

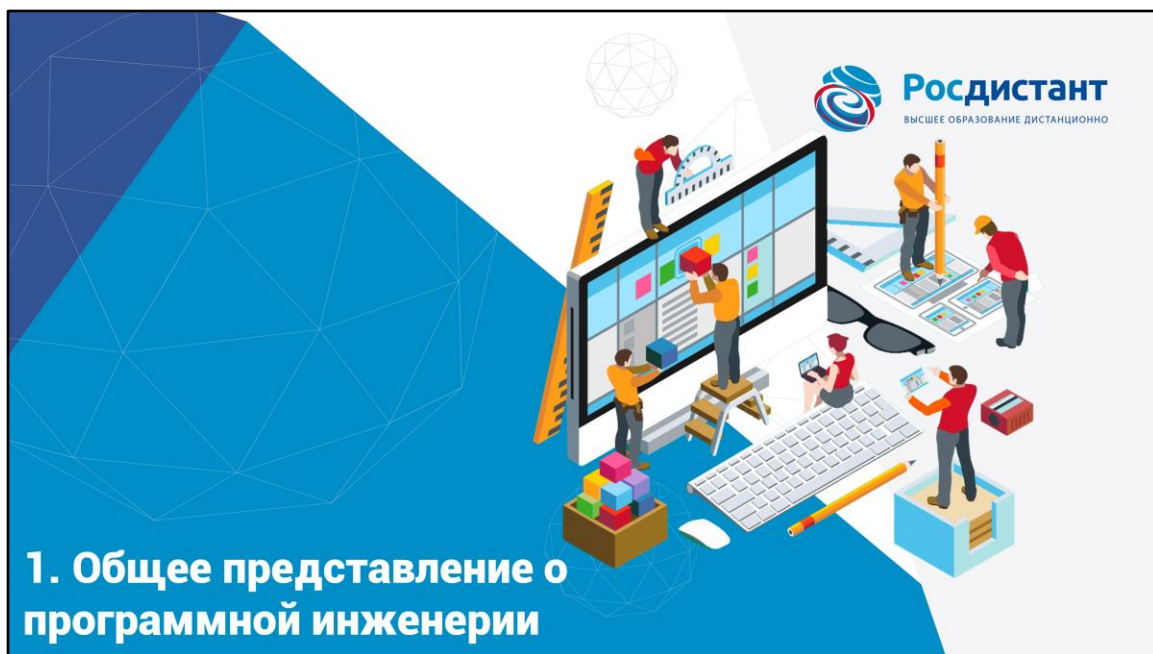


Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

ВВЕДЕНИЕ В ПРОГРАММНУЮ

ИНЖЕНЕРИЮ 1 ЧАСТЬ





1. Общее представление о программной инженерии

Тема 1. Общее представление о программной инженерии

В ходе изучения данной темы у вас должно сложиться общее представление о программной инженерии. Вы научитесь различать основные понятия: программа, программное обеспечение, программный продукт. Вы должны понимать общую концепцию и подходы к разработке программного обеспечения, значимость принципов программной инженерии в деятельности разработчиков программных продуктов.

ПОНЯТИЕ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Программная инженерия - процесс анализа требований пользователей с последующим проектированием, созданием и тестированием программного приложения

IEEE в стандарте 610.12-1990

- применение систематического, дисциплинированного, вычислимого подхода к разработке, эксплуатации и сопровождению программного обеспечения

Фриц Бауэр

- создание и использование стандартных инженерных принципов

Барри Бем

- практическое применение научных знаний для творческого проектирования и создания компьютерных программ



Программная инженерия определяется как процесс анализа требований пользователей с последующим проектированием, созданием и тестированием программного приложения, которое будет удовлетворять требованиям заказчика. Программная инженерия – это прежде всего процесс анализа требований пользователей с последующим проектированием, созданием и тестированием программного приложения.

Давайте посмотрим на различные известные определения программной инженерии:

IEEE в стандарте 610.12–1990 определяет программную инженерию как:

- применение систематического, дисциплинированного, вычислимого подхода к разработке, эксплуатации и сопровождению программного обеспечения;
- приложение инженерии к программному обеспечению.

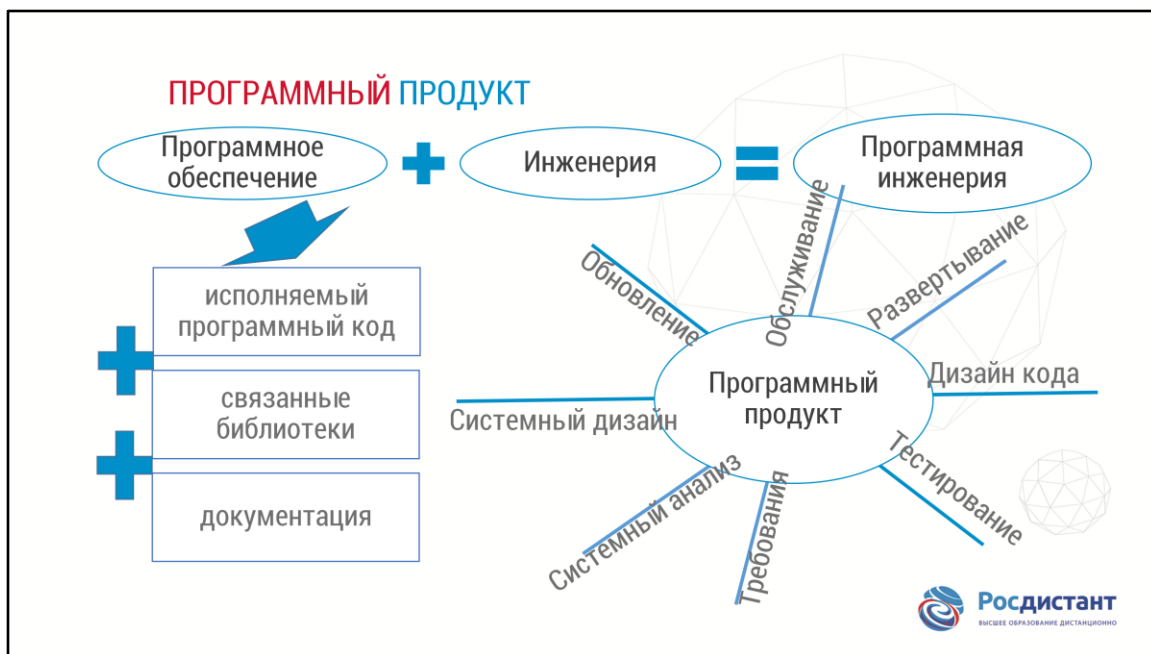
Фридрих Бауэр определил программную инженерию как «создание и использование стандартных инженерных принципов». Он считает, что это поможет получить экономически надежное программное обеспечение, которое эффективно работает на реальных машинах.

Барри Бозм говорит о программной инженерии как о включающей в себя «практическое применение научных знаний для творческого проектирования и создания компьютерных программ». По его мнению, к программной инженерии

также относится сопутствующая документация, необходимая для разработки, эксплуатации и обслуживания программного обеспечения.

Программная инженерия – это подробное исследование от инженерии до проектирования, разработки и сопровождения программного обеспечения.

Программная инженерия была введена для решения проблем некачественных программных проектов. Проблемы возникают, когда программное обеспечение превышает сроки, бюджет и снижает уровень качества. Решение этих проблем гарантирует, что приложение будет построено последовательно, правильно, в срок, в рамках бюджета и в рамках требований. Спрос на программную инженерию также возник, чтобы удовлетворить огромную скорость изменения требований пользователей и среды, в которой должно работать приложение.



Давайте сначала разберемся, что же означает программная инженерия. Этот термин состоит из двух слов: программное обеспечение и инженерия. Программное обеспечение – это больше, чем просто программный код. Программа – это исполняемый код, который служит некоторой вычислительной цели. Программное обеспечение считается набором исполняемого программного кода, связанных библиотек и документации. Программное обеспечение, созданное для конкретных требований, называется программным продуктом.

