Содержание:

Введение

Программное обеспечение представляет собой совокупность программ, предназначенных для решения задач на компьютере. Программа - это упорядоченный набор команд. Программное и аппаратное обеспечение работают взаимосвязано и в непрерывном взаимодействии. Любой аппаратное устройство управляется программно.

Программное обеспечение можно разделить на три вида:

- системное

- прикладное

- инструментальное

Приведенная классификация является достаточно условной. Интеграция программного обеспечения привела к тому, что практически любая программа имеет черты каждого класса.

Системное ПО предназначено для управления работой компьютера, распределения его ресурсов, поддержки диалога с пользователями, оказание им помощи в обслуживании компьютера, а также для частичной автоматизации разработки новых программ.

Основными компонентами общесистемного программного обеспечения являются операционные системы, которые решают задачи взаимосвязанного функционирования отдельных компонентов.

Системное программное обеспечение осуществляет управление работой вычислительной системы. Как правило, системные программы обеспечивают взаимодействие других программ с аппаратными составляющими, организации пользовательского интерфейса. Сюда относят операционные системы, сервисные системы.

Прикладное программное обеспечение предназначено для решения прикладных задач профессиональной деятельности человека (то есть, приложенное к

практике). Спектр таких программ чрезвычайно широк: от производственных и научных до учебных и развлекательных. Сюда относят расчетные, обучающие, моделирующие программы, компьютерные игры и тому подобное.

Инструментальное программное обеспечение предназначено для разработки всех видов информационно-программного обеспечения. При этом под информационным обеспечением понимают совокупность предварительно подготовленных данных, необходимых для работы программного обеспечения. Например, любая современная программа имеет встроенную справку для работы с этой программой. Файл справки представляет собой информационное обеспечение.

Целью курсовой работы является выявление видов программного обеспечения и основные требования, предъявляемые к программному обеспечению.

Задачи курсовой работы:

- рассмотреть системное программное обеспечение

- рассмотреть прикладное программное обеспечение

- рассмотреть инструментальное программное обеспечение

- исследовать требования к программному обеспечению

- определить функциональные и нефункциональные требования

- выявить принципы построения программного обеспечения

Предметом курсовой работы является теоретические особенности программного обеспечения.

Структура работы. Курсовая работа состоит из введения, двуз глав, заключения и библиографии.

1. Основные виды программного обеспечения

1.1. Системное программное обеспечение

К системным программам, прежде всего, относят:

- операционные системы;

- сервисные программы - драйверы, утилиты, программы управления сетями и т.д.

К системному программному обеспечению также относят широкий круг программ, которые необходимы для управления работой компьютера, поддержания диалога с пользователем, оказание помощи в обслуживании компьютера. Некоторые системные программы поставляются вместе с компьютером и документацией к нему [1].

Важнейшей составляющей системного программного обеспечения является операционная система.

Операционная система (ОС) - это комплекс программ, которые загружаются при включении компьютера и осуществляют диалог с пользователем, управление компьютером, запускают другие программы на выполнение. Операционная система создает удобные условия для работы пользователя с устройствами компьютера, обеспечивает взаимодействие между программами [4].

Дело в том, что привычные операции для работы с устройствами - это операции очень низкого уровня, и действия, которые необходимо выполнить, состоят из сотен или тысяч элементарных операций. Например, для выполнения такой, казалось бы, несложной действия, как копирование данных с дискеты на жесткий диск, компьютер выполняет тысячи операций по запуску накопителя, проверки его работоспособности, поиска и обработки информации [7].

Операционная система, с одной стороны, освобождает пользователя от 2 необходимости понимания внутренних процессов работы компьютера, а с другой - уведомляет его о ходе выполнения действий, ошибки и возможный путь их устранения. Операционная система согласовывает работу периферийных устройств, поддерживает диалог с пользователем. Именно с операционной системы начинается загрузка компьютера.

Операционная система осуществляет и загрузки приложений в оперативную память, передает им управление в начале работы, предоставляет необходимые ресурсы, а при завершении их работы освобождает оперативную память [2].

