

УДК 004.05 + 004.4 + 004.9

МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ТЕСТИРОВАНИЯ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ ДЛЯ ТЕХНОЛОГИЙ BIG DATA



С.М. Боровиков
Доцент кафедры
ПИКС БГУИР,
кандидат
технических наук



В.О. Казючиц
Аспирант кафедры
ПИКС БГУИР,
магистр
технических наук



С.К. Дик
Доцент кафедры ЭТТ
БГУИР, кандидат
физико-
математических наук,
доцент



С.С. Дик
Проект-менеджер
компании
«Itransition»,
магистр техники и
технологии

Белорусский государственный университет информатики и радиоэлектроники,
Республика Беларусь
Компания «Itransition», Республика Беларусь
E-mail: bsm@bsuir.by

С.М. Боровиков

Доцент кафедры проектирования информационно-компьютерных систем БГУИР. Основная область научных интересов – прикладные математические методы в проектировании изделий радиоэлектроники, включая алгоритмы статистического прогнозирования надёжности изделий электронной техники и оценку надёжности прикладного программного обеспечения на ранних этапах его разработки. Руководитель разработки программных комплексов по автоматизированному расчёту и обеспечению надёжности электронных устройств: система АРИОН (2008-2009 гг.), система АРИОН-плюс (2011-2015 гг.).

В.О. Казючиц

Окончил БГУИР (2017 г.), в настоящее время является аспирантом этого университета, магистр технических наук. Проводит научные исследования по прогнозированию надёжности полупроводниковых приборов большой мощности.

С. К. Дик

Окончил Минский радиотехнический институт по специальности «Радиотехника», руководит научными исследованиями в области лазерной медицины и биомедицинской оптики.

С.С. Дик

Окончил Белорусский государственный университет информатики и радиоэлектроники (2016 г.), и аспирантуру при этом университете (2020 г.), магистр техники и технологии. Работает в компании «Itransition» в должности проект-менеджера, занимается разработкой и внедрением программного обеспечения в различные сферы деятельности людей.

Аннотация. Прикладные компьютерные программы, используемые для технологий Big Data, содержат сотни тысяч-миллионы строк программного кода и поэтому после написания программ в них всегда имеются скрытые ошибки. Наличие этих скрытых ошибок и определяет уровень надёжности программ. Перед началом тестирования компьютерная программа характеризуется начальным уровнем надёжности, который обычно не отвечает требованиям практики. Задача этапа тестирования состоит в том, чтобы выявить и устранить наиболее критичные ошибки и свести долю скрытых ошибок к минимуму. Возникает вопрос, сколько времени необходимо выполнять тестирование. На основе статистических данных о тестировании и надёжности компьютерных программ разного функционального назначения получена модель, позволяющая определять прогнозируемое время тестирования, обеспечивающее для разрабатываемой программы заданный уровень её эксплуатационной надёжности.

Ключевые слова: прикладные компьютерные программы, эксплуатационная надёжность, время тестирования, модель времени тестирования, эффективность тестирования.

Работа проводилась в рамках договора № Ф20МВ-021 на выполнение НИР «Статистические модели надёжности прикладных программных средств и их использование для оценки ожидаемой безотказности компьютерных программ на ранних этапах их разработки» в соответствии с решением научного совета БРФФИ по результатам конкурса «БРФФИ–Минобразование М-2020 (протокол № 1 от 22.04.2020).

Введение.

Аналитические системы, используемые для обработки больших данных относятся к классу сложных информационно-компьютерных систем, в которых вклад программного обеспечения в ненадёжность систем может составлять 40 и более процентов [1]. Развитие и совершенствование технологий Big Data обуславливают необходимость разработки новых прикладных компьютерных программ по обработке больших объёмов данных. Успех в обработке больших объёмов данных зависит от надёжности используемых прикладных компьютерных программ. Для принятия решения о целесообразности разработки и уровне эффективности функционирования аналитической системы в заданных условиях необходимо на ранних этапах её проектирования оценить ожидаемую эксплуатационную надёжность планируемых к разработке прикладных компьютерных программ. Под эксплуатационной надёжностью будем понимать тот уровень надёжности, который компьютерная программа покажет на начальном этапе её эксплуатации, то есть после выполнения этапа тестирования и отладки.

