

## Автоматизированная система разработки тест-планов при проведении тестирования программного обеспечения

*И.С. Полевщиков, М.С. Чирков, А.В. Леванов*

*Пермский национальный исследовательский политехнический университет*

**Аннотация:** Рассмотрена проблема повышения эффективности составления тест-планов при тестировании сложных программных систем (ПС). Средства автоматизации разработки тест-планов позволяют описать данный документ достаточно обобщенно, без учета многих атрибутов, требуемых для успешности дальнейших этапов тестирования ПС. Применение разрабатываемой автоматизированной системы (АС), описанной в статье, позволяет снизить трудоемкость создания тест-планов за счет наличия четкой последовательности шагов при составлении документа, предоставления специалисту справочной информации и советующих воздействий. Данная АС в перспективе может использоваться как непосредственно при разработке тест-планов в ходе реальных программных проектов, так и при обучении начинающих специалистов навыкам заполнения данных документов.

**Ключевые слова:** тестирование программного обеспечения, тест-план, автоматизированная система.

### 1 Введение. Актуальность исследования

На начальной стадии тестирования сложной программной системы (ПС) необходимо создать документ (тест-план), описывающий и регламентирующий перечень работ, техники, подходы, области, ресурсы, контрольные точки тестирования и т.д. [1].

Написание документации, и, в частности, тест-плана – это рутинный процесс, занимающий достаточно много времени. Следует отметить, что у начинающих специалистов по тестированию программного обеспечения (ПО) могут возникать сложности в создании этого документа, поскольку необходимо знать особенности заполнения различных пунктов плана, в частности критерии тестирования, методы, техники, метрики.

Для повышения эффективности написания тест-планов, в частности, сокращения времени, используются различные средства автоматизации, например, TestRail [2], qTest [3], TestLink [4], TestLodge [5], EasyQA [6].

Существенным недостатком данных средств является то, что с их

---

применением процесс тестирования описывается достаточно обобщенно, не учитываются все возможные атрибуты тест-плана [1], необходимые для успешности дальнейших этапов тестирования системы. Для начинающих специалистов использование данных средств не дает полного понимания процесса тестирования.

Таким образом, разработка автоматизированной системы (АС) создания тест-планов является востребованной задачей. Применение АС уменьшит трудоемкость составления этого документа и будет полезна для начинающих специалистов в сфере тестирования ПО, т.к. данная АС не только «проведет» пользователя по всем необходимым шагам в тест-плане, но и позволит просматривать справочную информацию на каждом шаге.

## **2 Функциональные возможности автоматизированной системы составления тест-планов**

С учетом выявленных и указанных выше недостатков существующих средств автоматизации, реализован прототип АС, позволяющей специалисту составить тест-план. Процесс создания данной АС основан на развитии существующих работ в области тестирования ПО и, в частности, планирования как фазы тестирования сложных ПС. Среди данных работ следует отметить статьи [7,8], описывающие концепцию построения автоматизированной системы управления процессом тестирования ПО (и, в частности, подсистемы планирования), а также исследования [9,10], посвященные анализу различных методов верификации ПО и их применению при построении сложных ПС.

АС позволяет сформировать тест-план как документ, включающий следующие пункты [1]: цель тестирования, области, подверженные и не подверженные тестированию, тестовые стратегии и подходы, критерии тестирования, ресурсы тестирования, расписание тестирования, роли и их

---

ответственность, оценка рисков, документация и различные метрики.

Алгоритм составления специалистом тест-плана с применением данной АС представим последовательностью приведенных ниже шагов.

На *шаге №1* пользователю АС (специалисту) необходимо заполнить текстовое поле описанием *цели* тестирования. На рис. 1 приведен макет интерфейса для описания цели тестирования.

