, модель измерения качества пользовательских историй.*

## Введение

В процессе разработки программного обеспечения можно столкнуться с проблемами, вызванными изменением требований. При разработке программного обеспечения для систем критической инфраструктуры необходимо эффективно справляться с последствиями изменения требований для снижения рисков разработки [1-3]. Изменения требований часто влияют на разработку программного обеспечения (ПО), заставляя разработчиков возвращаться к более ранним этапам жизненного цикла и проверке соответствующих результатов. Возникшая ситуация не только требует дополнительных ресурсов и средств, но и может привести к отставанию от графика разработки [1, 2]. «Затронутая» проектная и конструкторская документация, которую невозможно эффективно «изолировать» в процессе изменения требований, повышает риски разработки программного обеспечения, не соответствующего новым требованиям. Кроме того, затронутая проектная и конструкторская документация, которую невозможно полностью и корректно изменить, значительно снизит вероятность успешной разработки системы. При этом существует множество факторов, влияющих на успешность разработки системы, включая программное обеспечение. Одной из ключевых проблем является разработка программного обеспечения систем, так как связанные с этим ПО документы не могут быть незамедлительно пересмотрены и эффективно адаптированы к текущему изменению требований. Поэтому для снижения рисков при изменении требований к проекту процесс разработки программного обеспечения должен обладать высокой степенью изоляции и гибкостью модификации.

Изменения требований зачастую являются ключевыми факторами, которые приводят к неудачам проектов. Модель итеративной инкрементной разработки (ИИР) широко используются во многих методологиях разработки программного обеспечения [4]. В итеративных методиках требования имеют много возможностей для изменений. Инкрементальные концепции могут эффективно снизить сложность требований.

Модель ИИР может снизить частичный риск изменения требований к программному обеспечению. Стоит отметить, что гибкая модель разработки — это популярная в настоящее время методология разработки программного обеспечения [5-7], которая использует итеративную и инкрементную модель разработки для уменьшения влияния изменения требований. Гибкая модель разработки отличается высокой адаптивностью и высокой устойчивостью к изменениям требований, снижает сложность требований за счет использования ИИР и уменьшает фактор влияния изменений требований за счет рефакторинга, что повышает гибкость изменений требований без опоры на документацию. При изменении требований гибкая модель разработки ПО может эффективно снизить риск изменения требований, значительно сократив задержки графика разработки, внебюджетные расходы и ошибки в требованиях к качеству [8]. Однако у гибкой разработки ПО есть существенный недостаток: игнорирование анализа этапов работ и документации, сосредоточенность на работоспособном продукте и отсутствие внимания к последующему сопровождению как программного обеспечения, так и системы в целом.

При гибкой разработке программного обеспечения формальное документирование требований не рассматривается как необходимый элемент. Пользовательские истории — это краткие описания требований заказчика. Стоит отметить, что пользовательские истории также используются в гибкой методологии определения требований к программному обеспечению.

Пользовательские истории становятся основой для гибкой разработки и работы с изменениями требований. Основываясь на количественном определении качества, в докладе предлагается модель измерения качества пользовательских историй (МИКПИ):

- МИКПИ измеряет и объединяет аргументируемое, контролируемое и сертифицируемое качество;
- применение механизма количественной оценки модели МИКПИ должно обеспечить своевременное обнаружение ошибок в пользовательских историях.

Внедренные решения и улучшения, основанные на правилах, могут особенно улучшить качество пользовательских историй и снизить риск неудач проектов разработки программного обеспечения для систем критической инфраструктуры.

## 1. Риски изменения требований и критическая модель разработки

Изменения требований [9] являются одним из ключевых факторов, приводящих к неудачам проектов разработки программного обеспечения.