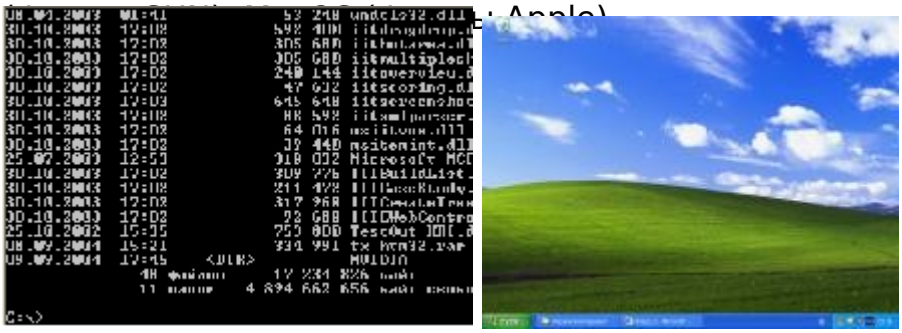
Для персональных компьютеров создано много операционных систем. Одной из первых популярных ОС была дисковая операционная система (сокращенно ДОС) фирмы Microsoft Corporation - MS-DOS (Microsoft Disk Operation System) (рис. 6.1а).

Как любые прикладные программные средства, операционные системы постоянно обновляются.

Так, первая версия MS-DOS была создана в 1981 г. Затем она была усовершенствована и положено в основу более развитых ОС. MS-DOS связана с аппаратной частью компьютера и считается наиболее надежной. Именно она часто используется для выполнения специальных технических операций. Например, при возникновении аварийной ситуации, когда другие средства не помогают, компьютер загружается с дискеты с записанной MS-DOS [6].

С помощью MS-DOS запускают и те программы, которые не работают иначе. С развитием аппаратной составляющей операционные системы тоже меняются. Сейчас наиболее распространены операционные системы компании Microsoft - Windows 98, Windows 2000, Windows NT, Windows XP (рис.1.1) [10].

В 2007 году вышла новая версия - Windows Vista. Именно Windows можно увидеть во многих пользователей дома, в офисах, в учреждениях и учебных заведениях. Операционная система Windows NT специально создана для работы в компьютерных сетях. Для организации работы сети часто пользуются операционными системами Unix или Linux, которые считаются надежными, стабильными и достаточно защищенными. Развиваются и находят своих сторонников и такие операционные системы, как OS / 2 (фирмы IBM), JavaOS



а) MS-DOS б) Windows

Рисунок 1.1 - Окна операционных систем [10]

Каждая из операционных систем имеет свои специфические особенности, однако все они выполняют следующие функции: управление аппаратными ресурсами компьютера; управления процессами ввода / вывода данных на внешние устройства; управления работой приложений; обеспечение удобного способа

взаимодействия пользователя с компьютером.

Операционные системы могут иметь своеобразные надстройки, которые называют оболочками. Они упрощают взаимодействие пользователя с компьютером, не требуют запоминания команд и их ручного ввода, обеспечивают отображение действий на экране. [3] Важной составляющей системных программ является драйверы. Они необходимы для правильной работы любого устройства. То есть каждому устройству соответствует своя программа-драйвер, которая организует обмен данными между устройством и компьютером. Так, для работы монитора должно быть 3 установлен драйвер монитора, для клавиатуры - драйвер клавиатуры для мыши - драйвер мыши, для накопителей - драйверы и т.д. [14]

Существует много вспомогательных программ, которые расширяют возможности операционных систем. Такие программы называются сервисными. Например, программы-архиваторы с помощью использования специальных методов «упаковки» позволяют уплотнять информацию на дисках. Антивирусные программы предназначены для диагностирования и удаления вирусов. Программы оптимизации дисков в порядок данные и таким образом обеспечивают быстрый доступ к информации на дисках.

1.2. Прикладное программное обеспечение

Для персональных компьютеров разработаны тысячи прикладных программ различного назначения. Наиболее распространенными среди них являются текстовые и графические редакторы, электронные таблицы, системы управления базами данных и т.д.

Текстовые редакторы позволяют готовить и оформлять текстовые документы. Они обеспечивают использование различных шрифтов при наборе текста, автоматическое расстановки переносов в словах, выравнивание краев текста, оформление текста в колонках, нумерацию страниц, создание таблиц и диаграмм, проверку правописания и подбор синонимов и тому подобное. Редакторы, предоставляющих дополнительные средства для оформления текстов, называют текстовыми процессорами.

Среди текстовых процессоров распространены Microsoft Word, OpenOffice.org Writer и др. Электронные таблицы (или табличные процессоры) позволяют решать задачи, связанные с числовыми расчетами, с большими таблицами данных [8].

Кроме того, эти программы помогают строить двух- или трехмерные графики, диаграммы. Среди табличных процессоров наиболее распространенными являются Microsoft Excel, OpenOffice.org Calc и др.

Системы управления базами данных (СУБД) предназначены для систематизации и обработки больших объемов информации. Они обеспечивают ввод данных, поиск, сортировку записей, создание отчетов [9].

