**ЗБ-ПИ20-2 Попов Дмитрий Сергеевич Экзамен**

**Билет №19**

**1 вопрос** Модели надёжности программных средств. Прогнозные модели.

**2 вопрос** Оценить надёжность по модели Шумана.

Дано: Общее число операторов: 10000. Оценка осуществляется после 10 прогонов.

Данные:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тчас | 0.5 | 0.4 | 0.5 | 0.75 | 0.2 | 0.5 | 0.3 | 0.3 | 0.1 | 0.4 |
| Кол-во ошибок | 2 | 0 | 5 | 3 | 4 | 1 | 3 | 2 | 0 | 1 |

**3 вопрос** Сравните второй и третий уровни зрелости организации согласно стандартам CMM/CMMI. Приведите задачи и процессы, требующие автоматизации на третьем уровне зрелости организации.

**Ответ на 1 вопрос:**

**Модели надёжности программных средств**

Модели надёжности программных средств можно разделить на две основные категории: аналитические и эмпирические.

**Аналитические модели** позволяют рассчитать количественные показатели надёжности на основе данных о поведении программы в процессе тестирования. Они включают в себя:

* **Динамические модели**, где поведение программы рассматривается во времени, позволяя анализировать частоту отказов и интервалы между ними.
* **Статические модели**, которые не связывают надёжность с временем, а учитывают зависимость количества ошибок от числа тестовых прогонов или характеристик входных данных.

**Эмпирические модели** базируются на анализе структурных особенностей программ. Они включают:

* **Феноменологические (эмпирические) модели**, предполагающие, что связь между надёжностью и другими параметрами программы является статической. Эти модели оценивают характеристики программного обеспечения, свидетельствующие о его надёжности.

Применение моделей надёжности позволяет оценить уровень надёжности программного обеспечения на разных этапах его жизненного цикла, что важно для управления качеством программных систем.

**Прогнозные модели** в управлении качеством программных систем используются для оценки и предсказания надежности программного обеспечения на основе анализа его характеристик и поведения. Эти модели помогают определить потенциальные проблемы и риски, связанные с надежностью системы, еще до начала ее эксплуатации.

Основные виды прогнозных моделей включают:

1. **Модель Мотли-Брукса**: Основана на длине и сложности структуры программы, а также на количестве ошибок на страницу операторов. Позволяет предсказать количество ошибок в программе.
2. **Модель Холстеда**: Учитывает объем программы, количество операций и операндов, что помогает предсказать количество ошибок в программе.

Эти модели помогают разработчикам и тестировщикам сосредоточиться на наиболее критических аспектах программы, повышая тем самым ее надежность и качество.

**Ответ на 2 вопрос:**

Модель надежности Шумана используется для оценки надежности программного обеспечения на основе данных о количестве обнаруженных ошибок и времени их появления.

Основная формула для расчета надежности по модели Шумана:

***R(t) = exp(−λt)***

где:

* ***R(t)*** — вероятность того, что система будет функционировать безотказно в течение времени t.
* ***λ*** — интенсивность отказов, рассчитываемая как отношение общего числа ошибок к общему времени.

**Решение:**

**Шаг 1: Суммируем общее время тестирования**

Так как у нас 10000 операторов и 10 прогонов, общее время работы всех операторов за каждый прогон будет:

Ttotal=10000×(0.5+0.4+0.5+0.75+0.2+0.5+0.3+0.3+0.1+0.4)×10=10000×3.95×10=395000 часов

**Шаг 2: Суммируем общее количество ошибок**

Etotal=2+0+5+3+4+1+3+2+0+1=21 ошибка

**Шаг 3: Рассчитываем интенсивность отказов λ**

λ=Etotal/Ttotal=21/395000≈0.0000532 ошибки/час

**Шаг 4: Оценка надежности**

Для оценки надежности R(t) через t часов, используем формулу:

R(t)=exp(−λt)

Для оценки надежности после 1 часа:

R(1)=exp(−0.0000532×1)=exp(−0.0000532)≈ 0.999947681

**Таким образом, вероятность того, что система будет функционировать безотказно в течение 1 часа, составляет примерно 0.999947681 или 99.9947681%. Это указывает на высокую надежность системы в течение одного часа при текущей интенсивности отказов.**

**Решение на Python:**

import math

# Данные

total\_operators = 10000

runs = 10

T\_hours = [0.5, 0.4, 0.5, 0.75, 0.2, 0.5, 0.3, 0.3, 0.1, 0.4]

errors = [2, 0, 5, 3, 4, 1, 3, 2, 0, 1]

