



Кабак И.С.

ОПТИМИЗАЦИЯ ЗАТРАТ НА РАЗРАБОТКУ И ЭКСПЛУАТАЦИЮ СЛОЖНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

Задачи информатизации образования связаны с разработкой сложного программного обеспечения, решающего как информационные, так и управляющие задачи. Разработка такого программного обеспечения приводит к существенным финансовым затратам. Кроме затрат на разработку программного обеспечения необходимо оценивать их затраты, связанные с устранением последствий его отказов. Все эти затраты желательно оценить еще на стадии разработки технического задания на программный продукт. Данная статья посвящена вопросам оценки и минимизации вышеупомянутых затрат.

Ключевые слова: моделирование, надежность программного обеспечения.

В условиях рыночной экономики главным фактором становится качество продукции. Качество включает ряд показателей, в том числе – надежность. Для производства качественной продукции требуются надежные системы автоматизации. Автоматизированные системы управления (АСУ) включают аппаратную часть и программное обеспечение (ПО). Надежность ПО АСУ – это комплексное свойство, где главным показателем является безотказность. Безотказность – это свойство, характеризующее способность сохранять работоспособное состояние при соблюдении заданных режимов и условий эксплуатации. В настоящее время возрастает значение АСУ техническими системами. Они используются в промышленности, на транспорте, в авиационно-космическом комплексе, в оборонных и др. областях

Современные АСУ техническими системами – это сложные комплексы оборудования, где используются компьютеры, сети передачи данных, контроллеры, датчики, исполнительные механизмы и другие устройства. Они включают в себя, как необходимый компонент, сложное ПО. Качество и надежность работы всей АСУ в существенной степени зависит от качества и надежности ПО. Требования к качеству и надежности ПО необходимо обеспечивать при выполнении всех процессов жизненного цикла АСУ. Для оценки надежности ПО АСУ необходимо собирать и обрабатывать данные об отказах как во время разработки ПО, так и при его эксплуатации. Причиной отказов ПО АСУ являются, в первую очередь, ошибки в программном коде.

Целью данной статьи является снижение затрат на разработку ПО и потерь при эксплуатации комплекса оборудования, управляемого этими программами, в ходе производственного, учебного или научно-исследовательского процесса.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести оценку параметра надежности сложного ПО – среднего времени наработки на отказ;
- на основе среднего времени наработки на отказ ПО определить ожидаемые суммарные потери при отказах для комплекса оборудования;
- построить модель зависимости стоимости разработки сложного ПО от его надежности (среднего времени наработки на отказ);
- определить экономически обоснованный уровень надежности сложного ПО.

Оценка затрат на разработку программного обеспечения

Разработка программного обеспечения включает несколько стадий, в том числе, разработку технического задания и алгоритмизацию, автономную и комплексную отладку и приемо-сдаточные испытания. В зависимости от типа программ и используемых инструментальных средств, языков и технологий программирования, доля затрат на эти стадии различна. Однако, при создании сложных управляющих и информационных аппаратно-программных систем, включающих дорогостоящее оборудование, особо велика доля комплексной отладки, требующей наибольших затрат. Комплексная отладка часто совмещается с приемо-сдаточными испытаниями, поэтому можно считать, что эта стадия является завершающей для процесса разработки. Решение вопроса прекращения разработки и сдачи готового ПО в эксплуатацию может быть принято только на этой стадии.

При комплексной отладке, используются не только локальные ресурсы, но и дополнительное удаленное оборудование. Кроме расходов на заработную плату сотрудников при поиске и устранении ошибок в программном продукте, амортизацию локальной вычислительной техники и средств телекоммуникаций, необходимо учитывать еще расходы на аренду оборудования производственного комплекса, а также затраты на использование вычислительной и другой вспомогательной техники, стоимость заготовок и инструмента при непроизводственных действиях, оплату труда удаленных специалистов, электрической и тепловой энергии и др.

Рассмотрим в качестве примера проект разработки ПО для дистанционного обучения инженеров по специальности автоматизация производственных процессов и инженеров-машиностроителей. В табл. 1 приведены данные о недельной стоимости комплексной отладки в течение 10 недель. Данные приведены

с учетом стоимости всех затрат, в том числе, с учетом амортизации оборудования.

