

Лабораторные работы №3, №4, №5

Введение

Необходимым условием создания программного продукта, показатели качества которого удовлетворяют требованиям технического задания, является назначение обоснованных норм надежности. Следует подчеркнуть, что часто разработчики технического задания формируют необоснованные требования к показателям надежности. Завышение требований к надежности приводит к необоснованным затратам на стадиях разработки программного продукта. Занижение требований к показателям надежности приводит к завышенным затратам на стадии эксплуатации изделия.

Цели и задачи лабораторной работы

Целью выполнения лабораторной работы является изучение моделей оценивания показателей надежности.

Теоретическая часть

1. Основные понятия и определения

Software Reliability Engineering (SRE) решает проблемы, источником которых являются потребности пользователей программных продуктов. *SRE* есть прикладное научное направление, занимающееся вопросами измерения, предсказания и управления (на тактическом уровне) надежностью систем, функциональность которых зависит от состояния входящих в их состав программных продуктов с тем, чтобы максимизировать удовлетворенность пользователей.

Вопросам *SRE* уделяется большое внимание в таких областях как телекоммуникационные системы; системы дистанционного образования; аэрокосмическая промышленность. Высокий интерес к *SRE* обусловлен тем, что выбор адекватного метода управления надежностью способствует повышению качества управления проектом.

К числу организаций, вкладывающих ресурсы в исследования по тематикам *SRE*, относятся: *Alkatel, AT&T, Hewlett-Packard, Hitachi, IBM Corp, Motorola, NASA, US Air Force, US Navy, US Army, Toshiba*. Хотя точные экономические данные получить не представляется возможным в силу их конфиденциального характера, можно утверждать, что использование методов *SRE* позволяет повысить эффективность программных проектов до шести раз.

Установлено, что затраты, связанные с обеспечением надежности, составляют лишь несколько процентов от стоимости проекта. Например, если в проекте участвуют от 40 до 100 исполнителей, то затраты на предпроектную стадию могут составить от одной до двух человеко-недель; определение состава работ (*WBS*) – от одного до трех человеко-месяцев. Затраты, связанные со сбором и анализом данных о дефектах проекта и усилий, направленных на их устранение, потребуют от половины до одного человеко-дня в неделю.

Деятельность по *SRE* включает в себя:

- измерение (оценку на основе фактических данных) и прогнозирование (оценку посредством моделей) значений показателей надежности;
- разработку метрик, характеризующих эффективность использования программных продуктов, а также процессов их реализации.
- выявление факторов, влияющих на функционирование программных продуктов;

- использование этих знаний при описании программных продуктов и руководстве их созданием; тестированием; официальной передаче пользователям; использовании и обслуживании.

Надежность определяется вероятностью того, что система, либо системная компонента будет обеспечивать требуемые функциональные и нефункциональные характеристики в течение заданного периода времени при заранее заданных характеристиках среды использования в предположении, что система нормально функционировала в начале этого периода. Пример: вероятность того, что система реального времени будет обеспечивать заранее определенную функциональность и производительность в течение десяти часов работы при условии, что она используется для тех целей, для которых предназначена.

В силу того, что надежность программного продукта зависит от способа его использования, информация об условиях фактического использования программного продукта является важной составляющей при оценивании надежности. Параметрами среды, в которой происходит фактическое использование продукта, являются:

- характеристики внешней, по отношению к программному продукту, среды (например, число обращений пользователей к программной системе);
- частота использования разных функций (свойств системы), предоставляемых программным продуктом.

Информация об условиях использования количественно характеризуется посредством «операционного профиля».

Время является внешней характеристикой производительности программного продукта, получаемой в результате его применения. Лучшей характеристикой производительности является «собственное время системы» - время выполнения операций центральным процессором (CPU). Вместе с тем, допускается изменение и других параметров для их последующего использования в моделях надежности – например, календарное время; время регистрации по часам; число перезапусков системы; время выполнения и др. В зависимости от того, «какое время» используется, определяются разные метрики: доступность для вычислений; чувствительность к ошибкам; среднее время наработки до отказа; среднее время восстановления после сбоя и т.д. Использование времени как косвенную объективную характеристику надежности. имеет следующие преимущества. Во-первых, имеет ясно физическое понимание этого показателя. Во-вторых, использование этого показателя дает комплексную характеристику надежности аппаратной и программной компонент. Подчеркнем, что время является лишь одной из возможных косвенных характеристик надежности. Допускается использование и других показателей, например. «число напечатанных страниц» (для принтера).

