

УДК 004.413.4:336

МЕТОДЫ АНАЛИЗА ОПЕРАЦИОННЫХ РИСКОВ ПРИ УПРАВЛЕНИИ РЕЛИЗАМИ БАНКОВСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ

Петросян Г.С.

ФГБОУ «Российский экономический университет имени Г.В. Плеханова», Москва,
e-mail: grantp@rambler.ru

Настоящая статья посвящена задачам оценки операционных рисков при планировании, сборке, тестировании и развёртывании IT-релизов. Автором рассмотрена проблема обеспечения возможности внесения изменений в IT-инфраструктуру без ухудшения качества сервиса, что является одной из основных целей процесса управления релизами. Рассмотрены мероприятия и практики, направленные на сокращение потерь от операционных IT-рисков. Также описывается математическая модель для прогнозирования ущерба от IT-риска в разрезе релизов, основанная на методологии Value at Risk. Для подтверждения результатов проведены расчёты с использованием языка программирования R. Данная работа и предложенные в ней методы могут оказаться полезными для менеджеров по управлению релизами, риск-менеджеров, специалистов по внедрению и сопровождению, а также специалистов по тестированию информационных систем.

Ключевые слова: IT-релиз, операционный риск, автоматизированная банковская система, тестирование программного обеспечения, стоимостная мера операционного риска

METHODS FOR OPERATIONAL RISK ANALYSIS IN RELEASE MANAGEMENT OF BANKING SOFTWARE

Petrosyan G.S.

Plekhanov Russian University of Economics, Moscow, e-mail: grantp@rambler.ru

This article is dedicated to the operational risk measurement across release planning, building, testing and deployment. The author considers the problem of ensuring that any changes in IT infrastructure can be performed without degradation of existing services, which is one of the main objectives of the release management process. Activities and practices aimed at the mitigation of IT operational risks across multiple releases are provided. Mathematical model for estimating the IT risk in the context of release management, based on Value at Risk approach, is provided in the research. To confirm the result, calculations are made by means of R programming language. Results of the paper can be useful for release managers, risk managers, support, deployment and test engineers.

Keywords: IT release, operational risk, automates banking system, software testing, operational value at risk

В настоящее время происходит интенсивное развитие и усложнение банковских информационных систем и технологий, также наблюдается тенденция к сокращению релизного цикла, повышение скорости разработки и обеспечение возможности выпуска релиза в любой момент. Растёт популярность гибких методологий (Agile) [1], а также методологии DevOps (development and operations) [2] среди российских банков и их технологических подразделений. В условиях сокращения скорости вывода новых продуктов на рынок и повышения гибкости внедрений (release anytime) банки сталкиваются с проблемой поддержки текущих показателей надёжности автоматизированных систем и минимизации операционных IT-рисков. По этой причине управление релизным циклом банковского программного обеспечения не может рассматриваться отдельно от соответствующих технологических рисков, которые может привести установка нового релиза.

Операционный IT-риск – это риск ущерба текущей деятельности банка в виде убытка или недополученного дохода, вы-

званный используемыми информационными технологиями и IT-процессами [3]. Примерами последствий инцидентов операционного IT-риска могут служить неверно рассчитанные платежи по кредитам, сбой в работе карточного процессинга в часы пиковой нагрузки, ошибки при расчете комиссионного вознаграждения банка при переводе денежных средств. Подобные ошибки способны привести к серьёзным денежным и репутационным потерям для различных организаций.

Согласно методологии ITIL [4] управление IT-релизами – это процесс, отвечающий за планирование и контроль сборки, тестирования и развёртывания релизов, а также за внедрение нового функционала без нарушения целостности существующих IT-сервисов.

В современной литературе по управлению IT-процессами, включая методологию ITIL, слабо отражена проблема интеграции процесса управления релизами и процесса управления рисками. Цель данной статьи заключается в описании основных подходов к управлению IT-рисками в рамках релиз-

ного цикла банковского ПО, а также в разработке математической модели прогнозирования операционных IT-рисков перед внедрением новых версий программного обеспечения. При построении математической модели использован аппарат теории вероятностей, математической статистики и методов Монте-Карло. Представленная модель реализована с использованием языка программирования R.

Методы предотвращения реализации IT-рисков

Приведём основные мероприятия и практики, направленные на выявление и предотвращение реализации IT-рисков при внедрении нового функционала ПО, а также на снижение вероятности наступления рисков события либо минимизации его последствий в случае реализации.

