Анализ моделей надежности технических и программных систем

Пакулин Н.В., Лаврищева Е.М.,

Конференция

OS : DAY - Надежность

17-18 мая 2018, Москва

Докладчик

д.ф.м.н. проф., г.н.с. ИСП РАН

Лаврищева Е.М.

Основные темы доклада:

1. Модели надежности и качества технических и программных систем.

2. Средства оценки надежности систем.

1. Модели надежности систем

Надежность техники и систем

Надежность – это способность системы сохранять устойчивость и работоспособность без отказов, сбоев и ошибок в заданный промежуток времени.

Многие системы реального времени, радарные, медицинские, бортовые предъявляют высокие требования к надежности (недопустимость ошибок, устойчивость, достоверность, защищенность и др.).

Надежность - это целевая функция от числа оставшихся и не устраненных ошибок, основанная на вероятности и времени безотказной работы с учетом возникающих отказов, рисков угроз из среды (вирусы, атаки и др.).

Оценка надежности проводится с помощью собранных статистических данных при тестировании и эксплуатации системы – времени безотказной работы, количестве ошибок, отказов, отказов и дефектов.

Надежность системы можно определить как вероятностную функцию $P(i) = P$, если не оказалось отказов в i - прогонах программы на тестах; $P(t) = P$, если не было отказов в интервале времени $(0, t)$ выполнения программы. Вероятность безотказной работы системы можно определить по формуле: $P(t) \approx \exp(-t/T)$, которая определяет рост надежности.

Надежность в широком смысле

Термин reliability (надежность) обозначает способность системы обладать свойствами, обеспечивающими качественное выполнении функций, заданных в требованиях к системе.

Термин dependability означает пригодноспособность системы к:

- **использованию (availability),**
- **непрерывному функционированию (reliability),**
- **безопасности (safety) работы без катастрофических последствий,**
- **конфиденциальности (confidentability), секретности информации,**
- **сохраности информации и устойчивости (integrity) к изменениям,**
- **эксплуатационной завершенности ПО (aintainability), способной к устранению ошибок и восстанавлению.**

Надежность технических систем зависит от двух факторов:

- **качества отдельных технических компонентов системы;**
- **отсутствия дефектов в конструкции изготовления и способности компонентов работать качественно.**

Надежность программных систем зависит от этих же факторов и от случайных изменений данных и маршрутов исполнения программ, которые могут привести к неверные результатам или отказам и нарушить работоспособность системы.

Базовые типы ошибок в системах

Отказ **(failure) – это переход системы из рабочего состояния в нерабочее при получении результатов, не соответствующих допустимым значениям или при случайных изменениях данных, элементов системы и др.**

Дефект **(fault) – это некоторое непредвиденное событие при выполнении элемента программы или последствие ошибок в спецификации требований, документации и т.п.**

Ошибка **(error) – это следствие недостатков в описании программы или процесса, приводящего к неправильной интерпретации информации или данных (fault – как причина ошибки, которая ее вызывает).**

Интенсивность отказов **– это частота появления отказов или дефектов в системе при ее тестировании или эксплуатации; причины отказов выясняются и исправляются.**

Повышение надежности программ достигается путем:

- восстановления системы от ошибочных сбоев средствами информационной безопасности и избыточности;

- прогнозирования показателей надежности отдельных компонент, в которых обнаружены ошибки, дефекты или отказы в процессе тестирования и эксплуатации;

- оперативной защиты от непредумышленных, случайных искажениях процесса и данных.

Цель науки теории надежности систем

В теории надежности систем главным понятием является работоспособность функций системы, заданными в требованиях.

Анализ надежности функционирования систем проводится с помощью:

- собранных ошибок, отказов и дефектов, влияющие на оценку показателей надежности;

- методов и средств контроля и защиты системы от искажений программ, процессов и данных с использованием различных видов помехозащиты;

- методов и средств прогнозирования характеристик надежности на этапах ЖЦ с учетом требований к разработке функций системы;

- диагностических средств проверки правильности функционирования системы и при переходе ее в нерабочее состояние.

Диагностика системы в тестовом или функциональном режиме состоит в:
- контроле правильности выполнения функций на тестовых данных;

- проверке работоспособности системы;

- поиске и исследовании причин сбоев, отказов и ошибок.