Отказ. Под отказом понимается неспособность системы предоставлять заявленные функции, либо не обеспечивать заявленные нефункциональные характеристики. Отказом является также несоответствие внешних результатов функционирования от требований к программному продукту, либо их несоответствие ожиданиям пользователей. Отказы могут быть обусловлены дефектами в аппаратной, либо в программной компонентах, а также ошибками пользователя.

Дефект (*fault, defect, bug*) – неполные, пропущенные, либо избыточные указания, либо последовательность взаимосвязанных инструкций, следование которым является причиной одной или более свершившихся, либо потенциальных отказов. Разделяют «врожденные» дефекты – те, которые изначально присутствуют в программном продукте. «Дефекты модификации» - те, которые внесены в результате выполнения корректировок. либо изменения проектных решений. В качестве метрической характеристики дефектов выступает плотность дефектов, например количество дефектов на тысячу строк исполняемого кода. Дефекты имеют субъективную природу – они являются следствием ошибок, допущенных людьми.

Если, например, допущена ошибка при изображении условного оператора на структурной схеме алгоритма, результатом будет дефект в коде. При выполнении программы будет иметь место ошибка обработки данных (вычисления пойдут не по той ветке), так что пользователь будет наблюдать на экране результат отличный от того, который должен быть. Иными словами, пользователь наблюдает *симптом дефекта* – отказ программного продукта.

Последствия ошибок. Для внешних правообладателей (пользователей, заказчиков программного продукта, спонсоров, кураторов разработки) последствия выражаются в их неудовлетворенности результатами функционирования программного продукта. Для внутренних правообладателей (команды проекта) последствия выражаются дополнительным объемом работ, который связан с поиском и исправлением дефектов.

Тяжесть отказов определяется их влиянием на работоспособность систем, в которых программный продукт используется в качестве инструмента поддержки управления. Тяжесть отказов выражается в снижении функциональности (потерях сервисов); экономических показателях (непосредственные потери и упущенная выгода); угрозе безопасности людей; репутационных потерях и др. Многие из этих характеристик не допускают непосредственного измерения (например, репутационные потери), поэтому для их характеристики используются экспертные оценки и балльные характеристики.

Операционный профиль. Представляет собою последовательность частот (или вероятностей) выполнения отдельных операций в процессе функционирования программного продукта. Системы, в которых программная компонента является критически важной, могут иметь более одного операционного профиля. Операционные профили служат основой назначения глубины тестирования (чем чаще используются какие-либо операции, тем больше им следует уделять внимания), а также формирования тестовых наборов.

Построению операционного профиля предшествует построение «профиля пользователя», «профиля заказчиков», «профиля режимов функционирования», «функционального профиля». Профили конструируются за счет создания детального иерархического списка заказчиков (внешних правообладателей обладающих необходимыми полномочиями при закупках и являющихся лояльными по отношению к программной системе); пользователей (лиц, непосредственно использующих систему); режимов использования программного продукта; функций и операций, которые должны реализовываться в каждом из режимов. Для каждой единицы перечисленных выше профилей необходимо определить вероятность появления и риска отказа. Это создаст возможности для количественного описания профиля.

2. Метрики и модели

2.1. Надежность.

Рассмотрим ситуацию, когда фиксация проблемы с последующим выделением дефектов и их устранением осуществляется в ходе создания программного продукта, системного тестирования и опытной эксплуатации, сопровождения в ходе промышленной эксплуатации. В этом случае мы рассчитываем на то, что исправления, вносимые в программный продукт, способствуют росту надежности.

Наиболее употребительной метрикой надежности является частота отказов. Частота отказов определяется как число отказов, зарегистрированных в единицу времени (секунду/час/день/неделю...). Синонимом «частоты отказов» является «относительное число отказов». Другой часто применяемой метрикой является среднее время между отказами. Между интенсивностью отказов и средним временем между отказами существует обратная зависимость. Интенсивность отказов является

хорошим показателем качества программного продукта с точки зрения пользователя. При росте надежности во времени, интенсивность отказов падает.

Если дефекты не устраняются, можно установить простые соотношения между интенсивностью отказов и надежностью:

$$R(t) = e^{-\lambda t}, \quad (1)$$

где $R(t)$ есть характеристика надежности;

t - определяемый. исходя из назначения системы, период наблюдения.