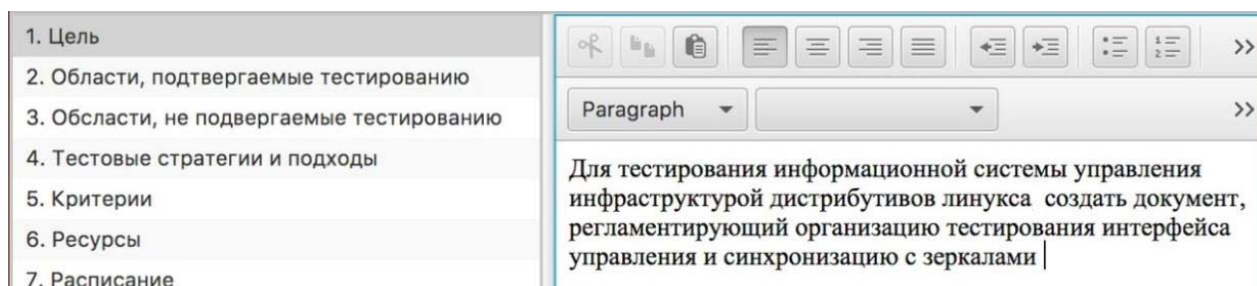


Рис. 1. – Описание цели тестирования

На *шаге №2* пользователем указываются *области, подвергаемые тестированию*, и для каждой области выбирается наиболее подходящий метод тестирования (рис. 2). Также можно вывести справочную информацию по каждому методу тестирования.

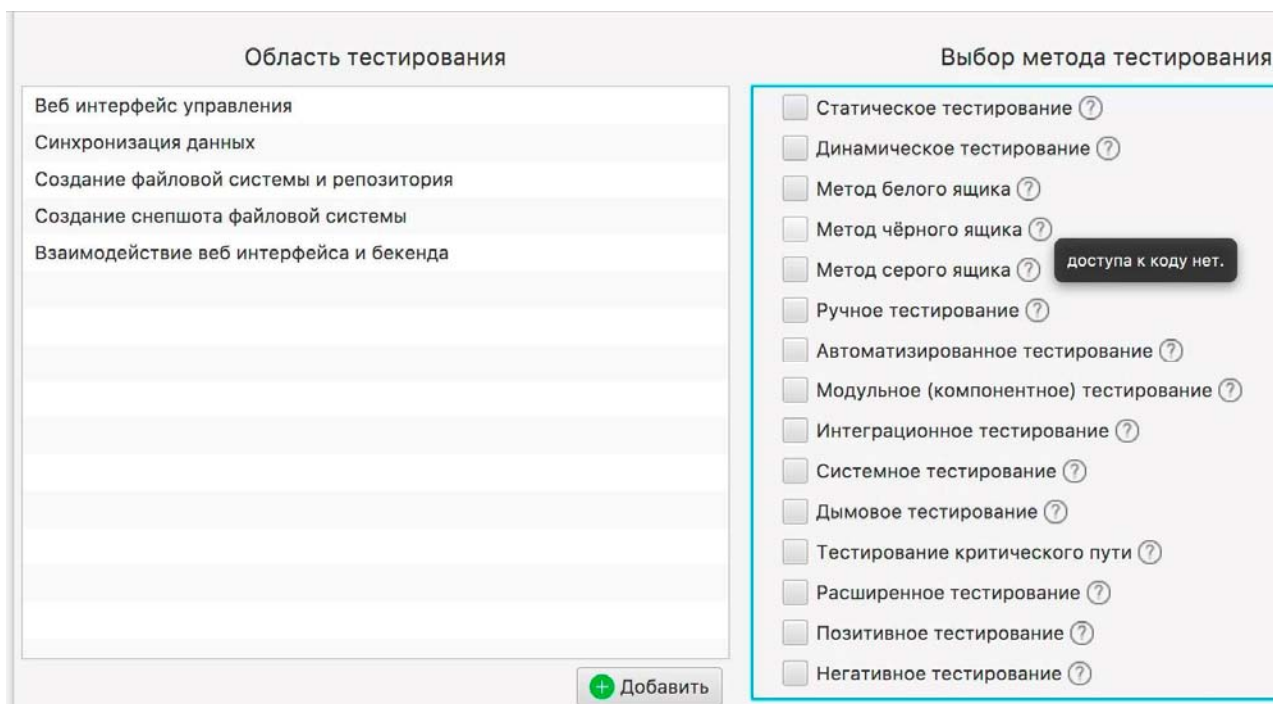


Рис. 2. – Указание областей, подвергаемых тестированию

На шаге №3 указываются области, не подвергаемые тестированию, и для каждой из них описывается, почему для данной области не будет производиться тестирование (рис. 3).

