Библиографические ссылки

1. Отчет по НИР по контракту с Министерством труда УР от 23 августа 2010 № 28/МТ-10 на тему «Разработка модели прогнозирования и управления рисками повреждения здоровья работающими» / исп.: Б. В. Севастьянов, А. П. Тюрин, Р. О. Шадрин, И. Г. Русяк, В. Г. Суфиянов, И. В. Васильева.

2. *Лялькина Г. Б.*, *Бердышев О. В.* Математическая обработка результатов эксперимента : учеб. пособие. – Пермь : Изд-во Перм. нац. иссл. политех. ун-та, 2013. – 78 с.

Получено 10.12.2014

УДК 681.3.06

С. В. Вологдин, доктор технических наук, ИжГТУ имени М. Т. Калашникова
Д. В. Попов, магистрант, ИжГТУ имени М. Т. Калашникова

ПРОГРАММНО-ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА АВТОМАТИЗАЦИИ ТЕСТОВЫХ СЦЕНАРИЕВ (НА ПРИМЕРЕ ДЫМОВОГО ТЕСТИРОВАНИЯ)

Для тестирования программного обеспечения используются различные виды тестирования, не последнее значение среди которых имеет дымовое, рассматривающееся как короткий цикл тестов, выполняемый для подтверждения того, что после сборки кода (нового или исправленного) устанавливаемое приложение стартует и выполняет основные функции [1].

Вывод о работоспособности основных функций делается на основании результатов поверхностного тестирования наиболее важных модулей приложения на предмет возможности выполнения требуемых задач и наличия быстро находимых критических и блокирующих дефектов. В случае отсутствия таковых дымовое тестирование объявляется пройденным, и приложение передается для проведения полного цикла тестирования, в противном случае дымовое тестирование объявляется проваленным, и приложение уходит на доработку.

Современные методологии разработки используют подход непрерывного слияния, который подразумевает ежедневную сборку программного продукта. Сборки не всегда бывают качественными и могут содержать в себе ошибки в работе критичного для конечного пользователя функционала, поэтому проверка ключевого функционала должна осуществляться непосредственно после сборки и перед передачей на тестирование программного продукта. Это позволяет сократить потерю времени на тестирование сборки, содержащей блокирующие ошибки [2].

Дымовое тестирование позволяет обнаруживать критичные ошибки заранее, правка которых начнется незамедлительно после их обнаружения. Автоматизация дымового тестирования позволяется собрать все ошибки на текущей версии собранного продукта и передать их в отдел разработки.

Для облегчения работы, экономии времени и людских ресурсов необходимо внедрить автоматизацию тестовых сценариев для дымового тестирования. Для автоматизации тестовых сценариев дымового тестирования необходимо использовать такой же язык программирования, какой используется при разработке системы с целью экономии времени и средств для интеграции автоматизированных тестов в систему [3]. В случае с BPMS ELMA (система управления бизнес-процессами ELMA) этим языком программирования является C# [4].

Автоматизированные тесты для данной системы разрабатывают, как правило, с помощью Microsoft Visual Studio или в редакторе сценариев дизайнера ELMA.

С целью экономии времени и средств на разработку, а в дальнейшем и внедрение автоматизированных тестов было принято решение использовать редактор сценариев дизайнера ELMA.

Дизайнер ELMA – программно-инструментальное средство для визуализации бизнес-процессов предприятия, а также для создания различного типа объектов системы, типов документов, организационной структуры, типов проектов и отчетов.

Дизайнер ELMA позволит визуально смоделировать бизнес-процесс, направленный на автоматический запуск необходимых тестов, с начальной формой, которая позволит выбирать те тесты, которые необходимо выполнить, а также позволит визуально отразить результаты в сводном отчете в виде таблицы, в котором будут указаны наименования тестов, их результат, ошибки и комментарии к этим ошибкам.

Из этих данных можно сделать вывод, что дизайнер ELMA позволяет наиболее эффективно создавать автоматические тесты, которые представлены визуально, а также присутствует возможность редактировать отчеты и начальную форму автоматических тестов.

При разработке программного обеспечения существует множество методологий разработки. Наиболее подходящей методологией разработки программного обеспечения в рамках автоматизации тестовых сценариев дымового тестирования является спиральная модель с элементами каскадной модели в секторах спирали. Преимуществами данной модели в поставленной задаче являются:

1. Разработка функционирующего прототипа или демоверсии в короткие сроки.

2. Возможность уточнения требований после каждой версии.

3. Постепенная реализация функционала посредством добавления новых тестов.

4. Из-за отсутствия конкретных сроков реализации окончательной версии программного обеспечения всегда имеется готовая версия автоматизированных тестов.