Используются такие СУБД:

Microsoft SQL Server;

- Oracle;

- MySQL;

- Microsoft Access и др. [12]

В состав прикладного программного обеспечения входят и графические редакторы. Как правило, пользователю предоставляются возможности изображения основных графических примитивов (отрезка, прямоугольника, круга, произвольной кривой), использование палитры цветов, преобразование рисованных объектов (перемещение, копирование, увеличение, уменьшение).

Современные графические 4 редакторы обрабатывать изображения, введенное с помощью сканеров, получать трехмерные изображения. Среди графических редакторов распространены Adobe Photoshop, Adobe Illustrator, Corel DRAW, 3D Studio, Macromedia Flash, Paint [10].

Существуют программные средства, которые сочетают в себе возможности текстовых, графических редакторов, электронных таблиц, систем управления базами данных. Эти программные средства называются интегрированными системами.

Главное преимущество их состоит в том, что они имеют схожие компоненты, предусматривают единые правила работы с программами, единственный способ использования функциональных клавиш.

Среди интегрированных систем популярны: Microsoft Office, Open Office [8]. В состав прикладного программного обеспечения относятся также средства специализированного назначения: математические пакеты, учебные пакеты,

системы деловой и научной графики, системы автоматизированного проектирования, бухгалтерские системы и т.д.

1.3. Инструментальное программное обеспечение

Инструментальные системы, или, как их еще называют, системы программирования, предназначенные для создания новых программ. Ведь пользователям часто требуется, чтобы программа выполняла специфические операции, не предусмотренные существующими прикладными программными средствами.

Для популярных языков программирования разработано много инструментальных систем. Естественно, что программисты предпочитают те системам, которые являются легкими в использовании, позволяют получить эффективные программы и имеют встроенные библиотеки готовых подпрограмм [1].

Системы программирования, прежде всего, различаются по языку программирования, которую они поддерживают. Например, большой популярностью пользуются системы программирования Visual C ++, Borland C ++ (для программистов, пишущих на языке C), Visual Basic (для тех, кто предпочитает языке Basic), Turbo Pascal, Borland Delphi (для поклонников языка программирования Pascal).

Инструментальное программное обеспечение предназначено для разработки всех видов информационно-программного обеспечения. При этом под информационным обеспечением понимают совокупность предварительно подготовленных данных, необходимых для работы программного обеспечения. Например, любая современная программа имеет встроенную справку для работы с этой программой. Файл справки представляет собой информационное обеспечение [10].

К инструментальному программному обеспечению относят:

- редакторы (текстовые, графические, музыкальные);

- системы табличной обработки данных (табличные процессоры), системы управления базами данных;

- трансляторы языков программирования;

- интегрированные системы делопроизводства и т.д. [13]

Таким образом, инструментальные программные средства могут оказать помощь на всех стадиях разработки программного обеспечения.

2. Основные требования к программному обеспечению

2.1. Требования к программному обеспечению

Программное обеспечение должно отвечать следующим характеристикам:

- модульность (выполнение условия позволяет снизить расходы по изменениям функциональной возможности системы за счет варьирования конфигурации ее отдельных элементов) [6];

- открытость (требование, направленное на обеспечение системы взаимодействовать с другим программным обеспечением по определенным стандартам) бинарная совместимость с предыдущими приложениями "что позволяет взаимодействовать с унаследованными системами;

- масштабируемость программного обеспечения для широкого спектра системных конфигураций от изоморфной рабочей станции к корпоративной информационной системе;

- независимость от платформы (ограниченное одной платформой применения накладывает ограничения на приобретение систем в будущем) [5].

Время от времени компании по ряду причин (производительность, масштабируемость, стоимость и т.п.) изменяют действующие операционные системы на другие. Изменяя платформы, большинство компаний рассчитывают перенести применения со старой платформы на новую. При этом применение, построенные на основе независимой от платформ системы, могут быть сразу же использованы на новой платформе.

Применение, что привязаны к операционной системе или сетевой операционной системы, в таком случае приходится переписывать или даже полностью заменять [3].

Если стоимость переписывания или замены приложений превышает доход от инвестиций в новую платформу, компания вынуждена сохранить свое далеко не оптимальное операционная среда) соответствие нормативной конфигурации компьютера (большинство организаций используют различные модели компьютеров различных производителей.

Этим моделям предварительно заданная конфигурация без учета специфики пользователя. Кроме того, они могут отличаться и по составу комплектующих. Через некоторое время, когда нужно будет пополнение или обновления драйверов и приложений, соответственно резко возрастут временные и финансовые затраты) совместимость с приложениями инфраструктуры, объединяет не только пользователей одного предприятия, но и основных его деловых партнеров и заказчиков (деловые организации уже обнаружили потенциальные возможности обработки данных в масштабе нескольких предприятий [14].