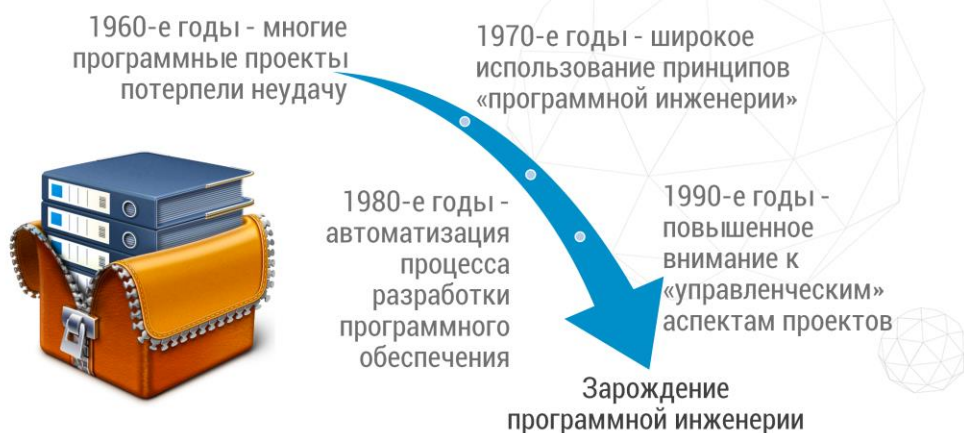
С другой стороны, инженерия – это разработка продуктов с использованием четко определенных научных принципов и методов. К ним относятся:

- обновление,
- обслуживание,
- развертывание,
- дизайн кода,
- тестирование,
- системный дизайн,
- системный анализ,
- требования.

Таким образом, программная инженерия – это инженерная отрасль, связанная с

разработкой программного продукта с использованием четко определенных научных принципов, методов и процедур. Результатом разработки программного обеспечения является эффективный и надежный программный продукт.

ИЗ ИСТОРИИ



В конце 1960-х годов многие программные проекты потерпели неудачу. Они вышли за рамки бюджета. На выходе получилось ненадежное программное обеспечение, обслуживание которого требует больших затрат. Программное обеспечение большого размера было трудным и довольно дорогим в обслуживании. При этом множество программного обеспечения не способно было удовлетворить растущие требования заказчика. Кроме того, с увеличением аппаратных возможностей возрастала сложность программных проектов; спрос на новое программное обеспечение увеличивался быстрее, чем возможность создания нового программного обеспечения.

Все вышеперечисленные проблемы привели к программному кризису, для которого решение заключалось в следующем: преобразование неорганизованного кодирования в дисциплину программной инженерии. Эти инженерные модели должны были помочь компаниям оптимизировать операции и предоставить программное обеспечение, отвечающее требованиям клиентов.

В конце 1970-х годов широко использовались принципы программной инженерии, хотя о самой программной инженерии еще не было речи. К этим принципам относятся:

- управлять, используя поэтапный план жизненного цикла;

- выполнять непрерывную проверку;
- поддерживать дисциплинированный контроль продукта;
- использовать современные методы программирования;
- четкая подотчетность за результаты;
- использовать меньше людей;
- сохранять приверженность улучшению процесса.

Более подробно мы чуть позже рассмотрим эти принципы.

В 1980-х годах произошли автоматизация процесса разработки программного обеспечения и рост автоматизированной разработки программного обеспечения.

Девяностые годы прошлого столетия стали свидетелями повышенного внимания к управленческим аспектам проектов, стандартов качества и процессов, таких как ISO 9001 .

Так зародилась программная инженерия как область, направленная на грамотное и качественное создание программного обеспечения.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ И КОМПЬЮТЕРНЫЕ НАУКИ

Программная инженерия

- процесс анализа требований пользователей с последующей разработкой программного приложения
- создание нового программного обеспечения
- работа с командой

Информатика

- проектирование и понимание компьютеров и вычислительных процессов
- изучение того, как обрабатываются данные и так далее
- использует абстрактный подход к компьютерному программированию
- исследование работы компьютеры

Попробуем сравнить программную инженерию с компьютерной наукой – информатикой. Как в программной инженерии, так и в информатике мы рассматриваем программное обеспечение. Так что же их связывает и чем они различаются?

Что такое программная инженерия? Программная инженерия – это процесс анализа требований пользователей с последующей разработкой программного приложения, которое будет удовлетворять этим требованиям.

Она занимается созданием нового программного обеспечения, которое будет использоваться в реальном мире. Для этого вам потребуется работать с командой над созданием нового приложения, которое ваши конечные пользователи сочтут полезным.

Что такое компьютерная наука? Информатика – это дисциплина, которая включает проектирование и понимание компьютеров и вычислительных процессов. Кроме того, изучение того, как обрабатываются данные, безопасность сетей, организацию баз данных, искусственный интеллект и так далее. Информатика использует абстрактный подход к компьютерам и компьютерному программированию. Она исследует, как работают компьютеры с точки зрения алгоритмов и вычислений, которые обрабатывают данные с помощью теорий.

Таким образом, мы видим, что программная инженерия, являясь областью информатики, ставит перед собой задачу создания программного обеспечения. В то время как информатика занимается вопросами компьютерных технологий и вычислительных процессов с использованием программного обеспечения.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ И КОМПЬЮТЕРНЫЕ НАУКИ

Причины для использования методов программной инженерии

- улучшает качество программных продуктов
- повышает производительность
- контролирует графики работы
- снижает стоимость разработки
- удовлетворяет требования клиентов
- поддерживает работу инженеров

Причины для использования информатики

- признана во всем мире за выдающиеся достижения
- помогает математически и логически изучить основы компьютера
- дает глубокие знания о функционировании компьютера



Рассмотрим основные причины, которые побуждают нас заниматься программной инженерией или же информатикой.

Вот причины для использования методов программной инженерии.

- Это позволяет улучшить качество программных продуктов.
- Это позволит повысить производительность и обеспечить удовлетворение работой профессионалов в области программного обеспечения.
- Это позволит вам контролировать графики работы программного обеспечения и эффективно планировать.
- Это помогает снизить стоимость разработки программного обеспечения.
- Это позволит удовлетворить потребности и требования клиентов.
- Это поможет вам систематически и эффективно поддерживать работу инженеров.
- Вот причины для использования информатики.
- Она признана во всем мире за выдающиеся достижения в области исследований, обучения и взаимодействия с промышленностью.
- Помогает вам изучить основы компьютера математически и логически.
- Дает глубокие знания о реальном функционировании компьютера.

Таким образом, мы видим разницу и в определении, и в причинах, которые побуждают нас заниматься либо компьютерной наукой, либо программной

инженерией. Программная инженерия – это исследование построения программных систем. Информатика – это исследование того, как компьютеры выполняют теоретические и математические задачи.

Вам следует выбрать программную инженерию, если вы хотите узнать общий жизненный цикл создания и поддержки конкретного программного обеспечения.

ПОПУЛЯРНОСТЬ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Большое программное
обеспечение

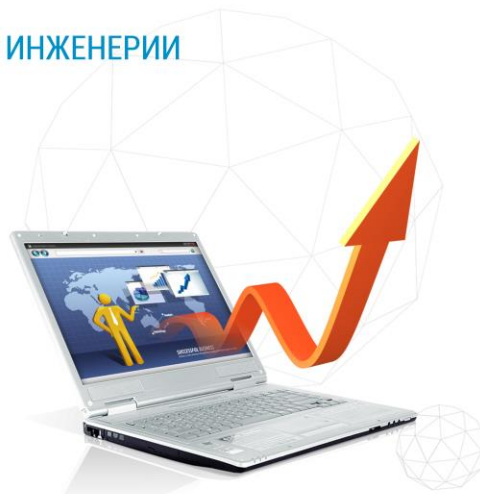
Масштабируемость

Адаптивность

Стоимость

Динамическая природа

Управление качеством



Почему же программная инженерия так популярна? Можно выделить несколько причин, которые определяют популярность программной инженерии.

Во-первых, большое программное обеспечение. В нашей реальной жизни построить стену намного удобнее, чем дом или здание. Когда размер программного обеспечения становится большим, лучше вести его разработку отдельными модулями.

Во-вторых, масштабируемость. Если бы процесс разработки программного обеспечения был основан на научных и инженерных концепциях, было бы легче воссоздать новое программное обеспечение для масштабирования существующего.

В-третьих, адаптивность. Всякий раз, когда процесс создания программного обеспечения основан на научных и инженерных принципах, нам легко воссоздать новое программное обеспечение с помощью программной инженерии.

В-четвертых, стоимость. Промышленность аппаратного обеспечения показала свои навыки. Огромное производство снизило стоимость компьютеров и электронного оборудования. Это повлекло за собой рассмотрение вопроса и стоимости программных продуктов.