Пример: Допустим, система функционирует в образцовых и неизменных условиях и выявленные дефекты не устраняются. Допустим, что за 10 000 часов зарегистрировано 7 отказов. В этом случае оценка интенсивности отказов составит $\hat{\lambda} = 7 * 10^{-4}$ в час. Средняя наработка между отказами составит $\hat{T} = 1/\hat{\lambda} = 1428$ часов. Оценка вероятности безотказной работы в течение ближайших 10 часов составит

$$\hat{R} = e^{-7 * 10^{-4} * 10} = 0.993$$

В случае, когда имеет место рост надежности, интенсивность отказов становится убывающей функцией (в предположении, что программный продукт тестируется и используется в представительных (образцовых, т.е. определенных в техническом задании на разработку) условиях).

Рассмотрим две из множества известных моделей, соответствующих случаю растущей надежности. В обеих моделях предполагается, что устранение отказов происходит мгновенно без внесения новых дефектов. В модели «основное время выполнения» («*Basic Execution Time - BET*») интенсивность отказов описывается выражением

$$\lambda(\tau) = \lambda_0 e^{-\frac{\lambda_0}{v_0} \tau} \quad (2)$$

где λ_0 - начальная интенсивность;

v_0 - общее ожидаемое число дефектов.

Эта модель может быть преобразована к линейному виду

$$\lambda(\tau) = \lambda(\mu) = \lambda_0 \left(1 - \frac{\mu}{v_0}\right) \quad (3)$$

где $\mu(\tau)$ есть среднее число отказов, зарегистрированных в течение времени τ , т.е. среднее значение функции

$$\mu(\tau) = v_0 \left(1 - e^{-\frac{\lambda_0}{v_0} \tau}\right) \quad (4)$$

В модели *LPET (Logarithmic-Poisson Execution Time)* интенсивность отказов при периоде наблюдения τ определяется соотношением

$$\lambda(\tau) = \frac{\lambda_0}{\lambda_0 Q \tau + 1}, \quad (5)$$

где λ_0 - начальная интенсивность отказов;

Q - параметр затухания (насколько уменьшается интенсивность отказов за единицу времени).

Выражение (5) можно преобразовать к виду

$$\lambda(\tau) = \lambda(\mu) = \lambda e^{-Q\mu}, \quad (6)$$

где $\mu(\tau)$ есть среднее число отказов, зарегистрированных в течение периода наблюдения τ , т.е.

$$\mu(\tau) = \frac{1}{Q} \ln(\lambda_0 Q \tau + 1), \quad (7)$$

Модель *BET* относится к классу «моделей с конечным числом отказов». Для этой модели среднее значение функции приближается к некоторому асимптотическому значению по мере роста значения периода наблюдения τ . Модель *LPET* относится к классу «моделей с неограниченным числом отказов». В действительности модели, относимые к упомянутым классам, предназначены для поддержки процессов устранения дефектов, которых на самом деле может быть очень ограниченное число.

Оценивание параметров упомянутых моделей может выполняться *методом наименьших квадратов*.

Важно отметить, что есть два различных пути использования моделей:

1. Модель как аккумулятор исторических данных.
2. Модель как инструмент прогнозирования состояния, наступления события. Например: «...когда интенсивность отказов достигнет целевой величины...», «...когда я могу закончить тестирование...»

Прогнозирование – одно из наиболее интересных приложений модели, но в то же время наиболее опасное. В литературе отмечается, что ни одна из моделей не обеспечивает точного решения задачи прогнозирования. Более того, результаты, полученные посредством разных моделей, могут достаточно сильно различаться.

После того, как определена структура модели и рассчитаны значения её параметров, появляется возможность рассчитать общее число дефектов в коде (после выполнения процедур поиска и устранения дефектов); предполагаемую интенсивность отказов; какое время потребуется для того, чтобы довести фактическую интенсивность отказов до целевого значения.