Прикладные компьютерные программы, используемые для технологий Big Data, содержат сотни тысяч–миллионы строк программного кода и поэтому после написания программ в них всегда имеются скрытые ошибки. Наличие этих ошибок определяют уровень надёжности компьютерных программ. После написания кода компьютерной программы и устранения нарушений правил языка программирования программа характеризуется начальным уровнем надёжности, который как правило не отвечает требованиям практики. Поэтому далее выполняют этап тестирования и отладки программы. Задача этого этапа состоит в том, чтобы выявить и устранить наиболее критичные ошибки, свести к минимуму долю скрытых ошибок, оставшихся в компьютерной программе, и в итоге обеспечить приемлемый уровень эксплуатационной надёжности компьютерной программы. Разработчики аналитических систем хотели бы знать ожидаемый уровень эксплуатационной надёжности разрабатываемой прикладной компьютерной программы ещё до написания кода программы и выполнения её тестирования. Разработчиков аналитических систем интересуют также финансовые затраты и/или затраты времени, которые потребуются для достижения заданного уровня надёжности прикладной компьютерной программы.

Актуальность.

Оценка надёжности прикладной компьютерной программы до написания её программного кода была рассмотрена в [2, 3]. Описанный подход основан на статистических моделях ожидаемого числа возможных ошибок в компьютерной программе (некоторые специалисты используют термин «дефекты проектирования программы»), но не принимает во внимание быстродействие компьютера, изменчивость входных данных и другие факторы. В работе [4] для оценки ожидаемой надёжности прикладной компьютерной программы, планируемой к разработке, предлагается модель, в которой используются некоторые параметры будущей компьютерной программы, данные о влиянии на её надёжность различных факторов и предполагаемое в дальнейшем тестирование компьютерной программы. Модель учитывает отрасль применения компьютерной программы, быстродействие процессора компьютера, изменчивость входных данных, а также рабочую нагрузку, которую будет воспринимать компьютерная программа со стороны эксплуатационной среды (ввод-вывод данных и нахождение этих операций в очереди, наличие состояний ожиданий, загрузка-выгрузка программы и/или её модулей из памяти и т.д.). Рассматривается получение параметров модели. Из работы [4] следует, что для достижения заданного уровня эксплуатационной надёжности компьютерной программы необходимо обеспечить нужное значение коэффициента эффективности тестирования Q , показывающего, во сколько раз уменьшается интенсивность отказов компьютерной программы после выполнения её тестирования относительно начальной интенсивности отказов λ_0 (до начала тестирования).

Большинство моделей надёжности компьютерных программ исходят из того, что интенсивность отказов λ прямо пропорциональна числу оставшихся в программе ошибок [5, 6]:

$$\lambda \approx C_{\Pi} \cdot N_{\text{ош}} = C_{\Pi} \cdot F \cdot L, \quad (1)$$

где C_{Π} – коэффициент пропорциональности, определяемый экспериментально; $N_{\text{ош}}$ – число оставшихся ошибок в компьютерной программе; F – плотность ошибок в программе (среднее число ошибок, приходящихся на строку кода: $F = N_{\text{ош}} / L$); L – объём компьютерной программы в исполняемых строках программного кода, англоязычный вариант понятия «строки кода» – LOC (от слов Lines of Code).

С учётом выражения (1) величина Q показывает, во сколько раз за счёт тестирования уменьшится число оставшихся ошибок в компьютерной программе относительно начального их числа (до выполнения тестирования).

Вопрос о длительности этапа тестирования прикладной компьютерной программы остался открытым. Важно спрогнозировать, сколько времени должно выполняться тестирование, чтобы достичь необходимого значения коэффициента эффективности тестирования Q , после чего компьютерная программа предположительно будет отвечать требованию эксплуатационной надёжности и её можно будет использовать для решения практических задач.

Исходные положения и предпосылки.

В соответствии с ГОСТ 27.205-1997 [7] об ожидаемой надёжности планируемой к разработке прикладной компьютерной программы будем судить по интенсивности проявления ошибок (λ), обусловленных дефектами её проектирования, то есть оставшимися в компьютерной программе скрытыми ошибками. Проявление ошибки при использовании программы приводит к невыполнению задачи пользователя, что равносильно отказу программы, поэтому будем употреблять слова «интенсивность отказов программы». Использование λ в качестве характеристики надёжности компьютерной программы удобно тем, что её можно оценивать с помощью прогнозируемой плотности ошибок F в компьютерной программе [8].