Случайный характер процессов

С точки зрения теории надежности в системе возникают случайные процессы во времени T на последовательности времен $t_k < t_{k+1}$ и образующие случайную величину ξ в зависимости от значения t ($t \in T$ – моменты времени).

Если случайная величина дискретна, т.е. принимает конечное число значений в виде $x_1, x_2, \ldots, x_n$, то закон распределения ξ описывается вероятностью $P(\xi = x_i)$ и в общем случае $F(x) = P(\xi < x_i)$ называется функцией распределения случайной величины.

Случайный процесс с непрерывным временем, который описывается однородными событиями, называется пуассоновским процессом.

Если функции оказывается неслучайной величиной, то вычисляется математическое ожидание или дисперсия, как среднее отклонение от реализации такой функции.

Поиск случайных величин осуществляется стохастическими методами, процесс соответственно является стохастическим, вероятностным.

Если на множестве времени T определяется случайный процесс, то для всех его точек вычисляется случайная величина $\xi(t)$, которая и называется ее значением.

Случайный характер моделей надежности

С точки зрения теории случайных процессов процесс возникновения отказов в системе является стохастическим, то **модели надежности** также являются **стохастическими.**

К случайным процессам относится Марковский процесс, при котором его поведение после момента времени *t* зависит от его значения. Если случайные величины распределены по показательному, эрланговскому или гиперэрланговскому законам, то поведение системы описывается **Марковским процессом**.

Тестирование обеспечивает поиск дефектов и отказов, которые возникают случайно в системе и определяются с помощью функции

$p(t, x, s)$,

где $t < s$ – момент времени и $x \in X$ - положение точки.

Случайная функция удовлетворяет соотношению

$$P(t, x, u) = p(s, p(t, x, s), u),$$

где $t < s < u$ означает, что в момент времени *t* в точке *x* система в состоянии $p(t, x, u)$, переходит в состояние $p(t, x, s)$.

Марковский процесс с дискретным временем и конечным числом состояний называется **Марковской цепью**.

Функция $p_{ij}(t, x, s)$, при которой система в момент *t* переходит из *i* в *j*-состояние момента *s*, называется Колмогорова и определяется решением системы уравнений:

$$d/ds\ p_{ij}(t, s) = \sum P_{ij}(t, s)\, a_{kj}(s).$$

Случайный характер дефектов

Отказы в системе считаются случайными, если они возникают из-за дефектов.

Количество дефектов, время их выявления и местонахождения способствует образованию случайной величины.

Зависимость количества отказов от времени выполнения может быть больше, чем вероятность прохождения по фрагменту кода, который содержит дефект.

Для проведения оценки показателей, основанных на статистике дефектов, отказов и интенсивности выявленных отказов в системе, используются математические модели надежности, которые исходят из предположения, что найденные дефекты устраняются и при этом новые дефекты не вносятся.

К таким моделям относится класс *моделей роста надежности.*

Эти модели дают рост надежности при уменьшении количества дефектов в системе. При этом кривая роста надежности системы задает зависимость между временной протяженностью используемых результатов тестирования программы в системе координат и суммарным числом обнаруженных ошибок.

Классификация моделей надежности[1]

Прогнозирующие модели **надежности основаны на характеристиках создаваемой программы: длина, сложность, число циклов и степень их вложенности, количество ошибок и др. Это**

- модель Мотли–Брукса основана на длине и сложности структуры программы, количестве и типах переменных и интерфейсов.

- модель Холстеда дает прогнозирование количества ошибок в программе в зависимости от объема и числа операций на процессах ЖЦ и др.

Измерительные модели **основаны на измерении свойств надежности для зафиксированной конфигурации ПО и обнаруженные ошибки не исправляются. Это**

- модель Нельсона, которая основывается на выполнении *k*-прогонов программы при тестировании и позволяет определить надежность по формуле: $R(k) = \exp[-\sum \nabla t_j \lambda(t)]$,

где t_j – время выполнения *j*-прогона, $\lambda(t) = -[\ln(1 - q_i)\ \nabla j]$ и при $q_i \leq 1$ она интерпретируется как интенсивность отказов.