Пример: Предположим, что в качестве базовой модели выбрана модель *BET*. Из формул (2)-(4) следует, что число ошибок, которые следует устранить для того, чтобы достичь требуемой интенсивности отказов, определяется соотношением:

$$\Delta\mu = \frac{V_0}{\lambda_0} (\lambda_P - \lambda_F), \quad (8)$$

где λ_P - фактическая интенсивность отказов. Дополнительное время, необходимое для устранения ошибок, составит

$$\Delta\tau = \frac{V_0}{\lambda_0} \ln \frac{\lambda_P}{\lambda_F} \quad (9)$$

В случае, если $\nu_0 = 120$ дефектов в коде; $\lambda_0 = 15$ отказов в час (например, CPU); при текущем значении $\lambda_p = 2.55$ отказа в час и при целевом значении $\lambda_F = 0.0005$ отказов в час, значение

$$\Delta\mu = \frac{120}{15}(2.55 - 0.0005) \approx 21 \text{ отказ.}$$

$$\Delta\tau = \frac{120}{5} \ln \frac{2.55}{0.0005} \approx 68.3 \text{ часов (например, CPU).}$$

Если известна стоимость одного часа (например, CPU), то можно оценить не только стоимость разработки тестов, необходимых для обеспечения требуемого качества, но и затраты, связанные с прогоном тестов (т.е. речь идет о построении экономической модели качества программного продукта).

Приведенные оценки относятся к классу «точечных» и дают значение «вероятных», т.е. средних величин. На практике, однако, важно вычислять также доверительные границы для оценивания параметров и получить меру (т.е. количественную характеристику) того, насколько можно доверять полученным цифрам. Вместо того, чтобы представлять ожидаемые значения в виде одной величины, представляется подходящий интервал (например, 70%, 90%, 95% доверительный интервал). В этом случае, вместо того, чтобы говорить о том, что ожидается 21 отказ, следует говорить о том, что с вероятностью 90% число отказов будет лежать в диапазоне от 17 до 25, т.е. [17,25].

Точность получаемых оценок в значительной степени определяется способом регистрации данных. Например, набор исходных данных может быть представлен значениями моментов времени, когда возникает отказ. Альтернативным вариантом является регистрация величины интервалов между отказами, либо число отказов, наблюдавшихся за период тестирования – группированные данные. Либо в виде описания каждого корректируемого отказа; либо в виде информации о части кода, в которой зафиксирован отказ; либо описанием модулей, на которые оказывают влияние вносимые изменения и т.д.

2.2. Готовность

Одной из часто используемых характеристик надежности является готовность. Например, *Bell core* целевым показателем простоя элементов телекоммуникационной сети определяет 3 минуты в год. *Готовность* есть вероятность того, что система (элемент системы) окажется работоспособным в заранее указанное время. Неготовность, напротив, означает вероятность того, что система (элемент) окажется неработоспособным в заранее заданный момент времени. Понятие готовности (неготовности) тесно связано с понятием *устранения* отказов. Количественно устранение отказов можно охарактеризовать *коэффициентом восстановления*, который определяется числом отказов, устраняемых в единицу времени. Например, отказ в программной компоненте, в среднем приводит к 10-минутному простоя аппаратно-программного комплекса. Коэффициент восстановления составит 1/10 на отказ в минуту (по сути коэффициент восстановления представляет собою интенсивность потока восстановления – см. *Уравнения Колмогорова*).

Готовность системы может быть охарактеризована разными показателями. Например, *мгновенная готовность* выражается вероятностью застать систему исправной в любой момент времени t в течение времени жизни системы. *Средняя готовность* есть доля заранее определенного временного интервала $[0, T]$, в течение которого система должна быть готовой к использованию.

Оценкой *средней готовности* системы является величина

$$\hat{A}_c(T) = \frac{\text{суммарное время в в тчении которого система исправна}}{T} \quad (10)$$

Соответственно *коэффициент простоя* (интенсивность отказов)

$$\hat{\lambda}_c(T) = \frac{\text{Общее число отказов(ссобытий)за период T}}{\text{Суммарное время работоспособного состояния системы за период T}} \quad (11)$$

Коэффициент восстановления (интенсивность восстановления) определен как

$$\hat{\zeta}_c(T) = \frac{\text{Общее число отказов за период T}}{\text{Общее время, в тчении которого система ремонтировалась за период T}}$$

Если период T достаточно длинный, средняя готовность преобразуется к виду «установившегося значения средней готовности» A_{SS} . Этому понятию соответствует соотношение (*коэффициент простоя*)

$$\hat{A}_{SS} = \frac{\hat{\zeta}}{\hat{\zeta} + \hat{\lambda}} \quad (13)$$

Из этого соотношения следует, что коэффициент простоя (коэффициент неисправности) связан с коэффициентом готовности системы, а также коэффициентом восстановления.