5. Использование каскадной модели в каждом секторе спиральной модели позволяет полностью завершить этап без прерывания процесса, результат которого будет гарантирован.

Процесс разработки будет разбит на четыре сектора:

1. Оценка и разрешение рисков.
2. Определение целей.
3. Разработка и тестирование.
4. Планирование.

Первый сектор подразумевает определение рисков на этапе разработки автоматизированного тестирования и их разрешение; во втором определяются и формулируются цели для программиста, занимающегося автоматизацией; в третьем осуществляется непосредственная разработка автоматизированного теста, а также его тестирование на работоспособность и отсутствие ошибок; планирование работ для следующей итерации осуществляется в четвертом секторе.

С точки зрения автора данный процесс разработки автоматизированного теста является оптимальным.

Библиографические ссылки

1. *Майерс Г.*, *Баджетт Т.*, *Сандлер К.* Искусство тестирования программ. – Вильямс : Диалектика, 2012. – 272 с.

2. *Ошероув Р.* Искусство автономного тестирования с примерами на C#. – ДМК Пресс, 2014. – 360 с.

3. *Рассел Д.* Автоматизированное тестирование. – VSD, 2013. – 102 с.

4. Система управления бизнес-процессами ELMA. – URL: http://www.elma-bpm.ru/ (дата обращения: 16.12.2014).

Получено 12.02.2015

УДК 001.8 : 658.5

Ю. Н. Терехова, аспирант, ИжГТУ имени М. Т. Калашникова
В. С. Клековкин, доктор технических наук, профессор, ИжГТУ имени М. Т. Калашникова

ДИНАМИЧЕСКОЕ ЭФФЕКТИВНОЕ УПРАВЛЕНИЕ ПРЕДПРИЯТИЕМ

Современные интегрированные системы менеджмента направлены на повышение эффективности деятельности предприятий за счет управления интеллектуальной собственностью и правильной оценкой рисков бизнес-процессов. В частности, на ОАО «Ижевский радиозавод» для всех процессов внедрены ключевые показатели результативности (KPI), которые в реальном режиме времени поддерживают эти процессы в заданных режимах. Но однажды настроенные процессы в динамично меняющихся условиях не всегда обеспечивают необходимый интегральный результат. В работе [1] показана попытка получения комплексного индекса, но и в этом случае не удается в режиме онлайн управлять динамической системой предприятия с целью максимизации его прибыли.

Настоящая статья направлена на решение обозначенной проблемы при рассмотрении предприятия как единой системы, в которой управляющие процессы стремятся обеспечить максимальный синергический эффект от правильного взаимодействия [2].

Представим предприятие как классическую систему [3]:

$$S = \left\langle \left\{ A_{\text{п}}; A_{\text{у}}; A_{\text{и}}; B_{\text{п}}; B_{\text{у}}; B_{\text{и}} \right\}; W; \Phi \right\rangle, \qquad (1)$$

где $A_{\text{п}}$, $A_{\text{у}}$, $A_{\text{и}}$ – множества материальных и интеллектуальных производственных, управленческих и изменяющих элементов системы; $B_{\text{п}}$, $B_{\text{у}}$, $B_{\text{и}}$ – множества материальных и интеллектуальных производственных, управленческих и изменяющих процессов системы; $W = \{W_1, W_2, \ldots, W_i\}$ – множество операций над множеством системы (1); $\Phi = \{\Phi_1, \Phi_2, \ldots \Phi_n\}$ – множество предикатов (утверждений) «ложь» или «истина» при выполнении множества операций W.

Из формулы (1) видно, что изменение одного элемента или процесса ведет к изменениям других. Необходимо определить, каким образом они зависят друг от друга. Нас в большей степени интересуют управленческие процессы, поскольку они позволяют организовать работу производства.

Выделим из множества (1) управляющие процессы и поддерживающие элементы и представим их в виде множеств (2) с точки зрения эффективного осуществления процессов.

$$\begin{cases} A_{\text{у}} = a_{\text{уп}1}; a_{\text{уп}1}; \ldots; a_{\text{уп}n}; a_{\text{уп}1}; a_{\text{уп}2}; \ldots; a_{\text{уп}n}, \\ B_{\text{у}} = b_{\text{уп}1}; b_{\text{уп}1}; \ldots; b_{\text{уп}n}; b_{\text{уп}1}; b_{\text{уп}2}; \ldots; b_{\text{уп}n}, \end{cases} \qquad (2)$$

где $a_{\text{уп}i}$ – материальные элементы, поддерживающие управляющие процессы; $a_{\text{уп}i}$ – интеллектуальные элементы, поддерживающие управляющие процессы; $b_{\text{уп}i}$ – элементарные материальные управляющие процессы; $b_{\text{уп}i}$ – элементарные интеллектуальные управляющие процессы.