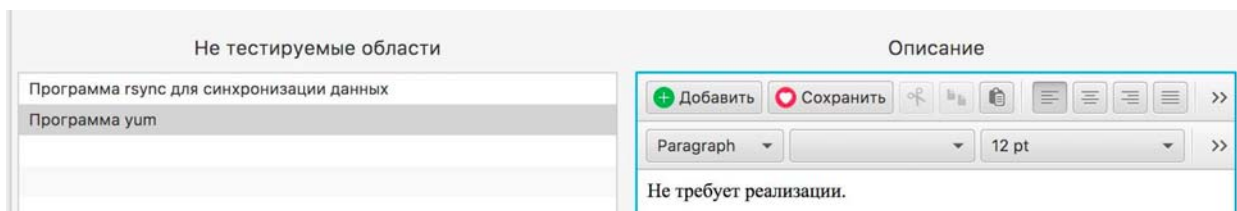


Рис. 3. – Указание областей, не подвергаемых тестированию

На шаге №4 заполняется информация о *тестовых стратегиях и подходах*. Необходимо описать в текстовом поле общий подход к тестированию (рис. 4), а также указать, каким образом будут применяться методы тестирования, выбранные на шаге №2.

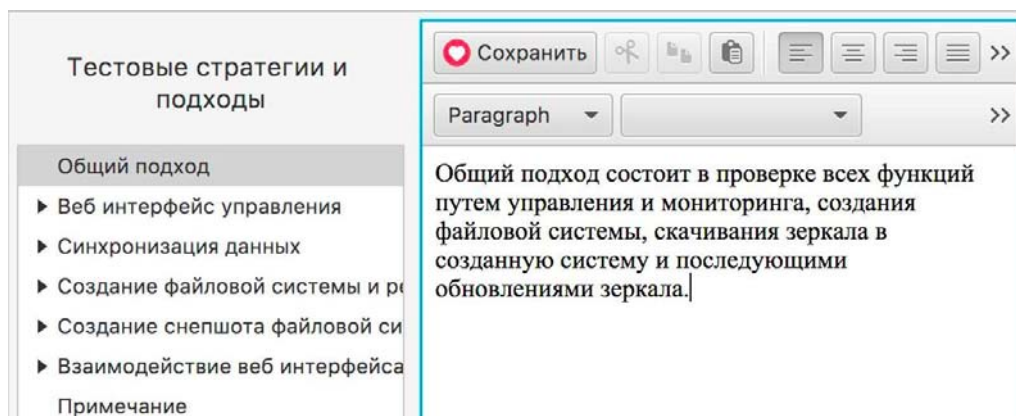


Рис. 4. – Описание тестовых стратегий и подходов

На шаге №5 необходимо описать различные *критерии* тестирования: приемочные критерии, критерии качества; критерии начала тестирования; критерии приостановки тестирования; критерии возобновления тестирования; критерии завершения тестирования (рис. 5).

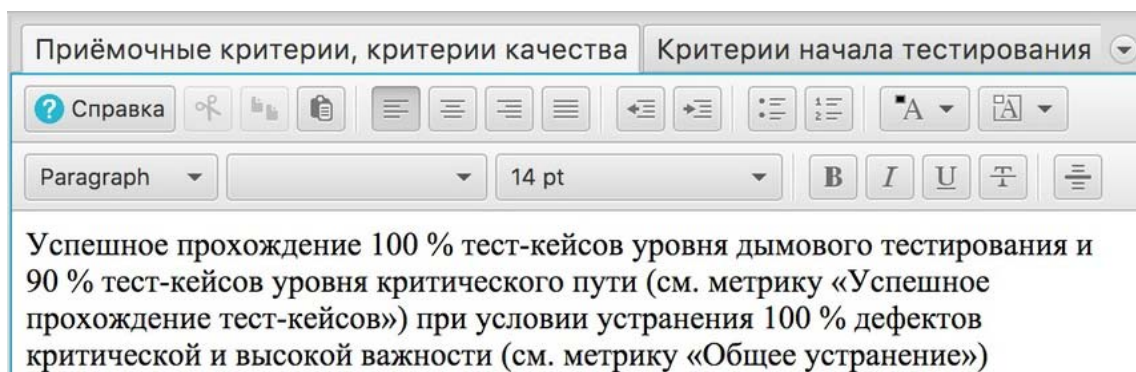


Рис. 5. – Описание одного из критериев

По каждому из критериев можно посмотреть справочную информацию (рис. 6).