Высокий риск неудачи проекта по разработке программного обеспечения связан с задержками в графике разработки, невыполненными вышестоящими требованиями и т.д. [3]. В Таблице 1 приведены критические события, являющиеся фундаментальными причинами создания программного проекта с высокими рисками неудачи. Основными событиями, увеличивающими риски неудачи проекта, являются: неполные системные требования, эволюция технологий и среды, запросы заказчиков на изменение и перераспределение ресурсов, необходимых для разработки. Для снижения коэффициента неудачи [10] проекта требования к программному обеспечению должны иметь обсуждаемую, оцениваемую, контролируемую, управляемую, подтверждаемую способность обработки всех видов изменений требований. Это связано с тем, что новые требования к программному обеспечению системы будут постоянно предлагаться до тех пор, пока разработка системы не будет прекращена или система будет выведена из эксплуатации.

*Таблица 1. Основные причины неудачи проекта разработки программного обеспечения*

| **Пройденный этап разработки** | **Причины неудачи** |
|---|---|
| Этап требований | Несоответствие между системными требованиями и требованиями пользователя |
| Этапы проектирования и внедрения | Невозможность быстрой адаптации к новым технологиям и средам |
| Все этапы разработки | Несоответствие постоянным изменениям требований |
| | Невозможность своевременно скорректировать распределение ресурсов |

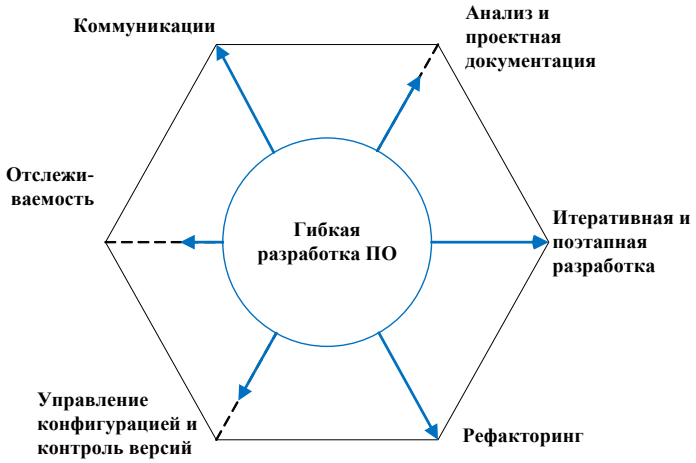


*Рис. 1. Преимущества и недостатки гибкой разработки программного обеспечения*

Стоит отметить, что при изменении требований гибкая разработка программного обеспечения значительно уменьшает задержки относительно графика работ, затраты и количество ситуаций с неудовлетворенными требованиями пользователей к качеству. И как следствие применения гибкой разработки программного обеспечения риск изменения требований к программному обеспечению может быть эффективно снижен. Однако гибкий процесс не затрагивает работы и документы на этапе анализа и проектирования, и применяется для разработки, не ориентированную на создание

документации, не уделяет внимания последующему сопровождению, что является серьезным недостатком этого метода. Преимущества и недостатки гибкого процесса разработки программного обеспечения показаны на рис. 1. Для преодоления последствий изменения требований необходимо или исправить или усилить недостатки гибкого процесса разработки.

## 2. Качество пользовательской истории

Гибкий процесс разработки программного обеспечения тесно связан с пользовательскими историями. И следовательно, качество пользовательских историй становится важным показателем, влияющим на риски проекта гибкой разработки программного обеспечения.

Пользовательская история — это краткое описание требований заказчика. В методологиях и фреймворках гибкого процесса разработки ПО, таких как Extreme Programming [11] или Scrum [12], пользовательские истории обычно используются как способ сбора требований. Пользовательская история — это форма требований к системе, содержащей программное обеспечение, которая стала необходимым элементом и популярна в гибких методологиях. В отличие от традиционных методов, таких как спецификация системных требований или диаграммы прецедентов, акцент в пользовательской истории делается на простоте и изменчивости. Карточка, обсуждение и подтверждение — три важнейших элемента представления пользовательских историй. В процессе гибкой разработки системы документы с формальными требованиями не являются необходимыми элементами. Поэтому пользовательская история становится важной основой для разработки программного обеспечения и изменения требований. Пользовательская история описывает небольшую часть функциональности модуля в простом и легко читаемом виде. В хорошо написанной пользовательской истории описывается требуемая функциональность, для чего она используется и почему она необходима. Качественная пользовательская история должна иметь независимые, обсуждаемые, ценные, оцениваемые и проверяемые характеристики. В соответствии с потребностями разработки описание пользовательской истории должно иметь единый формат. Для достижения согласованности требований следует удалить похожие и повторяющиеся пользовательские истории. Для избегания конфликта требований, пользовательские истории с высокой корреляцией должны быть объединены.