Построив применения на основе указанной инфраструктуры, компании всего мира смогли получить;

- неожиданно высокие доходы по своим инвестициям в информационные технологии);

- наличие встроенной диагностики вирусов на клиентских местах и серверах (на предприятии достаточно одного успешного вирусного нападения, чтобы восстановление информационной структуры "съело" не только годовой бюджет на информационные технологии, но и всю прибыль предприятия);

- наличие эффективной системы восстановления частичной работоспособности системы в форс-мажорных ситуациях и т.д. [2]

Классификация требований к программному обеспечению и программных систем:

Функциональные - перечень функций, которые система должна или не должна выполнять, его поведение в определенных условиях.

Не функциональные - характеристики системы и ее окружения, ограничения на функции, но не поведение [13].

Функциональные и нефункциональные требования предметной области - характеристики предметной области эксплуатации системы.

2.2. Функциональные и нефункциональные требования

Функциональные требования к ПС [1] - это те требования, которые отвечают на вопрос: "что делает система?" или "должна делать система?". Это перечисление всех основных функций обработки входящих и представления исходных данных, которые должны быть реализованы в ВС. Они описывают главную функциональность системы. Функциональные системные требования также называют набором взаимосвязанных вариантов использования ПО или прецедентов.

Прецедент (Use Case) - это набор взаимосвязанных сценариев, который описывает использование программной системы определенные акторы для решения одной из задач [11].

Актер (Actor) - сущность, имеет поведение: например, человек (пользователь), отдельный программный компонент или другая программная система [8].

Сценарий (Scenario) - последовательность действий или взаимодействий между актером и ПС [5].

Первичной формой описания прецедентов является текст, который должен быть сформирован в процессе выявления СВ (путем общения с будущими пользователями, экспертами с предметной области и т.д.). Для дальнейшей формализации (структуризации) текстового описания прецедентов существует несколько форм:

1) Свободная форма описывается неформальный стиль описания. Описание прецедента занимает несколько абзацев и охватывает различные возможные сценарии.

2) Краткое описание - аннотация в виде одного абзаца. Она описывает только главный успешный сценарий.

3) Развернутое описание - при таком подходе подробно описываются все шаги и варианты развития сценария.

В нотации UML - функциональные требования описывают с помощью диаграммы прецедентов (Use Case Diagram).

Получив представление об основных определениях функциональных требований и их основные способы описания, можно перейти к описанию основных атрибутов качества ПО или нефункциональных требований.

Нефункциональные требования к ПО [9] отвечают на вопрос: "как та или иная ПС реализует функциональные требования?». К этим требованиям относятся следующие шесть атрибутов качества ПО:

1. Производительность (perfomance) - свойство ПС выполнять причиненные функции за определенный промежуток времени.

2. Надежность (reliability) - свойство ПС обрабатывать данные с заданным уровнем отказов и ошибок.

3. Масштабируемость (scaleability) - свойство ПС сохранять производительность и надежность в заданных пределах в условиях значительного роста числа пользователей и / или аппаратной платформы системы; это предполагает влияние увеличения числа операторов-пользователей, так и возможность увеличения числа технических средств (датчиков, контроллеров, терминалов и т. д.), данные из которых должны быть обработаны в ВС с заданными значениями производительности и надежности [4].

4. Удобство использования (usability) - эта характеристика показывает, насколько данная ПС соответствует таким критериям конечного пользователя как:

- понятность функций интерфейса;

- возможность изучения системы;

- эргономичность интерфейса [1].

5. Сопровождение (maintenancability) - свойства ПС, характеризуются следующими атрибутами:

- анализ исходного текста;

- возможность внесения изменений (возможность конфигурирования)

- тестирование исходного кода.

6. Переносимость (portability) - свойство ПС корректно функционировать не только на исходной платформе, но также иметь возможность быть использованной на

другой операционной или аппаратной платформе [2].

Рассмотрев развернутое описание нефункциональных требований к ПО, можно перейти к описанию основных особенностей компонентного программирования и свойств компоненты, чтобы понять принципы, которые будут положены в основу компонентов.

Ниже будут рассмотрены основные атрибуты качества, необходимо положить в основу будущего программного обеспечения [6].

Во-первых, необходимо обеспечить удобство пользования программным продуктом для облегчения работы с ним. Пользователь не должен тратить много времени на ознакомление и изучение основных функций программы.

