

Возможные подходы к оценке функционирования сложных систем

© 2017 Епихин Максим Николаевич
научный сотрудник

© 2017 Данелян Тэя Яновна

кандидат экономических наук, доцент кафедры прикладной информатики
и информационной безопасности

Российский экономический университет имени Г.В. Плеханова

117997, г. Москва, Стремянный пер., д. 36

E-mail: mepihin@yandex.ru, tdanelan@yandex.ru

Рассмотрены положения оценки функционирования сложных систем (информационных систем), основные характеристики исследования таких систем, приведены примеры возможных подходов к оценке применимости и функционирования ЭИС. Выделены некоторые проблемы, часто возникающие при проектировании и эксплуатации систем такого рода.

Ключевые слова: системный анализ, оценка систем, теория систем.

Оценка качества ЭИС

Множество характеристик качества программного обеспечения может быть представлено в виде дерева, в котором более элементарные характеристики являются необходимым условием существования более обобщенных. Стрелки в нем указывают логическое отношение следования.

Характеристики, расположенные на нижнем уровне дерева, представляют собой множество элементарных свойств, которые имеют фундаментальные отличия друг от друга и группируются в наборы соответственно условиям, необходимым для существования конкретных характеристик промежуточного уровня.

Для определения качества программного обеспечения необходимо объявить для него целесообразную совокупность характеристик качества по приоритетам, установить для них один или более измеримых показателей. Качественные характеристики затем используются для вывода некой *интегральной оценки* достоинства всего программного обеспечения. При этом должны обнаруживаться также всевозможные аномалии характеристик качества, которые следует классифицировать. Серьезные недостатки должны анализироваться и исправляться либо игнорироваться, если выяснится, что данный критерий неприменим к оцениваемому программному продукту, что бывает довольно часто. Например, низкий уровень понятности может быть допустим в случае высокоэффективной программы, работающей в реальном масштабе времени.

Ниже предлагаются описания всех возможных промежуточных и элементарных свойств качественного программного обеспечения.

Понятность. Программное обеспечение обладает свойством понятности в той степени, в которой оно позволяет оценивающему лицу понять его назначение. Под этим определением подразумевается, что всякий программный продукт необходимо создавать с учетом нужд конечного пользователя, условий, оговоренных конкретным документом (“Соглашением о требованиях”, контрактом и т.п.).

Завершенность. Программный продукт обладает свойством завершенности, если в нем присутствуют все необходимые компоненты, каждый из которых разработан всесторонне. О документе говорят, что он завершен, если в нем присутствуют все элементы содержания, перечисленные в оглавлении, и это содержание с достаточной полнотой отражает аспекты функционирования систем, соответствующие всем другим характеристикам.

Осмысленность. Программный продукт обладает свойством осмысленности, если его документация не содержит избыточной информации. Лишние фразы и повторы затемняют основную мысль и не позволяют сосредоточить внимание на важных подробностях. Это относится как к документации, так и к самим программам.

Мобильность. Программный продукт обладает свойством мобильности, если он может легко и эффективно использоваться для работы на ЭВМ иного типа, чем та, для которой он предназначен. Свойство мобильности равнозначно свойству автономности, определяющему способность программных средств работать “на самообслуживании”, без привлечения дополнительных программных ресурсов.

Полезность. Программный продукт обладает свойством полезности, если он удобен для практического применения. *Это свойство имеет две стороны:*

1. Программы должны быть написаны так, чтобы была возможность их полного или частичного использования (в случае необходимости) в иных условиях. При этом речь идет не о свойстве мобильности, а об ответе на вопрос: полезна ли функция, выполняемая данным программным продуктом, для других проектов?

2. Необходимо надлежащая проработка вопросов обеспечения взаимодействия человека с машиной: должны быть четко определены входные параметры, форматы ввода данных, которые следует делать либо свободными, либо обеспечивающими определенную гибкость.

Машинонезависимость. Программный продукт обладает свойством машинонезависимости, если входящие в него программы могут выполняться на вычислительной машине иной конфигурации, чем та, для которой они непосредственно предназначены.

Надежность. Программный продукт обладает свойством надежности, если можно ожидать, что он будет удовлетворительно выполнять необходимые функции в течение определенного времени. Обеспечение надежности предполагает получение ответов на следующие две группы вопросов:

1. Способен ли программный продукт удовлетворить выдвинутым требованиям к нему? Если программный продукт - программа, то достигается ли необходимая точность процедур трансляции, загрузки и выполнения?

2. При функционировании в реальных условиях продолжает ли программа работать правильно в случае исходных данных, существенно отличающихся от тестовых? Как много будет выявлено скрытых ошибок после аттестации программы как работоспособной? Какова вероятность того, что результаты будут содержать необнаруженные ошибки?