В работе [4] модель оценки ожидаемой эксплуатационной интенсивности отказов прикладной компьютерной программы ($\lambda_{\text{экс}}$) получена в виде

$$\lambda_{\text{экс}} = \frac{60}{Q} K_{\Sigma} \frac{R}{B} F_0 L \cdot 10^{-6} = \frac{60}{Q} K_{\Sigma} V N_0 \cdot 10^{-6}, \text{ ч}^{-1}, \quad (2)$$

где Q – коэффициент эффективности тестирования компьютерной программы; K_{Σ} – суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу; R – пиковое быстродействие процессора, указываемое производителем в технической документации, размерность: операций в секунду; B – количество команд (операторов) в компьютерной программе; F_0 – начальная плотность ошибок в программе; L – объём компьютерной программы в исполняемых строках программного кода; V – скорость выполнения команд компьютерной программы (скорость для пикового быстродействия процессора R), определяемая отношением $V = R/B$; N_0 – число ошибок в компьютерной программе до выполнения тестирования.

Параметр F_0 в выражении (2) – это плотность ошибок компьютерной программы после написания её кода и устранения нарушений правил языка программирования, представляет собой среднее число ошибок, приходящихся на строку кода: $F_0 = N_0 / L$.

Согласно определению, предложенному в работе [4], коэффициент эффективности тестирования Q представляет отношение

$$Q = \frac{N_0}{N_{\text{п.тест}}}, \quad (3)$$

где $N_{\text{п.тест}}$ – число оставшихся ошибок в прикладной компьютерной программе после завершения её тестирования.

Модель прогнозирования.

Для получения модели прогнозирования, используемой для определения времени, требуемого для проведения тестирования планируемой к разработке прикладной компьютерной программы, необходимо в отношении (3) выразить величину $N_{\text{п.тест}}$ через время тестирования t и далее решить полученное уравнение относительно t .

Значение $N_{\text{п.тест}}$ можно найти как

$$N_{\text{п.тест}} = N_0 - n_{\text{обн}}, \quad (4)$$

где $n_{\text{обн}}$ – суммарное число ошибок, обнаруженных в компьютерной программе при её тестировании.

В соответствии с работой [5] величина $n_{\text{обн}}$ описывается экспоненциальной функцией суммарного времени функционирования программы при её тестировании:

$$n_{\text{обн}} = N_0 \left[1 - \exp\left(-\frac{\lambda_0}{N} t\right) \right], \quad (5)$$

где t – суммарное время выполнения компьютерной программы при её тестировании; N – общее число ошибок, которые гипотетически могут проявиться в процессе тестирования компьютерной программы.

Здесь и далее в качестве t рассматривается процессорное время выполнения компьютерной программы. В дальнейшем при необходимости время выполнения программы t может быть преобразовано в календарное время.

В источнике [9] значение N формулы (5) рекомендуется определять как

$$N = N_0 \cdot K_{\text{ош}},$$

где $K_{\text{ош}}$ – коэффициент, показывающий, какая доля начального числа ошибок компьютерной программы N_0 может проявиться в процессе тестирования, по умолчанию принимают $K_{\text{ош}} = 0,95$ [9].

Для простоты будем считать, что

$$N \approx N_0. \quad (6)$$

Подставляя в выражение (4) величину $n_{\text{обн}}$, определяемую по формуле (5), и учитывая примерное равенство (6), получим

$$N_{\text{п.тест}} = N_0 - N_0 \left[1 - \exp\left(-\frac{\lambda_0}{N_0} t\right) \right] = N_0 \exp\left(-\frac{\lambda_0}{N_0} t\right) \quad (7)$$

Тогда отношение (3) с учётом последней части выражения (7) примет вид:

$$Q = \frac{N_0}{N_{\text{п.тест}}} = \frac{N_0}{N_0 \exp\left(-\frac{\lambda_0}{N_0} t\right)} = \exp\left(\frac{\lambda_0}{N_0} t\right) \quad (8)$$

Интенсивность отказов прикладной компьютерной программы до начала тестирования (λ_0) с учётом бысродействия процессора компьютера может быть определена по модели Муса [5]:

$$\lambda_0 = C V N_0 = C \frac{R}{B} N_0 \quad (9)$$

где C – коэффициент пропорциональности, называемый некоторыми специалистами коэффициентом проявления ошибок компьютерной программы.