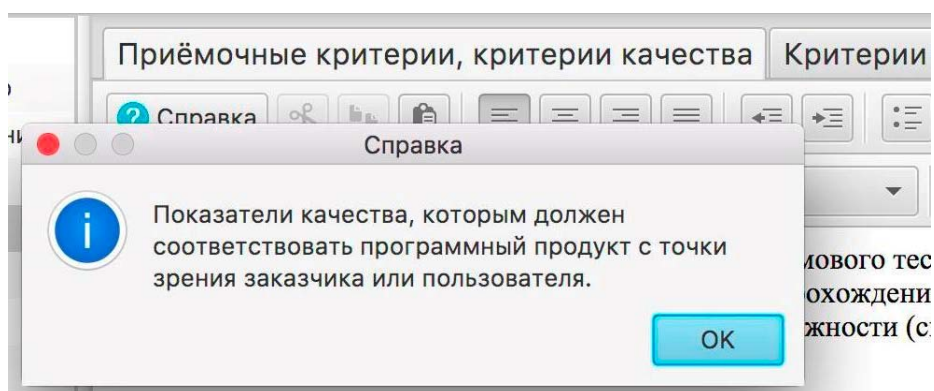


Рис. 6. – Справочная информация по одному из критериев

На *шаге №6* требуется описать различные ресурсы (программные, аппаратные, человеческие, временные, финансовые), необходимые для проведения тестирования (рис. 7). По каждому из ресурсов можно посмотреть справочную информацию (аналогично рис. 6).

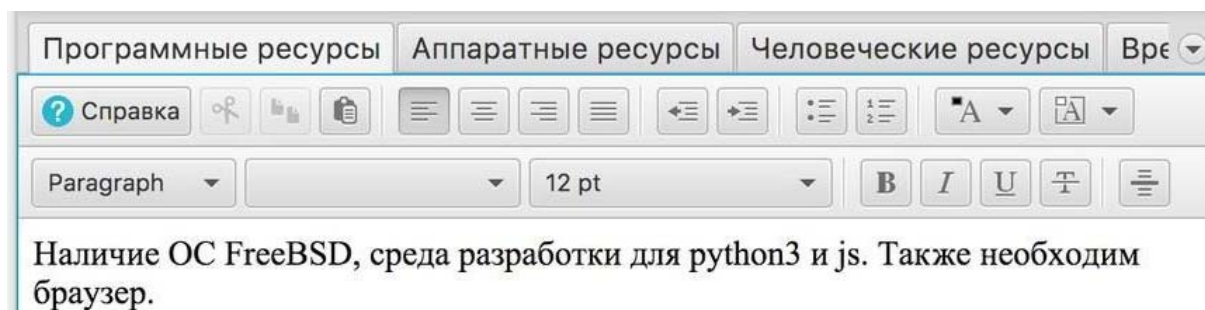


Рис. 7. – Описание одного из ресурсов

На *шаге №7* создается *расписание* (рис. 8). Для каждой контрольной точки необходимо заполнить, что должно произойти в ней, отметить даты



начала и окончания (выбрать даты из встроенного календаря).

