



Лабораторная работа №3

Надежность и качество программных средств

Выполнила: студентка ЭФ гр.

БИН-22(оз) Сидикова Д.А.

Проверила: Доцент кафедры ИСЭ

Петрова Л.В.

Список сокращений:

ПС- программное средство;

ПИ-программные изделия;

ИС-информационные системы.

Цель работы:

Изучить надежность и качество ПО

Основные *понятия и показатели*

- ***Качество*** программного обеспечения
- ***Требования*** для обеспечения качества и надежности ПС
- ***Процесс создания*** сложных ПС.
- ***Методы обеспечения*** надежности ПС.
- ***Факторы обеспечения*** надежности ПС

Основные понятия и показатели:

Надежность-это свойство объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах, соответствующих заданным режимам и условиям *использования, технического обслуживания, ремонта, хранения и транспортирования*

Определения характеристик и субхарактеристик качества :

- **Функциональные возможности** - способность программного средства обеспечивать *решение задач*, удовлетворяющих сформулированные потребности заказчиков и пользователей при применении комплекса программ в заданных условиях.
- **Функциональная пригодность** - набор и описания субхарактеристики и ее атрибутов, определяющие назначение, номенклатуру, основные, необходимые и достаточные функции программного средства, соответствующие техническому заданию и спецификациям требований заказчика или потенциального пользователя.
- **Правильность (корректность)** - способность программного средства обеспечивать *правильные* или приемлемые для пользователя *результаты* и внешние эффекты.
- **Способность к взаимодействию** - свойство программных средств и их компонентов *взаимодействовать* с одной или большим числом компонентов внутренней и внешней среды.
- **Защищенность** - способность компонентов программного средства *защищать программы и информацию* от любых негативных воздействий.

продолжение.

- **Надежность** - обеспечение комплексом программ достаточно низкой вероятности отказа в процессе функционирования программного средства в реальном времени.
- **Эффективность** - свойства программного средства, обеспечивающие требуемую производительность решения функциональных задач, с учетом количества используемых вычислительных ресурсов в установленных условиях.
- **Практичность (применимость)** - свойства программного средства, обуславливающие сложность его понимания, изучения и использования, а также привлекательность для квалифицированных пользователей при применении в указанных условиях.
- **Сопровождаемость** - приспособленность программного средства к модификации и изменению конфигурации и функций.
- **Мобильность** - подготовленность программного средства к переносу из одной аппаратно-операционной среды в другую.

Задачи теории и анализа надежности сложных программных средств:

- **формулирование основных понятий**, используемых при исследовании и применении показателей надежности программных средств;
- **выявление и исследование основных факторов**, определяющих характеристики надежности сложных программных комплексов;
- **выбор и обоснование критериев** надежности для комплексов программ различного типа и назначения;
- **исследование дефектов и ошибок**, динамики их изменения при отладке и сопровождении, а также влияния на показатели надежности программных средств;

продолжение.

- *исследование и разработка методов* структурного построения сложных программных средств, обеспечивающих их необходимую надежность;
- *исследование методов и средств* контроля и защиты от искажений программ, вычислительного процесса и данных путем использования различных видов избыточности и помехозащиты;
- *разработка методов и средств* определения, прогнозирования характеристик надежности в жизненном цикле комплексов программ с учетом их функционального назначения, сложности, структурного построения и технологии разработки.

Факторы обеспечения надежности ПС:

Факторы уязвимости:...

- *ошибки оперативного и обслуживающего персонала* в процессе эксплуатации ПС;
- *искажения в каналах телекоммуникации информации*, поступающей от внешних источников и передаваемой потребителям, а также недопустимые для конкретной информационной системы характеристики потоков внешней информации;
- *сбои и отказы в аппаратуре вычислительных средств*;
- *изменения состава и конфигурации комплекса взаимодействующей аппаратуры информационной системы за пределы, проверенные при испытаниях или сертификации и отраженные в эксплуатационной документации.*

Методы обеспечения надежности ПС:

- **Предупреждение ошибок**, методы позволяющие минимизировать или исключить появление ошибки за счет использования CASE-технологий
- **Обнаружение ошибок**, методы направленные на разработку дополнительных функций программного обеспечения, помогающих выявить ошибки: систематическое тестирование и обязательная сертификация.
- **Устойчивость к ошибкам**, дополнительные функции программного обеспечения, предназначенные для исправления ошибок и их последствий и обеспечивающие функционирование системы при наличии ошибок.

Модели надежности ПО.

Модель надежности программного обеспечения - относится к математической модели, построенной для оценки зависимости надежности ПО от определенных параметров.

Табл «Классификация моделей надежности ПО»

Модели надежности программных средств				
Аналитические				Эмпирические
Динамические		Статические		
Дискретные	Непрерывные	По области ошибок	По области данных	Модель сложности
Модель Шумана	Модель Джелинского – Моранды	Модель Миллса	Модель Нельсона	Модель, определяющая время доводки программ
Модель La Padula	Модель Муса	Модель Липова		
Модель Шика – Волвертона	Модель переходных вероятностей	Простая интуитивная модель		
		Модель Коркорэна		

Процесс создания сложных ПС.

Сложность - простая система характеризуется тем, что человек уверенно может перебрать все связи между её элементами, в сложной он этого сделать не в состоянии.