Коэффициент C формулы (9) зависит от функционального назначения прикладной компьютерной программы, области (сферы) деятельности людей. В качестве размерности интенсивности отказов λ_0 обычно используют $1/\text{ч} = \text{ч}^{-1}$. Для получения этой размерности единицей измерения, описывающей время в размерности параметров V и R формулы (9), должен быть выбран час (ч).

В работе [4] параметры C и B формулы (9) представлены в виде

$$C = C_B \cdot K_\Sigma; \quad B = L \cdot E_L, \quad (10)$$

где C_B – базовое значение коэффициента проявления ошибок компьютерной программы; K_Σ – суммарный коэффициент увеличения интенсивности отказов, обусловленный совместным действием изменчивости входных данных и рабочей нагрузки на компьютерную программу, $K_\Sigma \geq 1$; E_L – коэффициент расширения кода, показывающий увеличение числа команд компьютерной программы относительно числа строк программного кода L . Значение E_L зависит от используемого языка программирования, например для Си $E_L = 2,5$; для Fortran, Cobal $E_L = 3,0$; для Ada $E_L = 4,5$; для Си++ $E_L = 6,0$.

В работе [4] принято $C_B = 0,238 \cdot 10^{-7}$ 1/ошибка, что соответствует коэффициенту C в формуле (9) для прикладных компьютерных программ, используемых для мониторинга и обеспечения безопасности объектов и территорий. Коэффициент K_Σ в равенствах (10) был рассчитан на основе статистических (экспериментальных) данных об эксплуатационной надёжности прикладных компьютерных программ разного функционального назначения (областей применения) с учётом принятого значения C_B [4]. Значения коэффициента K_Σ приведены в таблице 1.

Таблица 1. Коэффициенты (на основе экспериментальных данных) для компьютерных программ различных областей применения

Область применения прикладной компьютерной программы	Коэффициент C , 1/ошибка	Коэффициент K_Σ	Средний процент времени выполнения компьютерной программы при её тестировании в течение календарной продолжительности $T_{\text{календ}}$
1. Авиация	$1,246 \cdot 10^{-7}$	5,23	8
2. Мониторинг и обеспечение безопасности	$0,238 \cdot 10^{-7}$	1,00	43
3. Телекоммуникации, мобильные устройства	$2,738 \cdot 10^{-7}$	11,5	3,5
4. Управление производственными процессами	$0,754 \cdot 10^{-7}$	3,17	14
5. Автоматизированные системы управления (технологии Big Data)	$4,563 \cdot 10^{-7}$	19,2	2,5
6. Разработка программ, моделирование, обучение	$3,353 \cdot 10^{-7}$	14,1	3
Среднее		8,83	12

Согласно работе [10] в случае неопределённости можно считать, что одна строка кода компьютерной программы транслируется в 10 машинных команд, т. е. значение E_L в

равенствах (10) может быть принято $E_L = 10$.

С учётом формулы (9), равенств (10) и последней части выражения (8) формула для расчёта коэффициента Q примет вид

$$Q = \exp\left(\frac{\lambda_0}{N_0} t\right) = \exp\left(\frac{C_B K_\Sigma \frac{R}{L E_L} N_0}{N_0} t\right) = \exp\left(\frac{C_B K_\Sigma R}{L E_L} t\right). \quad (11)$$

Преобразуем формулу (11), приняв во внимание значение $C_B = 0,238 \cdot 10^{-7}$ и следующие переводные и уточняющие коэффициенты, в том числе, обеспечивающие для времени t размерность в ч (часах):

множитель $3600 \cdot 10^{10^6}$ – коэффициент перевода быстродействия процессора R из размерности «операций в с» в размерность «миллионов операций в ч»;

множитель $0,7$ – средний поправочный коэффициент, учитывающий реальное быстродействие процессора относительно пикового значения R , указываемого производителем.

Тогда

$$Q = \exp\left(\frac{C_B K_\Sigma R \cdot 3600 \cdot 10^6 \cdot 0,7}{L E_L} t\right) = \exp\left(\frac{60 K_\Sigma R}{L E_L} t\right). \quad (12)$$

Из выражения (12) получим формулу для расчёта времени t выполнения компьютерной программы при её тестировании для обеспечения требуемого значения коэффициента эффективности тестирования Q :

$$t = \frac{L E_L \ln(Q)}{60 K_\Sigma R}, \text{ ч}. \quad (13)$$

Формулу (13) можно рассматривать в качестве модели для прогнозирования процессорного времени тестирования t планируемой к разработке прикладной компьютерной программы. В модель (13) пиковое быстродействие процессора компьютера R должно подставляться в размерности «миллионов операций в с». Прогнозное время t выполнения компьютерной программы при её тестировании будет получено в часах.