Приведенные в табл. 1 данные позволяют выдвинуть гипотезу о линейном характере зависимости затрат на отладку от ее длительности (от времени отладки в неделях). Используя метод Хи-квадрат, определим правомерность такого предположения. Обработка данных из табл. 1, позволяет говорить о справедливости гипотезы с вероятностью 99,5 %. Таким образом, подтверждается предположение о линейной зависимости роста затрат на комплексную отладку от времени. Эти затраты показаны на рис. 1 штриховой линией:

$$C^1 = c \cdot t, \quad (1)$$

где c – удельные затраты на разработку ПО в единицу времени; t – время отладки.

Каждый отказ ПО приводит к дополнительным финансовым потерям, связанным с приостановками технологического процесса и исправлением ошибок. Для определения потерь от отказа ПО, необходимо знать зависимость интенсивности отказов от времени отладки. Такая зависимость приведена в работах автора [1-3].

Поток отказов ПО считается ординарным. Интенсивность потока отказов совпадает с параметром потока отказов [1]. Среднее число отказов объекта с восстановлением в единицу времени совпадает с интенсивностью потока отказов. Интенсивность потока отказов определялась по модели надежности ПО [1]:

$$H(t) = \sum_{i=1}^N a_i \cdot \exp(-b_i \cdot t), \quad (1)$$

где N – количество различных групп операторов или классов; у которых одинаковые характеристики надежности; $a_1, a_2, a_3, \dots, a_N, b_1, b_2, b_3, \dots, b_N$ – коэффициенты модели.

Зададим среднюю стоимость отказа программного обеспечения C_1 и получим формулу, связывающую потери от отказа ПО C_0 с интенсивностью потока его отказов $H(t)$:

$$C_0 = C_1 \cdot H(t). \quad (2)$$

Подставив значение $H(t)$ из формулы (1) в итоге получим:

$$C_0 = C_1 \cdot \sum_{i=1}^N a_i \cdot \exp(-b_i \cdot t). \quad (3)$$

На рис. 1 кривая C_0 изображена непрерывной линией.

Таблица 1. Затраты на комплексную отладку программного обеспечения учебной гибкой производственной системы (МГТУ «СТАНКИН») в условных единицах

1 нед.	2 нед.	3 нед.	4 нед.	5 нед.	6 нед.	7 нед.	8 нед.	9 нед.	10 нед.
0,98	1,02	1,01	0,99	0,98	1,01	1,02	0,99	0,99	1,01

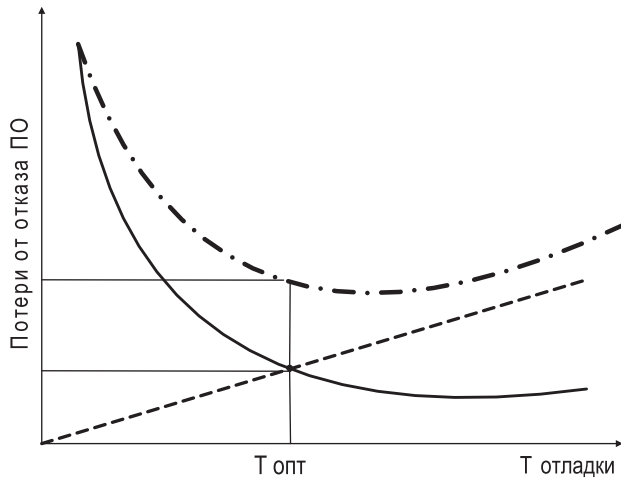


Рис. 1. Суммарные затраты на разработку ПО

Суммарные затраты на отладку ПО могут быть определены как сумма C^1 и C_0 :

$$C = C^1 + C_0 = c \cdot t + C_1 \cdot \sum_{i=1}^N a_i \cdot \exp(-b_i \cdot t) \quad (4)$$

На рис. 1 общие затраты C изображены штрихпунктирной утолщенной линией.