В-пятых, динамическая природа, то есть постоянный рост и адаптация

программного обеспечения. Это зависит от среды, в которой работает пользователь.

И наконец, управление качеством: предлагает лучший метод разработки программного обеспечения для предоставления качественных программных продуктов.

Таким образом, мы определили основные причины, которые делают программную инженерию популярной и востребованной в век цифровых технологий, когда практически все основано на использовании программных продуктов.



А теперь рассмотрим, с чего начинается программная инженерия и что необходимо понять, чтобы научиться разрабатывать качественное программное обеспечение.

Программная инженерия обычно начинается с первого шага – инициирование пользовательского запроса для конкретной задачи или результата. Он подает свое требование в организацию – поставщик услуг.

Команда разработчиков программного обеспечения разделяет требования пользователей, системные требования и функциональные требования.

Требования собираются путем проведения интервью с пользователем, обращения к базе данных, изучения существующей системы и так далее.

После сбора требований команда анализирует, можно ли сделать программное обеспечение таким, чтобы оно соответствовало всем требованиям пользователя.

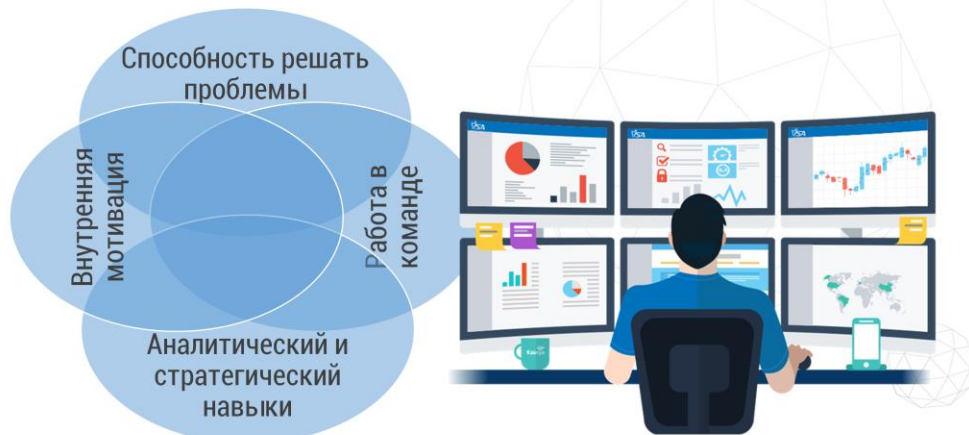
Затем разработчик определяет дорожную карту своего плана. Системный анализ также дает понимание ограничений программного продукта.

В соответствии с требованиями и анализом выполняется разработка программного обеспечения.

Реализация дизайна программного обеспечения начинается с написания программного кода на подходящем языке программирования.

Мы видим, что для того, чтобы научиться делать качественное программное обеспечение, нужно будет пройти большой путь – от понятия того, что нужно, до разработки программного продукта.

НАВЫКИ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Вот несколько рекомендаций для разработчиков программного обеспечения. Считается, что современные программные инженеры должны:

- действовать таким образом, чтобы это было выгодно как клиенту, так и работодателю.
- Создавать программные продукты и связанные с ними модификации, которые должны соответствовать самым высоким профессиональным стандартам. Это поможет сохранить честность и независимость в профессиональном подходе.
- Продвигать этический подход к разработке и обслуживанию программного обеспечения.

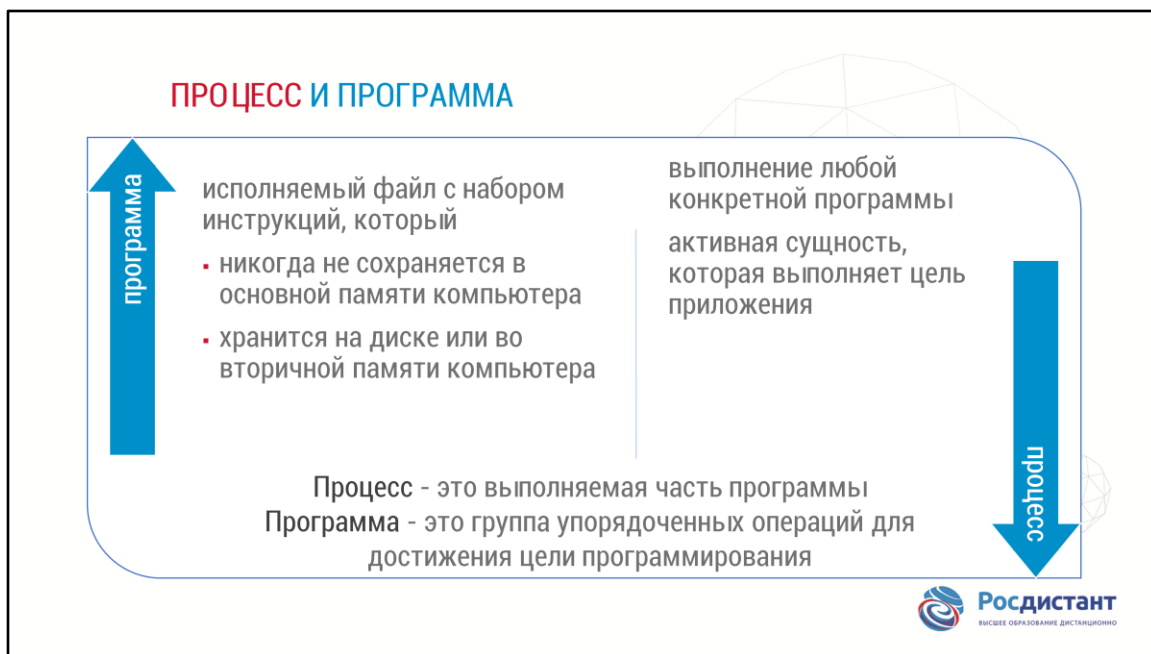
Разработчики программного обеспечения обладают не только техническими знаниями, но и определенными навыками, которые характеризуют их как профессионалов.

1. **Способность решать проблемы.** Вся суть разработки программного обеспечения заключается в создании программ, решающих проблему пользователя. Если вы из тех, кто склонен выявлять проблемы и выяснять, как лучше всего подходить к их решению, вы можете идеально подходить для карьеры разработчика программного обеспечения.
2. **Работа в команде.** Разработчики программного обеспечения проводят много

времени, объединяясь с другими разработчиками и программистами, чтобы создать лучший продукт для пользователя. Необходимы плодотворное общение, сочувствие и умение сотрудничать с другими.

3. Аналитический и стратегический навыки. Кодирование и программное обеспечение – это своего рода загадки, и если вы тот, кто любит разгадывать тайну, глядя на общую картину, а также на более мелкие шаги по пути, но вы сможете научиться разрабатывать программное обеспечение. Создание таких программ требует тщательного планирования.
4. Внутренняя мотивация. Создание программного обеспечения может быть сложной и утомительной работой, поэтому важно, чтобы вы не увязли и не разочаровались, если не добьетесь успеха с первого раза. Если вы относитесь к тому типу людей, которые по своей природе полны решимости завершить все, что начали, вы хорошо подойдете для карьеры в области разработки программного обеспечения.

Таким образом, мы определили психологические качества, которыми должны обладать разработчики. А теперь попробуем разобраться, что и как будем разрабатывать.



Для этого сначала разберемся в основных отличиях между процессом и программой.

Итак, программа – это исполняемый файл, который содержит определенный набор инструкций, написанных для выполнения определенного задания на вашем компьютере. Например, браузер Google chrome.exe – это исполняемый файл, в котором хранится набор написанных в нем инструкций, позволяющих просматривать веб-страницы.

Программы никогда не сохраняются в основной памяти компьютера, они хранятся на диске или во вторичной памяти компьютера. Они считываются из первичной памяти и выполняются ядром операционной системы.

А процесс – это выполнение любой конкретной программы. Считается активной сущностью, которая выполняет цель приложения. Несколько процессов могут быть связаны с одной и той же программой.

Например, если вы дважды щелкните значок браузера Google chrome на своем компьютере, вы запустите процесс, который в свою очередь запустит программу Google chrome. Когда вы открываете другой экземпляр chrome, вы, по сути, создаете два процесса.

Таким образом, подводим итог: процесс – это выполняемая часть программы, тогда как программа – это группа упорядоченных операций для достижения

цели программирования.