Для наглядности влияния коэффициента тестирования Q и объёма компьютерной программы в строках программного кода L на процессорное время выполнения компьютерной программы t при её тестировании получена номограмма для случая, когда указываемое производителем пиковое быстродействие процессора $R = 500$ млн операций / с, $E_L = 10$, коэффициент $K_\Sigma = 19,2$ (рисунок 1). На этом рисунке обозначение LOC означает число строк программного кода.

На основе анализа статистических (экспериментальных) данных о тестировании и надёжности прикладных компьютерных программам разного функционального назначения [8] определён средний процент процессорного времени выполнения компьютерных программ при их тестировании в случае использования общепринятых методов тестирования и рабочей календарной продолжительности $T_{\text{календ}}$, составляющей не менее 40 процентов от общего бюджетного времени, отводимого на разработку компьютерной программы (см. таблицу 1). Для прикладных компьютерных программ, используемых в составе автоматизированных систем управления, средний процент выполнения компьютерных программ при их тестировании в течение рабочего календарного времени $T_{\text{календ}}$ составил примерно 2,5 %.

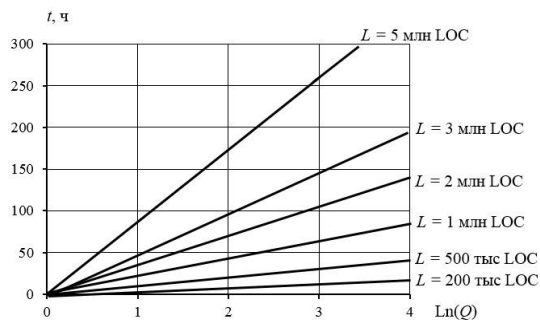


Рисунок 1. – Номограмма определения процессорного времени тестирования прикладной компьютерной программы

Предполагая, что прикладные компьютерные программы, используемые в технологиях Big Data, наиболее близки к этой области деятельности, получено выражение, которое рекомендуется для прогнозирования рабочей календарной продолжительности тестирования прикладных компьютерных программ, предназначенных для технологий Big Data:

$$T_{\text{календ}}^{(\text{BigData})} = \frac{100t}{2,5} = 40t . \quad (14)$$

При этом процессорное время t может быть определено по модели (13) в зависимости от требуемого значения коэффициента тестирования Q , который должен выбираться с учётом заданной эксплуатационной интенсивности отказов $\lambda_{\text{экс}}$, определяемой для компьютерной программы по формуле (2).

Тестирование одной и той же прикладной компьютерной программы может выполняться несколькими специалистами. Поэтому важно знать нормативную трудоёмкость процедуры тестирования $T_{\text{норм}}$, необходимую для достижения требуемого коэффициента тестирования программы:

$$T_{\text{норм}} = \frac{T_{\text{календ}}}{T_{1/\text{чел.-дн}}}, \text{ чел.-дн.}, \quad (15)$$

где $T_{1/\text{чел.-дн}}$ – средняя продолжительность одного рабочего дня (размерность: ч / чел.-дн.).

Процессорное время выполнения прикладных компьютерных программ t при их тестировании по всей совокупности компьютерных программ разных областей применения в среднем составляет 12 процентов в общей рабочей календарной продолжительности тестирования. С учётом этого среднее календарное (рабочее) время тестирования компьютерной программы ($T_{\text{календ}}$) можно находить как

$$T_{\text{календ}} = \frac{100t}{12} \approx 8,1t . \quad (16)$$

Моделью (16) рекомендуется пользоваться в случаях, когда сложно чётко установить область (сферу) применения прикладной компьютерной программы согласно перечню областей применения, приведённому в таблице 1.

Заключение.

На основе статистических (экспериментальных) данных о тестировании и надёжности прикладных компьютерных программ разных областей применения предложена модель прогнозирования времени тестирования компьютерных программ, в том числе, предназначенных для технологий Big Data. Модель позволяет спрогнозировать требуемое время тестирования планируемой к разработке компьютерной программы для обеспечения заданного уровня её эксплуатационной надёжности. Такое прогнозирование полезно, так как позволяет ещё до разработки прикладной компьютерной программы получить представление о её ожидаемой эксплуатационной надёжности и

оценить предполагаемые для этого затраты времени и средств.