Во-вторых, необходимо обеспечить надежность работы программы для многократного ее использования с минимальным количеством отказов в работе.

В-третьих, необходимо обеспечить возможность внесения изменений в программный продукт, поскольку данная предметная область имеет очень широкий спектр применения и способов применения.

В-четвертых, необходимо обеспечить достаточную скорость расчета поставленных задач, поскольку решение задач многомерной оптимизации очень ресурсомиске [10].

Следовательно, необходимо обеспечить следующие нефункциональные требования к программному продукту:

1) удобство;

2) надежность;

3) сопровождение;

4) производительность.

Функциональные требования для данного программного продукта обнаружены и представлены в свободной форме описания.

1. Пользователем данной компоненты, то есть основным актером прецедента, может быть исследователь в области расположения объектов в пространстве. Пользователь программы может не иметь опыта в программировании или

компьютерных науках, но нуждаюсь удобной и быстрой работы с программным средством для решения проблем расположения геометрических объектов на плоскости [9].

2. Обработка данных:

- программа должна обеспечить удобный ввод данных пользователем с помощью графического интерфейса;

- программа должна обеспечить поиск уже готовых шаблонов геометрических объектов;

- программа должна обеспечить получение необходимых данных для работы в удобном формате;

- программа должна обеспечить возможность сохранения отдельных объектов пользователем в удобном формате данных;

- программа должна обеспечить возможность сохранения программных решений пользователем в удобном формате данных;

- программа должна обеспечить возможность повторного использования программных решений и сохранившихся объектов [5].

3. Математические методы:

- необходимо реализовать алгоритм нахождения функции плотного размещения объектов;

- необходимо реализовать алгоритм нахождения функции плотного размещения объектов с учетом анизотропии геометрических объектов;

- алгоритм должен быстро и надежно рассчитывать поставленную задачу [2].

Ниже представлена UML-диаграмма прецедентов, отражающая функциональные требования.