# Шаг 1: Суммируем общее время тестирования для всех операторов и всех прогонов

T\_total = total\_operators \* sum(T\_hours) \* runs

# Шаг 2: Суммируем общее количество ошибок

E\_total = sum(errors)

# Шаг 3: Рассчитываем интенсивность отказов λ

lambda\_failure\_rate = E\_total / T\_total

# Шаг 4: Оценка надежности R(t) для заданного времени t

def reliability(t, lambda\_failure\_rate):

    return math.exp(-lambda\_failure\_rate \* t)

# Пример оценки надежности через 1 час

t = 1

reliability\_at\_t = reliability(t, lambda\_failure\_rate)

print(f"Общее время тестирования всех операторов: {T\_total} часов")

print(f"Общее количество ошибок: {E\_total}")

print(f"Интенсивность отказов (λ): {lambda\_failure\_rate:.10f} ошибки/час")

print(f"Надежность системы через {t} час: {reliability\_at\_t:.10f}")

Общее время тестирования всех операторов: 395000.0 часов

Общее количество ошибок: 21

Интенсивность отказов (λ): 0.000053165 ошибки/час

Надежность системы через 1 час: 0.999947681

**Ответ на 3 вопрос:**

Стандарты CMM (Capability Maturity Model) и CMMI

(Capability Maturity Model Integration) представляют собой модели зрелости процессов в организациях, разработанные для улучшения качества и эффективности работы. CMM был разработан в 1980-х годах, а его развитие привело к созданию CMMI, который объединяет различные аспекты управления процессами и качеством. CMMI определяет набор практик, которые организация может внедрить для достижения определенного уровня зрелости в различных областях деятельности, таких как разработка программного обеспечения, управление проектами, безопасность и другие. Эти стандарты помогают организациям систематизировать и оптимизировать свои процессы, что ведет к повышению качества продукции и услуг, снижению рисков и улучшению общей производительности.

Второй уровень зрелости организации по CMM/CMMI характеризуется как “управляемый” (managed), что означает наличие определенных процессов, которые документированы и применяются в рамках конкретных проектов. На этом этапе процессы сфокусированы на организации конкретного проекта, но не стандартизированы и могут различаться в разных проектах.

Третий уровень зрелости, “определенный” (defined), предполагает, что во всех проектах процессы следуют заданному корпоративному стандарту, так называемому стандартному процессу организации. Это означает, что процессы четко определены, документированы и применяются в масштабах всей организации.

Задачи и процессы, требующие автоматизации на третьем уровне зрелости организации:

1. **Управление требованиями**: Автоматизация процесса сбора, анализа и управления требованиями к продукту или услуге. Это включает в себя использование специализированного программного обеспечения для управления требованиями, которое позволяет отслеживать изменения требований, обеспечивать их соответствие и контролировать выполнение.
2. **Планирование проекта**: Автоматизация процесса планирования проектов, включая определение сроков, бюджета, ресурсов и задач. Это может включать использование инструментов проектного управления, таких как Microsoft Project или Jira, для создания и отслеживания планов проектов.
3. **Мониторинг и контроль проекта**: Автоматизация процесса мониторинга хода выполнения проектов, включая отслеживание сроков, бюджета и качества. Это может включать использование систем отслеживания ошибок, таких как Bugzilla или Jira, для регистрации и отслеживания дефектов в продукте.
4. **Управление качеством**: Автоматизация процесса контроля качества продукции или услуг, включая тестирование, анализ и улучшение качества. Это может включать использование автоматизированных тестов для проверки функциональности продукта или услуги.
5. **Выбор технических решений**: Автоматизация процесса выбора оптимальных технических решений для реализации проектов, включая выбор технологий, инструментов и платформ. Это может включать использование инструментов сравнения и выбора технологий, таких как Gartner Magic Quadrant или Forrester Wave.
6. **Обучение в организации**: Автоматизация процесса обучения и повышения квалификации сотрудников, включая разработку и проведение обучающих курсов и тренингов. Это может включать использование систем дистанционного обучения, таких как Moodle или Coursera, для предоставления доступа к обучающим материалам и проведения онлайн-курсов.

Автоматизация этих процессов на третьем уровне зрелости организации позволяет повысить эффективность работы, улучшить качество продукции или услуг и снизить риски возникновения ошибок.