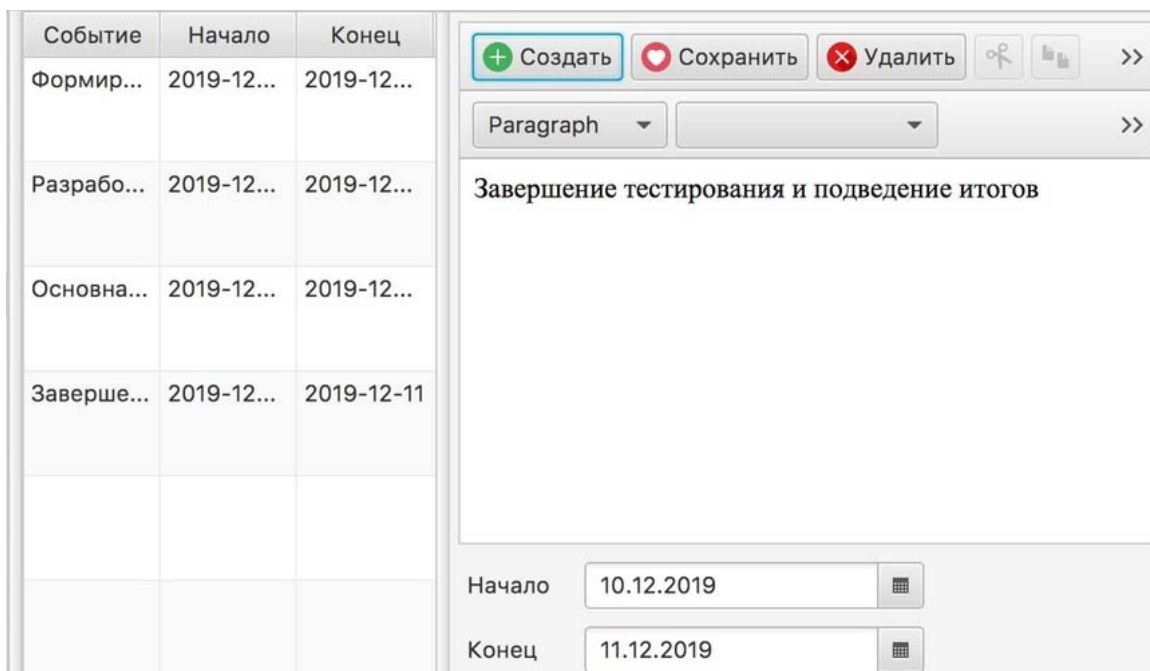


Рис. 8. – Создание расписания тестирования

На *шаге №8* требуется определить *роли* специалистов и обозначить область, за которую ответственен специалист с каждой из ролей (рис. 9).

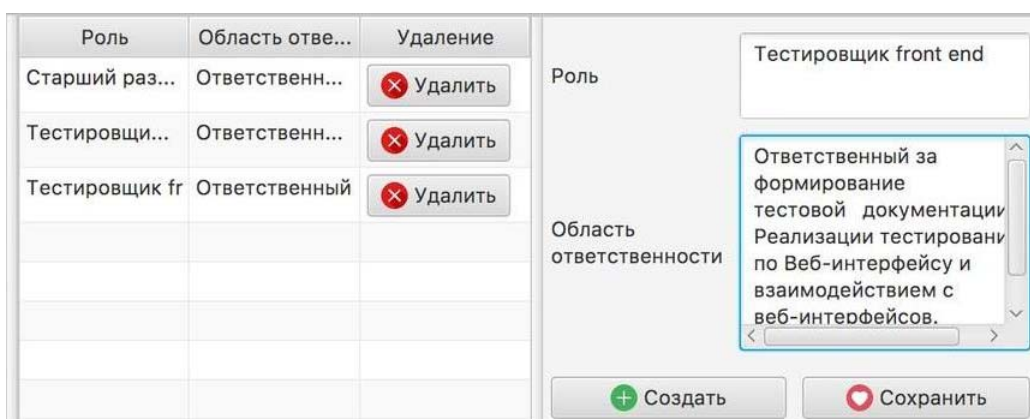


Рис. 9. – Указание ролей и областей их ответственности

На *шаге №9* представлена возможность описать информацию об *оценке рисков* (рис. 10).

Риск	Оценка риска	Удаление
В случае нет...	Низкая	Удалить
Выход за пре...	Средняя	Удалить
Выход за дат...	Высокая	Удалить

Риск

Оценка риска

В случае нетрудоспособности какого-либо из участников команды можно его заменить

Низкая

Создать Сохранить

Рис. 10. – Указание рисков и добавление описания к каждому из них

На шаге №10 требуется указать: наименования документации, кто из ролей за нее ответственный (согласно информации, полученной на шаге №8) и к какой дате документация должна быть готова (рис. 11).

Вид документа...	Ответственный	Дата готовности	Удаление
Отчет о тестир...	Тестировщик fr...	2019-12-12	Удалить
Отчет о тестир...	Тестировщик b...	2019-12-12	Удалить
Отчет о резуль...	Старший разра...	2019-12-18	Удалить

Вид документации

Отчет о результатах тестирования

Ответственный

Старший разработчик

Дата готовности

18.12.2019

Создать Сохранить

Рис. 11. – Описание документации

На шаге №11 необходимо указать информацию о метриках, позволяющих формализовано оценить качество процесса тестирования. Пользователю предоставлена возможность выбрать необходимые метрики из перечня возможных. Для каждой метрики представлена справочная информация и формула, по которой будет производиться расчет (рис. 12).