Список литературы

- [1] Чуканов, В. О. Методы обеспечения аппаратно-программной надёжности вычислительных систем / В. О. Чуканов, В. В. Гуров, Е. В. Прокопьева [Электронный ресурс]. – 2014. – Режим доступа : <http://www.mcst.ru/metody-obespecheniya-apparatnoprogrammnoj-nadezhnosti-vychislitelnykh-sistem>. – Дата доступа: 4 апреля 2021.
- [2] Боровиков, С. М. Прогнозирование ожидаемой надёжности прикладных программных средств с использованием статистических моделей их безотказности / С. М. Боровиков, С. С. Дик // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3 – 4, 2018 / editorial board: M. Vatura [etc.]. – Minsk, BSUIR, 2018. – P. 348 - 354.
- [3] Боровиков, С. М. Возможный подход к оценке надёжности прикладных программных средств для технологий Big Data / С. М. Боровиков, Лэ Ван Там, С. С. Дик // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня : сборник материалов V Международной научно-практической конференции, Минск, 13–14 марта 2019 г. В 2 ч. Ч. 2 / Белорусский государственный университет информатики и радиоэлектроники; редкол. : В. А. Богуш [и др.]. – Минск, 2019. – С. 77-83.
- [4] Borovikov S. M., Kaziuchyts V. O., Khoroshko V. V., Dick S. S., Klinov K. I. Assessment of expected reliability of applied software for computer-based information systems. Informatics, 2021, vol. 18, no. 1, pp. 84–95 (in Russian). <https://doi.org/10.37661/1816-0301-2021-18-1-84-95>
- [5] Шубинский, И. Б. Функциональная надёжность информационных систем. Методы анализа / И. Б. Шубинский. – М. : «Журнал Надёжность», 2012. – 296 с.
- [6] Karanta, I. Methods and problems of software reliability estimation: VTT Working Papers 63 / I. Karanta. – Espoo : VTT Technical Research Centre of Finland, 2006. – 57 p.
- [7] ГОСТ 27.205-97. Надёжность в технике. Проектная оценка надёжности сложных систем с учётом технического и программного обеспечения и оперативного персонала. Основные положения. – Минск : Госстандарт Республики Беларусь, 2005. – 22 с.
- [8] McCall, J. A. Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing [Electronic resource] / J. A. McCall [et al.]. – 1992. – Mode of access: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a256164.pdf>. – Date of access: 20.04.2020.
- [9] Модели оценки надёжности программных средств [Электронный ресурс]. – Режим доступа : <https://helpiks.org/4-73503.html> (дата обращения: 3.04.2021).
- [10] Чуканов, В. О. Надёжность программного обеспечения и аппаратных средств систем передачи данных атомных электростанций : учеб. пособие / В. О. Чуканов. – М. : МИФИ, 2008. – 168 с.

MODEL FOR PREDICTING THE TEST TIME OF APPLIED COMPUTER PROGRAMS FOR BIG DATA TECHNOLOGIES

S.M. BOROVIKOV
PhD
Associate professor of
the BSUIR

V.O. KAZIUCHYTS
Master of engineering
Postgraduate student of
the BSUIR

S.K. DICK
PhD
Associate professor of
the BSUIR

S.S. DICK
*Master of Engineering and
Technology*
Project manager of
"Itranshen" Company

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
Itransition Company, Republic of Belarus
E-mail: bsm@bsuir.by

Abstract. Applied computer programs used for Big Data technologies contain hundreds of thousands or millions of lines of program code and therefore, after writing programs, they always contain hidden errors. The presence of these hidden errors determines the level of reliability of programs. Before testing begins, a computer program has a certain level of reliability, which usually does not meet the requirements of practice. The task of the testing phase is to identify and eliminate the most critical errors and to reduce the proportion of latent errors to a minimum. The question arises about the duration of the test execution time. On the basis of statistical data on testing and the reliability of computer programs of various functional purposes, a model has been obtained that allows one to determine the predicted testing time, which provides a given level of its operational reliability for the developed program.

Keywords: applied computer programs, operational reliability, test time, test time model, test efficiency.