Пользовательская история должна иметь пять основных характеристик качества:

- *Возможность обсуждения*: ясное и простое содержание пользовательской истории может обеспечить легкое обсуждение между заказчиками, разработчиками, конечными пользователями и заинтересованными сторонами.
- *Оценка*: низкая сложность и высокая модульность пользовательской истории позволяют разработчикам легко оценивать время и стоимость разработки.
- *Контролируемость*: прикладное управление конфигурацией и контроль версий могут своевременно определить разницу между версиями и ревизиями.
- *Управляемость*: Создание таблицы перекрестных ссылок позволяет удобно управлять интегрированными отношениями между пользовательскими историями.
- *Подтверждаемость*: правильное, полное и последовательное содержание пользовательской истории может помочь в создании тестовых примеро для модульного тестирования, интеграционных и приемочных испытаний системы.

Измерение качества пользовательских историй должно учитывать следующие факторы качества:

(1) *Базовое (обсуждаемое и оцениваемое) качество*: в гибком процессе разработки программного обеспечения качество пользовательских историй может напрямую влиять на итеративную и инкрементальную операцию. Неоднозначные, сложные, пользовательские истории с высокой или низкой связностью часто становятся проблемами при разработке программного обеспечения, особенно при изменении требований. Для улучшения обсуждаемых и оцениваемых характеристик пользовательские истории должны иметь следующие факторы качества:

- *Ясность документов*: пользовательские истории являются основой для обсуждения функций разработки программного обеспечения. Ясность является необходимым фактором для обсуждения документов и связей между ними. Высокая ясность документов обеспечивает правильность, полноту, непротиворечивость, а также единый формат документации.
- *Низкая сложность*: сложность пользовательских историй можно измерить с помощью размера, логической сложности и сложности структуры данных. Сложность пользовательских историй позволяет определять график и стоимость разработки.

- *Высокая модульность*: пользовательские истории выполняют важную миссию итеративной и поэтапной работы. Модульность пользовательских историй позволяют разработчикам ПО создавать более понятные системы, которые проще настраивать и расширять в целях удовлетворения изменяющихся потребностей заинтересованных сторон. Модульность является основным фактором качества при изменении требований и поэтапной разработке. Низкая и высокая связность могут создать пользовательские истории с высокой модульностью.

(2) *Качество управления (контролируемость и управляемость)*: система управления конфигурацией может управлять всеми документами этапа разработки, контролировать версии продукта и взаимосвязь перекрестных ссылок документов. Механизм управления конфигурацией должен учитывать следующие три фактора:

- Система управления конфигурацией: удобная и практичная процедура управления конфигурацией используется для управления соответствующими операциями по регистрации и выдаче документов.
- *Контроль версий*: система управления конфигурацией должна сочетать подходящие и практичные инструменты контроля версий для обработки, записи и хранения различий между версиями пользовательских историй.
- *Таблицы перекрестных ссылок*. Таблицы перекрестных ссылок документов этапа разработки являются важным элементом для конкретного установления взаимосвязей и прослеживаемости пользовательских историй.

(3) *Приемочное (подтверждаемое) качество*. Одной из важных задач пользовательской истории является помощь в проведении приемочных испытаний. Следовательно, пользовательская история нуждается в квалифицированном содержании, обеспечивающем высокую тестируемости для проверки как программного обеспечения, так и системы в целом.