Оценочные модели **основаны на серии тестовых прогонов при тестировании систем в реальной среде и зависят от частоты проявления и устранения причин отказов и их интенсивности. Это**

- модель Мусы, которая учитывает интервалы между отказами, время между отказами распределяется по экспоненциальному закону и интенсивность отказов пропорционально числу ошибок.

---

[1]Мороз Г.Б., Лаврищева Е.М. Модели роста надежности ПО.- К.: 1992.

- **модель Гоэла** базируются на отказах и задает четыре класса моделей: без подсчета ошибок, с подсчетом отказов, с подсевом ошибок, модели с выбором областей входных значений.

- **модели Джелински** и Моранды, Шика Вулвертона и Литвуда–Вералла *без подсчета ошибок* основаны на измерении интервала времени между отказами и позволяют спрогнозировать количество ошибок, оставшихся в программе. После каждого отказа оценивается надежность и определяется среднее время до следующего отказа.

- **модель Муса–Окумото (**логарифмическая) допускает, что некоторые дефекты имеют большую вероятность проявления в виде отказов, снижение интенсивности отказов с каждым устраненным дефектом получает экспоненциальное распределение
Функция $m(t)$ зависит от времени и имеет вид:

$$m(t) = \ln(l_0 q\, t + 1),$$

где q – задает экспоненциальный спад интенсивности отказов с каждым устраненным дефектом, а функция интенсивности отказов $\lambda(t)$ имеет вид:

$$\lambda(t) = \lambda_0 \, / \, \lambda_0 \, \theta\, t + 1.$$

Модель Мусы устанавливает зависимость:

1) *среднего числа отказов* от времени функционирования *τ* (рис.3):

$$m=M_0\,[1-\exp\,(\frac{c\tau}{M_0T_0})],$$

где M_0 – общее число ошибок; T_0 – начальная наработка на отказ; *c*– коэффициент времени испытаний, т – время функционирования.

2) *средней наработки на отказ T* от времени функционирования (рис.4):

$T = T_0 \exp\,(\frac{c\tau}{M_0T_0})$, где M_0, T_0, *c* – величины, зависящие от наработки на отказ.

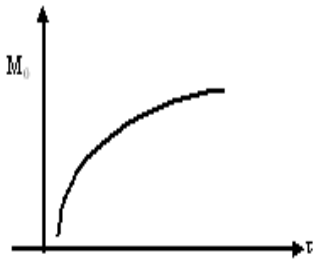

Рис.3

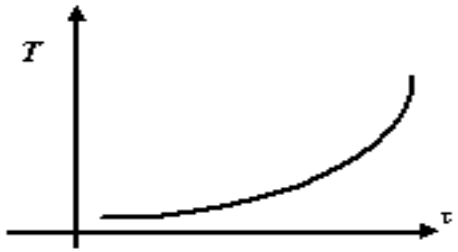

Рис.4

Модель Гоэла базируются на отказах и задает четыре класса моделей оценки надежности: без подсчета ошибок, с подсчетом отказов, с подсевом ошибок, модели с выбором областей входных значений.

(Модели Джелински и Моранды, Шика Вулвертона и Литвуда–Вералла (*без подсчета ошибок*) основаны на измерении интервала времени между отказами и позволяют спрогнозировать количество ошибок, оставшихся в программе.

Модель Гоело–Окумото **(экспоненциального роста) обеспечивает процесс обнаружения ошибок с помощью неоднородного пуассоновского процесса. В ней интенсивность отказов зависит от времени, а количество выявленных ошибок при тестировании трактуется как случайная величина. Исходные данные m , X_i и T аналогичны предыдущим моделям.**

Функция среднего числа отказов, обнаруженных к моменту t, имеет вид:

$$m(t) = N(1 - e^{-bt}),$$

где b – интенсивность обнаружения отказов;

$q(t) = b$ - показатель роста надежности.

Функция интенсивности $\lambda(t)$ зависит от времени работы системы до отказа:

$$\lambda(t) = Nbe^{-bt},\ t \geq 0,$$

где N и b решаются из уравнения:

$$m/N - 1 + \exp(-bT) = 0$$

Модели роста надежности

Модели Ямадою–Охбою–Осаки (Yamada–Ohba–Osaki) задаются неоднородным процессом Пуассона S-подобной кривой экспоненциальной модели с помощью двух моделей.

