

1.1 ПРАКТИЧЕСКАЯ РАБОТА № 1

ИССЛЕДОВАНИЕ АСПЕКТОВ ИНЖЕНЕРИИ ТРЕБОВАНИЙ К КОРПОРАТИВНОЙ СИСТЕМЕ

1. Цель работы:

- Изучить основные аспекты инженерии требований.
- Получить практические навыки в построении моделей требований к корпоративной информационной системе.

2. Теоретические сведения

Инженерия требований (Requirements Engineering, RE) — поддисциплина системной инженерии, которая занимается разработкой требований. Главная часть инженерии требований — это реверс-инжиниринг использующей (над)системы (using system) для того, чтобы получить описание модели "чёрный ящик".

Требования к программным продуктам или информационным системам (ИС) можно разделить на две большие группы. Это **функциональные требования** (описывающие, что необходимо реализовать в продукте или системе, в т.ч. какие действия должны выполнять пользователи при взаимодействии с ними) и **нефункциональные требования** (описывающие, как должна работать система или программный продукт, и какими свойствами или характеристиками она должна обладать).

Нефункциональные требования часто связаны с **атрибутами качества** (т.е. требованиями, определяющими качественные характеристики разрабатываемого программного обеспечения (ПО) или системы, такие как производительность, надежность, масштабируемость):

Ограничения – условия, ограничивающие выбор возможных решений по реализации отдельных требований или их наборов.

Бизнес-правила — политика, руководящие принципы или положения, которые определяют или ограничивают некоторые аспекты бизнеса, в т.ч. правила, определяющие состав и правила выполнения определенных бизнес-процессов. К бизнес-правилам относятся корпоративные политики, правительственные постановления, промышленные стандарты и вычислительные алгоритмы, которые используются при разработке продукта или системы либо непосредственно влияют на разработку.

Примеры бизнес-правил: «При отгрузке заказа менеджер должен запросить у бухгалтера товарно-транспортную накладную и счет-фактуру», «Если оплата по счету не поступила в течение 15 дней, заказ считается отменённым».

Внешние интерфейсы — описание аспектов взаимодействия с другими системами и операционной средой. К ним относятся требования к API продукта или системы, а также требования к API других систем, с которыми осуществляется интеграция.

Примеры внешних интерфейсов: «Обеспечить запись в журнал операционной системы следующих событий: сообщения о запуске и остановке сервиса ХХ»; «Обеспечить запись в журнал параметров модулей программы: сканера, ядра, антивирусных баз (информация должна заноситься в журнал при запуске программы и при обновлении модулей)».

Предложения по реализации — предложения, оценивающие возможность использования определенных технологических и архитектурных решений.

Предложения по тестированию разрабатываемого ПО — дополнения к требованиям, указывающие, каким образом то или иное требование должно быть протестировано.

Юридические требования — требования к лицензированию, патентной чистоте.

Одним из способов определения нефункциональных требований является составление шаблона.¹

Для групп по определению нефункциональных требований особенно важно привлечь к этой работе не только аналитиков и пользователей, но и **архитекторов и ключевых разработчиков продукта или системы, а также группу тестирования**. Архитектор воспринимает нефункциональные требования как входные данные для выбора и проектирования архитектуры приложения, а группа тестирования планирует те сценарии нагрузочного тестирования, которые будут использоваться для проверки выполнения нефункциональных требований (в основном это касается атрибутов качества).

Критерии качественных нефункциональных требований

Как к функциональным, так и к нефункциональным требованиям применяются *критерии качества требований* — т.е. описание тех качеств, которым должны удовлетворять качественные требования.

Ниже приведены основные характеристики качественных требований.

Полнота (отдельного требования и системы требований) — требование должно содержать всю необходимую информацию для его реализации. В него включается вся информация об описываемом параметре, известная на момент описания. Система требований также не должна содержать невыявленных и не определенных требований.

Однозначность — требование должно быть внутренне непротиворечиво и все работающие с ним должны понимать его одинаково. Требования следует выражать просто, кратко и точно, используя известные термины. Обычно базовые знания читателей спецификации требований к ПО различаются. Поэтому в ее состав нужно включить раздел с определением понятий прикладной области, используемых при определении требований. Пример, неоднозначного требования. «Период обновления экрана должен быть не менее 20 сек.»

Корректность отдельного требования и согласованность (непротиворечивость) системы требований — требование не должно содержать в себе неверной, неточной информации, а отдельные требования в системе требований не должны противоречить друг другу.

Необходимость — требование должно отражать возможность или характеристику ПО, действительно необходимую пользователям, или вытекающую из других требований.

¹ Книга Карла Вигерса "[Разработка требований к программному обеспечению](#)" — в разделе «Приложение Г» этой книги находятся примеры документации требований; [Материалы ГОСТ 34 серии](#)

Осуществимость — включаемое в спецификацию требование должно быть выполнимым при заданных ограничениях операционной среды.

Проверяемость — проверяемость требования означает, что существует конечный и разумный по стоимости процесс ручной или машинной проверки того, что ПО удовлетворяет этому требованию.

Существуют следующие наиболее распространенные модели качества:

[ISO 9126](#)

[ГОСТ 34](#)

[Модель качества по МакКоллу](#) (McCall's Quality Model)

[Модель качества по Бозму](#) (Boehm's Quality Model)

1061-1998 IEEE Standard for Software Quality Metrics Methodology

ISO 8402:1994 Quality management and quality assurance

Все атрибуты качества с точки зрения архитектуры ИС можно разделить на две большие группы: первая группа (runtime) – это атрибуты, относящиеся ко времени работы приложения или системы; вторая группа (design time) определяет ключевые аспекты проектирования приложения или системы. Многие из этих атрибутов взаимозависимы.