- *Гарантированное содержание*: для эффективного достижения общего консенсуса между заказчиками, разработчиками и заинтересованными сторонами пользовательская история должна иметь правильное, полное и последовательное содержание.
- *Тестируемость*: для подтверждения функций сситемы, пользовательские истории должны иметь высокую тестируемость. Тестируемая пользовательская история может помочь тестировщикам составить план тестирования и выполнить тестовые примеры, чтобы гарантировать приемлемые функции системы в целом.
- *Проверяемость*: в гибкой разработке программного обеспечения пользовательские истории являются основой поведенческой, а также итеративной и инкрементной разработки. Поддающиеся проверке пользовательские истории могут помочь обеспечить правильность поведения, полноту встроенной функциональности и проверку приемлемости разработанной системы.

Архитектура факторов качества пользовательских историй представлена на рис. 2.


*Рис. 2. Архитектура факторов качества пользовательских историй*

Для сбора полезных факторов необходимо разработать набор полных и четких контрольных списков для проведения обсуждения пользовательской истории. Контрольные списки могут обнаруживать и идентифицировать качественные и неполные дефекты пользовательской истории. Обсуждение и реализация пользовательской истории могут помочь собрать обсуждаемые, оцениваемые, контролируемые, управляемые и подтверждаемые факторы качества пользовательской истории.

## 3. Повышение качества пользовательских историй

Повышение качества пользовательских историй может эффективно уменьшить влияние изменений требований и снизить риски разработки системы.

При оценке пользовательских историй необходимо учитывать три характеристики качества, которые включают базовое качество, качество управления и качество приемки. На основе вышеизложенного можно сгенерировать интегрированный показатель качества пользовательской истории в соответствии со следующей последовательностью оценок:

(1) Измерение базового (обсуждаемого и оцениваемого) качества ($Q_{\text{о}}$) сочетает в себе ясность документов, низкую сложность и модульность трех показателей:

$$Q_{\text{о}} = Q_{\text{о}1}w_{\text{о}1} + Q_{\text{о}2}w_{\text{о}2} + Q_{\text{о}3}w_{\text{о}3}, \quad (1)$$

где $Q_{\text{о}1}$ – ясность содержания, $Q_{\text{о}2}$ – низкая сложность, $Q_{\text{о}3}$ – высока модульность, $w_{\text{о}i}, i = \overline{1,3}$ – вес соответствующего показателя, $w_{\text{о}1} + w_{\text{о}2} + w_{\text{о}3} = 1$.

(2) Измерение контролируемости и управляемости качества ($Q_{\text{к}}$) представляет собой комбинированную систему управления конфигурацией, инструменты контроля версий и показателя таблицы перекрестных ссылок:

$$Q_{\text{к}} = Q_{\text{к}1}w_{\text{к}1} + Q_{\text{к}2}w_{\text{к}2} + Q_{\text{к}3}w_{\text{к}3}, \quad (2)$$

где $Q_{\text{к}1}$ – система управления конфигурацией, $Q_{\text{к}2}$ – контроль версий, $Q_{\text{к}3}$ – таблица перекрестных ссылок, $w_{\text{к}i}, i = \overline{1,3}$ – вес соответствующего показателя, $w_{\text{к}1} + w_{\text{к}2} + w_{\text{к}3} = 1$.

(3) Измерение качества приемки ($Q_{\text{п}}$) сочетает в себе показатели гарантированного содержания, тестируемости и проверяемости:

$$Q_{\text{п}} = Q_{\text{п}1}w_{\text{п}1} + Q_{\text{п}2}w_{\text{п}2} + Q_{\text{п}3}w_{\text{п}3}, \quad (3)$$

где $Q_{\text{п}1}$ – гарантированное содержание, $Q_{\text{п}2}$ – метрика тестируемости, $Q_{\text{п}3}$ – метрика проверяемости, $w_{\text{п}i}, i = \overline{1,3}$ – вес соответствующего показателя, $w_{\text{п}1} + w_{\text{п}2} + w_{\text{п}3} = 1$.