Функция C имеет минимум, что проиллюстрировано на рис. 1. Таким образом, существует оптимальное время отладки ПО ($T_{\text{отл}}$), для которого суммарные финансовые потери минимизированы.

Постановка задачи оптимизации надежности программного обеспечения и выбор метода

В работе [1] приведена формула, связывающая интенсивность отказов ПО с его структурой, интенсивностью отказов его модулей, и вероятностью работы модулей системы. Согласно этой формуле, интенсивность отказов ПО составляет:

$$H(t_1, t_2, \dots, t_N) = \sum_{i=1}^N v_i \cdot h_i(t_i), \quad (5)$$

где $h_i(t_i)$ — интенсивность отказов модуля i -й группы при отладке его в течении времени t_i , v_i — частотный коэффициент, учитывающий структуру системы и вероятности работы ее модулей.

Зависимость $H(t)$ рассмотрена в работе [2], где приведены результаты математического моделирования надежности. Было установлено, что значение функции $H(t)$ определяется по формуле:

$$H(t) = - \sum_{i=1}^N v_i \cdot \frac{1}{\tau_i} \cdot \ln \left[1 - \left(1 - \alpha_0^{-h_i^0 \tau_i} \right) \alpha_i^{\left(\frac{t}{\sum_{j=1}^N v_j \tau_j} \right)} \right], \quad (6)$$

где v_i — вероятность работы модуля i -го типа; τ_i — время его работы; α_i — комплексный коэффициент, учитывающий скорость отладки и ряд других параметров.

Зависимость $H(t)$ интенсивности потока отказов от времени работы ПО является нелинейной характеристикой.

Будем считать, что на отладку программной системы отведено время T_0 . Это время будет складываться из времени отладки всех модулей. Время отладки каждого модуля является неотрицательной величиной:

$$\begin{cases} T_0 = \sum_{i=1}^N t_i \\ t_1 \geq 0, t_2 \geq 0, \dots, t_N \geq 0. \end{cases} \quad (7)$$

Изменяя значения времени отладки i -го модуля t_i , при соблюдении системы ограничений (7), будем получать различные значения: функции $H(t_1, t_2, \dots, t_N)$.

Поставим задачу минимизировать значение интенсивности отказов ПО $H(t_1, t_2, \dots, t_N)$ (6) при ограничениях (7).

Учитывая нелинейный характер функций $h_i(t_i)$, задача оптимизации может быть решена методами нелинейного математического программирования.

Большинство методов нелинейного программирования не имеет ограничений по числу оптимизируемых переменных. С увеличением N трудоемкость вычислений резко возрастает, что затрудняет их практическое применение. Реальное программное обеспечение автоматизированных систем управления состоит из 100-500, а иногда и больше программных модулей. При столь больших значениях N многие из методов нелинейного программирования становятся неэффективными.

Рассмотрим случай, когда ПО состоит из большого числа программных модулей, которые характеризуются малым объемом кода и простой структурой.

Для оптимизации надежности такого ПО предлагается использовать алгоритм, основанный на принципе Беллмана [8,9].

Решение оптимизационной задачи на основе принципа Беллмана

Интенсивность отказов ПО в зависимости от времени отладки можно оценить по формуле [2]:

$$h(t) = -\frac{1}{\tau} \cdot \ln \left[1 - P^0 \cdot \alpha_i^{\frac{t}{\tau}} \right]. \quad (8)$$

Разложим логарифмическую функцию $\ln \left[1 - P^0 \cdot \alpha_i^{\frac{t}{\tau}} \right]$ в степенной ряд, подставим в выражение (8) и получим:

$$h(t) \approx \frac{1}{\tau} \cdot \sum_{j=1}^{\infty} \frac{\left(P^0 \cdot \alpha^{\frac{t}{\tau}}\right)^j}{j}. \quad (9)$$

Отметим, что значение P^0 весьма мало и составляет 0,01-0,03. Значение α не превосходит единицы. При возведении в степень j произведение $\left(P^0 \cdot \alpha^{\frac{t}{\tau}}\right)^j$ будет быстро убывать. Второй член степенного ряда будет не менее чем в 100 раз, а третий не менее, чем в 15000 раз меньше первого члена ряда.