Успешное прохождение тест-кейсов

$$T^{SP} = \frac{T^{Success}}{T^{Total}} \cdot 100\% \quad ?$$

$T^{Success} \quad ?$   
 $T^{Total} \quad ?$

Начальная фаза проекта	20	▲	▼
Основная фаза проекта	50	▲	▼
Финальная фаза проекта	90	▲	▼

Общее устранение дефектов

$$D_{Level}^{FTP} = \frac{D_{Level}^{Closed}}{D_{Level}^{Found}} \cdot 100\% \quad ?$$

$D_{Level}^{Closed} \quad ?$   
 $D_{Level}^{Found} \quad ?$

		Важность дефекта			
		Низкая	Средняя	Высокая	Критическая
Фаза проекта	Начальная	10	40	50	80
	Основная	20	50	70	90
	Финальная	30	60	100	100

Рис. 12. – Выбор метрик и выставление границ для каждой из них

Пользователю необходимо установить границы значений для каждой из выбранных метрик. В частности, для ряда метрик (например, успешное прохождение тест-кейсов, покрытие требований тест-кейсами, плотность покрытия требований, покрытие классов эквивалентности, покрытие граничных условий) пользователь выставляет (в %) минимальные границы значений, что можно представить множеством  $A = \{a_i \mid i = \overline{1, N_{ph}}\}$ , где  $a_i$  – минимальная граница значения для  $i$ -й фазы проекта ( $N_{ph}$  – общее число фаз). Например,  $A = \{a_1, a_2, a_3\}$ , где  $a_1, a_2, a_3$  – минимальные границы значений для начальной, основной и финальной фаз проекта соответственно.

Для некоторых метрик (например, общее устранение дефектов, текущее устранение дефектов) необходимо заполнить (в %) минимальные границы значений в форме матрицы  $D = (d_{ij})$ , где  $d_{ij}$  – минимальная граница значения для  $i$ -й фазы проекта применительно к дефектам  $j$ -й важности ( $j = \overline{1, N_{imp}}$ , причем  $N_{imp}$  – число возможных градаций важности дефектов). Например,



можно выделить  $N_{imp.} = 4$  градации [1]: низкую, среднюю, высокую и критическую важность дефектов.

В ряде случаев (например, для метрики «стоп-фактор») пользователю представлена форма, где он сможет составить требуемую формулу.

После выполнения всех перечисленных шагов специалист может сформировать тест-план как документ формата «.docx» или «.pdf».

### 3. Заключение

Таким образом, преимуществами данной АС, способствующими повышению эффективности составления тест-планов (в частности, сокращению времени и числа ошибок при создании документа), являются:

1) Пользователю АС заранее известно, какие пункты включены в тест-план, а также при необходимости предоставляется справочная информация по всем критериям, тестовым стратегиям, методам тестирования, метрикам оценки процесса тестирования и т.д. Специалисту необходимо только заполнять предложенные ему поля и переключаться между шагами.

2) Возможность заполнять пункты плана не только произвольным текстом с помощью полей, но и выбирать значения из предложенного списка (с возможностью «запоминания» выбранных данных для применения в следующих шагах), выбирать необходимые метрики из перечня возможных, заполнять границы метрик, строить формулы метрик в «конструкторе».

3) Предоставление пользователю советуемых воздействий на основе анализа хранящихся в системе данных о программных проектах [8].

4) Автоматическая генерация тест-плана в виде текстового документа с возможностью выбора формата файла.

Повышение эффективности применения АС может быть достигнуто объединением с другими средствами тестирования, например, системами, предназначенными для: поддержки жизненного цикла (ЖЦ) тест-кейсов, ЖЦ

---

дефектов; подготовки отчетов о результатах тестирования; разработки набора требований к ПО. В частности, области, подвергаемые и не подвергаемые тестированию могут быть получены на основе перечня требований к ПО.

Рассмотренная АС, с учетом ее дальнейшего развития, может применяться не только непосредственно для составления тест-планов в реальных программных проектах, но и для обучения молодых специалистов навыкам составления тест-планов (при наличии модулей оценки знаний и навыков, аналогичных описанным в работе [11]).

