Министерство образования Республики Беларусь

ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра технологий программирования

**Методические указания для выполнения практического занятия №6**

**по курсу «Основы программной инженерии»**

«Формирование функциональных и технологических требований к программному средству»

Полоцк, 2021

**Цель:** научиться формировать функциональные и нефункциональные требования к программному обеспечению.

**Теоретические сведения**

**Требования к программному обеспечению** — совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Создаются в процессе разработки требований к программному обеспечению, в результате анализа требований.

Требования могут выражаться в виде текстовых утверждений и графических моделей.

В классическом техническом подходе совокупность требований используется на стадии проектирования ПО. Требования также используются в процессе проверки ПО, так как тесты основываются на определённых требованиях.

Этапу разработки требований, возможно, предшествовало технико-экономическое обоснование, или концептуальная фаза анализа проекта. Фаза разработки требований может быть разбита на выявление требований (сбор, понимание, рассмотрение и выяснение потребностей заинтересованных лиц), анализ (проверка целостности и законченности), спецификация (документирование требований) и проверка правильности.

**Свойства требований:**

* полнота,
* ясность,
* корректность,
* согласованность,
* верифицируемость,
* необходимость,
* полезность при эксплуатации,
* осуществимость,
* модифицируемость,
* трассируемость,
* упорядоченность по важности и стабильности,
* наличие количественной метрики.

**Классификация требований по К. Вигерсу.**

Для управления изменениями требований сегодня зачастую применяется подход, сформулированный американским программным инженером и ИТ-консультантом Карлом Вигерсом. Основное содержание подхода К. Вигерса сформулировано в его книге «Разработка требований к программному обеспечению».

При создании ИС выделяются два вида требований: **функциональные и нефункциональные**. Вигсрс выделяет три вида функциональных требований:

1) **бизнес-требования**. Данный вид требований формируется заказчиком ИС и основывается, прежде всего, на целях создания заказываемого продукта. Бизнес-требования определяют, какие преимущества должен получить заказчик при получении готового продукта, а также какие проблемы или задачи будут решены в результате его применения. В результате формулирования бизнес-требований очерчиваются границы создаваемой ИС, а также создается общий образ проекта. Например, на уровне бизнес- требований могут быть сформулированы требования к поддержке бизнес- процессов. Если в качестве проекта выступает CRM-система, то в качестве бизнес-требований к ней будут выделены коммуникационные, отчетные и управленческие процессы;

2) **пользовательские требования**— это задачи, которые будет решать ИС для поддержки пользователей. Функциональные требования данного уровня представляются в виде сценариев (user journey), алгоритмов и таблиц «событие — отклик». Также формирование пользовательских требований может вестись на основе ключевых ролей, которые будут использеваться для работы ИС. Возможности каждой роли, будь то «Клиент», «Инвестор», «Партнер» и др., будут различаться в дальнейшем;

3) **функциональные требования** — это основные требования по функциональности ИС, которые далее детально описываются в виде технического задания и передаются на реализацию разработчикам.

Но, как показала практика, одних только функциональных требований к ИС недостаточно, поскольку созданная лишь на их основе ИС не будет удовлетворять всем требованиям бизнеса. В связи с этим К. Вигерс выделяет три вида нефункциональных требований к ИС:

1) **бизнес-правила**, включающие требования регуляторов (например, экологические нормативы или нормы безопасности), промышленные стандарты, корпоративные стандарты и другие ограничения, которые неизбежно налагаются внешней средой или политикой компании;

2) **атрибуты качества**, которые не относятся к функциональности системы, однако являются обязательным условием для эффективного применения создаваемой ИС в дальнейшем. В качестве требований вида «Атрибуты качества» может выступать возможность интеграции с другими ИС, интероперабельность, поддержка программных продуктов и т.п.;

3) **ограничения**, к которым, как правило, относятся вынужденные технические или ресурсные ограничения (уровни производительности, технические протоколы и пр.).

Стоит отдельно отметить, что в своих работах К. Вигерс указывал на разницу в источниках для формирования функциональных и нефункциональных требований. Так, основные источники для формирования функциональных требований — это функциональные описания объектов автоматизации, подразделений компании, ролей и т.п. В свою очередь нефункциональные требования могут быть сформулированы на основе качественных характеристик. Процесс формирования требований строится на основе обследования предприятия, включающего интервью с сотрудниками и заинтересованными сторонами, наблюдение за рабочим процессом, анкетирование и т.п.

**Функциональные требования** задают «что» система должна делать; нефункциональные – с соблюдением «каких условий» (например, скорость отклика при выполнении заданной операции); часто функциональные требования представляют в виде сценариев (вариантов использования) Use Сase.

1. Функциональные требования. Это перечень сервисов, которые должна выполнять система, причем должно быть указано, как система реагирует на те или иные вход­ные данные, как она ведет себя в определенных ситуациях и т.д. В некоторых слу­чаях указывается, что система не должна делать.
2. Нефункциональные требования. Описывают характеристики системы и ее окружения, а не поведение системы. Здесь также может быть приведен перечень ограничений, накладываемых на действия и функции, выполняемые системой. Они включают временные ограничения, ограничения на процесс разработки системы, стандарты и тд.
3. Требования предметной области. Характеризуют ту предметную область, где будет эксплуатироваться система. Эти требования могут быть функциональными и не­функциональными.

