



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004

## РАЗРАБОТКА СЕРВИСА АВТОМАТИЧЕСКОЙ ПРОСТАНОВКИ ТЕГОВ НА ЭЛЕМЕНТЫ БЭКЛОГА ИТ-ПРОЕКТА

**Гадельшин И.Р.**

*ФГБОУ ВО "Уфимский государственный нефтяной технический университет", Уфа, Россия (450064, Республика Башкортостан, г. Уфа, ул. Космонавтов, д. 1) e-mail: insar2000@yandex.ru*

При разработке сервиса автоматизации постановки тегов на задачи в бэклогах команд разработки, который рассчитан на большое количество ИТ-проектов, возникает ряд технических сложностей. Во-первых, необходимо обеспечить гибкость условий тегирования рабочих элементов для разных бэклогов. Во-вторых, нужно обеспечить автономную работу сервиса на протяжении длительного времени. В статье описаны технические решения, которые позволили реализовать сервис универсальным и тиражировать его на различные команды разработки. Приведена диаграмма последовательности, описывающая последовательность действий сервиса при постановке тегов.

Ключевые слова: Гибкие методологии, ИТ-проект, разработка ПО, управление разработкой.

## DEVELOPMENT OF A SERVICE FOR AUTOMATIC TAGGING OF IT PROJECT BACKLOG ELEMENTS

**Gadelshin I.R.**

*Ufa State Petroleum Technological University, Ufa, Russia (450064, Republic of Bashkortostan, Ufa, Cosmonauts St,1) e-mail: insar2000@yandex.ru*

When developing a service for automating tagging of work items in development teams' backlogs, which is designed for a large number of IT projects, a number of technical difficulties arise. Firstly, it is necessary to provide flexibility in the conditions for tagging work items for different backlogs. Secondly, we have to ensure the autonomous operation of the service for a long time. The article describes technical solutions that made it possible to implement a universal service and replicate it to various development teams. A sequence diagram is provided that describes the sequence of actions of the service during the tagging process.

Keywords: Agile, IT project, software development, development control.

Качество постановки задач является одним из ключевых факторов успешности процесса разработки программного обеспечения, так как от него зависит количество возникающих ошибок и доработок при реализации функционала. Сервис автоматической постановки тегов на рабочие элементы в бэклоге программного продукта помечает пользовательские истории и ошибки, не соответствующие критериям готовности к взятию в работу, тегами. Данные теги повышают прозрачность бэклога проекта и наглядно показывают недостающие атрибуты рабочих элементов. Благодаря этому сервис способствует повышению качества проработки задач, а также снижению трудозатрат на контроль качества бэклога.

В процессе разработки данного сервиса возникла проблема его масштабирования на бэклоги всех проектов компании ввиду различия методологий разработки (Scrum и Kanban), названий рабочих элементов (ТТ – техническое требование и User Story), а также требований команд к правилам простановки тегов. Необходим сервис, который будет гибким и настраиваемым под каждый проект. В статье описана реализация сервиса для бэклогов, которые ведутся в Azure DevOps Server (ADS), но с небольшими доработками он может быть использован с другими трекерами задач.

Правила простановки тегов описывают, какие проверки рабочих элементов мы проводим с помощью системы автоматизации. Они представляют собой условия, которым должен соответствовать рабочий элемент для простановки на него определенного тега.

Для достижения гибкости сервиса необходимо разделить структуру правила на механизм (общая часть правила для всех проектов) и параметры (настраиваются под каждый проект). В качестве примера рассмотрим правило, по которому сервис проставляет тег "нет ТК" на User Story без тестового сценария.

Проставлять тег [название тега] на User Story, у которых:

- не стоят теги tech и meth;
- нет ссылки типа «Tested by»;
- статус совпадает с [статусы правила].

В данном случае проверка наличия тегов tech и meth, а также ссылки на тест-кейс являются механизмом правила. В качестве параметров выступают название тега, а также статусы рабочих элементов, которые проверяются с помощью данного правила.

