

ВЕРИФИКАЦИЯ КАЧЕСТВА ПО И АВТОМАТИЗИРОВАННАЯ СИСТЕМА МОНИТОРИНГА РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

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

Соколовская М. А.

Сторожев Д. А. – преподаватель

Качество программного обеспечения - это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности. Общеизвестными стандартами в области качества являются: Capability Maturity Model (CMM) – модель оценки уровня зрелости процессов разработки вместе с его производными; Capability Maturity Model Integrated (CMMI) – интегрированная модель оценки уровня зрелости процессов разработки; стандарты ISO 9000 – обширная и наиболее распространенная во всем мире серия стандартов качества; модель шести сигм; ITIL (IT Infrastructure Library) - ориентирована на обеспечение функционирования IT-инфраструктуры. 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.

Согласно международному стандарту ISO 14598: метрика - это количественный масштаб и метод, который может использоваться для измерения. Выделяют следующие метрики качества: метрики по тестовым случаям; метрики по багам / дефектам; метрики по задачам.

Обеспечение качества - это совокупность мероприятий, охватывающих все технологические этапы разработки, выпуска и эксплуатации программного обеспечения (ПО) информационных систем, предпринимаемых на разных стадиях жизненного цикла ПО, для обеспечения качества выпускаемого продукта.

Верификация качества - это совокупность действий проводимых над объектом тестирования в процессе разработки для получения информации об актуальном состоянии объекта тестирования в разрезе: "готовность продукта к выпуску", "соответствие зафиксированным требованиям", "соответствие заявленному уровню качества продукта".

Тестирование программного обеспечения - это одна из техник контроля качества, включающая в себя

активности по планированию работ, проектированию тестов, выполнению тестирования и анализу полученных результатов.

Выполнением тестирования и предоставлением результатов при полу автоматизированной 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, etc.). Недостатком CI 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 различных тестовых фреймворков (JUnit, TestNG, HtmlUnit, etc.) трансформируется в единый интерфейс Result, который обрабатывается входным сервисом.

На рисунке 1 представлена архитектура созданной системы мониторинга результатов тестирования.

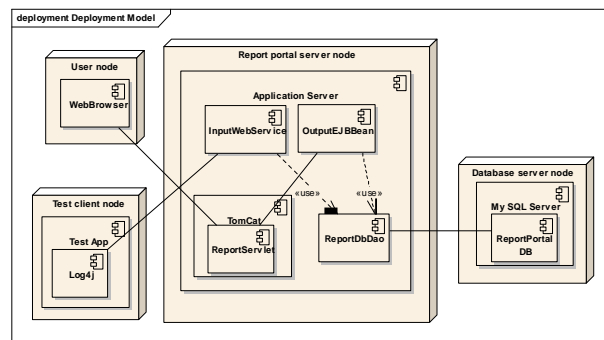


Рисунок 1 – архитектура системы мониторинга результатов тестирования

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 рисунке 2 представлена работа системы мониторинга результатов тестирования для одного пользователя.

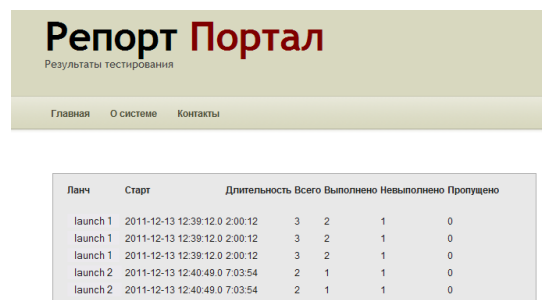


Рисунок 2 – результаты тестовых запусков пользователя

Для каждого сьюта доступны результаты тестов, которые в нем выполнялись с возможностью детализации логгирования по шагам. Кроме того, пользователь может осуществить поиск/фильтрацию по дате, имени сьюта, теста, запуска, номеру билда. Также доступна опция построения гистограммы тестирования билдов и тестовых сьютов.



Рисунок 3 – гистограмма тестирования билдов

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 использованных источников:

1. Фаулер М. Архитектура корпоративных программных приложений.: Пер. с англ. - М.: Издательский дом "Вильямс", 2006. - 544с.: ил. - Парал. тит. англ.
2. Хорстманн К. С, Корнелл Г. Библиотека профессионала. Java 2. Том 1, Том 2. Основы.: Пер. с англ. — М.: Издательский дом "Вильямс", 2003. — 848 с: ил. — Парал. тит. англ.