Следует обратить внимание на то, что с момента поставки очередной версии программного продукта наблюдается разброс (иногда значительный) в значениях коэффициента простоя. Это так называемая «переходная зона». В дальнейшем значения коэффициента простоя снижается и стабилизируется. На практике значение коэффициента простоя обычно зависит как от профиля использования, так и от способа разрешения проблем. Достаточно нормальной практикой является то, что $\zeta_c(T) \gg \lambda_c(T)$.

На практике модели надежности и готовности используются для прогнозирования состояния системы.

3. Практическое использование формального аппарата надежности

Практически невозможно обеспечить надежность программного продукта не имея логичного и реалистичного плана верификации процессов и активностей, относящихся ко всем стадиям жизненного цикла. Пример такого плана представлен в IEEE Std – 1012; ESA-PSS – 05-10. SRE предписывает использование современных технологий ориентированных на предупреждение, идентификацию дефектов, обеспечение устойчивости к ним; уменьшение обусловленных ими негативных последствий. SRE предписывает увязывать эти технологии с бизнес-процессами. Это включает в себя конструирование и квантификацию профилей пользователей, а также обеспечение баланса между надежностью программного продукта и другими ограничениями. Практика SRE предписывает сбор и анализ данных о свойствах программного продукта; оценивание и определение направления изменения

надежности; управление процессами создания программного продукта в терминах ресурсов, а также критерия «когда остановить тестирование»; отслеживать состояние деятельности, связанной с обеспечением надежности программного продукта. Отслеживание уровня надежности и направления её изменения используется для совершенствования системы управления и поддержки процесса создания прикладного программного продукта и повышения уровня удовлетворенности потребителей. IEEE&ESA-PSS- стандарты V&V (Verification & Validation) предполагают, что на стадии формирования и анализа требований будут решены следующие задачи:

- анализ взаимосвязи функциональных и нефункциональных требований, соответствующих различным иерархическим уровням представления системы (см. «Дом качества»);
- классификация требований по значимости (см. «Ранжирование проектов»);
- анализ взаимовлияния требований (см. «выявление противоречий в требованиях»);
- планирование генерации системных тестов;
- планирование генерации тестов приемлемости.

SRE аргументирует необходимость этих действий тем, что разработчики, как и пользователи (заказчики) нуждаются в том, чтобы:

1. выявить и классифицировать все дефекты в программном продукте;
2. определить реальные требования к надежности со стороны покупателей и определить с экономической точки зрения компромиссное решение в пространстве трех показателей: сроки – качество – цена;
3. определить профиль пользователей программного продукта;
4. определить целевые показатели надежности программного продукта.

Идентификация и классификация отказов программных продуктов, а также тяжести их последствий, зависит от области приложений, а также особенностей сопровождения. Например, *U.S. Federal Communications Commissions (FCC)* требует, чтобы в случае нарушения сервисного обслуживания более чем на 30 минут, либо в случае, когда нарушение сервисного обслуживания затрагивает интересы более чем 50 000 абонентов, информация об этом в течение не более чем 30-ти минут после наступления события, должна поступать в *Federal Communications Commission*. В этом контексте, отказ коммуникационной системы может быть классифицирован следующим образом:

1. докладываемый в *FCC*;
2. простой системы менее 30 минут при отказе в обслуживании менее чем 50 000 абонентам;
3. потеря одного или более обязательных сервисов;
4. потерю функциональности можно компенсировать вспомогательными средствами, либо схожими опциями;
5. незначительные неудобства, либо незначительные потери эффективности.

Для того, чтобы понять первопричину отказов, необходимо собрать данные об отказах. Отказы, которые приводят к полному простоя системы, в свою очередь могут быть разбиты (в зависимости от фактически выявленных, либо предполагаемых причин) на:

- отказы аппаратной компоненты;
- отказы программной компоненты;

- ошибки персонала;
- нерасчетные воздействия окружающей среды;
- неизвестные.

Внутри каждого подкласса можно выделить дополнительные категории. Например, если с точки зрения особенностей использования системы важна её готовность (время восстановления), может оказаться полезной классификация отказов по продолжительности устранения. Например, менее 2-х минут; от пяти до десяти минут; от десяти до 30 минут; более чем 30 минут (см. лаб. работу по классификации на основе выборочных данных).

