

ОЦЕНКА НАДЕЖНОСТИ МОДУЛЬНЫХ ПРОГРАММНЫХ СРЕДСТВ

А.В. Анцыпов, В.В. Бахтизин

Белорусский государственный университет информатики и радиоэлектроники,
220027, Минск, Республика Беларусь, ул. П. Бровки, 6, тел.: +375 29 7566009
E-mail: ancipov@gmail.com

Предлагается методика оценки надежности программного средства, основанная на оценке надежности отдельно взятых модулей. Предложены метрики для оценки веса модуля. Предложена методика для оценки качества документации. Предложены метрики для оценки качества документации и установлена зависимость между качеством документации и надежностью программного средства.

This paper suggests a method of reliability estimation of software. It is based on: single module reliability estimation. It suggests metrics for module weight estimation. It suggests a method of estimation of quality of documentation. It suggests metrics for quality of documentation estimation. The relationship was established between the quality of documentation and software reliability.

Введение

В настоящее время деятельность многих организаций и предприятий напрямую зависит от правильной обработки информации соответствующими программными средствами.

К настоящему времени более 90 стран мира адаптировали стандарты ISO 900x на национальном уровне [1]. Во многих случаях заказчикам необходимо знать, каким образом на предприятии обеспечивается качество разрабатываемого программного средства. В этом случае соответствие стандартам является гарантией качества программного средства.

Одним из основных стандартов качества в области инженерии программного обеспечения в настоящее время является стандарт ISO/IEC 9126-1:2001 – Software engineering – Product quality (Информационная технология. Качество программного продукта) [2].

ISO/IEC 9126 регламентирует шесть основных характеристик качества программных средств: функциональность, надежность, эффективность, практичность, сопровождаемость, мобильность.

Одной из основных характеристик качества программных средств является надежность [2, 3]. Оценка надежности разрабатываемого программного средства является одной из важных задач, стоящих перед разработчиком программного средства.

На надежность программного средства влияет очень много факторов, одним из которых является качество документации. Под документацией понимается совокупность документов определяющих процедуры и ресурсы, необходимые для управления и обеспечения программного проекта. В данной статье исследуется зависимость надежности разрабатываемого программного средства от качества документации.

Оценка качества документации

Для количественной оценки качества документации предлагаются следующие метрики.

1. Соответствие документации программному средству. Документация должна соответствовать разрабатываемому программному средству. Если обозначить A – количество документов (спецификация требований, планы тестирования и т. п.) соответствующих разрабатываемому программному средству, B – общее количество документов, которые должны соответствовать разрабатываемому программному средству, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B$, $0 < X < 1$.

2. Документирование выходных результатов. При разработке программного средства разработчик должен документально оформить выходные результаты. Если обозначить A – количество документально оформленных результатов, B – общее число выходных результатов, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B$, $0 < X < 1$.

3. Документирование возникших проблем. При разработке программного средства разработчик должен документально оформить возникающие проблемы и устранить несоответствия. Если обозначить A – количество документально оформленных возникших проблем, B – общее число возникших проблем, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B$, $0 < X < 1$.

4. Документирование используемых стандартов и методов. При разработке программного средства разработчик должен выбрать, адаптировать и использовать те стандарты, методы, языки программирования, которые документально оформлены. Если обозначить A – количество документально оформленных стандартов, методов, языков программирования, B – общее количество используемых стандартов, методов, языков программирования, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B, 0 < X < 1$.

5. Документирование планов проведения работ. При разработке программного средства разработчик должен разработать планы проведения работ процесса разработки. Планы должны включать все требования связанные с разработкой, включая безопасность и защиту. Если обозначить A – количество документально оформленных требований, B – общее количество требований, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B, 0 < X < 1$.

Необходимо учитывать, что каждая метрика в различной степени влияет на качество документации. В мировой практике принято использовать методы ранжирования с целью определения весовых коэффициентов метрик. Каждый эксперт оценивает вес каждой метрики по шкале относительной значимости в диапазоне от 0 до 1. В качестве общей меры оценки в данной работе используется средневзвешенное значение веса i -й метрики V_i . Условие нормирования для V_i имеет вид $\sum_{i=1}^{N_m} V_i = 1$.

$$V_i = \frac{\sum_{j=1}^{N_e} v_{ji}}{\sum_{i=1}^{N_m} \sum_{j=1}^{N_e} v_{ji}}$$

где N_m – количество метрик, N_e – количество экспертов, v_{ii} и v_{ji} – соответственно экспертная оценка i -й и j -й метрики i -го эксперта.