(4) Объединение $Q_{\text{о}}$, $Q_{\text{к}}$ и $Q_{\text{п}}$ в интегрированный показатель качества пользовательской истории ($Q_{\text{и}}$):

$$Q_{\text{и}} = Q_{\text{и}1}w_{\text{и}1} + Q_{\text{и}2}w_{\text{и}2} + Q_{\text{и}3}w_{\text{и}3}, \quad (4)$$

где $w_{\text{и}i}, i = \overline{1,3}$ – вес соответствующего показателя, $w_{\text{и}1} + w_{\text{и}2} + w_{\text{и}3} = 1$.

Трехуровневая комбинация соотношений для $Q_{\text{и}}$ является моделью измерения качества пользовательской истории (МИКПИ).

$Q_{\text{и}}$ — это механизм относительной оценки, который также является основой для выявления проблем или дефектов пользовательских историй. В МИКПИ основные факторы качества объединяются в измерения качества высокого уровня. Измерения качества высокого уровня объединяются в $Q_{\text{и}}$. Следовательно, если $Q_{\text{и}}$ не удовлетворяет критерию качества, это означает, что в пользовательских историях существовали некоторые связанные дефекты. В соответствии с комбинированными формулами измерения качества сопоставляются трем уравнениям измерений и нескольким факторам качества. Связанные действия должны тщательно проверены для выявления проблемы или дефект, и на их основе можно предложить корректирующие действия.

Для повышения качества пользовательских историй рекомендуется следовать следующим правилам:

**<Правило 1>** ЕСЛИ $Q_{\text{и}}$ не удовлетворяет критерию качества пользовательской истории, ТО следует проанализировать $Q_{\text{о}}$, $Q_{\text{к}}$ и $Q_{\text{п}}$ для выявления проблем и дефектов соответствующих документов или действий.

**<Правило 2>** ЕСЛИ $Q_{\text{о}}$ не удовлетворяет обсуждаемому и оцениваемому критерию качества, ТО следует проанализировать факторы качества ясности содержания, низкой сложности и высокой модульности для выявления дефектов связанных документов или действий.

**<Правило 3>** ЕСЛИ $Q_{\text{к}}$ не удовлетворяет контролируемому и управляемому критерию качества, ТО система управления конфигурацией, инструменты контроля версий и факторы качества таблицы перекрестных ссылок должны быть проанализированы для выявления дефектов связанных документов или действий.

**<Правило 4>** ЕСЛИ $Q_{\text{п}}$ не удовлетворяет качеству приемки, ТО должны быть проанализированы факторы качества коммуникационных возможностей, тестируемости и верифицируемости для выявления дефектов связанных документов или действий.

## 4. Процедура улучшения качества пользовательской истории

Изменение пунктов требований – это проблема, с которой может столкнуться проект разработки программного обеспечения, основанный на методологии гибкой разработки. Способность преодолеть влияние изменения требований позволяет эффективно снизить риски разработки программного проекта. На основе модели МИКПИ разработана процедура улучшения качества пользовательской истории (элементов требований) для своевременного исправления дефектов качества пользовательской истории и конкретного улучшения гибкой разработки и качества управления изменениями. Процедура улучшения делится на пять этапов (рис. 3):

