

Надежность программных продуктов

Контроль принимаемых решений

Надежность ПП – это его способность безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.

Отказ ПП - проявление ошибки в работе пп.

Таким образом, надежный ПП не исключает наличия в нем ошибок – важно лишь, чтобы эти ошибки при практическом применении этого ПП в заданных условиях проявлялись достаточно редко.

Для определения характеристик и качества надежности применяют:

- тестирования пп на различных этапах жц

ПП может обладать различной степенью надежности. При оценке степени надежности *ПП* следует учитывать последствия каждого отказа. Некоторые ошибки в *ПП* могут вызывать лишь незначительные неудобства при его применении, тогда как другие ошибки могут иметь катастрофические последствия.

Поэтому для оценки надежности *ПП* иногда используют дополнительные показатели, учитывающие стоимость (вред) для пользователя каждого отказа.

Таким образом:

Обеспечение надежности – является одним из основных мотивов разработки программных средств.

Общими принципами обеспечения надежности можно считать:

1. предупреждение ошибок;
2. самообнаружение ошибок;
3. самоисправление ошибок;
4. обеспечение устойчивости к ошибкам.

1). Предупреждение ошибок –

Цель этого подхода - *не допустить* ошибок в пп.

На основе анализа природы ошибок при разработке пп были сформулированы следующие варианты реализации этого подхода:

- борьба со сложностью
- обеспечение точности перевода
- преодоление барьера между пользователем и разработчиком
- обеспечение контроля принимаемых решений.

Этот подход связан с организацией процессов разработки пп, т.е. с технологией программирования. И хотя гарантировать отсутствие ошибок в пп невозможно, но в рамках этого подхода можно достигнуть приемлемого уровня надежности пп.

Остальные три подхода связаны с организацией самих продуктов технологии – программ. Они учитывают возможность ошибки в программах.

Остальные три подхода связаны с организацией самих продуктов технологии – программ. Они учитывают возможность ошибки в программах.

2). Самообнаружение ошибки в программе означает, что программа содержит средства обнаружения отказа в процессе ее выполнения.

3). Самоисправление ошибки в программе означает не только обнаружение отказа в процессе ее выполнения, но и исправление последствий этого отказа, для чего в программе должны иметься соответствующие средства.

4). Обеспечение устойчивости программы к ошибкам означает, что в программе содержатся средства, позволяющие локализовать область влияния отказа программы, либо уменьшить его неприятные последствия, а иногда предотвратить катастрофические последствия отказа.

Эти подходы используются реже, предупреждение ошибок.

Это связано со следующими причинами:

- 1. С тем, что многие простые методы, используемые в технике в рамках этих подходов, неприменимы в программировании, например, дублирование отдельных блоков и устройств (выполнение двух копий одной и той же программы всегда будет приводить к одинаковому эффекту – правильному или неправильному).
- 2. Добавление в программу дополнительных фрагментов приводит к ее усложнению (иногда – значительному), что в какой-то мере мешает методам предупреждения ошибок.

В большей степени в классической технологии программирования детализован подход предупреждения ошибок.

1). Методы борьбы со сложностью

Известны два общих метода борьбы со сложностью систем:

- обеспечение независимости компонент системы;
- использование в системах иерархических структур.

Обеспечение независимости компонент означает разбиение системы на такие части, между которыми должны остаться по возможности меньше связей.

2).Обеспечение точности перевода.

Обеспечение точности перевода направлено на достижение однозначности интерпретации документов различными разработчиками, а также пользователями пп.

Это требует придерживаться при переводе определенной дисциплины. Допускается использовать общую дисциплину решения задач, рассматривая перевод как решение задачи. В соответствии с этим весь процесс перевода можно разбить на следующие этапы:

1. Понять задачу;
2. Составить план (включая цели и методы решения);
3. Выполнить план (проверяя правильность каждого шага);
4. Проанализировать полученное решение.

3).Преодоление барьера между пользователем и разработчиком.

Как обеспечить, чтобы пп выполнял то, что пользователю разумно ожидать от нее?

Для этого разработчикам необходимо правильно понять:

- во-первых, чего хочет пользователь

- во-вторых, его уровень подготовки и окружающую его обстановку.

Ясное описание соответствующей сферы деятельности пользователя или интересующей его проблемной области во многом облегчает достижение разработчиками этой цели.

При разработке пп следует привлекать пользователя для участия в процессах принятия решений, а также тщательно освоить особенности его работы.

3). Контроль принимаемых решений.

Обязательным шагом в каждом процессе (этапе) разработки пп должна быть проверка правильности принятых решений. Это позволит обнаруживать и исправлять ошибки на самой ранней стадии после ее возникновения, что существенно снижает стоимость ее исправления и повышает вероятность правильного ее устранения. С учетом специфики разработки пп необходимо применять везде, где это возможно,

- смежный контроль
- сочетание как статических, так и динамических методов контроля.

Смежный контроль означает, проверку полученного документа лицами, не участвующими в его разработке, с двух сторон: во-первых, со стороны автора исходного документа, и, во-вторых, лицами, которые будут использовать полученный документ в качестве исходного в последующих технологических процессах. Такой контроль позволяет обеспечивать однозначность интерпретации полученного документа.

Сочетание статических и динамических методов контроля означает, что нужно не только контролировать документ как таковой, но и проверять, какой процесс обработки данных он описывает. Это отражает одну из специфических особенностей ПП (статическая форма, динамическое содержание).