- Значения целевых показателей надежности устанавливаются исходя из потребностей пользователей;
- ограничений используемых технологий разработки прикладного программного обеспечения;
- возможностей разработчиков;
- возможностей получения (сбора) данных; организационной модели бизнес-процесса; стоимости разработки (см. лаб. раб. по специальным статистическим методам обработки малых выборок);
- расписания выполнения работ.

Обычно, для того, чтобы достичь определенной категории по показателям надежности, выделяется совокупность частных целей. Например, Wellcome установило общее требование для оценки производительности телекоммуникационных систем. Целевое значение для сетевых переключательных элементов – простой не более 3-х минут в год (при этом время восстановления после отдельного отказа не превышает 30 секунд); фиксируется состояние полного отказа, если сервис недоступен более чем 30 секунд. Это, конечно, не требование надежности, но это требование готовности. Фактически средняя недоступность системы (из-за отказа переключательных элементов) составляет $\frac{3}{(60 * 24 * 365)} = 0.00000571$ (мин.) Для вычисления целевого показателя надежности, также необходимо установить целевое значение показателя, связанного с восстановлением системы. Например, приняв упрощающее предположение о том, что система находится в установившемся состоянии, можно воспользоваться соотношением (13) для расчета целевого значения характеристики восстановления. Пусть среднее значение частоты восстановления составляет $\zeta = 0.3$ отказа в минуту. В этом случае, зная требуемое значение готовности $1 - 0.00000571 = 0.99999429$ и зная частоту восстановления, посредством (13) можно определить, что средняя частота отказов λ не должна превышать 0.00000171 отказов в минуту.

В стандарте IEEE Std-1012 также предполагается, что следующие задачи решаются в ходе проектирования, кодирования, помодульного и интеграционного тестирования:

- анализ трассируемости проектных решений и кодов;
- количественная оценка качества проектных решений, кодов и документации;
- разработка планов тестирования для программных компонент, а также плана тестирования интеграции;
- проектирование, генерация и выполнение тестовых наборов.

SRE аргументирует необходимость этих действий требованиями того, чтобы разработчики прикладных программных продуктов:

- определились с функциональностью и сформировали операционный профиль;

- получили количественные оценки надежности при повторном (многократном, в составе разных программных продуктов) использовании компонентов;
- четко распределили требования к характеристикам надежности среди программных компонентов и создали продукт, характеристики надежности которого соответствуют целевым значениям показателей;
- использовали модели распределения ресурсов модели расписаний при выделении объемов работ, в соответствии с функциональными профилями;
- отслеживали и управляли процессами искусственного внесения дефектов и их устранения (для оценки эффективности систем диагностики).

Важным видом деятельности на стадии проектирования является определение целевых показателей надежности для подсистем и компонентов так, чтобы обеспечивалось целевое значение показателя надежности для системы в целом. Назначение целевых показателей надежности для подсистем и компонент носит итерационный характер, и должно основываться на сравнительном анализе альтернативных вариантов реализации программного продукта. Это обусловлено необходимостью обеспечения «баланса» между целевыми значениями показателей надежности системы; стоимостью и временем реализации системы.

На всех стадиях создания прикладного программного продукта настоятельно рекомендуется применение инспектирования и критического анализа промежуточных результатов (англ.- *inspections and review*). Их использование позволяет оценить среднее количество дефектов на конструктивную единицу (оператор/модуль/функцию...). Иными словами, инспектирование и критический анализ (сквозная проверка и контроль) позволяют выявить дефекты на тех стадиях, когда ещё не создана работающая версия программы (*для понимания значимости раннего обнаружения дефектов вспомните кривую стоимости устранения ошибки*).

Практически полезной метрикой является количество дефектов, приходящихся на один человеко-час работы (с учетом классов дефектов). При этом следует учитывать время, затрачиваемое на подготовку и проведение инспектирования.

В литературе указывается, что если значение этой метрики лежит в диапазоне от 3 до 7, то как процессы инспектирования, так и процессы разработки программного продукта являются управляемыми. В противном случае, необходимо проанализировать организацию программного проекта и, возможно, внести соответствующие изменения.

Задание на работы

Лабораторная работа №3 (модель РЕТ)

Дано: Модули программной системы соединены последовательно, что определяет интенсивность отказов всего изделия как $\lambda_{сист} = \sum_{i=1}^N \lambda_i^{(доп)}$. Допустимая интенсивность отказов системы составляет $\lambda_{сист} = 10^{-4} / час$.