### Литература

1. Куликов С.С. Тестирование программного обеспечения. Базовый курс. Минск: Четыре четверти, 2017. 312 с.
2. TestRail: Modern Test Case Management Software for QA and Development Teams. URL: [gurock.com/testrail](http://gurock.com/testrail) (accessed 19/12/2019).
3. qTest Manager: Transform Test Case Management. URL: [qasymphony.com/software-testing-tools/qtest-manager/test-case-management/](http://qasymphony.com/software-testing-tools/qtest-manager/test-case-management/) (accessed 19/12/2019).
4. QATestLab: Глоссарий системы TestLink. URL: [training.qatestlab.com/blog/course-materials/testlink-glossary/](http://training.qatestlab.com/blog/course-materials/testlink-glossary/) (дата обращения: 19.12.2019).
5. TestLodge: Online test case management tool. URL: [testlodge.com](http://testlodge.com) (accessed 19/12/2019).
6. EasyQA: Инструмент для управления тестированием. URL: [geteasyqa.com/ru/](http://geteasyqa.com/ru/) (дата обращения: 19.12.2019).
7. Полевщиков И.С., Тютюных А.А. Автоматизация управления фазой планирования при тестировании сложных программных систем // Научно-технический вестник Поволжья. 2018. №12. С. 281-283.
8. Полевщиков И.С., Файзрахманов Р.А. Автоматизированное

управление тестированием программных систем с применением нейронных сетей // Инженерный вестник Дона. 2018. №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5283.

9. Бурякова Н.А., Чернов А.В. Классификация частично формализованных и формальных моделей и методов верификации программного обеспечения // Инженерный вестник Дона. 2010. №4. URL: ivdon.ru/ru/magazine/archive/n4y2010/259.

10. Яловой И.О. Анализ требований и управление изменениями программных проектов // Инженерный вестник Дона. 2008. №4. URL: ivdon.ru/ru/magazine/archive/n4y2008/102.

11. Fayzrakhmanov R.A., Polevshchikov I.S. The Use of Mathematical Methods to Automate the Control of Skills in the Study of software Testing Algorithms // Proceedings of 2019 XXII International Conference on Soft Computing and Measurements (SCM): St. Petersburg, Russia, May 23–25, 2019 / IEEE Russia North West Section, Saint Petersburg Electrotechnical Univ. «LETI» (ETU «LETI»). [S. l.] : IEEE, 2019. pp. 107-110.

### References

1. Kulikov S.S. Testirovanie programmnoho obespecheniya. Bazovyy kurs [Software Testing. Basic course]. Minsk: Chetyre chetverti, 2017. 312 p.

2. TestRail: Modern Test Case Management Software for QA and Development Teams. URL: gurock.com/testrail (accessed 19/12/2019).

3. qTest Manager: Transform Test Case Management. URL: qasymphony.com/software-testing-tools/qtest-manager/test-case-management/ (accessed 19/12/2019).

4. QATestLab: Glossariy sistemy TestLink [QATestLab: TestLink Glossary]. URL: training.qatestlab.com/blog/course-materials/testlink-glossary/ (accessed 19/12/2019).



5. TestLodge: Online test case management tool. URL: [testlodge.com](http://testlodge.com) (accessed 19/12/2019).
6. EasyQA: Instrument dlya upravleniya testirovaniem [EasyQA: Test Management Tool]. URL: [geteasyqa.com/ru/](http://geteasyqa.com/ru/) (accessed 19/12/2019).
7. Polevshchikov I.S., Tyutyunykh A.A. Nauchno-tekhnicheskiy vestnik Povolzh'ya. 2018. №12. pp. 281-283.
8. Polevshchikov I.S., Fayzrakhmanov R.A. Inzhenernyj vestnik Dona, 2018, №4. URL: [ivdon.ru/ru/magazine/archive/n4y2018/5283](http://ivdon.ru/ru/magazine/archive/n4y2018/5283).
9. Buryakova N.A., Chernov A.V. Inzhenernyj vestnik Dona, 2010, №4. URL: [ivdon.ru/ru/magazine/archive/n4y2010/259](http://ivdon.ru/ru/magazine/archive/n4y2010/259).
10. Yalovoy I.O. Inzhenernyj vestnik Dona, 2008, №4. URL: [ivdon.ru/ru/magazine/archive/n4y2008/102](http://ivdon.ru/ru/magazine/archive/n4y2008/102).
11. Fayzrakhmanov R.A., Polevshchikov I.S. Proceedings of 2019 XXII International Conference on Soft Computing and Measurements (SCM): St. Petersburg, Russia, May 23–25, 2019. IEEE Russia North West Section, Saint Petersburg Electrotechnical Univ. «LETI» (ETU «LETI»). [S. 1.] : IEEE, 2019. pp. 107-110.