Процесс имеет минимальный срок жизни, тогда как программа имеет более длительный срок.

При выполнении процесса задействовано множество ресурсов, таких как адрес памяти, диск, принтер, в то время как программе требуется место в памяти на диске для хранения всех инструкций.

Процесс – это динамический или активный объект, тогда как программа – пассивный или статический объект.

Но нас интересуют понятия программного обеспечения, программного приложения, то есть тех объектов, созданием которых занимается программная инженерия.



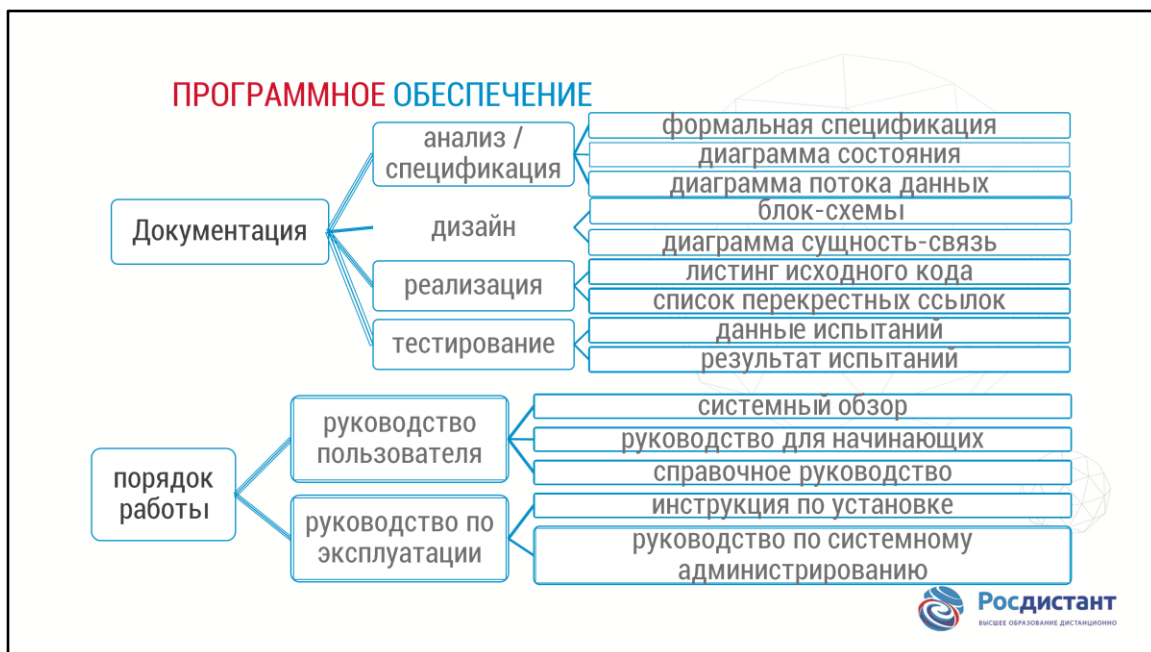
Итак, программное обеспечение – это набор инструкций, данных или программ, используемых для управления компьютерами и выполнения определенных задач. Программное обеспечение – это общий термин, используемый для обозначения приложений, сценариев и программ, которые выполняются на устройстве.

Программное обеспечение можно рассматривать как изменяемую часть компьютера, а аппаратное обеспечение – как неизменяемую часть.

Под прикладным программным обеспечением понимаются загружаемые пользователем программы, которые удовлетворяют его желание или потребность. Примеры приложений включают офисные пакеты, программы баз данных, веб-браузеры, текстовые процессоры, инструменты разработки программного обеспечения, редакторы изображений и коммуникационные платформы.

Многие люди приравнивают термин «программное обеспечение» к компьютерным программам. Программы разрабатываются отдельными лицами для личного пользования. Как правило, они имеют небольшой размер и ограниченную функциональность. Сам автор программы использует и поддерживает свою программу, она обычно не имеет хорошего пользовательского интерфейса и надлежащей документации. В то время как

программные продукты, программные приложения имеют несколько пользователей и, следовательно, должны иметь хороший пользовательский интерфейс, надлежащие рабочие процедуры и хорошую документацию. Программное обеспечение – это программа плюс хороший пользовательский интерфейс, плюс рабочие процедуры и плюс документация. Программный продукт должен быть правильно спроектирован, тщательно реализован и должным образом протестирован.



Любая программа является подмножеством программного обеспечения и становится программным обеспечением только в том случае, если подготовлены документация и руководства по эксплуатации. Программа состоит из набора инструкций, который представляет собой комбинацию исходного кода и объектного кода.

Документация состоит из различных видов руководств.

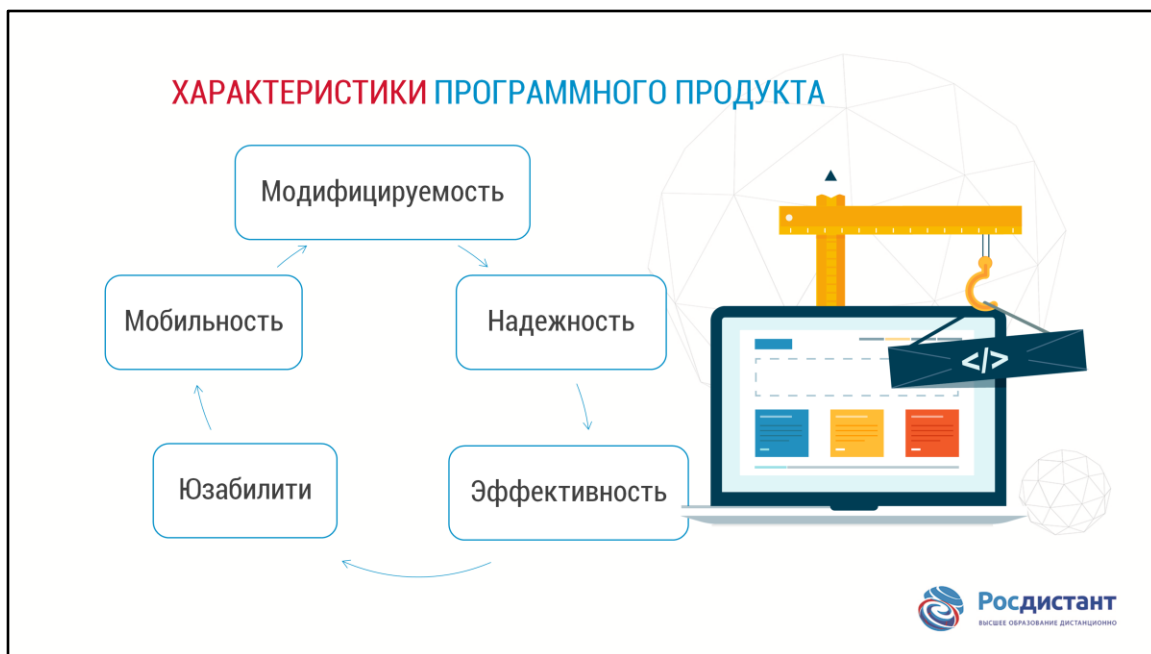
- Спецификации, которые включают описание формальной спецификации, диаграмму состояния, диаграмму потока данных.
- Дизайн – здесь обязательно нужно подготовить блок-схемы, диаграммы сущность – связь.
- Реализация – включает листинг исходного кода и список перекрестных ссылок.
- Тестирование – описывает данные испытаний и результат испытаний.

С данными документами вы будете знакомиться в процессе изучения этого курса и других учебных дисциплин.

Если рассмотреть документы, которые нужно описать для рабочих процедур, то они состоят из инструкций по настройке и использованию системы и инструкций по устранению сбоя системы.

Обычно программные продукты слишком велики, и их не может разработать

один программист. Поэтому программные продукты разрабатываются группой инженеров-программистов. Инженеры-программисты занимаются разработкой программного продукта, а программисты – написанием программ. Разработка программного продукта требует четко определенной и систематической методологии, без которой команда разработчиков программного обеспечения не смогла бы создать соответствующий программный продукт.



Мы кратко рассмотрели понятие программного продукта, а теперь попробуем определить основные его характеристики.

Характеристики любого программного продукта включают функции, которые отображаются в продукте при его установке и использовании. При этом мы должны понимать, что это не услуги, которые предоставляет продукт.

Характеристики неразрывно связаны с динамическим поведением продуктов и использованием продукта.