- Выбор пользовательских историй:
  - Выбор пользовательских историй;
  - выбор из карточек пользовательских историй наиболее важное и подходящее содержание истории в зависимости от функций;
  - прохождение выбранной пользовательской истории процедуры отбора пользовательских историй.
- Настройка пользовательской истории на основе поведенческой разработки:
  - для достижения согласованности удаление похожих и повторяющихся пользовательских историй;
  - для избегания конфликта объединение пользовательских историй с высокой корреляцией;
  - приведение описания пользовательских историй к единому формату.
- Измерение качества:
  - сбор факторов качества пользовательской истории при обсуждении пользовательской истории;
  - количественная оценка факторов качества пользовательской истории;
  - применение МИКПИ для измерения качества пользовательской истории.
- Выявление дефектов на основе предопределенных критериев качества для определения качества пользовательской истории:
  - определение дефектов качества пользовательской истории;
  - подтверждение дефектов качества пользовательской истории.
- Исправление дефектов качества:
  - применение способа коррекции качества на основе правил для пересмотра пользовательской истории;
  - проведение формальной процедуры проверки пересмотренной пользовательская истории;
  - сохранение новой версии пользовательской истории в исходном репозитории для контроля версий.

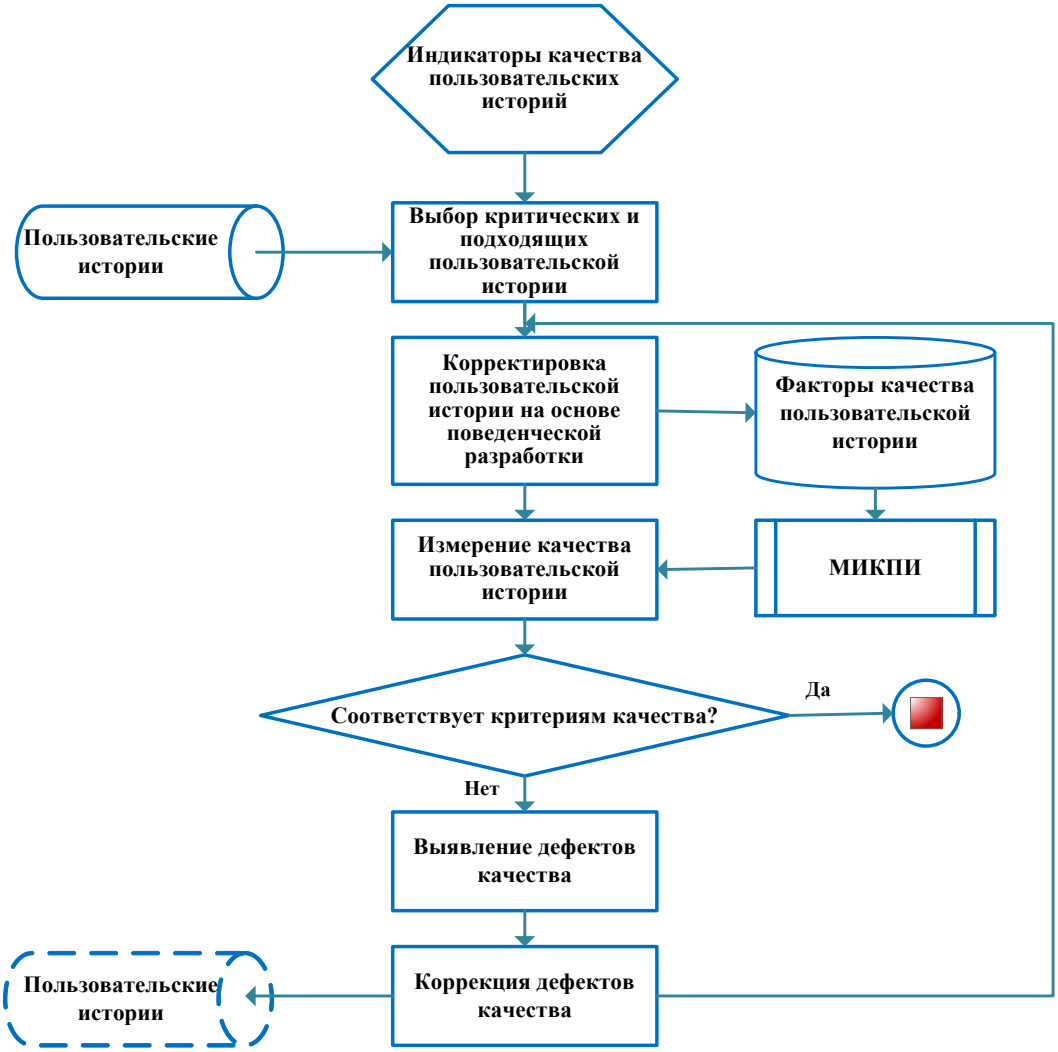


*Рис. 3. Архитектура факторов качества пользовательских историй*

## 5. Заключение

Изменения требований часто приводят к неудаче проекта, связанного с разработкой программного обеспечения. Для снижения рисков проекта разработки программного обеспечения необходимо преодолевать трудности, связанные с изменением требований. Создание пользовательской истории становится важной основой для гибкой разработки программного обеспечения и управления изменениями требований. Чтобы справиться с изменением требований и снизить риск разработки программного обеспечения, в докладе проанализированы и собраны критические факторы качества пользовательских историй и предложена модель измерения качества пользовательских историй (МИКПИ). Применяемая модель МИКПИ может повысить качество требований к разработке и эффективно обрабатывать изменения требований, чтобы снизить риск разработки программного обеспечения.

На основе модели МИКПИ процедура улучшения качества пользовательской истории применяется для повышения качества пользовательской истории и снижения рисков гибкой разработки программного проекта.