Продолжение

Модель S-образного замедленного роста надежности **имеет S-подобную кривую роста надежности:**

$m(t) = N(1-(1+\beta)t)\exp(-\beta t))$, $N, \beta > 0$,

где N **– количество дефектов;** β **– коэффициент скорости выявления и устранения дефекта. Функция интенсивности отказов** $\lambda(t)$ **имеет вид:** $\lambda(t) = N\beta^2 t \exp(-\beta t)$.

Модель S-образного роста надежности с перегибами **анализирует отказы и дает оценку роста надежности системного ПО с данными:**

– взаимозависимые дефекты;

– вероятность выявления отказов пропорционально дефектам;

– частота выявления отказов возрастает;

– дефекты устраняются и новые не вносятся.

Уравнения *S-образного* кривой имеет вид

$\mu(t) = N(1 - \exp(-\beta t) / (1 + \psi \exp(-\beta t))$.

Прогнозирование надежности модуля системы проводится с помощью моделей Мусы и Мусы–Окумоты.

При появлении отказа в моменты времени $t_1, t_2, \ldots, t_n$ **, определяется вероятность** $P\{T>t_n\} = (1 - q\,t_n)^n$ **и среднее время ожидания** $T = t / qt_n$ **, как непрерывная распределенная экспоненциально величина.**

Инженерия надежности ПО

Software reliability engineering ориентирована на количественное изучение поведения компонентов системы:

1) сбор данных в процессах инспектирования, верификации, валидации тестирования и эксплуатации;

2) прогнозирование надежности по модели Мусы;

3) оценки надежности и качества элементов и системы в целом.

Сбор статистики проводится на этапах ЖЦ.

***Этап спецификации требований* - определение требований к системе, выбор метрик оценки надежности (интенсивности отказов, вероятности безотказного функционирования, восстановливасть и др.).**

***Этап проектирования* – спецификация модулей, тестирование, накопление данных об ошибках, дефектах и др.**

***Этап реализации* – комплексное тестирование систем, верификация и валидация систем и фиксация ошибок.**

***Этап испытаний* – оценка надежности по мат. *моделям* и принятие решения о степени готовности системы.**

***Этап сопровождения* – анализ частоты и серьезности отказов и оценка надежности и качества готовой системы.**

Поддержка инженерии качества систем

Стандарты:

- IEEE610.12 1990. Glossary of Software Engineering terminology.
- ISO/IEC 9126 (1-4) 2001 Information Technology – Software Engineering Guality: characteristics, External, Internal metrics, mesurement.
- В.В. Липаев. Процессы и стандарты ЖЦ сложных программных средств. Справочник. Синтег.-Москва, 2006.

Научные работы:

1. Липаев В.В. Надежность ПО, ПС.-1981, 1998.
2. Майерс Г. Надежность ПО.М.: Мир.-1980.-280с.
3. Липаев В.В. Качество ПО, ПС.-1983, 2001.
4. Холстед М.Х. Начала науки о программах.- М.: Мир.- 1981.
5. Липаев В.В. Надежность программных средств.-Синтег.-Москва.- 1998.
6. Тайер Т., ЛиповМ., Нельсон Э.Надежность ПО.- М.: Мир.- 1981.
7. ISO 9126-1-4.2001.- ИТ, ТО. Качество ПС. Ч.1. Модели качества. Ч.2. Внешние метрики. Ч.3. Внутренние метрики. Ч.4. Метрики качества в использовании.
8. Основы инженерии качества программных систем//Андон Ф.И., Коваль Г.И. и др.-К.: 2007.
9. Липаев В.В. Программная инженерия сложных ПП.-2014 (Инженерия качества и стандарты).

Инструментальные средства (ИС) оценки качества систем

ИС включают: ТМ «Тест инженер», ТМ «Надежность» и ТМ «Качество» [1,2].

ТМ «Тест инженер» отслеживает время и частоту выполнения компонентов в процессе тестирования, фиксирует отказы и дефекты, риски отказов модулей и оптимальное время тестирования.