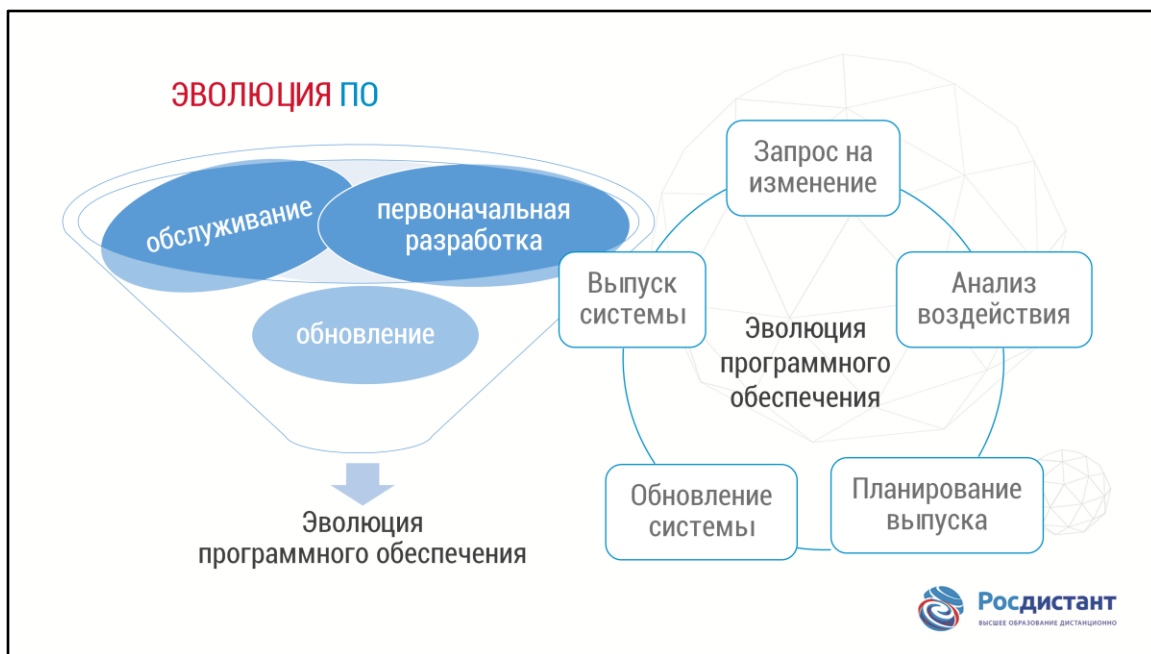
О любом программном обеспечении следует судить по тому, что оно предлагает и какие методы помогают вам его использовать. К основным характеристикам мы отнесем эффективность, надежность, модифицируемость и другие. Однако относительная важность этих характеристик варьируется от одной программной системы к другой.

Рассмотрим основные характеристики программного продукта. Наибольшее внимание уделяют следующим.

- **Модифицируемость.** Программное обеспечение должно развиваться, чтобы соответствовать меняющимся требованиям клиентов.
- **Надежность** включает различные характеристики. Надежное программное обеспечение никогда не должно причинять физического или экономического ущерба во время сбоя системы.

- Эффективность. Программное обеспечение должно чрезмерно использовать системные ресурсы, такие как память и цикл процессора.
- Юзабилити. Программное приложение должно иметь специальный интерфейс и документацию.
- Мобильность означает независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и тому подобное.

Придерживаясь выделенных критериев, можно рассчитывать на качественный программный продукт.



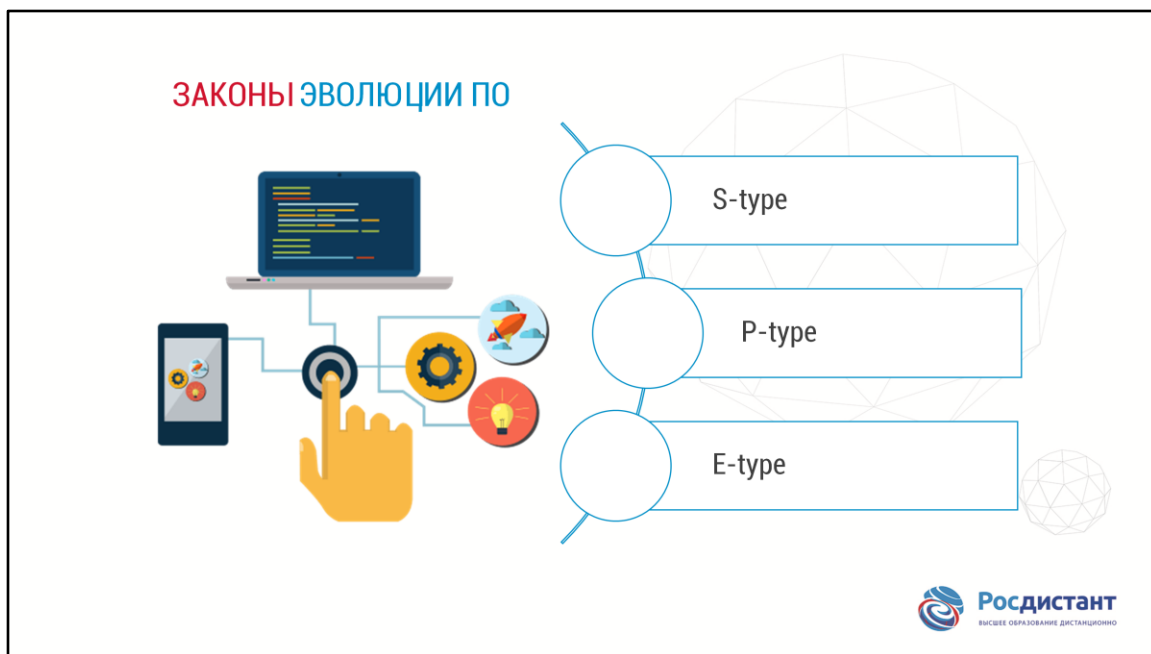
Процесс разработки программного продукта с использованием принципов и методов программной инженерии называется эволюцией программного обеспечения. Это включает первоначальную разработку программного обеспечения, его обслуживание и обновление до тех пор, пока не будет разработан желаемый программный продукт, удовлетворяющий ожидаемым требованиям.

Эволюция начинается с процесса сбора требований. После чего разработчики создают прототип предполагаемого программного обеспечения и демонстрируют его пользователям, чтобы получить их отзывы на ранней стадии разработки программного продукта. Пользователи предлагают свои изменения. Этот процесс продолжается до тех пор, пока желаемое программное обеспечение не будет выполнено.

Даже после того, как у пользователя есть желаемое программное обеспечение, прогрессивные технологии и меняющиеся требования вынуждают программный продукт меняться соответствующим образом. При воссоздании программного обеспечения с нуля точное выполнение требований нецелесообразно.

Единственное осуществимое и экономичное решение – обновить существующее программное обеспечение, чтобы оно соответствовало последним требованиям.

Данный процесс может включать несколько итераций, до тех пор, пока не будет создан тот продукт, который удовлетворяет требованиям пользователя – заказчика.



Законы эволюции программного обеспечения

Мейр Мэнни Леман сформулировал законы эволюции программного обеспечения. Он разделил программное обеспечение на три разные категории.

- S-type – статический тип – это программное обеспечение, которое работает строго по определенным спецификациям и решениям. Решение и метод его достижения сразу понимаются перед написанием кода. Программное обеспечение s-типа меньше всего подвержено изменениям, поэтому оно является самым простым из всех. Например, программа-калькулятор для математических вычислений.
- P-type – практический тип – это программное обеспечение с набором процедур. Это определяется тем, что именно могут делать процедуры. В этом программном обеспечении можно описать спецификации, но решение не сразу становится очевидным. Например, игровой софт.
- E-type – встроенный тип – это программное обеспечение работает в соответствии с требованиями реальной среды. Оно имеет высокую степень развития, поскольку в реальных ситуациях происходят различные изменения в законах, налогах и т. д. Например, программное обеспечение для онлайн-торговли.

Законы описывают баланс между силами, движущими новые разработки, с

одной стороны, и силами, замедляющими прогресс, с другой. Однако, имея дело с линиями программных продуктов, нам необходимо иметь дело с модификацией нескольких продуктов, которые включают общие, переменные и специфические для продукта активы.