Концепции в борьбе со сложностью ПО :

Первая – независимость. В соответствии с этой концепцией для минимизации сложности необходимо максимально усилить независимость компонентов системы. По существу это означает такое *разбиение системы, чтобы высокочастотная динамика ее была заключена в единых компонентах*, а межкомпонентные взаимодействия представляли лишь низкочастотную динамику системы.

продолжение.

- **Вторая концепция – иерархическая структура.** Каждый уровень представляет собой совокупность структурных отношений между элементами нижних уровней. *Концепция уровня позволяет понять систему, скрывая несущественные уровни детализации.*
- **Третья - проявление связей всюду, где они возникают.** Основная проблема многих больших программных систем – *огромное количество независимых побочных эффектов, создаваемых компонентами системы.* Из-за этих побочных эффектов систему невозможно понять. И можно быть уверенным, что систему, в которой нельзя разобраться, было очень трудно спроектировать хотя бы с минимальной гарантией надежности.

Приведем один из методов решения задачи:

1. **Поймите задачу. *Изучите данные.*** Изучите неизвестные. Достаточно ли данных для решения? Непротиворечивы ли они?
2. ***Составьте план.*** Чего вы должны добиваться? Какие методы проектирования будут использоваться? Встречалась ли вам уже такая задача? Не знаете ли вы близкой задачи? Можете ли вы воспользоваться ее результатом? Можете ли вы решить более специализированную или аналогичную задачу? Можете ли вы решить часть задачи?
3. ***Выполните план.*** Следуйте своему плану решения задачи. Проверяйте правильность каждого шага.
4. ***Проанализируйте решение*** Все ли данные вы использовали? Проверьте правильность решения. Можете ли вы воспользоваться полученным результатом или примененным методом при решении других задач?

Характеристики программного продукта:

- *функциональные возможности;*
- *надёжность;*
- *практичность;*
- *эффективность;*
- *сопровождаемость;*
- *мобильность.*

Требования разработки:

Требования для обеспечения качества и надежности ПС рекомендуется формулировать требования:

- *к объекту разработки на данном этапе* - к его программным и информационным компонентам, а также к интерфейсу между ними и с внешней средой;
- *к процессу, технологии и организации выполнения* совокупности работ и документов каждого этапа;
- *к методам и характеристикам средств автоматизации* выполнения работ, обеспечивающим необходимую надежность функционирования и качество ПС;
- *к методам и средствам контроля, измерения и документирования* качества процессов и результатов выполненных работ."

Качество программного обеспечения

Качество - способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям

Выделяют группы факторов : *Можно выделить три большие группы факторов, влияющих на качество программного обеспечения:*

- **функциональная** – связана с полнотой и удобством использования реализованных функций программного средства;
- **административная** – связана с квалификацией персонала, организационной структурой и управлением персоналом;
- **программно-архитектурная** – связана с процессом разработки программного обеспечения, выбранными методологиями, инструментальными средствами, использованными на различных этапах жизненного цикла программного обеспечения, а также архитектурой программного средства.

Факторов влияющие на качество

ПО :

- **Понятность:** Назначение ПО должно быть понятным, из самой программы и документации.
- **Полнота:** Все необходимые части программы должны быть представлены и полностью реализованы.
- **Краткость:** Отсутствие лишней, дублирующейся информации. Повторяющиеся части кода должны быть преобразованы в вызов общей процедуры. То же касается и документации.
- **Портируемость:** Лёгкость в адаптации программы к другому окружению: другой архитектуре, платформе, операционной системе или её версии.
- **Согласованность:** По всей программе и в документации должны использоваться одни и те же соглашения, форматы и обозначения.
- **Сопровождаемость:** Насколько сложно изменить программу для удовлетворения новых требований. Это требование также указывает, что программа должна быть хорошо документирована, не слишком запутана, и иметь резерв роста по использованию ресурсов (память, процессор).

продолжение

- **Тестируемость:** Позволяет ли программа выполнить проверку приёмочных характеристик, поддерживается ли возможность измерения производительности.
- **Удобство использования:** Простота и удобство использования программы. Это требование относится прежде всего к интерфейсу пользователя.
- **Надёжность:** Отсутствие отказов и сбоев в работе программ, а также простота исправления дефектов и ошибок.
- **Структурированность**
- **Эффективность:** Насколько рационально программа относится к ресурсам (память, процессор) при выполнении своих задач.
- **Безопасность**

Рекомендуется следующая общая схема процессов оценки характеристик качества программ:

- ***установка исходных требований для оценки*** - определение целей испытаний, идентификация типа метрик программного средства, выделение адекватных показателей и требуемых значений атрибутов качества;
- ***селекция метрик качества***, установление рейтингов и уровней приоритета метрик субхарактеристик и атрибутов, выделение критериев для проведения экспертиз и измерений;
- ***планирование и проектирование процессов оценки*** характеристик и атрибутов качества в жизненном цикле программного средства;
- ***выполнение измерений для оценки***, сравнение результатов с критериями и требованиями, обобщение и оценка результатов.

Выводы по работе:

Изучили:

- Основные *понятия и показатели*
- *Качество* программного обеспечения
- *Требования* для обеспечения качества и надежности ПС
- *Процесс создания* сложных ПС.
- *Методы обеспечения* надежности ПС.
- *Факторы обеспечения* надежности ПС



Благодарим за внимание!