ТМ «Надежность» обеспечивает:

- прогнозирование и распределение надежности по компонентам;

- верификацию, тестирование для сбора данных об ошибках и дефектах и хранение данных в специальной БД;

- оценку плотности дефектов и представления ее в виде графика надежности средствами HUGIN Lite 6.5 API;

- представление данных в модели распределения надежности модулей в виде матрицы с целью решения ее методом нелинейной оптимизации MATLAB 6.

---

1. Сборочное программирование. 1991.- (с.192-199).
2. Сборочное программирования. Основы индустрии ПП (с.278-296).

ПТМ Надежности

В состав ТМ надежности входит четыре ПТМ.

1. ПТМ «Распределения надежности» - реализует метод распределения надежности по компонентам методом парного сравнения модулей и построения квадратной матрицы А размером n x n. По матрице А вычисляются собственный вектор W = (w_1, w_2, … w_n) и собственные значения матрицы.

2. ПТМ «Прогнозирование надежности» реализует метод прогнозирования распределения значения надежности по каждому модулю по специальной формуле.

Коэффициент дефектов (к) и значение характеристик среды (ρ_i и φ_i) известны на момент начала прогнозирования надежности.

3. ПТМ «Прогнозирования плотности дефектов» реализует набор моделей надежности для оценки плотности дефектов и построения графической модели.

4. ПТМ «Оценка надежности ПС» согласно классификации дефектов (Orthogonal Defects classification) содержит:
тип дефекта, триггер дефекта, влияние дефекта.
На этих данных и данных по отказам отдельных модулей проводится оценка прогнозного значения надежности и системы в целом.

Надежность и качество аппаратуры и ПО ВПК (1960-1970, 1970-1990)

В работах (1-3) разработан метод обеспечения качества и надежности программных средств (техники и ПО) для бортовых вычислительных космических аппаратов в среде «Прометей» (1960-1980) и ПО реального времени для БЭМ-6, ЕС ЭВМ и СМ, а также отечественных ПЭВМ.

Тестирование модулей и систем включает:

- выделение маршрутов программ и наборов тестов, покрывающих структуру программ;

- проверка и контроль устойчивости, восстановление ПО после ошибок и отказов;

- анализ дефектов и устранения их в ПО.

Инструментальные средства тестирования обеспечивают:

- проверку спецификаций модулей и их паспортов (система РУЗА);

-имитацию среды движения самолетов и ракет, контроль программ аппаратуры, анализ безопасности полетов (система КИМС);

- технология создания комплексов программ (ПРОТВА – 5мил. команд) для автоматизации ПО специализированных ЭВМ), внедрение в 32 организациях ВПК (без внедрения в АН СССР и гражданских организациях), премия КМ СССР «Технология разработки бортовых систем» (1985).

Оценка качества систем

Модель качества Мкач = (*Q*, *A, M, W), где

Q **– Quility (качество),**

A- Attributes **(атрибуты),**

M- Metrics **(метрики),**

Weights **(весовые коэффициенты).**

Показатели качества (Quility) стандарта ISO 9126 (1-4):
q1**: функциональность (functionality),**
q2**: надежность (realibility),**
q3**: эффективность (efficiency),**
q5**: сопровождаемость (maitainnability),**
q6**: переносимость (portability).**

К подхарактеристикам надежности ПО относятся:

- **безотказность,**
- **устойчивость к ошибкам,**
- **восстанавливаемость, т.е. способность программы к перезапуску для повторного выполнения и восстановления данных после отказов.**

Модель качества

Свойства показателей качества программных систем

№	Наименование характеристики	Определение характеризуемых свойств ПС
1	Функциональность (functionality)	Свойства ПС, обусловливающие ее способность выполнять функции, соответствующие установленным и предполагаемым потребностям, при использовании в указанных условиях
2	Надежность (reliability)	Свойства ПС, обусловливающие ее способность сохранять уровень функционирования при работе в указанных условиях
3	Удобство применения (usability)	Свойства ПС, обусловливающие ее способность быть легко понимаемой, осваиваемой, удобной и привлекательной для пользователя при использовании в указанных условиях
4	Эффективность (efficiency)	Свойства ПС, обусловливающие ее способность обеспечивать рациональное использование выделенных ресурсов при работе в установленных условиях
5	Сопровождаемость (maintainability)	Свойства ПС, обусловливающие возможность ее эффективной модификации. Модификация может включать корректировку, усовершенствование или адаптацию ПС к изменениям среды, требований и функциональных спецификаций.
6	Переносность (portability)	Свойства ПС, обусловливающие ее способность быть переносимой из одной среды в другую