ЭВОЛЮЦИЯ ПО E-TYPE

Постоянные изменения	система E-типа адаптирована к изменениям реального мира
Возрастающая сложность	увеличение сложности программы по мере ее развития
Сохранение осведомленности	демонстрация новых возможностей программного обеспечения
Непрерывный рост	добавление новых функций в программный продукт
Снижение качества	ухудшение качества продукта без поддержки
Обратная связь	добавление системы многоуровневой обратной связи
Саморегулирование	самоконтроль процессов эволюции систем
Организационная стабильность	повышенная активность на протяжении всего срока службы продукта



Мэйр Мэнни Лёман ввел восемь законов эволюции программного обеспечения E-type. Он считал, что эти законы, вероятно, будут верны для всех типов крупных организационных программных систем.

Первый закон гласит, что обслуживание системы – это неизбежный процесс. По мере изменения системной среды возникают новые требования, и систему необходимо модифицировать.

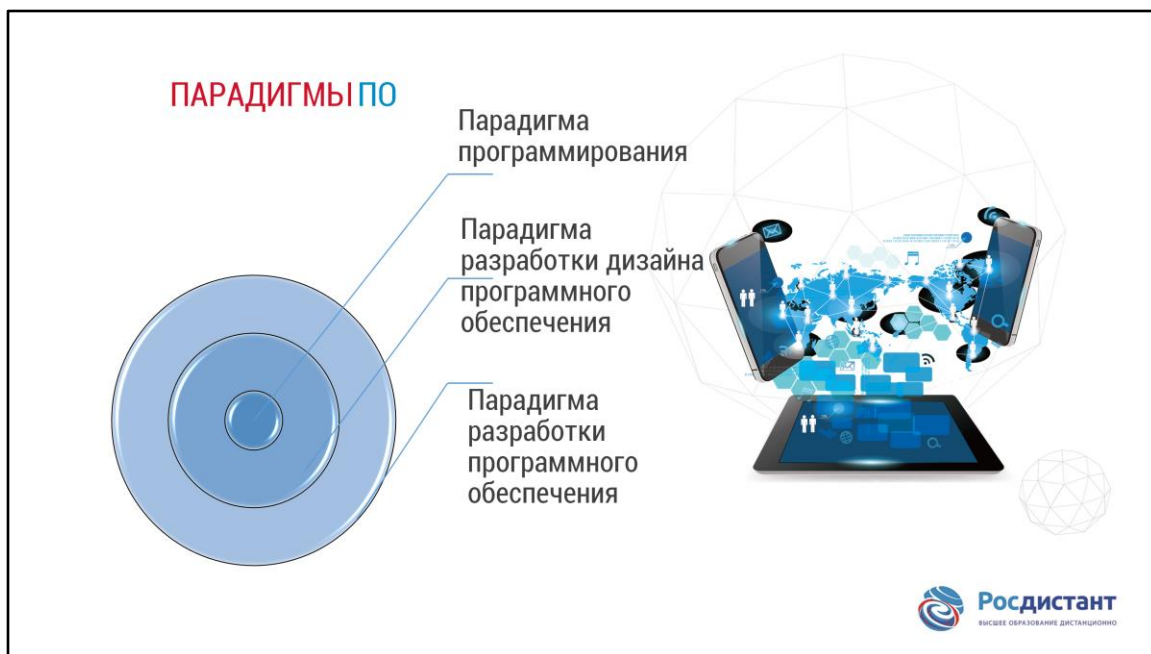
Второй закон говорит о том, что при изменении системы ее структура деградирует. Единственный способ избежать этого – инвестировать в профилактическое обслуживание.

Третий закон предполагает, что у систем есть собственная динамика, которая устанавливается на ранней стадии процесса разработки. Это определяет общие тенденции процесса обслуживания системы и ограничивает количество возможных изменений системы. По мере того, как вы изменяете и расширяете программу, ее структура имеет тенденцию к ухудшению. Внесение небольших изменений снижает степень структурной деградации и, таким образом, снижает риски возникновения серьезных проблем с надежностью системы.

Четвертый закон устанавливает, что большинство крупных программных проектов работает в «насыщенном» состоянии. То есть изменение ресурсов или персонала оказывает незаметное влияние на долгосрочное развитие системы.

Пятый закон касается приращения изменений в каждой версии системы. Добавление новой функциональности в систему неизбежно приводит к новым системным сбоям. Чем больше функциональных возможностей добавляется в каждом выпуске, тем больше ошибок будет. Следовательно, значительное увеличение функциональности означает, что за ним должен последовать следующий выпуск, в котором будут устранены новые системные сбои. Шестой и седьмой законы схожи и, по сути, говорят, что пользователи программного обеспечения будут становиться все более недовольными им, если оно не будет поддерживаться и не будет добавляться новых функций. Восьмой закон отражает самую последнюю работу по процессам обратной связи, хотя пока не ясно, как это можно применить в практической разработке программного обеспечения.

В программном обеспечении для персональных компьютеров мы часто наблюдаем, что за новой важной версией приложения следует обновление для исправления ошибок.



Парадигмы программного обеспечения относятся к методам и шагам, которые предпринимаются при разработке программного обеспечения. Предлагается множество методов, которые работают сегодня, но нам нужно увидеть, на каком этапе разработки программного обеспечения находятся эти парадигмы. Их можно объединить в различные категории, хотя каждая из них содержится одна в другой.

Парадигма программирования – это подмножество парадигмы проектирования программного обеспечения, которое также является подмножеством парадигмы разработки программного обеспечения.

Парадигма разработки программного обеспечения известна как парадигма программной инженерии, в которой применяются все инженерные концепции, относящиеся к разработке программного обеспечения. Она включает:

- сбор требований;
- разработку программного обеспечения;
- программирование.

Парадигма дизайна программного обеспечения является частью разработки программного обеспечения и включает:

- дизайн,
- обслуживание,

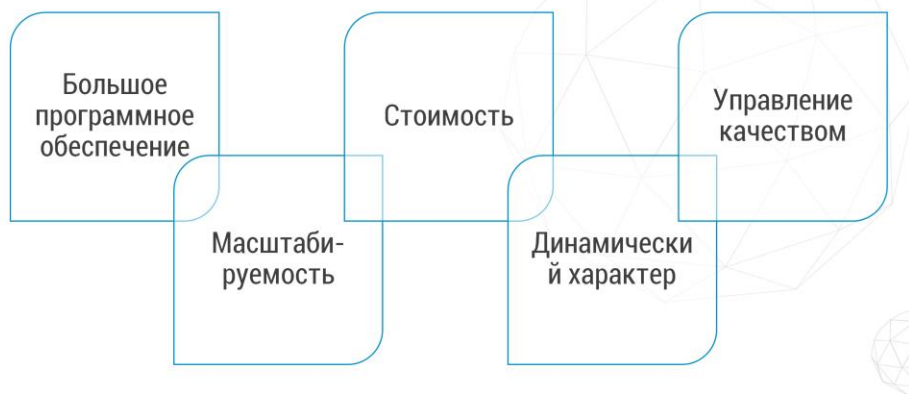
- программирование.

Парадигма программирования тесно связана с программным аспектом разработки программного обеспечения и включает:

- кодирование,
- тестирование,
- интеграцию.

Таким образом, мы рассмотрели основные парадигмы и сделали вывод, что все они являются частью разработки программного обеспечения.

ПОТРЕБНОСТЬ В РАЗРАБОТКЕ ПО



Потребность в разработке программного обеспечения возникает из-за более высокой скорости изменения требований пользователей и среды, в которой работает программное обеспечение. Можно выделить несколько ключевых моментов, которые определяют потребность в создании программного продукта. К ним относятся следующие.

- Масштабируемость. Если бы программный процесс не основывался на научных и инженерных концепциях, было бы легче воссоздать новое программное обеспечение, чем масштабировать существующее.
- Большое программное обеспечение.
- Стоимость остается высокой, если не адаптировать соответствующие процессы.
- Динамический характер. Постоянно растущий и адаптирующийся характер программного обеспечения во многом зависит от среды, в которой работает пользователь. Если природа программного обеспечения постоянно меняется, необходимо внести новые улучшения в существующее. Именно здесь программная инженерия играет важную роль.
- Управление качеством – лучший процесс разработки программного обеспечения сделает возможным лучший и качественный программный продукт.

Таким образом, мы рассмотрели те моменты, которые показывают нам необходимость разработки программного обеспечения.

ПРИНЦИПЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ



В программной инженерии для творческого решения проблем применяется систематический процесс, который включает следующие шаги:

- понять проблему;
- спланировать решение;
- осуществить этот план;
- проверить свои результаты на точность.

Таким образом, сценарий очень простой, а значит, нужны простые и ясные практические правила. Фактически все, что нужно знать о разработке программного обеспечения, можно резюмировать с помощью принципов, правил и сокращений. В приведенном ниже списке они начинаются с высокого уровня, а к концу становятся более специфичными для кода.

Наиболее важные из них, которые нужно усвоить, следующие.

1. Стартовый высокий уровень – вы сможете вернуться позже и обдумать некоторые из программ, более специфичные для программного обеспечения, когда у вас будет немного больше времени.
2. Размышления о процессе – рассматриваются проблемы анализа самого процесса. Нужно оценить и понять, что будет являться решением.
3. Размышления о кодировании решения. Само название говорит, что будут рассматриваться некоторые проблемы, связанные с переводом решения в

программный код.

4. Эмпирические правила. Мы понимаем, что обнаружение и устранение проблемы программного обеспечения в производственной среде в 100 раз дороже, чем поиск и устранение ее на этапе определения требований и проектирования.

Почти все время, потраченное на устранение проблем, можно отнести к небольшому количеству модулей проблем.

СТАРТОВЫЙ ВЫСОКИЙ УРОВЕНЬ

Четкое определение проблемы

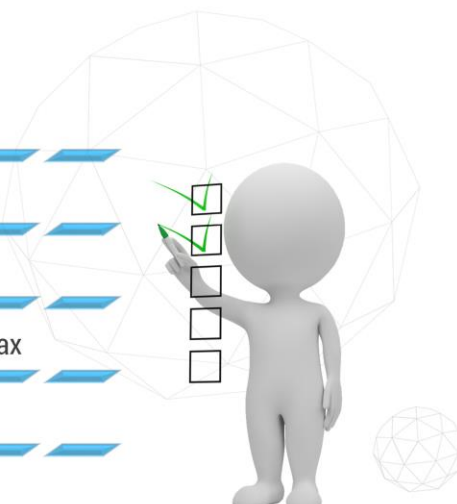
Разделяй и властвуй

KISS

Обучение, особенно на своих ошибках

Охват общей концепции решения

Создание не для себя



К стартовому высокому уровню отнесем следующие принципы и методы, которые позволят рассчитывать на разработку качественного программного обеспечения.

- Полностью продумайте проблему, прежде чем пытаться реализовать решение. Это самое важное, что вы можете сделать.
- Разделяй и властвуй. Разбейте глобальные проблемы на несколько небольших, над которыми можно работать независимо. Вот так серьезные проблемы становятся управляемыми.
- KISS – может быть девизом инженера. В мире существует тенденция создания излишне сложных систем, и это заставляет признать, что простые решения *намного лучше сложных*. Это не значит, что простота – это просто!
- Учитесь, особенно на своих ошибках – инженерия постоянно меняется, и разработка программного обеспечения, в частности, может быть одной из самых быстро меняющихся областей. Вы должны всегда как учиться у других людей, так и признавать собственные ошибки.
- Помните, почему существует программное обеспечение. Есть бесчисленное множество мелких решений, с которыми вы столкнетесь во время работы в качестве разработчика. И если вы не знаете общей картины, вы можете потратить время на неправильный путь. Это в значительной степени

основополагающий принцип.

- И наконец, помните, что Вы не будете использовать программное обеспечение! Всегда есть соблазн создавать решения для себя. Думайте о пользователе, а не о себе. Не думайте, что ваш пользователь будет иметь такой же уровень технической компетенции или информированности о программном продукте, как и вы.

Это небольшой перечень правил, которые вы всегда должны помнить при разработке программного обеспечения.

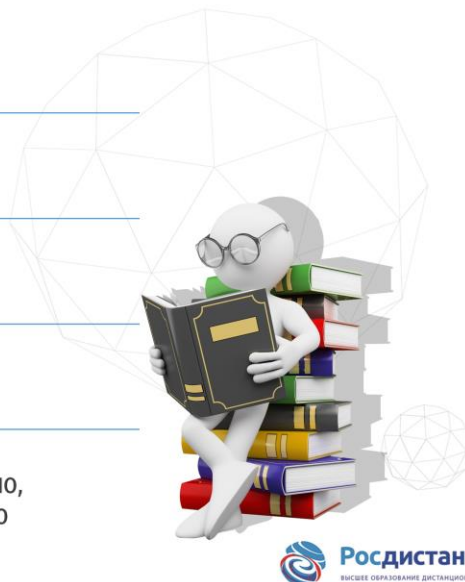
РАЗМЫШЛЕНИЯ О ПРОЦЕССЕ

Четкое видение проекта

Строгий процесс

Быстрая разработка приложений

Правило: сначала работать,
затем работать правильно,
затем выглядеть красиво



Ко второй группе «Размышления о процессе» мы отнесем следующие принципы, которых вы должны придерживаться при разработке программных продуктов.

- Имейте четкое видение проекта. Если вы точно не знаете, что пытаетесь построить, вы создадите не то. Необходимо задавать важные вопросы клиентам, чтобы не получилось так: «Вы построили именно то, что мы просили, но не то, что нам нужно».
- Иметь точный процесс. Разработка программного обеспечения – это творческая деятельность в области дизайна, но она должна выполняться систематически. При решении сложных проблем нужно помнить о логическом и вдумчивом подходе к их выполнению и управлению своими проектами.
- Быстрая разработка приложений – здесь стоит отметить, что требования к программному проекту постоянно меняются. Чем быстрее будет работать процесс, тем лучше вы сможете отреагировать на эти изменения.
- Заставьте программное приложение сначала РАБОТАТЬ, затем работать ПРАВИЛЬНО, а затем выглядеть КРАСИВО. Это адаптация быстрой разработки приложений, которая подчеркивает философию «пробуй сначала». Вы должны сначала построить что-то, что работает, а затем посмотреть, стоит ли

тратить время на то, чтобы все работало правильно. Только на заключительной фазе процесса вы действительно должны сделать свое решение хорошо выглядящим, например, рефакторинг кода. Таким образом, это еще одна группа требований, на которую нужно обратить внимание при разработке программного продукта.

РАЗМЫШЛЕНИЯ О КОДИРОВАНИИ РЕШЕНИЯ

YAGNI (You Aren't Gonna Need It)

DRY (Don't Repeat Yourself)

Абстракция

DRITA (Не изобретайте колесо заново)

Код, который хорошо выполняет одну задачу

Отладка сложнее написания кода

Кайдзен



К третьей группе «Размышления о кодировании решения» мы отнесем следующие принципы, которых вы должны придерживаться при разработке программных продуктов.

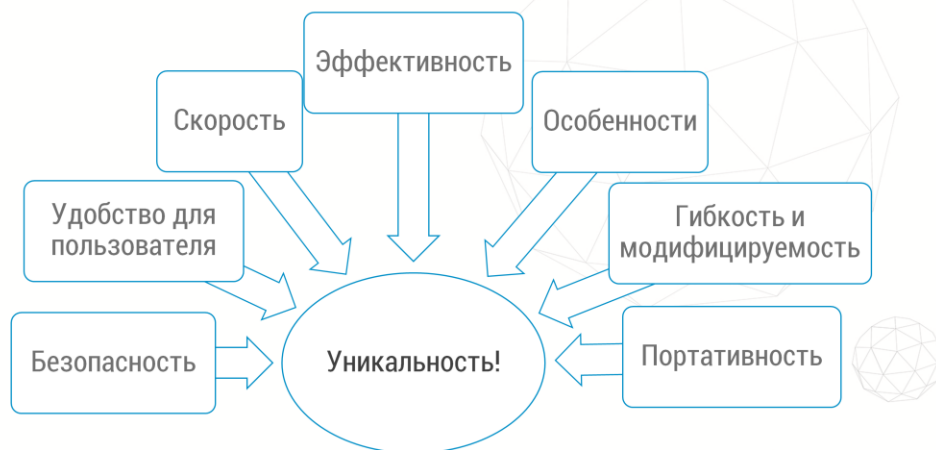
- YAGNI – вам это не понадобится! Существует действительно сильное искушение создать код, который мог бы реагировать на любые будущие события, будучи невероятно гибким и совершенным. Не делайте этого! Вы тратите силы зря, потому что вам действительно не понадобятся все эти дополнительные функции, опции или гибкости. Просто создайте то, что вам нужно.
- DRY . Одна из лучших особенностей кода – это то, насколько он пригоден для повторного использования. Если вы пишете классный фрагмент кода, который решает полезную проблему в одном месте, вернитесь к нему, когда проблема возникнет и в других местах.
- Примите абстракцию. Разработка программного обеспечения – это абстракция, игнорирование деталей и решение проблем более высокого порядка. Вам не нужно писать машинный код или ассемблерный код по какой-либо причине – современные языки программирования позволяют вам просто сказать компьютеру, что вы хотите, и он выполнит. Сосредоточьтесь на том, чтобы убедиться, что программа функционирует должным образом, без

необходимости знать детали реализации каждой составляющей части.