Без существенного ухудшения точности можно ограничиться только первым членом ряда:

$$h(t) = \frac{P^0 \cdot \alpha^{\frac{t}{\tau}}}{\tau}.$$

Приведем правую часть последнего равенства к экспоненциальному виду:

$$h(t) = \frac{P^0}{\tau} \cdot \exp\left(-\frac{\ln\left(\frac{1}{\alpha}\right)}{\tau} \cdot t\right).$$

Для упрощения дальнейших выкладок, обозначим:

$$\frac{P^0}{\tau} = \beta,$$

$$\frac{\ln\left(\frac{1}{\alpha}\right)}{\tau} = \gamma.$$

Тогда получим:

$$h_i(t) = \beta_i \cdot \exp(-\gamma_i \cdot t). \quad (10)$$

Подставим (10) в (5):

$$H(t_1, t_2, \dots, t_N) = \sum_{i=1}^N v_i \cdot \beta_i \cdot \exp(-\gamma_i \cdot t). \quad (11)$$

Необходимо найти минимум интенсивности отказов (11) при системе ограничений (7). Для оптимизации функции интенсивности отказов, задаваемой выражением (11), воспользуемся методом динамического программирования, т.е. составим последовательность рекуррентных формул:

$$f_k(T_0) = \min_{0 \leq t \leq T_0} [f_{k-1}(T_0 - t) + v_k \cdot \beta_{ki} \cdot \exp(-\gamma_k \cdot t)], \quad (12)$$

$$f_1(T_0) = v_1 \cdot \beta_1 \cdot \exp(-\gamma_1 \cdot t). \quad (13)$$

В формуле (12) вычисляется минимум суммы двух функций, одна из которых является экспонентой. Можно показать, что функции $f_k(t)$ при любом допустимом

значении k также будут экспонентами. Доказательство этого утверждения производится в [1].

Проведение операции минимизации, необходимое для вычислений по формулам (12) и (13), связано с определенными трудностями. Для перехода к следующему шагу – приращению значения k , необходимо определить функцию $f_k(t)$ на отрезке $0 \leq t \leq T_0$.

Для получения расчетных рекуррентных формул продифференцируем выражение (12), стоящее в квадратных скобках (при $k = 2$) и приравняем результат дифференцирования к нулю, как это принято при решении экстремальных задач классического математического анализа.

Решив полученное уравнение относительно переменной t_1 и подставив его в выражение (12), получим аналитическое выражение $f_2(t)$. Приведем выражение для $f_2(t)$ к виду (13) и определим соответствующие значения новых коэффициентов при экспоненте и аргументе t .

Доказательство правомочности описанных выше математических действий и подробный вывод рекуррентных формул приводится в работе [1].

Для оценки практических результатов оптимизации надежности ПО были проанализированы значения интенсивностей отказов реального ПО. Значения интенсивности отказов оценивали для двух методов:

- для традиционного распределения времени отладки, когда все программные модули отлаживаются до одинаковой величины интенсивностей отказов;
- для оптимального распределения времени отладки, которое вычисляют с использованием соотношений (11) – (13).

В результате сравнения двух методов распределения времени отладки получили, что использование оптимизации сокращает время отладки на 9–20% при том же уровне надежности, что и без оптимизации.

Выводы

В работе была поставлена и решена задача рационального планирования времени отладки ПО. Рациональное планирование включает две оптимизационные задачи: определение минимального фонда времени отладки сложного ПО для достижения им заданного уровня интенсивности его отказов и задачу достижения заданной интенсивности отказов программного комплекса за отведенное для разработки время. При этом интенсивность отказов представлена как функция от многих переменных, от времени отладки и интенсивностей отказов программных модулей – компонентов ПО.

Показано, что большинство методов нелинейного программирования неприменимы для оптимизации надежности ПО ГПС.

При незначительном упрощении математической модели надежности ПО, без существенной потери точности, возможно применение принципа Беллмана. На основании этого принципа выведены рекуррентные формулы, позволяющие решить обе поставленные задачи.