В действительности четкой границы между этими типами требований не существует. Например, пользовательские требования, касающиеся безопасности системы, можно отнести к нефункциональным. Однако при более детальном рассмотрении такое требование можно отнести к функциональным, поскольку оно порождает необходимость включения в систему средства авторизации пользователя.

**Сбор требований** — это один из самых важных этапов процесса создания любой информационной системы, будь то десктопное, веб или мобильное приложение или же просто доработка уже существующего решения. Прежде, чем начать собирать требования, необходимо выявить всех заинтересованных лиц (стейкхолдеров), которые будут пользоваться системой. Чем точнее будет этот список, тем полнее будут требования.

**Стейкхолдерами** могут быть любые физические лица и/или организации, которые активно участвуют в нашем проекте, и чьи интересы могут быть затронуты не только в процессе создания системы, но и непосредственно по завершению самого проекта. Ими могут быть менеджеры, начальники отделов, директора, любые сотрудники организации, которые будут хоть как-то взаимодействовать с готовым решением, и чьи требования (пожелания, идеи, потребности, проблемы) будем собирать.

Существует множество различных техник сбора требований, которые помогут лучше понять, что же хочет заказчик.

Рассмотрим основные из них чуть подробнее:

**Анкетирование**

Данный способ подразумевает под собой составление листа-опросника (анкеты, брифа), который может содержать открытые (требуют от опрашиваемого сформулировать его ответ) и закрытые (требуют от опрашиваемого выбрать ответ из предложенных вариантов) вопросы.

Анкетирование используется для того, чтобы подтвердить или детализировать ранее известные требования, выбрать параметры для решений.

Преимущества:

1. Высокая скорость получения результатов.
2. Сравнительно небольшие материальные затраты.

Недостатки:

1. Данный метод не подходит для выявления неявных требований.
2. При составлении опросника физически невозможно учесть все необходимые вопросы.

**Интервью**

Этот метод известен многим, беседа с заинтересованным лицом, тет-а-тет.  
Необходимо задавать открытые вопросы для получения информации и закрытые для того, чтобы подтвердить или опровергнуть конкретные варианты требований.

Данный способ применяется, в основном, для получения информации по какой-либо конкретной теме и/или для уточнения требований.

Многим может показаться этот способ достаточно легким, но это не так. Провести хорошее интервью достаточно сложно. Вы должны гибко реагировать на реакцию интервьюируемого и в случае необходимости изменять порядок заготовленных вопросов или их формулировку. Не забудьте во время интервью вести заметки.

Из плюсов:

1. Возможность задавать вопросы в произвольной последовательности.
2. Возможность использовать вспомогательный материал.
3. Анализ невербальной реакции опрашиваемого человека, позволит сделать дополнительный вывод о достоверности его ответов.

Из минусов:

1. Интервью отнимает достаточно много времени и сил.
2. Дополнительной сложностью является получение одинаковых ответов от интервьюируемых.

**Автозапись**

Данный метод подразумевает под собой работу с записями, письмами (электронными письмами), а также с любым другим документом, автором которого является Заказчик или конечный пользователь (т.е. стейкхолдер).  
В действительности это может быть и документ и наговоренная на диктофон последовательность действий или самая обычная салфетка, на которой заказчик набросал свои идеи / проблемы /хотелки, которые необходимо превратить в полноценные требования, согласовать и передать в разработку.

Примером такого метода может быть работа с концепцией или видением проекта (сам Заказчик любит называть это — «ТЗ»), которую он прислал вам на момент начала работ по аналитике.

Преимущество:

1. Помогает лучше понять сложные процедуры или процессы.

Недостаток:

1. Данный метод сильно зависит от опыта Заказчика, а также от его умения формулировать и выражать свои мысли.

**Изучение существующей документации**

Данная методика может быть использована при наличии в организации документации, которая может помочь в определении потребностей Заказчика. Примеры документации включают в себя: регламенты,  
описания процессов, структура организации, спецификации продукта, различные процедуры, стандарты и инструкции, шаблоны документов и т.д.  
Выявленные требования являются основой для дальнейшего анализа и должны быть детализированы.  
Данная методика применима, например, при автоматизации устоявшихся в организации регламентированных бизнес процессов.

Плюс:

1. Быстрое получение информации.

Минус:

1. Данный способ не применим при наличии в компании только базовых документов (или при их полном отсутствии) или, если в компании Заказчика не поддерживается актуальность документации.

**Повторное использование спецификации**

Повторно использовать спецификации можно в том случае, если есть уже завершенные один или несколько подобных проектов.

Техническое задание, подготовленное на предыдущем проекте может быть использовано для другого проекта с целью сократить продолжительность сбора, анализа и разработки требований, что позволит быстрее начать разработку.  
Например, ТЗ для интернет магазинов похожи друг на друга и содержат одинаковые требования.