Требуется:

- Назначить значение целевой интенсивности для i – го модуля $\lambda_p^{(i)}$.
- Оценить суммарное время, необходимое для обеспечения приемлемых значений интенсивностей отказов.

Исходные данные

№ варианта	Число модулей	Начальное число дефектов в i -м модуле $v_0^{(i)}$	Начальная интенсивность отказов $\lambda_0^{(i)}$	Фактическая интенсивность отказов $\lambda_F^{(i)}$
1	5	120*i	15*i	2.55*i
2	10	75*i	17*i	3*i
3	7	120*i	15*i	2.55*i
4	7	75*i	17*i	3.55*i
5	5	100*i	14*i	2.15*i
6	15	100*i	17*i	5.5*i
7	10	80*i	10*i	2.5*i
8	16	90*i	5*i	4.55*i
9	5	120*i	17*i	1.55*i
10	15	92*i	15*i	3.55*i
11	8	220*i	25*i	4.55*i
12	9	82*i	18*i	3.55*i
13	15	120*i	19*i	7.55*i
14	6	320*i	35*i	8.55*i
15	5	520*i	75*i	6.55*i
16	8	220*i	35*i	2.55*i
17	8	320*i	45*i	6.55*i
18	5	120*i	25*i	0.55*i
19	15	220*i	25*i	3.55*i
20	8	60*i	5*i	0.5*i
21	10	520*i	15*i	8.55*i
22	9	120*i	17*i	4.55*i
23	10	220*i	25*i	3.55*i
24	15	150*i	16*i	0.55*i
25	16	250*i	20*i	0.7*i

Лабораторная работа №4 (модель LPET)**Дано:** а) Целевое значение интенсивности отказов $\mu(\tau) = 0$ б) Начальное число интенсивности отказов $\lambda_0 = k * 150$ (k - номер варианта)

в) Исходные данные о периодах наблюдения и числе зарегистрированных отказов (см. таблицу).

Требуется:

а) МНК определить параметр затухания

б) Определить время достижения целевого показателя интенсивности отказов

Номер варианта	Периоды наблюдения	Число зарегистрированных отказов		
1	1	100		
	2	30		
	3	20		
	4	25		
	5	5		
	6	10		
2	1	150		
	2	35		
	3	23		
	4	15		
	5	10		
	6	12		
3	1	250		
	2	45		
	3	13		
	4	15		
	5	10		
	6	5		
4	1	250		
	2	45		
	3	13		
	4	65		
	5	10		
	6	5		
5	1	250		
	2	45		
	3	13		
	4	15		
	5	25		
	6	5		
6	1	250		
	2	45		
	3	20		
	4	15		
	5	10		

	6	7		
7	1	250		
	2	45		
	3	13		
	4	67		
	5	10		
	6	5		
8	1	250		
	2	98		
	3	13		
	4	15		
	5	10		
	6	5		
9	1	400		
	2	45		
	3	30		
	4	15		
	5	10		
	6	5		
10	1	350		
	2	45		
	3	13		
	4	15		
	5	10		
	6	5		
11	1	450		
	2	45		
	3	13		
	4	45		
	5	10		
	6	5		
12	1	550		
	2	45		
	3	13		
	4	78		
	5	10		
	6	5		
13	1	250		
	2	120		
	3	13		
	4	15		
	5	10		
	6	5		
14	1	250		
	2	98		
	3	13		
	4	75		
	5	10		
	6	5		

15	1	250		
	2	45		
	3	13		
	4	38		
	5	10		
	6	5		
16	1	250		
	2	45		
	3	29		
	4	15		
	5	10		
	6	5		
17	1	250		
	2	45		
	3	45		
	4	15		
	5	10		
	6	5		
18	1	250		
	2	45		
	3	67		
	4	15		
	5	10		
	6	5		
19	1	250		
	2	45		
	3	13		
	4	15		
	5	10		
	6	17		
20	1	250		
	2	45		
	3	65		
	4	15		
	5	10		
	6	5		
21	1	250		
	2	45		
	3	13		
	4	15		
	5	17		
	6	5		
22	1	250		
	2	45		
	3	74		
	4	15		
	5	10		
	6	5		
23	1	250		

	2	45		
	3	7		
	4	15		
	5	10		
	6	5		
24	1	250		
	2	45		
	3	77		
	4	15		
	5	10		
	6	5		
25	1	250		
	2	45		
	3	3		
	4	15		
	5	10		
	6	5		

Лабораторная работа №5 (характеристики готовности)

