

минимальному пороговому значению, выносятся в группу учащенно встречающихся.

В зависимости от размера наиболее длинного часто встречающегося набора алгоритм Apriori сканирует базу спектров определенное количество раз. Вариации алгоритма Apriori, являющиеся его оптимизацией, предложены для сокращения количества сканирований, количества наборов-кандидатов или же того и другого. Например AprioriTID и AprioriHybrid [2].

Как результат, после анализа и преобразования данных спектрограмм к бинарному виду это позволит предоставлять рекомендации с указанной достоверностью при попытке обнаружения автора неизвестного художественного произведения. Однако, стоит отметить, что данный алгоритм и результат его работы не следует использовать для 100% идентификации автора и использовать только в роли первичной идентификации.

Список использованных источников:

1. Усиков А. В. Решение задачи поиска ассоциативных правил как часть аналитического сервиса, предоставляющей рекомендации // Информационные технологии и системы 2015 (ИТС 2015): материалы международной научной конференции (БГУИР, Минск, Беларусь, 28 октября 2015). - Information Technologies and Systems 2015 (ITS 2015): Proceeding of the International Conference (BSUIR, Minsk, Belarus, 28th October 2015) / редкол. : Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2015. – С. 142-143
2. Agrawal, R., Imielinski T. and Swami A. Mining association rules between sets of items in large databases. / R. Agrawal// In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., – May 1993.

## **ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ РАБОЧИХ ПРОЦЕССОВ СТУДИИ ДИЗАЙНА**

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Хадасевич А.И.*

*Болтак С.В. – ассистент каф. ПОИТ*

Для современных компаний, которые производят определенный продукт или оказывают услуги, жизненно необходим контроль и организация бизнес-процессов. С этой целью компании используют средства автоматизации, которые позволяют следовать процессам с наибольшей скоростью, наименьшими затратами и максимальной эффективностью.

Крупные студии дизайна, которые работают над большими проектами сталкиваются с определенными проблемами. Это проблемы управления и контроля проектов, эффективного распределения рабочего времени сотрудников, получения отзыва и обратной связи. Различная информация и количество версий накапливаются по экспоненциальному закону, что приводит к тому, что невозможно найти то, что нужно. Все это приводит к увеличению времени на создание конечного продукта, и как следствие его стоимости.

Приложение рабочего процесса (от англ. Workflow application) – это программное средство, которое автоматизирует, по меньшей мере, некоторую часть процесса или процессов. Процессы, как правило, связаны с бизнес-процессами, но могут быть любым процессом, который требует серии шагов для автоматизации с помощью программного обеспечения. На некоторых этапах процесса может потребоваться вмешательство человека, например, утверждение или разработка пользовательского текста, но функции, которые могут быть автоматизированы, должны обрабатываться приложением.

Для студий дизайна приложения данного типа актуальны, если выстроены рабочие процессы в команде. Например, если у компании заказ на создание брошюры или буклета, то такими процессами могут быть написание текста, поиск или создание нужных изображений и создание общего дизайна. Эти процессы должны быть структурированы и выполняться по определенным правилам. Например, эти правила могут определять, какие исходные данные должны быть у сотрудника, какой должен быть результат его работы, и кто оценивает и одобряет конечный результат.

Важнейшей особенностью данных приложений является поддержка управления процессами, содержащими как автоматизированные - выполняемые средствами информационных систем, так и неавтоматизированные - выполняемые вручную операции.

Одной из специфик студий дизайна, является работа с множеством версий различных файлов. К тому времени, когда текст буклета или изображение будет одобрено, могут быть созданы десятки версий одного файла. Как правило, в рамках одного бизнес-процесса идет работа с одним набором файлов. Также над одним и тем же файлом в разное время могут работать разные люди. Решением этой проблемы может являться интеграция системы контроля версий в приложения рабочего процесса. Такие системы позволяют хранить несколько версий одного и того же файла, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение.