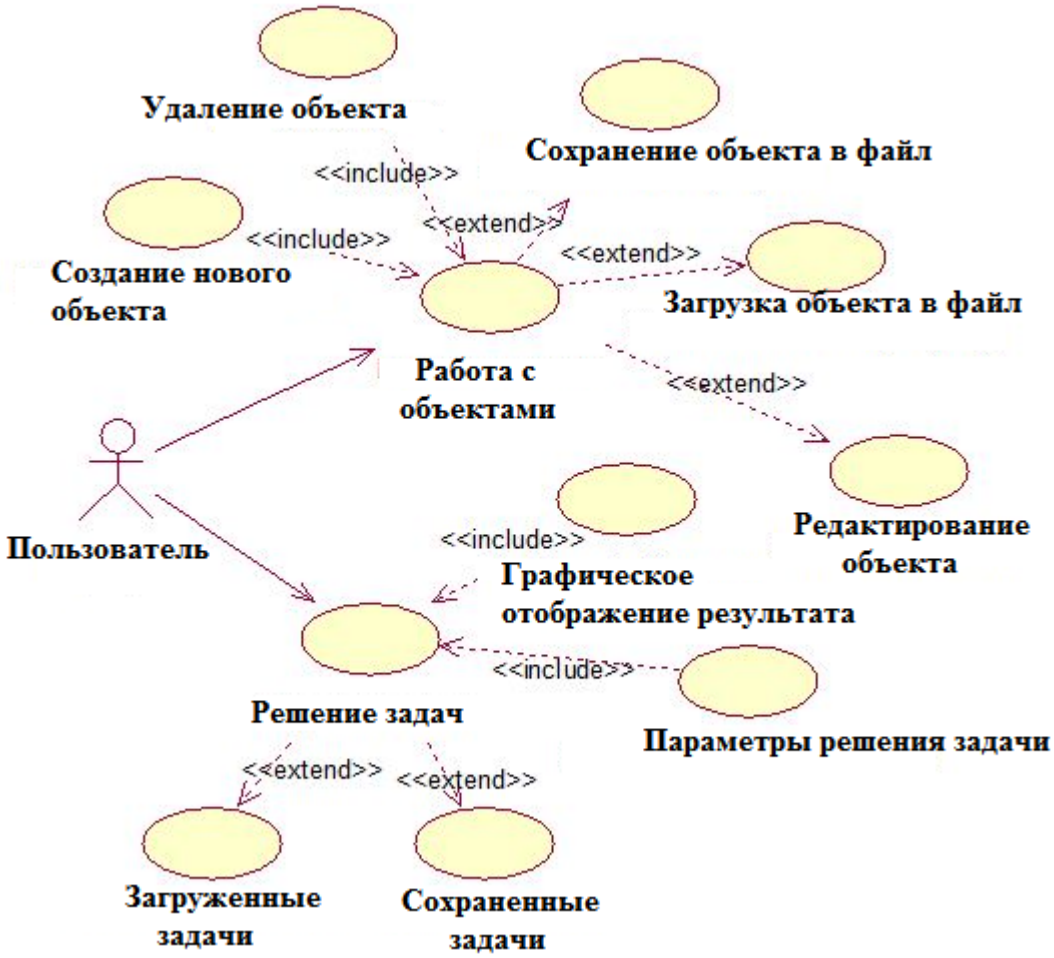


Рисунок 2.1 - Диаграмма вариантов использования [2]

Таким образом, были обнаружены все функциональные и нефункциональные требования к программному обеспечению, с учетом полученных функциональных и нефункциональных требований от существующей программной системы.

2.3. Принципы построения программного обеспечения

Процесс предоставления архитектуры, компонентов, интерфейсов и других характеристик системы или ее компонентов называется проектированием. Результат процесса проектирования – дизайн [4]. Как процесс, проектирования является инженерная деятельность в рамках жизненного цикла, в которой должным образом анализируются требования для создания описания внутренней структуры программного обеспечения, которая является основной для конструирования программного обеспечения. Программный дизайн должен

описывать архитектуру программного обеспечения, то есть представлять декомпозиции программной системы в виде организованной структуры компонент и интерфейсов между компонентами. Важнейшей характеристикой готовности дизайна является тот уровень детализации компонентов, который позволяет заняться их конструированием [8].

Проектирование программных систем можно рассматривать как деятельность, результат которой состоит из двух составных частей:

- Архитектурный и высокоуровневый дизайн - описание высокоуровневой структуры и организации компонентов системы;

- Детализированная архитектура - описание каждого компонента в том объеме, который необходим для конструирования.

В 1999 году Том ДеМарко предложил терминологическое разделение различных видов дизайна:

D - дизайн - декомпозиция структуры программного обеспечения в виде набора фрагментов или компонент;

FP - дизайн - семейство архитектурных представлений, основанных на шаблонах;

I - дизайн - создание высокоуровневой концепции; данный вид дизайна является результатом процесса анализа требований и их трансформации в подходы к реализации [7].

Если рассматривать данную область знать в терминах ДеМарко, то проектирование программного обеспечения в понимании программной инженерии может содержать в себе только D-дизайнта FP- дизайн. И-дизайн относится к работе с программными требованиями.

Область знаний по проектированию программного обеспечения представлена в виде 6 секций, структурированных по темам, изображенных на рисунке 2.2. [11]