- Перед выпуском каждая доработка должна проходить обязательное тестирование. Также должен быть протестирован старый функционал, который мог быть затронут изменениями (регрессионное тестирование). Задача тестирования ПО – проверка соответствия реального и ожидаемого поведения программного обеспечения. Обязательным условием проведения тестирования критичного банковского ПО является развёртывание и использование отдельных тестовых сред [5].

- Пилотирование нового функционала на промышленном контуре без тиражирования на все географические локации, отделения банка или устройства самообслуживания [3]. Данный подход предназначен для проверки работоспособности каждой следующей версии релиза в условиях реальных данных и ведет к уменьшению операционных рисков, а также к сокращению сроков вывода на рынок новых банковских услуг.

- Использование систем мониторинга за сбоями в финансовых транзакциях, а также обеспечение возможности отката нового функционала с промышленного контура в случае обнаружения сбоев.

- Непрерывный сбор статистических данных по инцидентам операционного риска, последующий анализ и выявление типовых рисков, которые имеют сходные сценарии реализации и могут быть предотвращены общими мероприятиями. При функциональном тестировании и при приемосдаточных испытаниях каждого релиза рекомендуется проводить тесты, покрывающие типовые риски.

Модель прогнозирования ущерба от IT-инцидентов

Считается, что на начальных фазах жизненного цикла ПО на 1000 строк исходного

кода приходится 1–3 ошибки, которые приводят к дефектам вычислительного процесса [6]. Программа, состоящая из нескольких сотен строк, может иметь тысячи альтернативных путей выполнения. По этой причине отделы тестирования сталкиваются с проблемой неосуществимости исчерпывающего тестирования (exhaustive testing). Это означает, что невозможно найти 100% дефектов в той или иной системе. Однако существует возможность, используя статистические данные, приближенно оценить число пропущенных дефектов и, как следствие, ущерб.

Опишем метод оценки, который базируется на модели Value at Risk (VaR).

Модели VaR, изначально разработанные в конце XX в. для оценки рыночного риска, могут также быть использованы в управлении операционными рисками для расчёта стоимостной меры операционного риска OpVar (Operational Value at Risk) [3].

Определим стоимостную меру операционного риска (operational value at risk, OpVar) IT-релиза t как значение потерь RL_t (release losses) от инцидентов операционного риска в данном релизе, которая не будет превышена с вероятностью α .

$$OpVar_{\alpha}(R_t) = \sup \{u \mid P(RL_t \leq u) \leq \alpha\}. \quad (1)$$

В формуле (1) множество $\{u, u \geq 0\}$ – это множество всех возможных значений ущерба u в рассматриваемом релизе.

Потери от операционных рисков – это дискретный во времени случайный процесс, что является фундаментальным отличием OpVar от рыночных моделей VaR, в которых цена актива рассматривается как непрерывный случайный процесс.

Введём понятия интенсивности и критичности операционного риска.

Интенсивность (frequency) риска определим как количество N_t дефектов, привнесённых релизом t , $t = 1, 2, \dots, T$.

Под критичностью (severity) риска будем понимать величину L_i финансовых потерь, которые повлечёт за собой дефект i , $i = 1, 2, \dots, N_t$.

Предположим, что интенсивность риска имеет распределение Пуассона с математическим ожиданием λ , а частные потери от дефекта i (критичность риска) подчинены логнормальному [8] распределению с параметрами μ и σ^2 :

$$N_t \in \text{Пуас}(\lambda),$$

$$S_i \in \text{LogN}(\mu, \sigma^2).$$

Также сделаем предположение, что случайные величины N_t и L_i являются независимыми.

Совокупные потери за релиз t могут быть рассчитаны по следующей формуле:

$$RL_t = \sum_{i=1}^{N_t} L_i. \quad (2)$$

Функция распределения случайной величины RL_t может быть представлена в следующем виде:

$$F_{RL_t}(x) = P(RL_t \leq x) = P(L_1 + L_2 + \dots + L_{N_t} \leq x).$$

Вывод точной формулы для $F_{RL_t}(x)$ является трудноразрешимой задачей [7], и в данной статье будет предложена модель оценки OpVar, основанная на методе Монте-Карло.