## Литература


1. *Kretsou M., Arvanitou E.-M., Ampatzoglou A., Deligiannis I., Gerogiannis V.C.* Change impact analysis: A systematic mapping study // Journal of Systems and Software. – 2021. – Vol. 174. – Article 110892.
2. *Логинов И.В., Фролов В.А.* Метод анализа динамики изменения требований назначения многофункциональных программно-технических систем // Экономика. Информатика. – 2021. – Т. 48, N 3. – С. 528–542.
3. *Maâzoun J., Bouassida N., Ben-Abdallah H.* Change impact analysis for software product lines // Journal of King Saud University - Computer and Information Sciences. – 2016. – Vol. 48, Iss. 4. – P. 364–380.

4. *Jharko E.* Life Cycle and Quality Assurance of Software for Systems of Critical Information Infrastructure Facilities // 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). – Sochi, Russia, 2021. – P. 440-444.
5. *Butt S., Khan S.U.R., Hussain S., Wang W.-L.* A conceptual model supporting decision-making for test automation in Agile-based Software Development // Data & Knowledge Engineering. – 2023. – Vol. 144. – Article 102111.
6. *Stray V., Hoda R., Paasivaara M., Lenarduzzi V., Mendez D.* Theories in Agile Software Development: Past, Present, and Future Introduction to the XP 2020 Special Section // Information and Software Technology. – 2022. – Vol. 152. – Article 107058.
7. *Palopak Y., Huang S.-J., Ratnasari W.* Knowledge diffusion trajectories of agile software development research: A main path analysis // Information and Software Technology. – 2023. – Vol. 156. – Article 107131.
8. *Tøndel I.A., Cruzes D.S., Jaatun M.G., Sindre G.* Influencing the security prioritisation of an agile software development project // Computers & Security. – 2022. – Vol. 118. – Article 102744.
9. *Жарко Е.* Управление требованиями, верификация и валидация программного обеспечения АСУ ТП АЭС // Материалы 30-й Международной конференции «Проблемы управления безопасностью сложных систем» (Москва, 2022). – М.: ИПУ РАН. – С. 162-166.
10. *Зайцев А.С.* Оценка актуальности развития Российской экономики в инновационном направлении // Интеллект. Инновации. Инвестиции. – 2010. – N. 4. – С. 10-14.
11. *Arifin N.F., Purwandari B., Setiadi F.* Evaluation and Recommendation for Scrum Implementation Improvement with Hybrid Scrum Maturity Model: A Case Study of A New Telco Product // 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). – Jakarta, Indonesia, 2020. – P. 178-183.