|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполняемой работы | Подпись руководите ля |
| 27.04.2022 | **Практическая работа №3**  **Тема**: Формализация требований  **Цель:** Выяснить требования к системе и преобразовать их на язык формальных моделей  **Задание 1.**  Изучил материал «Формализация требований на практике». Основные моменты законспектировал в дневник-отчет.  Основой успеха при создании надежного и полезного программного обеспечения (ПО) всегда является четкое понимание потребностей его пользователей. В наше время, когда технологии разработки программ все время совершенствуются и наращивают свои возможности, неправильное понимание нужд пользователей создаваемого ПО все еще остается одной из часто встречающихся причин провальных результатов проектов по его разработке. Самое неприятное свойство таких провалов — то, что неудачными могут оказаться результаты даже проекта, вполне успешного с точки зрения его участников и внутренних характеристик, и это станет ясно только в самом его конце.  Чтобы обеспечить более адекватный учет нужд пользователей при создании ПО, в рамках процесса его разработки обычно выделяют особую деятельность, называемую анализом требований, и включающую, как минимум, следующие действия.  - Предварительный анализ предметной области. Это выделение ее сущностей, связей между ними, а также типовых задач в ее рамках, позволяющее впоследствии понимать пользователей.  - Cбор пожеланий и выявление действительных нужд всех заинтересованных лиц.  - Формулировка на их основе требований к создаваемой программе.  - Фиксация полученной информации в виде ряда документов и моделей, которые уже используются непосредственно разработчиками программных систем.  Фиксация всей этой информации в документах и моделях должна помочь перейти от неясных, часто неосознанных, противоречивых и постоянно изменяющихся проблем и потребностей пользователей к четко сформулированным, непротиворечивым и однозначно понимаемым требованиям, которые уже можно использовать для разработки программных систем. Причем, независимо от конкретных разработчиков и организаций, участвующих в их создании на основе выделенных требований, итоговые системы должны получаться «одинаковыми», т.е. взаимозаменяемыми при решении любой задачи.  Как же добиться нужного качества формулировки требований? Как обеспечить однозначность их понимания различными людьми?  Эту задачу может решить формализация требований, т.е. представление их в рамках некоторого математического формализма, которое часть указанных выше свойств требований обеспечивает только за счет выбранной формы их представления, а другие позволяет проконтролировать, затратив на это приемлемые усилия. В частности, само по себе формальное представление некоторой информации всегда однозначно. Существуют строгие и выверенные на практике методы проверки формальных систем на полноту и непротиворечивость  Стандарты IEEE 830 и IEEE 1233 описывают как раз те характеристики, которые должны иметь требования, чтобы их можно было с успехом использовать для разработки ПО, и которые должны проверяться во время валидации по SWEBOK.  Первый стандарт перечисляет следующие необходимые свойства требований к ПО.  - Корректность, под которой имеется в виду адекватность.  - Однозначность.  - Полнота.  - Непротиворечивость или согласованность.  - Ранжированность по важности для пользователей и стабильности.  - Проверяемость.  - Удобство внесения изменений.  - Прослеживаемость.  Второй стандарт, IEEE 1233 определяет характеристики требований к более общим программно-аппаратным системам, во многом похожие на перечисленные выше.  Характеристики набора требований в целом.  - o Уникальность каждого требования, отсутствие смысловых  - пересечений с другими.  - o Четкое описание взаимосвязей.  - o Полнота.  - o Непротиворечивость или согласованность.  - o Четкое описание контекста, в раках которого эти требования имеют смысл.  - o Удобство внесения изменений.  - o Поддержка вариантов и версий.  - o Подходящий для выбранных целей уровень детализации формулировок.  **Задание 2.**  Изучил материал «Алгоритм формализации требований». Основные моменты законспектировал в дневник-отчет.  