Пусть имеется массив статистических данных по IT-инцидентам для серии релизов. Методом максимального правдоподобия [8] можно оценить математическое ожидание $\hat{\lambda}$ для распределения Пуассона, а также параметры $\hat{\mu}$ и $\hat{\sigma}$ для логнормального распределения:

$$\hat{\lambda} = \frac{\sum_{k=1}^n z_k}{n}, \quad (3)$$

$$\hat{\mu} = \frac{\sum_{k=1}^m \ln w_k}{m}, \quad (4)$$

$$\hat{\sigma} = \sqrt{\frac{\sum_{k=1}^m (\ln w_k - \hat{\mu})^2}{m}}. \quad (5)$$

В формуле (3) z_1, z_2, \dots, z_n – статистические данные по количеству дефектов на релиз, в формулах (4) и (5) w_1, w_2, \dots, w_m – данные по величине ущерба на дефект (в руб.).

Очевидно, что статистические данные по инцидентам будут отсутствовать на ранних этапах проекта по разработке банковского ПО. Для ПО, которое используется в промышленной эксплуатации больше 3–5 лет, статистика также может отсутствовать или может быть неточной. В таких случаях для прогнозирования параметров интенсивности и критичности операционного риска (то есть параметров $\hat{\lambda}$, $\hat{\mu}$ и $\hat{\sigma}$) следует использовать статистические данные по смежным проектам либо метод экспертных оценок.

Используя метод Монте-Карло, сгенерируем N случайных величин, распределенных по Пуассону с параметром $\hat{\lambda}$: x_1, x_2, \dots, x_N .

Затем смоделируем потери, подчиненные логнормальному распределению с параметрами $\hat{\mu}$ и $\hat{\sigma}$, и, таким образом, получим совокупную матрицу интенсив-

ности и критичности операционного риска (aggregated frequency and severity):

$$AFS = \begin{pmatrix} l_{11} l_{12} \dots l_{1x_1} \\ l_{21} l_{22} \dots l_{2x_2} \\ \dots \\ l_{N1} l_{N2} \dots l_{Nx_N} \end{pmatrix}.$$

Заметим, что разные строки матрицы AFS содержат различное число элементов, причём некоторые строки могут не содержать ни одного элемента.

Сложим элементы матрицы AFS по строкам и получим вектор совокупных потерь в разрезе N релизов:

$$\begin{pmatrix} rl_1 \\ rl_2 \\ \dots \\ rl_N \end{pmatrix} = \begin{pmatrix} l_{11} + l_{12} + \dots + l_{1x_1} \\ l_{21} + l_{22} + \dots + l_{2x_2} \\ \dots \\ l_{N1} + l_{N2} + \dots + l_{Nx_N} \end{pmatrix}.$$

Упорядочим получившиеся суммы по возрастанию:

$$rl^{(1)} < rl^{(2)} < \dots < rl^{(N)}.$$

Для расчёта OpVar найдём номер k^* , такой, что $OpVar_\alpha(R) = rl^{(k^*)}$.

Исходя из (1) и из того, что все сценарии равновероятны:

$$P(rl \leq rl^{(i)}) = \frac{i}{N}.$$

Отсюда

$$\frac{k^*}{N} \leq \alpha, \quad \frac{k^* + 1}{N} > \alpha,$$

иначе $OpVar_\alpha(R) = rl^{(k^*+1)}$.

Поэтому:

$$\alpha N - 1 < k^* \leq \alpha N.$$

Окончательно получим, что

$$k^* = [\alpha N]. \quad (6)$$

Также можно рассчитать OpCVar (operational conditional value at risk) – математическое ожидание ущерба при условии, если он превысит значение VaR:

$$OpCVar_\alpha(R) = \frac{1}{N - k^*} \sum_{k=k^*+1}^N rl^{(k)}. \quad (7)$$

Расчёт показателя Value at Risk с использованием языка R

Продemonстрируем предложенный алгоритм расчёта OpVar на примере. Пусть имеются статистические данные за 14 релизов банковской автоматизированной систе-

мы (таблица), и требуется спрогнозировать потери от IT-инцидентов для внедряемого в настоящий момент релиза 15.

Для оценки параметров $\hat{\lambda}$, $\hat{\mu}$ и $\hat{\sigma}$ выполним скрипт (рис. 1) на языке программирования R [9].

Получим:

$$\hat{\lambda} = 2,143, \hat{\mu} = 10,181, \hat{\sigma} = 1,893.$$

Рассчитаем показатели OpVar и OpCVar, используя R.