Зная значения метрик и их веса, качество документации разрабатываемого программного средства $Q_{пс}$ можно найти по формуле

$$Q_{пс} = \sum_{i=1}^{N_m} (V_i * X_i),$$

где X_i – значение i -й метрики.

Оценка надежности программного средства

Зная качество документации разрабатываемого программного, следующим шагом является оценка его надежности. Как правило, современные программные средства имеют модульную структуру. Оценка надежности всего программного средства является достаточно сложной задачей. Существенно проще оценить надежность отдельных взятых модулей, а потом дать общую оценку надежности всего программного средства.

При оценке надежности модуля необходимо учитывать не только количество ошибок, но также и последствия, которые могут произойти в результате возникновения ошибки, и вероятность ввода k -го набора данных, приведшего к ошибке. В соответствии с международным стандартом IEEE 1044.1-1995 Guide to Classification for Software Anomalies (Классификация ошибок программного средства) [2] все ошибки можно разбить на 6 основных групп, представленных в таблице.

Таблица. Классификация ошибок программного средства в соответствии со стандартом IEEE 1044.1-1995.

Название ошибки	Описание
Priceless	Ошибка, после возникновения которой модуль перестает функционировать
High	Ошибка, после возникновения которой модуль функционирует, но функционирует неверно
Medium	Ошибка, после возникновения которой пользователь может продолжать использовать модуль, но данная ошибка может нанести существенный урон данным
Low	Ошибка, последствия которой носят незначительный характер
None	Ситуация, когда одна часть пользователей считает, что это нормальное поведение модуля, а вторая часть, что произошла ошибка
Detrimental	Ситуация, когда возможно появление ошибки (примером такой ситуации может быть постоянно растущий объем оперативной памяти, используемой модулем)

Кроме того, следует учесть ситуацию, когда ошибка при вводе k -го набора данных может не возникнуть.

Предлагается следующая методика определения оценки надежности модуля. Каждому типу ошибки экспертным путем присваивается вес E_j , условие нормирования для E_j имеет вид $\sum_{j=0}^6 E_j = 1$. Чем выше вес ошибки, тем более серьезные последствия она несет. E_0 соответствует ситуации отсутствия ошибки. Очевидно, что значение E_0 равно нулю. Тогда оценка надежности i -го модуля p_i определяется по формуле

$$p_i = 1 - \sum_{k=1}^D (E_j * b_k), \quad (1)$$

где D – количество наборов данных, введенных пользователем; E_j – вес произошедшей k -й ошибки; причем $j \in \{0,1,2,3,4,5,6\}$; b_k – вероятность ввода k -го набора данных.

Условие нормирования для b_k имеет вид $\sum_{k=1}^D b_k = 1$.

Приведем пример. Оцениваемый модуль должен выполнять три операции: получение всех записей из базы данных, добавление новой записи в базу данных и удаление не нужной записи из базы данных. Пусть вероятность получения всех записей равна 0.5, вероятность добавления новой записи равна 0.3 и вероятность удаления не нужной записи равна 0.2. Допустим, что получение всех записей прошло без ошибок, добавление новой записи прошло с ошибкой типа High (пользователь получил ответ об успешном завершении операции, хотя на самом деле запись не была добавлена), а удаление не нужной записи прошло с ошибкой типа Detrimental (запись была удалена, но процесс удаления занял продолжительный период времени). Веса ошибок Priceless, High, Medium, Low, None, Detrimental соответственно равны 0.5, 0.3, 0.1, 0.05, 0.035, 0.015. Тогда оценка надежности модуля будет равна

$$p_i = 1 - (0 * 0.5 + 0.3 * 0.3 + 0.015 * 0.2) = 0,907.$$

При оценке надежности всего программного средства необходимо учитывать, что различные модули в различной степени влияют на надежность программного средства (например, ошибка, возникшая при сохранении информации более критична, чем ошибка, возникающая при ее отображении).

Введем понятие веса модуля, как степени влияния отказа модуля на отказ всего программного средства.

Обозначим вес i -го модуля как s_i . Условие нормирования для s_i имеет вид $\sum_{i=1}^n s_i = 1$,

где n – количество модулей в программном средстве. Чем ближе значение s_i к единице, тем больше отказ i -го модуля влияет на отказ всего программного средства.