Командам, работающим по Scrum, при проверке тегов целесообразно ориентироваться на принадлежность рабочего элемента определенной итерации [1]. В проверке участвуют задачи текущего и предстоящего спринта. Для команд, использующих Kanban, проверка рабочих элементов осуществляется с учётом их статусов. В зависимости от текущего статуса рабочего элемента у него могут проверяться различные атрибуты.

Подключение разрабатываемого сервиса к бэклогу каждого из проектов потребует наличия в настройках персонального токена доступа к проекту в ADS и URL-адреса проекта для формирования API-запросов, а также запроса в формате WIQL для получения id рабочих элементов [3].

Для реализации возможности запуска сервиса для проектов с параметрами, описанными выше, необходимо создать конфигурационный файл с параметрами каждого проекта. Данный файл было решено реализовать в формате XML (Extensible Markup Language). Это язык разметки, который используется для хранения и передачи данных [4]. В структуры данного файла «Параметры проектов» — корневой элемент. Он содержит в себе элементы типа «Проект» и их атрибуты (имя проекта, токен, WIQL-запрос и URL бэклога). Проекты содержат в себе элементы типа «Правило», а также их атрибуты. Атрибуты включают в себя тип правила для сопоставления параметров правила с его структурой, название проставляемого тега и статусы рабочих элементов, на которые распространяется данное правило.

В рамках разработки сервиса было выдвинуто требование: проверка бэклогов проектов должна осуществляться в автоматическом режиме. Для реализации данного требования было решено реализовать сервис простановки тегов на основе службы рабочих ролей .NET [5]. Их основное преимущество — возможность длительного выполнения фоновых процессов без

участия пользователя. Данные службы также поддерживают механизм логирования, что обеспечивает возможность отслеживания выполнения задач и быстрое выявление ошибок в процессе работы.

На Рисунке 1 приведена диаграмма последовательности простановки тегов с помощью описанного сервиса в формате UML.