Листинг программы имеет следующий вид:

```
events_number<-10000 #1. Количество разыгрываемых сценариев
lambda<-fitdist(freq_hist, distr=»pois«, method=»mle«)$estimate #2. Параметр λ распределения Пуассона
lnorm_mean<-coef(fitdist(sev_hist, distr=»lnorm«, method=»mle«))[1] #3. Параметр μ лог-нормального распределения
lnorm_sigma<-coef(fitdist(sev_hist, distr=»lnorm«, method=»mle«))[2] #4. Параметр σ лог-нормального распределения
alpha=0.95 #5. Доверительный уровень
k<-floor(alpha*events_number) #6. Параметр k* для расчёта OpVar
freq_sev_matrix<-list() #7. Генерация матрицы AFS (aggregated frequency and severity)
for(i in 1:events_number){
  sev_vector <- c();
  t<-rpois(1, lambda)
  if(t>=1){
    for(j in 1:t){
      sev_vector<-append(sev_vector, rlnorm(1, meanlog=lnorm_mean, sdlog=lnorm_sigma));
    }
  }
  freq_sev_matrix[[i]]<-sev_vector
}
total_sev_vector <- c(); #8. Вектор из сумм строк матрицы AFS
for(i in 1:events_number){
  total_sev_vector<-append(total_sev_vector, sum(freq_sev_matrix[[i]]))
}
sorted_total_sev_vector<-sort(total_sev_vector) #9. Сортировка координат вектора из сумм строк матрицы AFS
var<-sorted_total_sev_vector[k] #10. Расчёт Var/CVar
above_var<-sorted_total_sev_vector[c((k+1):events_number)]
cvar<-mean(above_var)
print(var)
print(cvar)
```

Статистические данные по IT-инцидентам

№ релиза	Кол-во дефектов	Ущерб от дефектов, руб.
1	4	1586,74; 17379,30; 7416,71; 777,15
2	2	2863,54; 380513,40
3	2	314524; 15612,34
4	1	14857,74
5	4	141454,40; 97545,28; 3145,53; 6556,40
6	1	11428,18
7	4	83540,83; 24519,78; 14995,63; 908,42
8	1	136772,20
9	2	37441,59; 137692,50
10	2	22281,86; 1505342
11	0	–
12	3	132279,50; 64807,78; 365711,60
13	1	9411,29
14	3	176826,20; 4368,12; 7258,91

На рис. 2 изображены результаты выполнения программного кода в среде RStudio.

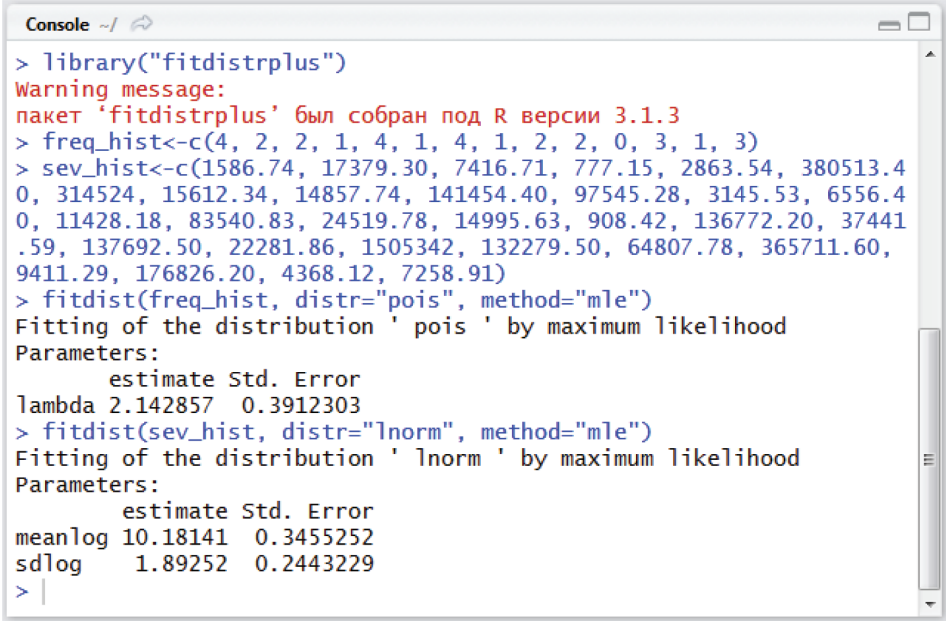
Исходя из рис. 2 потери для внедряемого релиза 15 могут быть выражены в следующих показателях:

$$OpVar_{0,95}(R_{15}) = 1335348 \text{ руб.}$$

$$OpCVar_{0,95}(R_{15}) = 3293289 \text{ руб.}$$

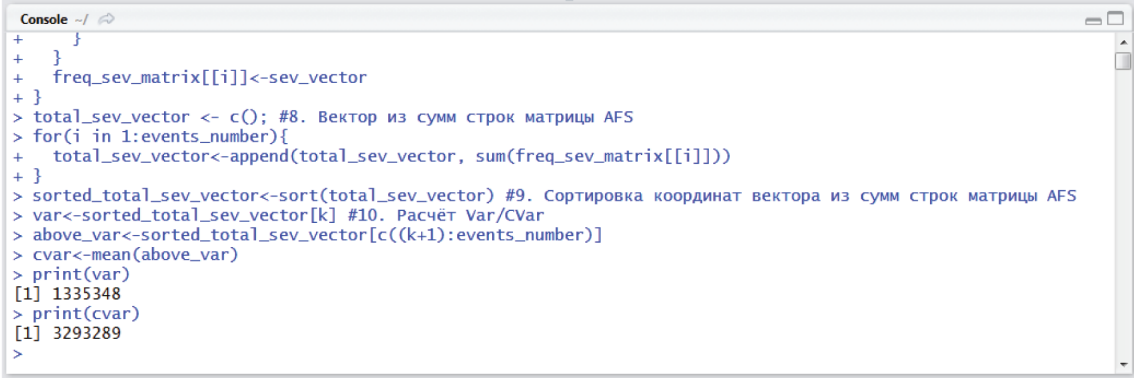
Изобразим графически упорядоченные значения потерь $rl^{(1)} < rl^{(2)} < \dots < rl^{(10000)}$, а также величину OpVar (рис. 3). Для этого необходимо выполнить следующий скрипт:

```
plot(sorted_total_sev_vector, axes = FALSE, type="l", ylim=c(0,2000000), xlab="номер сценария", ylab="величина ущерба (руб.)»,) # 1. Построение основного графика
axis(1, pos=0) # 2. Построение оси OX
axis(2, pos=0, las=1) # 3. Построение оси OY
points(k,0, pch=19) # 4. Отметим на графике OpVar
text(k, 0, k, pos=3)
points(k, var, pch=19)
text(k, var, «OpVar», pos=2)
```



```
Console ~/
> library("fitdistrplus")
Warning message:
пакет 'fitdistrplus' был собран под R версии 3.1.3
> freq_hist<-c(4, 2, 2, 1, 4, 1, 4, 1, 2, 2, 0, 3, 1, 3)
> sev_hist<-c(1586.74, 17379.30, 7416.71, 777.15, 2863.54, 380513.4
0, 314524, 15612.34, 14857.74, 141454.40, 97545.28, 3145.53, 6556.4
0, 11428.18, 83540.83, 24519.78, 14995.63, 908.42, 136772.20, 37441
.59, 137692.50, 22281.86, 1505342, 132279.50, 64807.78, 365711.60,
9411.29, 176826.20, 4368.12, 7258.91)
> fitdist(freq_hist, distr="pois", method="mle")
Fitting of the distribution ' pois ' by maximum likelihood
Parameters:
      estimate Std. Error
lambda 2.142857  0.3912303
> fitdist(sev_hist, distr="lnorm", method="mle")
Fitting of the distribution ' lnorm ' by maximum likelihood
Parameters:
      estimate Std. Error
meanlog 10.18141  0.3455252
sdlog   1.89252  0.2443229
> |
```

Рис. 1. Оценка максимального правдоподобия в среде RStudio



```
Console ~/
+ }
+ }
+ freq_sev_matrix[[i]]<-sev_vector
+ }
> total_sev_vector <- c(); #8. Вектор из сумм строк матрицы AFS
> for(i in 1:events_number){
+   total_sev_vector<-append(total_sev_vector, sum(freq_sev_matrix[[i]]))
+ }
> sorted_total_sev_vector<-sort(total_sev_vector) #9. Сортировка координат вектора из сумм строк матрицы AFS
> var<-sorted_total_sev_vector[k] #10. Расчёт Var/CVar
> above_var<-sorted_total_sev_vector[c((k+1):events_number)]
> cvar<-mean(above_var)
> print(var)
[1] 1335348
> print(cvar)
[1] 3293289
>
```