Оценка надежности разрабатываемого программного средства $P_{ПС}$ определяется по формуле

$$P_{ПС} = \sum_{i=1}^n (s_i * p_i). \quad (2)$$

Предлагается следующий набор метрик для оценки веса модуля. При этом в метриках основанных на сравнении с базовыми аналогами (метрики 5, 6), будем считать, что аналоги разработаны персоналом такой же квалификации и при той же организации процесса разработки.

Пусть программа содержит несколько модулей.

1. Метрика “Потребляемые ресурсы модулем”. Как правило, более весомый модуль потребляет и больше ресурсов.

Если обозначить A – объем потребляемых ресурсов модулем (объем оперативной памяти и т. п.), B – общий объем потребляемых программным средством ресурсов, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B$, $0 < X < 1$.

2. Метрика “Время работы модуля”. Как правило, чем большее время находится модуль в работе, тем он более весом.

Если обозначить A – время работы модуля, B – общее время выполнения всей задачи, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 < A < B$, $0 < X < 1$.

3. Метрика “Частота использования модуля”. Вероятность появления ошибки возрастает с ростом количества вызовов модуля, поэтому часто вызываемые модули более весомы нежели модули вызываемые реже.

Если обозначить A – количество вызовов модуля, B – общее количество всех вызовов модулей, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $1 \leq A < B$, $\frac{1}{B} < X < 1$.

4. Метрика “Зависимость модулей”. Как правило, в программном средстве модули связаны в древовидную структуру, где неправильная работа одного модуля приводит к неправильной работе других модулей.

Если обозначить A – количество модулей зависящих от данного модуля, B – общее количество всех модулей, то числовую оценку метрики можно найти по формуле $X = \frac{A}{B}$. При этом справедливо: $0 \leq A < B, 0 < X < 1$.

5. Метрика “Объем модуля”. Как правило, наиболее весомые модули имеют больший объем кода.

Если обозначить A – объем модуля (например, число строк кода), B – объем кода аналогичного по функциональности модуля, взятый из статистических данных организации-разработчика программного средства, то числовую оценку метрики можно найти по формуле $X = \begin{cases} \frac{A}{B}, & \text{если } A < B \\ 1, & \text{если } A \geq B \end{cases}$.

При этом справедливо: $0 < A, 0 < X \leq 1$.

6. Метрика “Команда разработки модуля”. Как правило, в разработке наиболее весомых модулей принимает участие большее количество разработчиков.

Если обозначить A – количество разработчиков, принимающих участие в разработке модуля, B – количество разработчиков, принимавших участие в создании аналогичного по функциональности модуля, взятое из статистических данных организации-разработчика программного средства, то числовую оценку

метрики можно найти по формуле $X = \begin{cases} \frac{A}{B}, & \text{если } A < B \\ 1, & \text{если } A \geq B \end{cases}$. При этом справедливо: $0 < A, 0 < X \leq 1$.

Вес i -го модуля s_i определяется по формуле

$$s_i = \frac{\sum_{j=1}^{m_i} (V_{ij} * X_{ij})}{\sum_{j=1}^{m_i} V_{ij}}, \quad \sum_{j=1}^m V_{ij} = 1, \quad (3)$$

где m – общее количество метрик; m_i – количество метрик используемых для расчета значения веса i -го модуля ($m_i \leq m$); X_{ij} – значение j -ой метрики, относящейся к i -му модулю; V_{ij} – весовой коэффициент j -ой метрики, относящейся к i -му модулю, определяющийся по статистическим данным организации разработчика программного средства или экспертным путем.

Зная надежность модулей p_i и их веса s_i , по формуле (2) можно найти надежность всего программного средства.

Зависимость надежности программного средства и качества документации

Проведен эксперимент, в рамках которого разрабатывалось программное средство, используемое при проектировании оптоволоконных сетей. Разработка программного средства носила итерационный характер. От первой и до финальной версии программного средства было выпущено сто промежуточных версий. Для каждой версии программного средства была найдена его надежность и качество документации (представленные на рисунке).

На рисунке видно, что с ростом качества документации, также возрастает и надежность разрабатываемого программного средства.

Для установления зависимости качества документации и надежности разрабатываемого программного средства проведен корреляционный анализ и определен нормированный коэффициент корреляции Брауэ – Пирсона r [5].

$$r = \frac{\sum_{i=1}^{100} [(x_i - \bar{x}) * (y_i - \bar{y})]}{\sqrt{\sum_{i=1}^{100} (x_i - \bar{x})^2 * \sum_{i=1}^{100} (y_i - \bar{y})^2}} \quad (4)$$

где x_i – качество документации i -й версии программного средства, \bar{x} – среднее значение качества документации, y_i – значение надежности i -й версии программного средства, \bar{y} – среднее значение показателя надежности.