Группа runtime. К этой группе относятся следующие атрибуты качества: **Доступность** — атрибут качества, определяющий время непрерывной работы приложения или системы. Чтобы определить этот параметр, обычно указывают максимально допустимое время простоя системы.

Надежность — требование, описывающее поведение приложения или системы в нештатных ситуациях (примеры: автоматический перезапуск, восстановление работы, сохранение данных, дублирование важных данных, резервирование логики)

Требования к времени хранения данных (например, использование БД в качестве постоянного хранилища данных, продолжительность хранения данных)

Масштабируемость — требования к горизонтальному и/или вертикальному масштабированию приложения или системы. Говоря о вертикальной масштабируемости, мы определяем требования к вертикальной архитектуре системы или приложения. К требованиям вертикальной масштабируемости могут относиться, например, возможность переноса приложений на более мощные SMP-системы, поддержка большого объема памяти и файлов. Говоря о горизонтальной масштабируемости, мы определяем требования к горизонтальной архитектуре системы или приложения. К требованиям горизонтальной масштабируемости могут относиться, например, возможность использования технологий кластеризации. Следует особо заметить, что вертикальное масштабирование обычно направлено на повышение производительности системы. Горизонтальное масштабирование, помимо производительности, позволяет повысить отказоустойчивость системы.

Требования к удобству использования системы/приложения (с точки зрения пользователя) и требования к удобству и простоте поддержки (Usability).

Требования к безопасности, как правило, включают в себя три большие категории: требования, связанные с разграничением доступа, требования, связанные с работой с приватными данными, и требования, направленные на снижение рисков от внешних атак.

Требования к конфигурируемости приложения, взаимодействия и расположения компонентов можно условно разделить на четыре уровня:

1. конфигурируемость на основе предопределенного набора параметров (predefined configurability), когда необходимый уровень модификации достигается путем изменения значений параметров из предопределенного набора;

2. конфигурируемость на основе предопределенного набора базовых объектов (framework constrained configurability), когда необходимый уровень модификации достигается путем перекомпоновки предопределенного набора процессов, сущностей и служебных процедур;

3. конфигурируемость путем реализации новых базовых объектов (basis reimplementation), когда обеспечивается расширение набора процессов и сущностей;

4. конфигурируемость путем новой реализации системы (system reimplementation), когда система должна устанавливаться и настраиваться с нуля.

Требования к производительности решения, определяемые в терминах количества одновременно работающих пользователей, обслуживаемых транзакций, времени реакции, продолжительности вычислений, а также скорости и пропускной способности каналов связи

Ограничения, накладываемые на объем доступной памяти, процессорного времени, дискового пространства, пропускную способность сети, при которых приложение должно эффективно выполнять возложенные на него задачи.

Группа design time. К этой группе относятся следующие выборочные атрибуты качества:

Требования к повторному использованию реализации или компонентов приложения или системы (Reusability).

Требования к расширяемости (Extensibility) приложения или системы в связи с появлением новых функциональных требований, тесно связанное с таким архитектурным атрибутом качества, как переносимость кода.

Требования к переносимости (Portability) приложения или системы на другие платформы.

Требования к взаимодействию между компонентами решения, между внешними компонентами, использование стандартных протоколов и технологий взаимодействия (Interoperability). Например, к таким требованиям можно отнести возможность использования нескольких стандартных протоколов для обмена данными между одной из подсистем разрабатываемой системы и внешней системой-поставщиком данных (на [примере](#) ArcGIS).

3. ХОД РАБОТЫ

1. Построить классификацию требований для предметной области, определенной в НИР. Использовать различные модели качества, описанные, например: <http://vspu2014.ipu.ru/proceedings/prcdngs/4585.pdf>

2. Сравнить применимость моделей качества к выбранной предметной области (задаче) НИР.

3. Оформить результаты в виде таблицы, сделать выводы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Понятие инженерии требований.
2. Стандарты инженерии требований.
3. Определить назначение функциональных и нефункциональных требований.
4. Назовите источники сведений для задания требований.
5. Как обеспечить полноту и непротиворечивость требований к системе?
6. Каким образом документируются требования?
7. Понятие корпоративной системы.
8. Примеры корпоративных систем.
9. Основные модели инженерии требований к проекту корпоративной системы.
10. Сравнения и применимость моделей инженерии требований к корпоративной системе.

5. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Боэм Б.В. Характеристики качества программного обеспечения [Текст]: / Б.В. Боэм. – М.: Мир, 1981.
2. Вигерс К.И. Разработка требований к ПО М.: Русская редакция Microsoft, 2004. – 575 с.
3. Липаев В.В. Методы обеспечения качества крупномасштабных программных систем М.: СИНТЕГ. – 2003. – 510 с

Методологии и стандарты, регламентирующие работу с требованиями

1. IEEE 1362 "Concept of Operations Document".
2. IEEE 1233 "Guide for Developing System Requirements Specifications".
3. IEEE Standard 830-1998, "IEEE Recommended Practice for Software Requirements Specifications"
4. IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990
5. IEEE Guide to the Software Engineering Body of Knowledge (1) - SWEBOOK®, 2004.
6. ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания.
7. ГОСТ 34.602-89. Информационная технология. Техническое задание на создание автоматизированной системы
8. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.