Структурированность. Программный продукт обладает свойством структурированности, если его взаимосвязанные части организованы в единое целое определенным образом. Структурированность программы может иметь в своей основе самые различные причины. Например, она может быть разработана в соответствии со специальными стандартами, определенными руководящими принципами и требованиями к интерфейсам, или может быть написана с использованием языка структурного программирования, или может отражать в своей структуре процесс постепенного эволюционного развития на осно-

ве целенаправленных и систематизированных изменений.

Эффективность. Программный продукт обладает свойством эффективности, если он выполняет требуемые функции без лишних затрат ресурсов и времени. Термин "ресурсы" здесь понимается в широком смысле: это может быть оперативная память, общее количество выполняемых команд на одну итерацию решаемой задачи или на один прогон, внешняя память, пропускная способность канала и т.п. Часто эффективность приобретает ценой ухудшения других характеристик, так как нередко является машинозависимой характеристикой и определяется свойствами конкретного используемого языка программирования.

Точность. Программный продукт обладает свойством точности, если выдаваемые им результаты имеют точность, достаточную с точки зрения основного их назначения.

Доступность. Программный продукт обладает свойством доступности, если он допускает селективное использование отдельных его компонентов.

Модифицируемость. Программный продукт обладает свойством модифицируемости, если он имеет структуру, позволяющую легко вносить требуемые изменения.

Открытость. Программный продукт обладает свойством открытости, если его функции и назначения соответствующих операторов легко понимаются в результате чтения текста программы.

Коммуникативность. Программный продукт обладает свойством коммуникативности, если он дает возможность легко описывать входные данные и выдает информацию, форма и содержание которой просты для понимания и несут полезные сведения.

Информативность. Программный продукт обладает свойством информативности, если он содержит информацию, необходимую и достаточную для понимания читающим лицом назначения программных средств, принятых допущений, существующих ограничений, исходных данных, результатов, отдельных компонентов и текущего состояния программ при их функционировании.

Расширяемость. Программный продукт обладает свойством расширяемости, если он позволяет увеличивать при необходимости объем памяти для хранения данных или расширять его функции.

Учет человеческого фактора. Программный продукт учитывает человеческий фактор, если он способен выполнять свои функции, не требуя излишних затрат времени со стороны пользова-

теля, неоправданных усилий пользователя по поддержанию процесса функционирования программ и без ущерба для морального состояния пользователя.

Возможные подходы к оценке применимости и функционирования ЭИС

Первый подход к оценке ЭИС

Создание любой информационной системы направлено на автоматизацию выполнения определенной задачи (задач) в какой-либо организации или структуре. Поэтому при оценке применимости ЭИС для решения определенного круга задач надо установить:

- 1) функциональную полноту ЭИС;
- 2) непротиворечивость множества функций в ЭИС;
- 3) независимость реализуемых ЭИС задач.

Определение указанных параметров составляет основу для одного из подходов к оценке применимости ЭИС.

1. Функциональная полнота ЭИС

В рамках данного подхода при определении функциональной полноты ЭИС необходимо выполнить следующее:

1. Осуществить анализ объекта применения ЭИС, т.е. предметной области ЭИС. В результате этого анализа будет выявлено:

F - функциональное множество задач, требующих решения;

I - характеристики этих задач (поток входящей и выходящей информации по каждой задаче);

$f_i(I_{вх}, I_{вых})$, причем $f_i \in F$ или $\{f_i\}_1^n = F$;

U - предметная область (система);

a_i - алгоритм, или метод решения каждой из поставленных задач,

$$a_i \frac{df}{df_i} = df_i(I_{вх}, I_{вых}).$$

2. На основе полученного анализа выяснить, является ли данная ЭИС функционально полной, непротиворечивой и независимой.

Определим функциональную полноту ЭИС.

2.1. Множество задач F предметной области должно поэлементно совпадать с множеством P задач (программ), решаемых данной системной ЭИС;

$P(F) ? F$ или $\{f_i\} ? \{a_i\}$. Если ДА, то переход к п. 2.2.

2.2. Входящий и выходящий потоки информации объекта применения должны совпадать с входящим и выходящим потоками информации в ЭИС.

$$\{D_{вх}, D_{вых}\}U? \{I_{вх}, I_{вых}\}ЭИС.$$

2.3. Алгоритмы работы в рамках предметной области должны совпадать с алгоритмами соответствующих процессов в ЭИС.

ЭИС можно назвать функционально полной, только если выполняются все три пункта. Если же наблюдаются отклонения, то требуется либо частичная доработка системы, либо отказ от данной ЭИС.