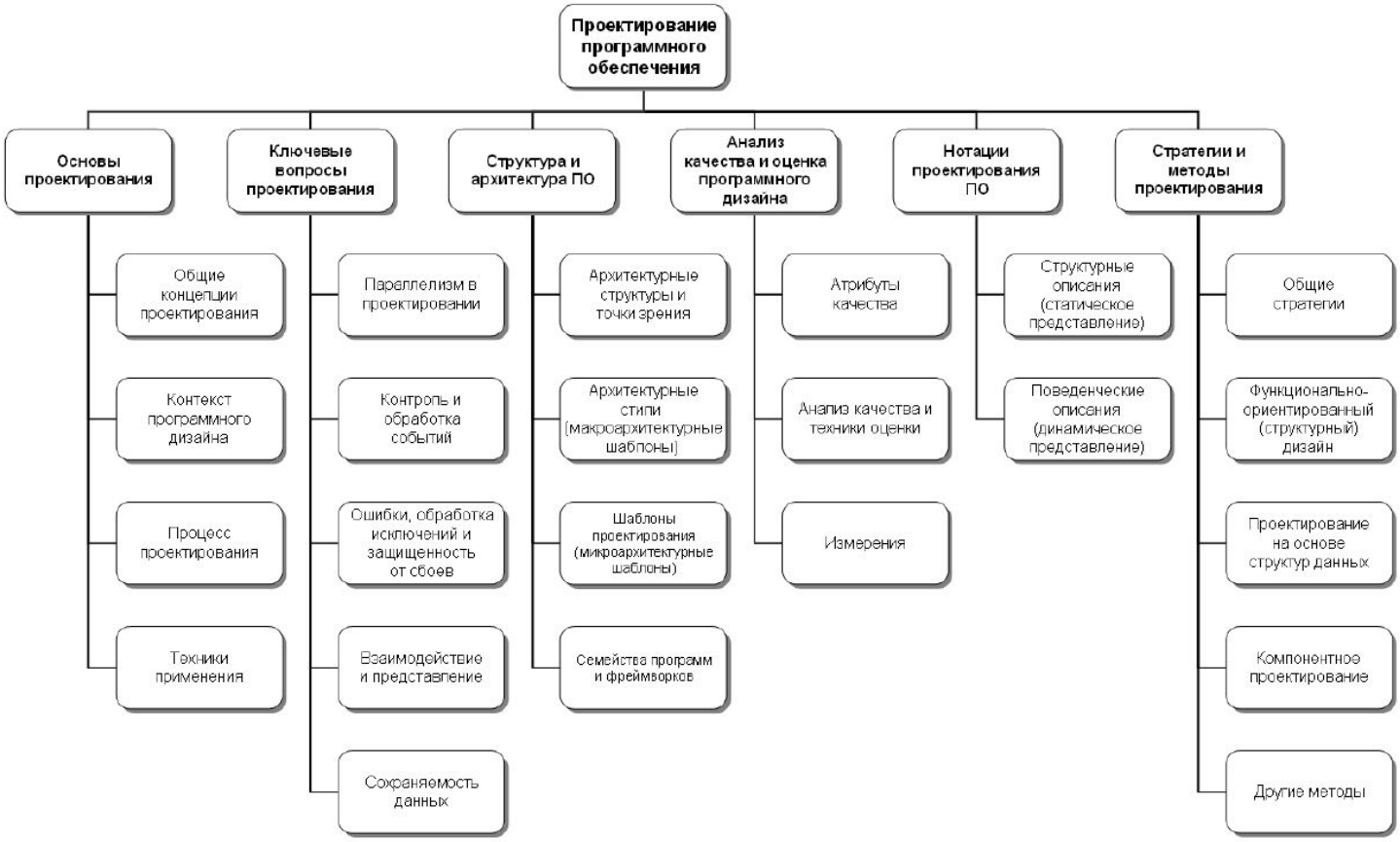


Рисунок 2.2 - Область знаний «проектирование программного обеспечения» [11]

Принципы построения программного обеспечения:

1. Модульность ПО - проведение декомпозиции алгоритмов и программ на модули с целью выделения общих типичных функций и компонентов.

2. Интеллектуальность ПО - наличие знаний о предметной области и умение использовать их при решении задач.

3. Черный ящик - ПО должно скрывать от пользователей сложный механизм организации и взаимодействия программ.

4. Умолчания - замалчивание однажды заданных параметров, если они имеют смысл в других аналогичных задачах [3].

7. Критические вопросы процесса разработки программного обеспечения качества [3].

Людям свойственно ошибаться. Каждая ошибка, когда она найдена, является сюрпризом, исправление которого или дорого стоит, или выбивает из ритма разработки.

Если контроль качества в организации ослаблен, нужно запланировать в самом процессе разработки ряд инспекций проектной документации и инспекций кодов, разрабатывая планы качества, систему измерений, программу тестирования, то есть более интенсивную работу по подтверждению качества программного обеспечения (Software Quality Assurance).

Производительность технологии.

Очень часто при новых разработках не ясно, как разработать алгоритм, достичь целей разработки или ограничить функциональность [14]. Поскольку позже "латание дыр" может быть безуспешным или трудоемким, или приводит к срывам темпов разработки, разумно в процессе предусмотреть заранее эксперименты для получения нужных сведений. Дальнейшая разработка будет развиваться более упорядоченным и эффективным образом.

Нестабильность (изменчивость) требований.

Для того, чтобы спроектировать, построить и оттестировать необходимую программу нужно, чтобы ее функции, интерфейсы и системный базис должны были стабильны. Так как возможны изменения при разработке, эти изменения должны быть временно заморожены. В планируемые периоды могут быть рассмотрены пакеты изменений и, соответственно, подделать программа. Если таким путем не управлять изменениями, процесс разработки становится нестабильным [12].

Существуют три основных типа изменений требований.

Неизвестные требования. Пользователь думает, что он знает, чего хочет, но при первоначального использования открывает для себя, что реальные потребности другие, чем он думал. Это нормальное явление при автоматизации человеческой деятельности. С ним можно бороться или рано прототипирование или планированием множества релизов, которые постепенно разрабатываются, используются, оцениваются и наращиваются до уровня следующего релиза.

Нестабильные требования. В то время как общие требования известны, в частности продолжают "плавать". В бортовых системах, для которых hardware и software часто проектируется одновременно, изменения в hardware приводят к изменению требований к программному обеспечению. Изменение hardware, диктуемых программными изменениями, является нонсенсом, аналогичным требованиям поменять фундамент при строительстве здания. Тем не менее, с данным явлением можно бороться путем предсказания нестабильности и изоляции их [4].