Одной из актуальных проблем на начальных этапах жизненного цикла разработки информационных систем является формирование требований к системе, корректно и точно отражающих цели и задачи заказчика. Для успешной реализации проекта разработки информационной системы, отвечающей целям и задачам заказчика, нужно выяснить требования заказчиков к системе и преобразовать их на язык формальных моделей так, чтобы обеспечить соответствие целям и задачам организации. Для этого необходимо использовать наиболее эффективные методы преобразования требований в формальные модели. Объектом исследования являются подходы и методы преобразования требований в формальную модель при разработке информационных систем. Предложен алгоритм преобразования требований на естественном языке в формальную модель в виде гибридной диаграммы деятельности. Предложенный алгоритм позволяет получать основные и предметные знания из текстов требований, описывать структуру и поведение системы, преобразовывать созданную формальную модель в ряд других моделей UML, также предложенное табличное представление текста можно рассматривать как базу знаний при проектировании информационной системы. Практическая ценность результатов исследований определена их применением для решения проблем формализации требований на первом этапе жизненного цикла разработки информационных систем.  Технологии разработки информационных систем; преобразование требований; формальная модель; методы преобразования требований.  Для успешной реализации проекта разработки информационной системы, отвечающей целям и задачам заказчика, нужно преобразовать требования на язык формальных моделей так, чтобы обеспечить соответствие разработанной информационной системы целям и задачам организации.  Рассмотрим алгоритм преобразования требований в формальную модель.  На первом шаге преобразований требований необходимо применить правила для трансформации предложений на естественном языке в табличное представление, которое состоит из пяти колонок: словосочетание вначале предложения, подлежащее, сказуемое, дополнение и словосочетание в конце предложения.  На втором шаге преобразований необходимо заполнить все пустые ячейки в полученном табличном представлении текста. Эти пустые ячейки образовались из-за того, что в требованиях содержаться предложения в страдательном залоге. Пассивной формой (страдательным залогом) называется такая конструкция предложения, при которой подлежащее не является действующим лицом (или предметом), а само подвергается действию со стороны дополнения (при этом дополнение может лишь подразумеваться, не будучи выражено в предложении). Для этих случаев необходимо преобразовать форму глагола из страдательного залога в действительный, и осуществить некоторые перестановки между колонками «Подлежащее» и «Дополнение» в табличном представлении, если это требуется. Как только заполнены все ячейки в табличном представлении, необходимо определить рабочие точки в системе – актеров. Актеры являются абстрактными компонентами системы, и они соответствуют подлежащему в предложениях. Необходимо рассматривать подлежащее лишь в тех предложениях, которые имеют глаголы, отличные от «быть».  Алгоритм, представленный в третьем шаге, применяется к табличному представлению, в результате чего создается лист актеров.  На следующем шаге необходимо вписать действия, выполняемые каждым из актеров, в их рабочие точки в том порядке, в котором они появились в тексте. Для этого необходимо для каждого актера последовательно пройти колонку «сказуемое» в табличном представлении и выписать действия, выполняемые одним «подлежащим». Наречия времени может изменить порядок действий, но сейчас они не являются объектом нашего исследования.  На пятом шаге, используя таблицу элементов, необходимо спроектировать граф элементов и установить пути между последовательно выполняемыми действиями. Связи в данном графе устанавливаются с помощью элемента «соединительный путь». Следует отметить, что данный элемент отличается от термина «сообщение», используемого в UML-диаграмме последовательности. «Сообщение» в UML обозначает то, что один актер инициализирует действия другого. Инициализация выражается через глаголы, но для того, чтобы отличать, какие глаголы являются сообщениями, а какие являются действиями, нам необходимы знания и интуиция. Автоматический анализ нуждается в интуиции человека эксперта, и для того, чтобы сделать работу проще, во внимание принимается следующая логика: все действия выполняются в рабочих точках. Путем соединения действий фактически представляется результат действий между рабочими точками.  **Задание 3.**  Требования, выделенные в Практических работах 1 и 2 записал в виде табличного представления и построил семантическую сеть, оформил в Приложении 3.1 |  |