В большинстве случаев только часть документации актуальна для нового проекта, поэтому потребуется тщательная проверка требований на соответствие текущим целям и задачам Заказчика.

Преимущество:

1. Сокращение времени на разработку документации.

Недостатки:

1. Высокая стоимость первого проекта.
2. Излишняя детализация требований, может привести к их дорогостоящим изменениям в будущем.

**Представитель заказчика в компании разработчика**

Один из наиболее эффективных методов сбора требований, поскольку позволяет получать от представителя Заказчика своевременную оценку прогресса, корректности реализации, в короткие сроки получать обратную связь (фидбек) и дополнительную информацию для корректировки и разработки требований.  
Метод часто применяется для сбора и управления требованиями при итерационной разработке, позволяет оперативно собирать, согласовывать и дорабатывать требования.

В дополнение можно сказать, что наличие представителя заказчика в компании разработчика является одним из главных правил Agile.

Преимущество:

1. Быстрое получение обратной связи и информации от Заказчика.

Недостатки:

1. Достаточно высокая цена для Заказчика.
2. Затраты по времени на адаптацию сотрудника.

**Мозговой штурм**

Мозговой штурм — наиболее часто используемый метод получения требований, которые связанны с новыми или плохо изученными направлениями деятельности организации Заказчика или функциями системы.  
Он позволяет собрать множество идей от различных заинтересованных лиц (стейкхолдеров) в кратчайшие сроки и практически бесплатно.  
Во время мозгового штурма участники «накидывают» любые идеи, касающиеся решения данной проблемы.

С помощью этой методики можно проработать несколько различных вариантов решения заданной проблемы, а также разрешить конфликты требований.

Плюсы:

1. Позволяет генерировать множество (в том числе и нестандартных) вариантов решений, а также позволяет участникам развивать идеи друг друга.

Минусы:

1. Участники мозгового штурма должны быть мотивированы на идеи.
2. Трудно применим в распределенных командах.

**Совещание**

Совещание — встреча, ориентированная на обсуждение конкретных вопросов, которые были определены и озвучены участникам заранее.  
На такие встречи привлекаются люди, которые придерживаются различных точек зрения по текущей проблеме и могут помочь описать требования, основываясь на взглядах с разных сторон. В процессе совещания уточняется общий список требований, выявляются скрытые требования и решаются конфликты требований.  
Совещания являются одной из ключевых практик в Agile, т.к. в них участвуют все стороны, заинтересованные в развитии проекта и решении проблемы.

Плюсы:

1. Позволяет развить и детализировать требования, определить приоритеты.

Недостатки:

1. Сложности в организации встречи, если команда географически разделена, могут возникнуть трудности с присутствием всех необходимых людей на совещании.
2. Консенсус необязательно будет достигнут.

**Use case**

Use cases или варианты использования позволяют собрать и сформировать функциональные требования от лица участников [Диаграммы вариантов использования](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2) определяют границы решения и показывают связи с внешними системами и участниками. Метод позволяет детализировать требования с точки зрения пользователей, а также помогает уточнить и систематизировать функционал, который требуется реализовать.

Плюсы:

1. Позволяет проработать все варианты развития сценария (основной и альтернативные сценарии)

Минусы:

1. Метод не применим для сбора нефункциональных требований.

**Ход работы**

**Задание 1.** Осуществить в сети Интернет поиск готовых программных продуктов (не менее пяти), решающих задачу из предметной области, в соответствии с вариантом. Представить результат в виде списка информационных систем (таблица 1).

Таблица 1 – Программные продукты

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Название продукта | Название производителя | Требования к системе | Возможности | Стоимость | Достоинства | Недостатки |
|  |  |  |  |  |  |  |  |

Варианты:

1. Видеоредакторы
2. Текстовые редакторы
3. Архиваторы
4. Планировщики
5. Среды разработки
6. Интернет-магазины
7. Игры
8. Веб-браузеры
9. Антивирусы
10. Аудиоредакторы
11. Графические редакторы
12. Редакторы презентаций
13. Аудио проигрыватели
14. Видеопроигрыватели
15. Программы для создания презентаций

На основании таблиц сделать вывод, каким должно быть программное обеспечение, чтобы учитывать все достоинства и недостатки готовых программных продуктов. Результат представить в виде списка главных достоинств.

**Задание 2.** Для вашей системы (варианты из практических работ 1-2) составить список тех пользователей, которые будут иметь дело с разрабатываемым программным продуктом.

**Задание 3.** Для каждого пользователя определить список его возможностей в вашей информационной системе (описание должно быть сделано на языке, понятном пользователю!).

**Задание 4.** Выделите главные требования к вашей информационной системе (разделив их на функциональные и нефункциональные).

Практическая работа оформляется в текстовом файле (в программе MS Word или аналогах) в электронном или печатном виде.

**Контрольные вопросы:**

**Источники:**

**https://studme.org/226099/informatika/upravlenie\_trebovaniyami\_swebok**

**https://intuit.ru/**