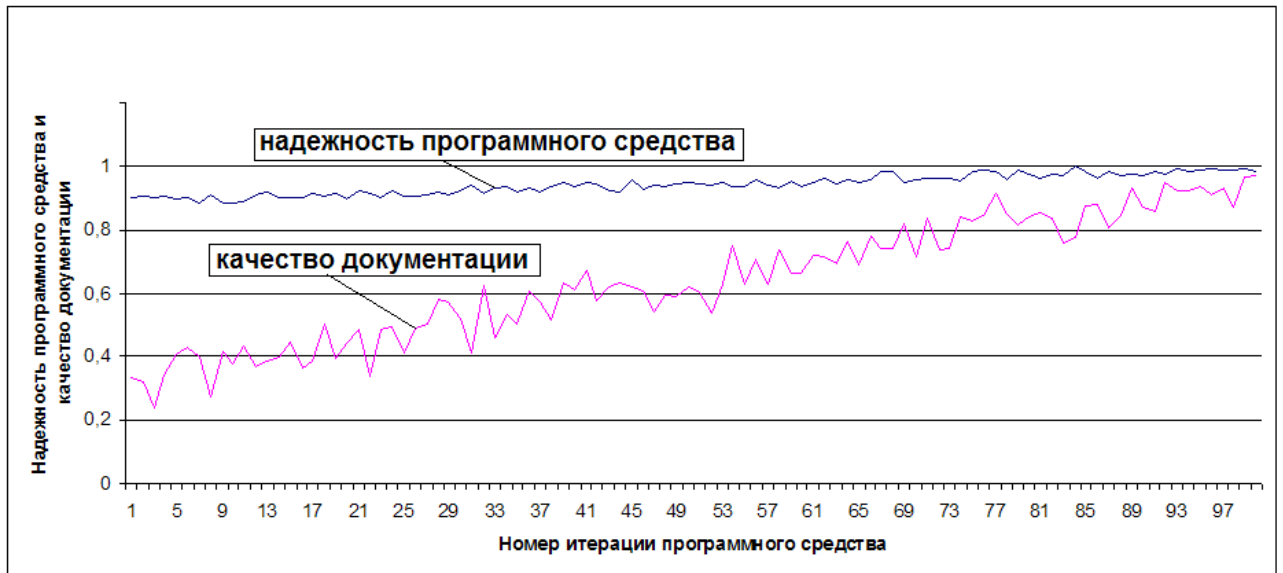


Рисунок. Сравнение значений надежности программного средства и качества документации

Коэффициент корреляции качества документации и надежности программного средства составил 0.913, что говорит о высокой степени зависимости надежности программного средства и качества документации разрабатываемого программного средства.

Для проверки значимости коэффициента корреляции находится показатель связности t [5]

$$t = \frac{|r|}{1/\sqrt{n}},$$

где r – коэффициента корреляции, n – количество версий разрабатываемого программного средства.

Для $\alpha = 0,05$ получаем $t_{кр} = 2.58$ и t равное 9.13, так как $t > t_{кр}$, следовательно зависимость между качеством документации и надежностью программного средства следует считать значимой.

В данной работе предлагается методика оценки надежности программного средства, основанная на оценке надежности отдельно взятого модуля, предложены метрики для оценки веса модуля, предложена методика для оценки качества документации, предложены метрики для оценки качества документации и установлена зависимость между качеством документации и надежностью программного средства.

Используя предложенную методику, у разработчика появляется возможность оценивать и контролировать надежность разрабатываемого программного средства. Достоинствами предложенной методики является то, что она основана на общепризнанных мировых стандартах и проста в использовании.

1. *Лунаев В.В.* Надежность программных средств. – М.: СИНТЕГ, 1998. – 280 с.
2. *ISO/IEC 9126-1:2001* Software engineering – Product quality.
3. *Бахтизин В.В., Глухова Л.А.* Стандартизация и сертификация программного обеспечения. – Мн.: БГУИР, 2006. – 200 с.
4. *IEEE Std 1044.1-1995* Guide to Classification for Software Anomalies.
5. *Гайдьшев И.* Анализ и обработка данных. – Санкт-Петербург: Питер, 2001. – 232 с.