- DRITA – не изобретайте колесо заново. Каждый раз, когда вы создаете код для выполнения чего-то общего, не имеющего прямого отношения к основам вашего приложения, кто-то, вероятно, уже написал этот код и лучше. Учитесь на их коде и используйте его вместо того, чтобы тратить время на изобретение колеса.
- Напишите код, который хорошо выполняет одну задачу. Единичный фрагмент кода должен делать только одно и делать это хорошо.
- Отладка сложнее написания кода. Читаемый код лучше компактного.
- Кайдзен – оставьте его лучше, чем, когда вы его нашли. Исправьте не только ошибку, которую вы пытаетесь решить, но и код, связанный с ней.

Мы рассмотрели основные принципы, на которые надо обратить внимание, прежде чем начинать писать программное приложение.

ХАРАКТЕРИСТИКИ УНИКАЛЬНОСТИ ПРОГРАММНОГО ПРОДУКТА



На рынке представлены тысячи продуктов для различных отраслей. Хотя большинство разработчиков создают отличные работающие программные продукты, многие, кажется, забывают о других аспектах, которые могут сделать продукт действительно качественным. Элементы хорошего программного обеспечения имеют такое же значение, как и его продажи. Следовательно, отличный программный продукт должен содержать базовые характеристики, заложенные в основу существования вашего программного обеспечения. Вы можете быть разработчиком программного обеспечения, стремящимся усовершенствовать свой продукт, или вы заинтересованы в программном обеспечении, но не уверены, будет ли оно удобным для пользователя. Просто взгляните на следующие характеристики, которые сделают ваш программный продукт уникальным среди остальных, и определите, какие из них все еще отсутствуют в вашем.

- Безопасность;
- удобство для пользователя;
- скорость;
- эффективность;
- особенности;
- гибкость и модифицируемость;

- портативность.

Разработчики программного обеспечения лучше разбираются в дизайне, и это находит отражение в характеристиках программного продукта. Итак, убедитесь, что у вас есть ресурсы самого высокого качества для разработки вашего программного обеспечения, которое станет уникальным продуктом на рынке.

БЕЗОПАСНОСТЬ И УДОБСТВО ДЛЯ ПОЛЬЗОВАТЕЛЯ

Безопасность	Использование передовых методов обеспечения информационной безопасности
	Отсутствие отрицательного воздействия на данные или оборудование
	Помощь в организации защиты данных от внешних угроз
Удобство для пользователя	Дружественный интерфейс
	Удобная навигация
	Эргономические требования



Рассмотрим более подробно выделенные характеристики.

Безопасность – любое программное обеспечение должно быть создано с использованием передовых методов обеспечения безопасности. На протяжении многих лет мы видели, как какое-то программное обеспечение выполняло свою работу, но функции безопасности сильно подвели продукт. Именно безопасность оказывается наиболее важной, когда речь идет о хорошем программном продукте. И с увеличением числа угроз безопасности в настоящее время любое программное обеспечение должно учитывать этот фактор. Кроме того, программное обеспечение не должно оказывать отрицательного воздействия на данные или оборудование. Программное обеспечение должно помочь пользователю принимать надлежащие меры для защиты своих данных от внешних угроз. Поскольку обеспечение безопасности должно быть постоянным процессом в программном обеспечении, то проверка на наличие проблем и сбоев должна быть регулярной. Безопасное программное обеспечение должно защищать данные пользователя от раскрытия, изменения и уничтожения.

Удобство для пользователя. Дружественное к пользователю программное обеспечение означает, насколько оно простое в использовании. Навигация очень важна, поскольку она определяет путь человека при использовании

программного обеспечения. Удобство использования хорошего программного обеспечения можно определить через пользовательский интерфейс. Любое программное обеспечение с необычным дизайном UI , пользовательским интерфейсом – это продукт, который будет пользоваться большим вниманием, чем тот, у которого этого нет.

Все программное обеспечение должно быть удобным для пользователя; в противном случае пользователи не смогут достаточно легко перемещаться, чтобы использовать некоторые функции, предлагаемые программным обеспечением. Количество усилий или времени, необходимых для изучения того, как использовать программное обеспечение, должно быть небольшим. Конечная цель разработки программного обеспечения заключается в том, чтобы оно было удобным даже для несведущих в области информационных технологий людей.



Следующая характеристика – это скорость. Ваше программное обеспечение загружается бесконечно? Если да, значит, оно не было разработано должным образом. Наряду со всеми последними стандартами кодирования, функциями и библиотеками, к которым имеет доступ разработчик программного обеспечения, нетрудно создать адаптивное программное обеспечение.

Существует множество программ, которые могут предлагать те же функции и при этом проявлять инициативу в ответах и реакциях.

Превосходное программное обеспечение должно быть быстрым как в функциональности, так и в интерфейсе. В выполняемых действиях необходима легкость. Хорошее программное обеспечение с минимальным отставанием или минимальным запаздыванием по-прежнему будет казаться более быстрым, чем большинство других программ на рынке. Быстрое программное обеспечение демонстрирует сосредоточенность и решимость выполнять задачи в соответствии с требованиями.

Следующее – это эффективность, которая важна для успеха любого программного обеспечения. Оно не только должно выполнять ту работу, для которой было создано, но также должно иметь возможность предлагать отличные функции, которые могут помочь пользователям быстро выполнять задачи. Эффективность в основном связана с тем, как программное

обеспечение использует доступные ресурсы.

Программное обеспечение должно быть в состоянии удовлетворить потребности пользователя в нем, которые позволяют эффективно его использовать. На рынке представлено множество программных продуктов, но лишь немногие из них наиболее востребованы.

ХАРАКТЕРИСТИКИ ПРОГРАММНОГО ПРОДУКТА

Особенности программного продукта

- Функции – то, ради чего создается ПО
- Продукт должен иметь четкое восприятие по внешнему виду, компонентам и возможностям

Гибкость и модифицируемость

- Редактируемость ПО
- Корректировка ошибок

Переносимость программного обеспечения

- Кроссплатформенность в различных конфигурациях



Особое внимание уделяем особенностям программного продукта. Функции, которые предлагает программное обеспечение, являются основной целью его существования. Продукт должен быть четко определен по внешнему виду, компонентам и возможностям.

Немаловажной характеристикой являются гибкость и модифицируемость. Гибкость программного обеспечения определяется тем, насколько легко вы можете добавлять, изменять или удалять функции программного обеспечения, не мешая текущей работе. Изменения в разработке программного обеспечения неизбежны; это может произойти во время самого процесса разработки или может потребовать корректировок в соответствии с будущими требованиями. Модифицируемость в чем-то похожа на гибкость. В то время как гибкость ориентирована на основные функциональные возможности расширения, модифицируемость больше связана с исправлением ошибки и незначительными изменениями. Вот почему важна надлежащая документация, которая включает определения интерфейсов, документацию по коду и многое другое.

Важна также переносимость программного обеспечения. Переносимость программного обеспечения – это удобство использования программного обеспечения в различных конфигурациях.

Признаком отличного программного обеспечения является то, что оно может быть реализовано на трех или более различных конфигурациях оборудования в течение срока его службы.

Кроме того, вы должны знать, что перенос программного обеспечения в новую среду сравнительно дешевле, чем разработка аналогичного программного обеспечения с нуля. Можно с уверенностью сказать, что мобильность является решающим фактором, когда речь идет о сокращении затрат на разработку.

Таким образом, мы рассмотрели общее представление о вопросах, которыми занимается программная инженерия. Мы определили спектр признаков хорошей разработки и качественного программного обеспечения.