Дано: Выборочные данные о времени наработки между проявлениями дефектов и времени устранения дефектов (см. таблицу)

Требуется: Определить:

- а) показатель средней готовности системы
- б) коэффициент простоя
- в) коэффициент восстановления
- г) коэффициент простоя

Номер варианта	№ события	Время наработки между проявлениями дефектов	Время локализации и устранения дефектов
1	1	1	10
	2	2	11
	3	3	9
	4	4	7
	5	5	3
2	1	11	17
	2	12	11
	3	13	12
	4	14	7
	5	15	3
3	1	11	7
	2	2	1
	3	13	2
	4	4	7
	5	15	3
4	1	15	1
	2	12	1
	3	17	2
	4	14	7
	5	15	3

5	1	11	10
	2	16	11
	3	17	12
	4	4	7
	5	15	30
6	1	110	17
	2	12	115
	3	19	12
	4	14	75
	5	15	3
7	1	11	9
	2	120	11
	3	18	16
	4	14	7
	5	19	15
8	1	11	170
	2	120	110
	3	110	12
	4	140	80
	5	15	35
9	1	11	170
	2	120	110
	3	114	120
	4	148	90
	5	153	40
10	1	21	27
	2	22	11
	3	33	32
	4	44	47
	5	55	53
11	1	31	37
	2	42	41
	3	53	52
	4	64	17
	5	15	115
12	1	41	77
	2	52	81
	3	163	92
	4	74	21
	5	155	134
13	1	51	27
	2	72	31
	3	93	52
	4	24	77
	5	85	93
14	1	71	37
	2	92	51
	3	83	72
	4	114	79

	5	154	165
15	1	21	97
	2	42	110
	3	73	12
	4	114	99
	5	65	108
16	1	51	170
	2	18	110
	3	93	22
	4	34	9
	5	45	37
17	1	91	170
	2	120	115
	3	141	126
	4	169	79
	5	158	35
18	1	117	27
	2	128	41
	3	139	62
	4	145	87
	5	159	93
19	1	11	17
	2	12	11
	3	13	12
	4	14	7
	5	15	3
20	1	18	189
	2	129	134
	3	168	116
	4	199	67
	5	189	93
21	1	114	170
	2	126	1186
	3	136	128
	4	147	273
	5	159	283
22	1	161	147
	2	172	511
	3	183	162
	4	124	779
	5	175	382
23	1	101	117
	2	182	211
	3	164	212
	4	144	37
	5	135	43
24	1	111	117
	2	212	211
	3	313	132

	4	144	74
	5	165	355
25	1	191	187
	2	182	191
	3	143	182
	4	154	755
	5	125	344

Методика выполнения задания

На основе расчетных соотношений определить значения нефункциональных требований в соответствие с темой задания

Требования к содержанию и оформлению отчета

Отчет должен содержать краткие теоретические сведения, обосновывающие необходимость расчета мер нефункциональных требований, исходные данные для расчетов, значения мер нефункциональных требований, ответы на контрольные вопросы.

Контрольные вопросы

1. Содержание понятия «нефункциональные требования», отличия функциональных и нефункциональных требований.
2. Как обосновывается выбор номенклатуры нефункциональных требований
3. Как назначаются значения нефункциональных требований на разных стадиях жизненного цикла изделия
4. Как доказывается соответствие фактических значений нефункциональных требований тем, которые определены в спецификациях.

Дополнительные вопросы

1. Процессы жизненного цикла программных средств согласно ISO 12207:1995
2. Содержание работ стадий программного проекта согласно РД 50-34.698-90
3. Характеристики качества программного продукта согласно ISO 9126:1991
4. Стадии создания автоматизированных систем согласно ГОСТ 34.601-90
5. Работы стадий создания автоматизированных систем согласно ГОСТ 34.602-89

Критерии оценки выполнения лабораторной работы

Задание считается выполненным, если выполнены работы, определенные в разделе «Методика выполнения задания», оформлен отчет, осуществлена его защита.

Список литературы

1. Милошевич Д. Набор инструментов для управления проектами – М.: Компания АйТи: ДМК Пресс, 2008. – 729с.
2. ESA PSS-05-02
3. ESA PSS-05-03
4. ESA PSS-05-10
5. Гвоздев В.Е., Бежаева О.Я., Блинова Д.В. и др. Практическое руководство по реализации программных проектов-Уфа, УГАТУ,2015