Литература

1. **Кабак И.С., Рапопорт Г.Н.** Оценка надежности программного обеспечения по его математической модели // Проблемы создания гибких автоматизированных производств под. Ред. И.М. Макарова, К.В. Фролова, П.Н. Белянина. – М.: Наука. – 1987. – С. 236-245.
2. **Кабак И.С.** Математическая модель для прогнозирования и оценки надежности программного обеспечения // Вестник МГТУ «СТАНКИН». – 2014. – №1 – С. 123-126.
3. **Кабак И.С., Суханова Н.В.** О моделировании и оценке надежности сложных программных комплексов // Известия кабардино-балкарского государственного университета. – 2012. – № 5. – С. 74-76.
4. **Гроссман К., Каплан А.А.** Нелинейное программирование на основе безусловной минимизации. Новосибирск: Наука. – 1981. – 84 с.
5. **Полак Э.** Численные методы оптимизации. Единый подход. М: Мир. – 1974. – 376 с.
6. **Сea Ж.** Оптимизация. Теория и алгоритмы. М.: Мир. – 1973. – 244 с.
7. **Фиакко А., Мак-Кормик Г.** Нелинейное программирование. Методы последовательной безусловной минимизации. М.: Мир.- 1972. – 240 с.
8. **Беллман Р.Е., Дрейфус С.Е.** Прикладные задачи динамического программирования М.:Наука. – 1965. – 460 с.
9. **Беллман Р.** Динамическое программирование пер. с англ. М.: Иностран. литер. – 1960. – 400 с.
10. **Колмогоров А.Н., Фомин С.В.** Элементы теории функций и функционального анализа. М.: Наука. – 1976. – 544 с.
11. Заявка на изобретение 2013133304 Российская Федерация Способ и устройство технической диагностики сложного технологического оборудования на основе нейронных сетей [Текст]/ Соломенцев Ю. М., Шептунов С.А., Кабак И.С., Суханова Н.В.: заявитель

и патентообладатель Федеральное государственное бюджетное учреждение науки Институт конструкторско-технологической информатики Российской академии наук (ИКИ РАН.-№ ; заявл. 18.07.2013; уведомление о положительном результате формальной экспертизы.

12. Патент на полезную модель 75247 Российская Федерация МПК 7 G06F15/16 Модульная вычислительная система [Текст]/ Кабак И.С., Суханова Н. В.: заявитель и патентообладатель Кабак И.С., Суханова Н.В. – № 2008106859; заявл.26.02.2008; опубл. 27.07.2008, Бюл. № 21- 2 с.: ил.

13. Патент на изобретение 2398281 Российская Федерация МПК 7 G06N 3/06 Многослойная модульная вычислительная система [Текст]/ Соломенцев Ю.М., Шептунов С.А., Кабак И.С., Суханова Н. В.: заявитель и патентообладатель Федеральное государственное бюджетное учреждение науки Институт конструкторско-технологической информатики Российской академии наук (ИКИ РАН.-№ 2008143737 ;заявл.07.11.2008; опубл. 27.08.2010, Бюл. № 24- 8с.: ил.

14. Патент на изобретение 2417442 Российская Федерация МПК 7 G06N7/02 Способ построения систем нечеткой логики и устройство для его реализации [Текст]/ Соломенцев Ю. М., Шептунов С.А., Кабак И.С., Суханова Н.В.: заявитель и патентообладатель Федеральное государственное бюджетное учреждение науки Институт конструкторско-технологической информатики Российской академии наук (ИКИ РАН.-№ 2008150203; заявл. 19.12.2008; опубл. 27.04.2011, Бюл. № 12- 8 с.: ил.

15. **Кабак И.С.** Нейросетевая модель для прогнозирования и оценки надежности программного обеспечения// Вестник МГТУ «СТАНКИН». – 2014. №1 – С.107-111.

16. **Кабак И.С.** Создание больших аппаратно-программных нейронных сетей для систем управления. //Авиационная промышленность. – 2012. №4. – С. 57-61.