Модель завершенности компонентов ПС

Модель требований к завершенности компонентов ПС построена с учетом дифференцированного подхода к ним и необходимости достижения установленных целевых значений завершенности компонентов ПС, адекватных потребностям заказчика ПС.

Мера эксплуатационного качества ПС **определена как *функция полезности* вида:**

$$Q_{nc} = \sum_{i=1}^{k} a_i \cdot R_i$$

где a_i – мера важности *i*-й функции ПС для делового процесса, R_i – надежность (безотказность) выполнения функции в заданном периоде *t* эксплуатации системы.

Задача определения уровня *завершенности*

Задача определения оптимального целевого уровня *завершенности* компонента определяется на параметрах:

u_i – коэффициент относительного веса функции F_i эксплуатационного качества *Qпс*, *i = 1,…,*

kvi_j – коэффициент относительного веса *j*-го ПрП в обеспечении выполнения *i*–й функции, *u = 1,…, k; j = 1,…, l;*

wj_s – коэффициент относительного веса *s*–го ПрП при выполнения *j*–го приложения, *s* = 1,…, *m*; *j = 1*,…, *l*;

r_s – безотказность модуля *Ms* в период эксплуатации *t*;

E_j – множество номеров всех модулей, необходимое для выполнения *j*-го компонента ПС;

α_s – нижняя граница безотказности модуля *Ms*;

β_s – верхняя граница безотказности модуля *Ms*;

G – общая цена ПС;

C – себестоимость создания ПС организацией–разработчиком;

***cs* – накладные расходы, связанные с разработкой модуля *Ms* ;**

***ds* – расходы, необходимые для достижения единичного уровня безотказности модуля *Ms*;**

δ – доля прибыли в цене ПС.

Вычисление завершенности *Qпс*

Функция максимальной полезности *Qпс* вычисляется так:

$$Q_{nc}(r_1,...,r_m) = \sum_{j=1}^{l}(\sum_{i=1}^{k} u_i v_{ij} \cdot \prod_{n \in E_j} r_n) \to \max \quad (1)$$

при ограничениях **s = *1,…, m*** **(2)**

$$c_s + d_s \cdot r_s \le (1-\delta) \cdot G \cdot \sum_{j=1}^{l}\sum_{i=1}^{k} u_i v_{ij} w_{js} \quad (3)$$

$$\sum_{s=1}^{m} (c_s + d_s \cdot r_s) \le C \quad (4)$$

Данная задача нелинейной оптимизации с линейными ограничениями (2) – (4) практически решается с помощью пакета MATLAB.

Организация вычисления *Qпс*

Проводится с помощью параметров в формулах (1)- (4)

u_i, v_{ij}, w_{js} , где

$u = 1,\ldots, k$; $j = 1,\ldots, l$; $s = 1,\ldots, m$

находятся методом анализа иерархий (МАИ) путем парного сравнения и последовательного определения локальных приоритетов компонентов ПС в пределах каждого уровня иерархии по отношению к компонентам предыдущего (высшего) уровня. При этом

- **ограничения (2) задают допустимые нижние αs и верхние βs границы безотказности модулей, исходя из оценок важности каждого модуля;**
- **ограничения (3) устанавливают взаимосвязь общих расходов на разработку модуля для установления линейной зависимости между стоимостью модуля и уровнем его безотказности;**
- **ограничение (4) устанавливает взаимосвязь суммарных расходов на разработку всех модулей и себестоимости создания ПС.**

Таким образом, данная модель устанавливает завершенность каждого модуля (r_i) на основе требований, а затем и и для каждого приложения q_i с учетом независимости модулей в структуре системы. После чего определяются целевые уровни завершенности всех компонентов и модулей системы.

Спасибо за внимание