2. Непротиворечивость ЭИС

Необходимо проверить ЭИС на наличие противоречащих друг другу и излишних блоков (программ).

3. Независимость блоков ЭИС

Каждая программная реализация ЭИС должна быть представлена отдельным модулем, который может быть заменен или отдельно отлажен.

Второй подход к оценке ЭИС

Второй подход к оценке применимости ЭИС связан с доказательством экономической необходимости приобретения и внедрения ЭИС. Критериями оценки параметров технологического процесса обработки данных (ТПОД) или ЭИС являются: соотношение затрат и выпуска ЭИС

$$K = B - Z,$$

где K_{max} - оптимальное ТПОД (при Z_{min} или B_{max});
Z - затраты на разработку и функционирование ЭИС;

B - стоимостная оценка выпуска.

$$Z = Z_p E + Z_3 + Z_m,$$

где Z_p - разовые затраты на разработку, отладку и внедрение ЭИС;

E - коэффициент эффективности капитальных вложений;

Z_3 - затраты на эксплуатацию по ТП (технологическому процессу);

Z_m - затраты на модификацию и адаптирование ТПОД.

Третий подход к оценке ЭИС

Третий подход к оценке применимости ЭИС предполагает определение технико-экономического обоснования необходимости применения ЭИС.

Критерием оценки использования новой технологии может служить формула расчета эффективности по трудовым и стоимостным затратам по старой и новой технологиям:

$$T = T_n - T_c = (+/-)\Delta \text{ чел.-ч,}$$

$C = C_n - C_c = (+/-)\Delta \text{ ден. ед. без учета стоимости самой ЭИС.}$

Прямая экономия средств в данном случае получается следующим образом:

$$(+)\Delta - (\text{Стоимость ЭИС})/t \cdot \text{ЭИС,}$$

где $t \cdot \text{ЭИС}$ - время окупаемости.

Кроме прямой экономии, следует учитывать при приобретении ЭИС и косвенную эффективность.

Четвертый подход к оценке применимости ЭИС

Оценка в аспекте интеллектуальности (степени автоматизации работ на объекте применения).

Оценка дается с учетом и других качеств ЭИС (в том числе доброжелательности и удобства). *Интеллектуальность (оценка дается по балльной системе):*

- 1) степень формализации задач;
- 2) совместимость с другими системами;
- 3) степень автоматизации решения задач с помощью ЭИС;
- 4) простота использования (удобство, достоверность) и способ освоения ЭИС пользователем;
- 5) отображение сложных задач;
- 6) разнообразность пользовательского интерфейса;
- 7) степень иерархичности и структурируемости системы;
- 8) степень отображения ручных операций;
- 9) степень контроля над достоверностью информации.

Пятый подход к оценке ЭИС

Пятый подход к оценке ЭИС основан на оценке языковых и программных средств и КТС (комплекса технических средств), используемых ЭИС. Оценка языковых средств:

- 1) адекватность описания маршрута вычисления на языке;
- 2) формализуемость, т. е. возможность применения трансляторов и компиляторов;
- 3) гибкость и глубина охвата, т.е. возможность описания всех необходимых действий в задаче;
- 4) объем работ, необходимый для написания программы и доступа к ней;

5) объем памяти, занимаемый ЭИС вместе с языком и ее интерфейсом;

6) наглядность ПИ (пользовательского интерфейса);

7) требуемая квалификация пользователя;

8) общая полезность ЭИС.

Заключение

Основную проблему оценки качества сложных систем представляет тот факт, что многие его частные качественные характеристики противоречивы.

Когда возникают подобные конфликтные ситуации, пользователи обычно затрудняются указать, какие же характеристики с их точки зрения являются более существенными.

Повышение жизнеспособности программного обеспечения можно осуществлять следующими четырьмя способами:

- 1) установлением в явном виде целевых параметров качества и их относительных приоритетов;
- 2) применением контрольных таблиц и вопросников по анализу качества;
- 3) введением специальных мер, гарантирующих высокое качество программ;
- 4) использованием специальных средств и методов повышения качества.

Если пользователь предпочитает эффективность программы ее мобильность и удобство эксплуатации, то важно довести это до сведения разработчика, причем, сделать это в такой форме, которая позволяла бы в дальнейшем определить, до какой степени желаемые свойства реализованы в готовой системе программного обеспечения.

1. Данелян Т.Я. Структурное моделирование // Экономика, статистика и информатика. Вестник УМО. 2014. □ 6.

2. Данелян Т.Я. Применение математико-статистических методов для функционального моделирования экономических систем // Образование. Наука. Научные кадры. 2016. □ 1.

Поступила в редакцию 02.07.2017 г.