Таким образом можно сформулировать основные цели создания программного средства:

- 1) описание и автоматизация бизнес-процессов, которые присутствуют в компании;

- 2) упрощение работы сотрудников, которые непосредственно ответственны за создание продукта и сотрудников, которые контролируют и управляют проектом;
- 3) организация системы контроля версий для файлов, которые участвуют в процессах;
- 4) организация взаимодействия и общения сотрудников.

Программное средство автоматизации рабочих процессов студии дизайна предусматривает выполнение основных функций:

- 1) создание и управление различными рабочими процессами;
- 2) наличие готового списка самых распространенных процессов;
- 3) наличие возможности добавления и изменения версий файлов, которые участвуют в процессах;
- 4) распределение ролей в рамках одного проекта (менеджеры и исполнители);
- 5) возможность контроля за состоянием проекта со стороны менеджеров;
- 6) возможность добавить комментарии и пометки к файлам, участвующим в процессе.

Разработанное программное средство позволяет студиям дизайна эффективно и качественно создавать различные информационные продукты. Благодаря автоматизации процессов сокращается время и соответственно цена конечного продукта, что выгодно выделяет данную компанию на фоне конкурентов.

Список использованных источников:

1. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#/ Рихтер Дж., Пер. с англ. – М.: Издательский дом «Вильямс», 2012. – 896 с.
2. Carstensen, P.H.; Schmidt, K. Computer supported cooperative work: new challenges to systems design/ Grudin, J. – Tokyo, 1999. – 19 с.

## АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Хильчук А.С.*

*Куликов С.С. – канд. техн. наук, доцент*

В настоящее время все больше появляется приложений, взаимодействующих с пользователем с помощью веб-интерфейса, за отображение которого отвечает веб-браузер. Как правило, страницы для браузера генерируются программным продуктом во время его работы. За продолжительное время использования в такие приложения добавляются новые функциональные возможности, изменяются существующие или даже наоборот – удаляются функции, которые уже не нужны. Такие изменения могут приводить к появлению нарушений в отображении страниц, а также к отказам самого ПО.

Тестирование ПО – это процесс исследования, испытания программного продукта, по результатам которого можно выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации. [1]

Тестирование, направленное на то, чтобы удостовериться в отсутствии ошибок в ранее работоспособной функциональности, которые могли возникнуть в результате изменений в приложении или среде его функционирования, называется регрессионным. [2] Этот вид тестирования практически всегда присутствует в плане по обеспечению качества ПО, т.к. по статистике исправление одного дефекта с высокой вероятностью влечёт появление нового. Причинами появления одних и тех же ошибок могут быть сами разработчики, слабая техника управления системами контроля версий ПО, неактуальность исправления дефекта в последующих сборках и т.д.

Регрессионное тестирование может занимать до 90% от общего объёма времени, затрачиваемого на проверку очередной сборки ПО. Для снижения данных затрат внедряется автоматизация проведения тестирования. Автоматизированное тестирование ПО – это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически, что помогает сократить время тестирования и упростить его процесс. [3]

Также с помощью автоматизированного тестирования можно осуществить нагрузочное тестирование, которое является актуальным для образовательных ресурсов. Данный вид тестирования направлен на исследование способности приложения сохранять заданные показатели качества при нагрузке в допустимых пределах и некотором превышении этих пределов. [2] Автоматизация его выполнения позволяет имитировать нагрузку на приложение посредством выполнения разного рода действий большого количества виртуальных пользователей.

Одним из наиболее популярных фреймворков для создания автоматизированных тестов веб-приложений является Selenium, использующий веб-драйвер для взаимодействия с браузером, который, в свою очередь, взаимодействует с самим веб-приложением. Данные веб-драйверы разрабатываются для браузеров Firefox, Internet Explorer и Safari, Opera, Microsoft Edge, а также для мобильных браузеров Android и iOS. Аналогично и с фреймворком – он доступен для языков Java, .Net (C#), Python, Ruby, JavaScript.

Частым архитектурным решением при проектировании автоматизированных тестов является паттерн PageObject. Основными его преимуществами является инкапсулирование работы с отдельными элементами веб-страницы, что позволяет уменьшить количество строк кода, а также его поддержку. Также к плюсам можно