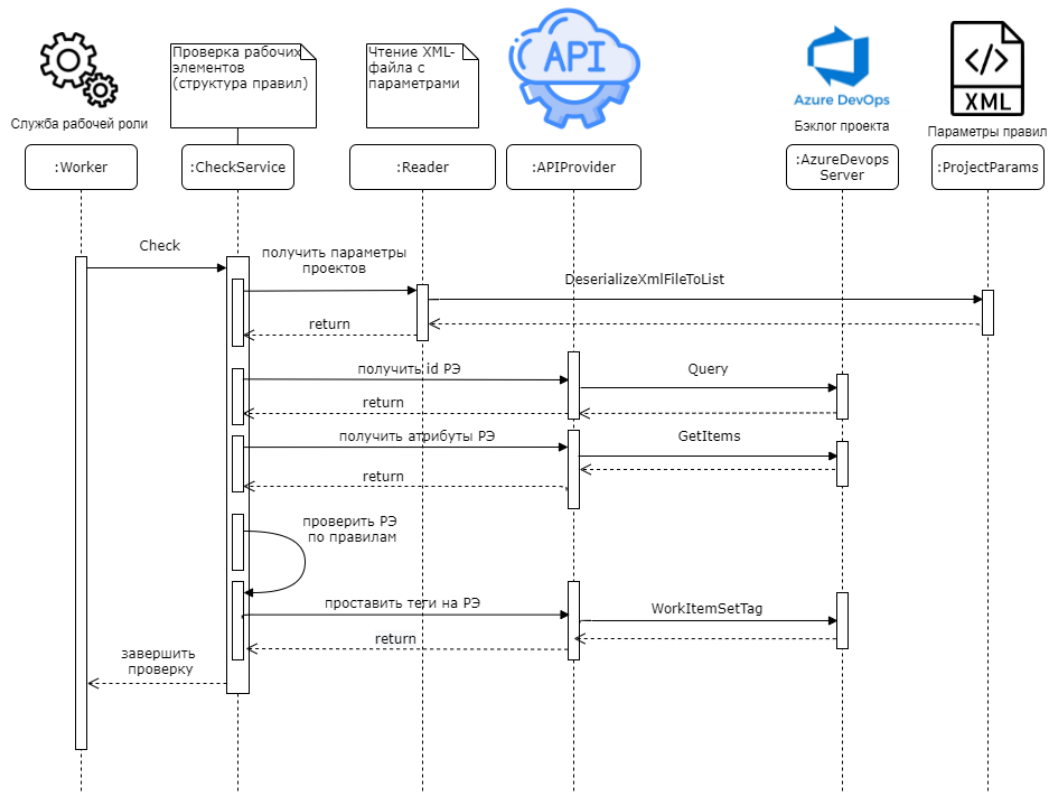


Рисунок 1 – Диаграмма последовательности простановки тегов

Источник: работа автора

Класс Worker содержит метод, который запускает метод Check проверки рабочих элементов. Метод Check, в свою очередь, обращается к методу DeserializeXmlFileToList для получения параметров проекта, а также параметров каждого его правила из XML-файла. Затем метод Check запускает метод Query класса APIProvider, с помощью которого отправляется запрос в AzureDevOps Server для получения id рабочих элементов бэклога. В теле данного запроса передается запрос в формате WIQL, который получен из параметров проекта. Затем метод Check запускает метод GetItems, с помощью которого сервис получает атрибуты рабочих элементов по их id. Теперь сервис располагает всеми данными, необходимые для проверки рабочих элементов по правилам. Если рабочий элемент соответствует условию правила, то на него проставляется соответствующий тег с помощью метода WorkItemSetTag. После проверки всех рабочих элементов выбранного проекта процесс простановки тегов завершается, и сервис переходит к следующему проекту.

На основе технических решений, описанных в данной статье, был разработан сервис автоматической простановки тегов, позволяющий отслеживать качество проработки задач, благодаря чему команда разработки может проводить мероприятия по дополнению задач

необходимыми атрибутами до их передачи разработчику. Благодаря этому повышается качество задач, а также снижается риск возникновения ошибок в готовом продукте.

### Список литературы

1. Сазерленд, Д. Scrum. Революционный метод управления проектами, 2-е изд. / пер. с англ. М. Гескина - М.: «Манн, Иванов и Фербер», 2017.- 288 с.
2. Дэвид Андерсон Канбан. Альтернативный путь в Agile / Андерсон Дэвид - М.: Манн, 2010. - 216 с.
3. Справочник по синтаксису языка запросов рабочих элементов (WIQL) // статья от 07.03.2023г.// [Электронный ресурс] URL: <https://learn.microsoft.com/Ru-Ru/azure/devops/boards/queries/wiql-syntax?view=azure-devops-2022> (дата обращения: 10.03.2023).
4. Старыгин А.А. XML: разработка web-приложений. -СПб.: БХВ-Петербург, 2003.
5. Службы рабочих ролей в .NET // статья от 09.03.2023г.// [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/dotnet/core/extensions/workers> (дата обращения: 12.03.2023).

### References

1. Sutherland, D. Scrum. The revolutionary method of project management, 2nd ed. / translated from the English by M. Geskin - M.: "Mann, Ivanov and Ferber", 2017.- p. 288
  2. David Anderson Kanban. An alternative path to Agile / Anderson David - M.: Mann, 2010. - p.216
  3. Handbook on the syntax of the working element Query language (WIQL) // article dated 07.03.2023// [Electronic resource] URL:<https://learn.microsoft.com/Ru-Ru/azure/devops/boards/queries/wiql-syntax?view=azure-devops-2022> (accessed 10.03.2023).
  4. Starygin A.A. XML: web application development. -St. Petersburg: BHV-Petersburg, 2003.
  5. Work role services in .NET // article from 09.03.2023// [Electronic resource] URL: <https://learn.microsoft.com/ru-ru/dotnet/core/extensions/workers> (accessed 12.03.2023).
-