УДК 004.415.5

ВЕРИФИКАЦИЯ КАЧЕСТВА ПО И АВТОМАТИЗИРОВАННАЯ СИСТЕМА МОНИТОРИНГА РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

Д.А. СТОРОЖЕВ

Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
d.storozhev@gmail.com

Качество программного обеспечения (ПО) – это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности. Верификация качества - это совокупность действий проводимых над объектом тестирования в процессе разработки для получения информации об актуальном состоянии объекта тестирования в разрезах: готовность продукта к выпуску, соответствие зафиксированным требованиям, соответствие заявленному уровню качества продукта.

Ключевые слова: качество ПО, верификация ПО, тестирование ПО, автоматизированная система.

На данный момент наиболее распространена и используется многоуровневая модель качества программного обеспечения, представленная в наборе стандартов ISO 9126. На верхнем уровне выделено 6 основных характеристик качества ПО: функциональность, надежность, удобство использования, эффективность, портативность, удобство сопровождения.

Выполнением тестирования и предоставлением результатов при полуавтоматизированной системе занимается специалист по тестированию, который запускает тесты и получает результаты в виде JUnit или TestNG отчетов. Если система автоматизирована полностью, выполнением тестирования и предоставлением результатов занимается сервер процесса непрерывной интеграции (Hudson, Jenkins и др.). Недостатком таких серверов является небольшой объем хранения с невозможностью фильтрации\поиска полученных результатов.

Разработанная система мониторинга результатов тестирования устраняет данный недостаток, предоставляя возможность отображать результаты автоматических тестов в удобной для пользователя форме с дополнительной функцией фильтрации\поиска. Система связывает модуль выполнения тестов (как при запуске вручную, так и с помощью CI-сервера) с модулем представления результатов тестирования.

Система функционирует в два потока – входной и выходной. Входной поток принимает результаты выполненных тестов, выходной поток эти результаты отображает в веб-браузере. Ядром системы являются два сервиса, один из которых принимает результат, обрабатывает его и записывает в базу, а другой занимается получением результатов из базы по запросу пользователя.

Связь системы с модулем выполнения осуществляется через специальный модуль логгирования, поставляемый вместе с системой. Основой модуля является стандартная log4j библиотека, а специальная надстройка, или аппендер, на каждое тестовое событие (onStart, onFinish, onSkip) вызывает методы обработки результатов выполнения тестов. Преимуществом модуля является то, что результат тестов, созданных с помощью API различный тестовых фрэймворков (JUnit, TestNG и др.) трансформируется в единый интерфейс Result, который обрабатывается входным сервисом.

На рис. 1 представлена архитектура созданной системы мониторинга результатов тестирования.

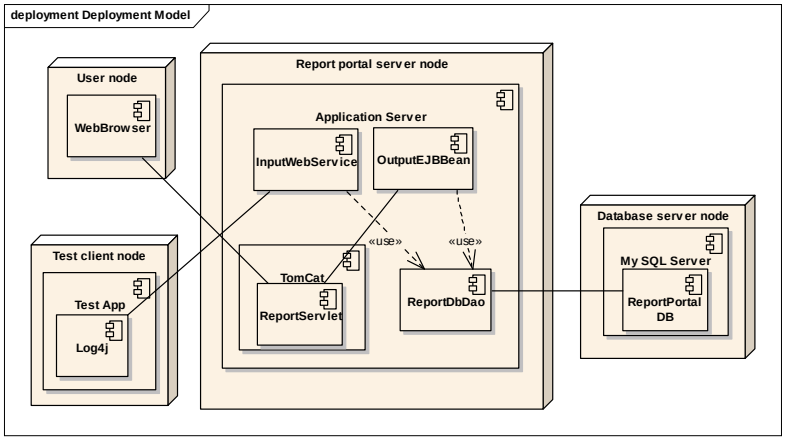

Рис. 1. Архитектура системы мониторинга результатов тестирования

Пользователь запускает наборы функциональных тестов любым образом, ничего не зная о внутреннем устройстве системы мониторинга. Единственной опцией, необходимой для оповещения системы является cmd-параметр, подключающий модуль логгирования. Тесты выполняются, результаты заносятся в базу входным сервисом. Далее пользователь для получения представления результатов заходит в веб-браузер, авторизуется и получает результаты выполнения тестов по сьютам. На рис. 2 представлена работа системы мониторинга результатов тестирования для одного пользователя.

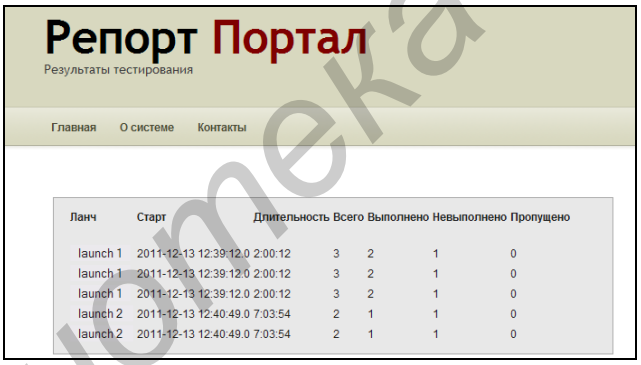

Рис. 2. Результаты тестовых запусков пользователя

Таким образом, система мониторинга результатов тестирования решает задачу хранения и представления результатов выполнения автоматических тестов, представляет возможность поиска и фильтрации результатов в удобном для пользователя виде, встраивается в процесс непрерывной интеграции, связывая модуль выполнения тестов с модулем представления результатов, тем самым повышая эффективность тестирования и улучшая качество разработки программного обеспечения в целом.

Список литературы

1. *Фаулер М.* Архитектура корпоративных программных приложений: пер. с англ. М.: Изд. дом «Вильямс», 2010.

2. *Хорстманн К. С., Корнелл Г.* Библиотека профессионала. Java 2. Том 1, Том 2. Изд. дом «Вильямс», 2009.