Рис. 2. Выполнение скрипта для расчёта OpVar и OpCVar

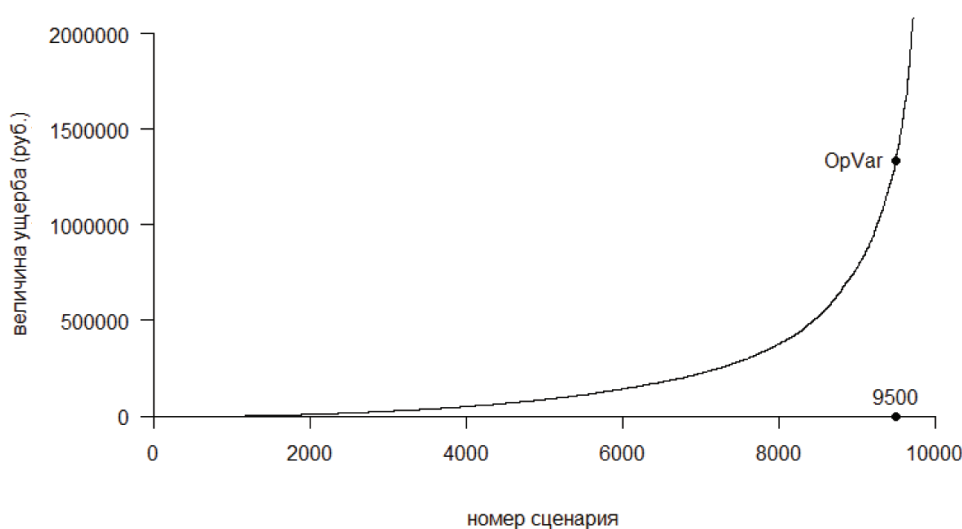


Рис. 3. Упорядоченные значения потерь по 10000 сценариям

В рамках рассмотренного нами примера предположим, что в течение 10 последующих релизов (релизы 15–24) внедрялись новые методы тестирования и, как следствие, была уменьшена частота инцидентов:

$$\hat{\lambda}^{(1)} = 1,664 .$$

Рассчитаем OpVar для релиза 25:

$$OpVar_{0,95}(R_{25}) = 1098100 \text{ руб.}$$

$$\Delta OpVar_{0,95} = 237248 \text{ руб.}$$

Отсюда можно сделать вывод, что новые методы тестирования привели к сокращению максимального ожидаемого ущерба на 237 248 руб.

Заключение

В работе сделан вывод о том, что для критичных банковских систем установка каждого нового релиза в промышленную среду может привести к возникновению инцидентов операционного риска и, как следствие, к существенным финансовым потерям. Этим объясняется необходимость интеграции процесса управления релизами и процесса управления рисками при разработке банковского ПО.

В статье описаны способы минимизации IT-рисков на всех этапах жизненного

цикла IT-релизов; разработана математическая модель оценки рисков, основанная на методе Value at Risk. Предложенная автором модель сопровождается практическим примером, а также программным кодом R.

Список литературы

1. Петросян Г.С., Титов В.А. Специфика тестирования в условиях гибких методологий разработки программного обеспечения // Международный журнал прикладных и фундаментальных исследований. – 2016. – № 9–3. – С. 486–487.
2. Jennifer Davis, Katherine Daniels. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale. – USA: O’Reilly Media, 2016. – 410 p.
3. Cruz M.G. Modeling, measuring and hedging operational risk – Chichester: John Wiley & Sons, Inc., 2002. – 346 p.
4. Rance S. ITIL Service Transition. – London: TSO, 2011. – 360 p.
5. Black R. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing. – USA: Wiley, 2009. – 676 p.
6. Липаев В.В. Тестирование программ / В.В. Липаев. – М.: Радио и связь, 1986. – 296 с.
7. Klugman S., Panjer H., Wilmot G. Loss Models: From Data to Decisions (4th ed.) – New York: Wiley, 2012. – 536 p.
8. Sheldon M Ross. Introduction to probability and statistics for engineers and scientists (5th ed.) – Los Angeles: Associated Press, 2009. – 686 p.
9. The R Manuals [Электронный ресурс]. – Режим доступа: <https://cran.r-project.org/manuals.html> (дата обращения: 05.09.2017).