Непонятные требования. Даже если требования известны и стабильны, разработчики часто не понимают их настолько подробно, чтобы сделать удовлетворительный программный продукт. Типичным примером является пользовательский интерфейс. Здесь полезно тестирования пользователем прототипов интерфейса или ранних версий пользовательской документации.

Сложность.

Программы приложений часто легче разработать, чем системы, поскольку их платформа и окружения более стабильны. Это не означает, что разработка приложений требует меньшей квалификации. Это означает, что есть меньше источников для одновременных изменений [6].

Например, при разработке новой операционной системы все ее элементы находятся в состоянии постоянных изменений: управляющая программа, компиляторы, система управления данными и т.д., в том числе и архитектура компьютера [8].

Для того, чтобы достичь хотя бы какого-то результата, нужно обеспечить хотя бы промежуточную стабильность. Обычно стабильность достигается путем использования приемов модульного программирования, в частности, выделение модулей и определения интерфейсов. Далее управление изменениями позволяет для каждого модуля иметь ощутимые и стабильные требования на определенном интервале между сменами.

Заключение

В ходе выполнения данной курсовой работы были исследованы основные виды и требования к программному обеспечению. Были пройдены следующие этапы для решения поставленной задачи:

Были подготовлены теоретические материалы и краткий обзор программного обеспечения. Была рассмотрена краткая характеристика функциональных и нефункциональных требований программного обеспечения. Было обнаружено функциональные и нефункциональные требования к программе,

Все программное обеспечение компьютера можно условно разделить на системное программное обеспечение, прикладное программное обеспечение и инструментальные системы.

Важнейшей составляющей системного программного обеспечения является операционная система, которая предназначена для организации взаимодействия пользователя с компьютером, управление ресурсами компьютера, запуска других программ на выполнение.

Кроме того, к системным программам относятся драйверы, утилиты, архиваторы и др. Для персонального компьютера разработаны тысячи прикладных программ различного назначения.

Наиболее распространенными среди них являются текстовые и графические редакторы, электронные таблицы, системы управления базами данных. Инструментальные системы предназначены для создания новых программ.

Выходными артефактами данной курсовой работы является в полной мере спроектировано решение поставленной задачи.

Библиография

1. Боэм Б. У. Инженерное проектирование программного обеспечения: Пер. с англ. под ред. А. А. Красилова М.: Радио и связь, 1985.
2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем / А.М. Вендров. - М: «Финансы и статистика», 2000. – 380 с.
3. Гагарина Л.Г. Технология разработки программного обеспечения / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. — М.: ИД «ФОРУМ»; ИНФРА-М, 2008. — С. 400.
4. Информатика. Учебное пособие / Под ред. Б.Е. Одинцова, А.Н. Романова - М.: Вузовский учебник: ИНФРА-М, 2012. – 300 с.
5. Леонтьев В. Твой компьютер 2010. Все новое "железо" года / В.Леонтьев. - Олма Медиа Групп, 2010.
6. Леффингуэлл Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл, Д. Уидриг. - М.: Вильямс, 2002. - 448 с.
7. Мартю В.М.Информатика для: Учебник/ Под редакцией В.М. Мартю. - 2009. - 880с.
8. Можаров Р.В. Программное обеспечение персональных компьютеров / Р.В. Можарова, Н.Р. Можарова, В.В. Евтеев, О.А. Кузьменко, М.О. Шевченко // Учебное пособие для вузов. – М.: Финстатинформ, 2003. – 450 с.

9. Орлов С.А. Технологии разработки программного обеспечения: Учебник для вузов. 3-е изд./ С.А. Орлов. – СПб.: Питер, 2004. – 527 с.
10. Сенкевич Г. Вторая жизнь старого компьютера / Г.Сенкевич. - БХВ-Петербург, 2010. – 159 с.
11. Скопин И.Н. Понятия и модели жизненного цикла программного обеспечения: Учебное пособие Новосиб. гос. ун-т / И.Н. Скопин. - Новосибирск, 2003. – 225 с.
12. Соломенчук В. Железо ПК 2010 / В.Соломенчук, П.Соломенчук. - БХВ-Петербург, 2010. – 264 с.
13. Соммервилл И. Инженерия программного обеспечения: 6-е издание. М.: Вильямс / И. Соммервилл, 2002. – 301 с.
14. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г.Буч, Дж. Рамбо. - СПб: Питер, 2002. - 496 с.