

Тема 6. Этапы разработки программных продуктов

- 1. Разработка программного обеспечения как бизнес-процесс**
- 2. Инфраструктура ИТ-отделов, команд, роли сотрудников**
- 3. Основные типы ПО, специфика разработки, стоимость**
- 4. Средства, сервисы и продукты оптимизации процесса разработки ПО**
- 5. Услуги аутсорсинга и консалтинга в ИТ**

Оценить всю ту пользу, которую принесли ИТ в современный бизнес и его процессы невозможно, ведь на сегодняшний день уже практически нельзя представить в себе успешный бизнес без использования ИТ-автоматизации, а если таковые и имеются, то сильно уступают своим «сотоварищам», бизнес-процессы которых автоматизированы средствами ИТ. Помимо автоматизации процессов, ИТ делает свой вклад и в экономическую составляющую производства, ведь информационные системы способны заменить работу сотен и даже тысяч других специалистов, что приводит к повышению рентабельности.



Отдельную часть ИТ занимает такое понятие, имеющее название как **«программное обеспечение»**, именуемое как «software», а сам процесс его создания **«разработка программного обеспечения»**.

Именно программное обеспечение (далее ПО) является самым дорогим продуктом в ИТ и его разработка.

На сегодняшний момент **разработка ПО** - это совокупность мероприятий по выполнению реализации задачи, стоящей перед **программным обеспечением**.

Вопрос №1: Разработка программного обеспечения как бизнес-процесс

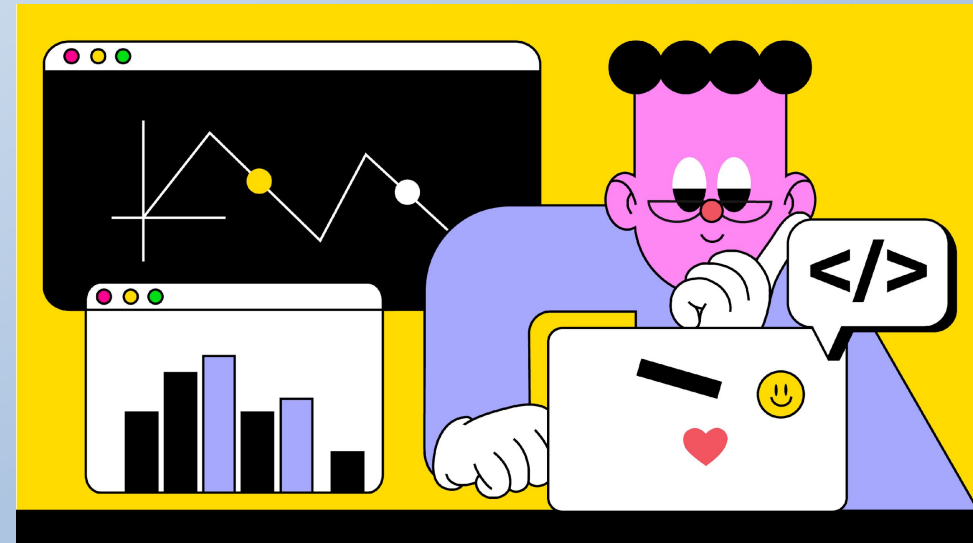


На сегодняшний день ПО имеет длинный список своих типов. Например основные типы ПО выглядят приблизительно следующим образом:

- **системное** (работа с аппаратной частью),
- **Прикладное** (не работает с аппаратной частью ЭВМ, лишь использует ее API),
- **Web** (работа с механизмами, присутствующими в браузерах),
- **сетевое** (обработка сетевых данных, получение сведений о сети) и др.

Разработка ПО требует не только знания специфики разработки программного продукта для того или иного типа приложения и стоящими перед ним задачи, но и изучения предметной области системы. Т.о. в процессе разработки ПО **востребованы не только ИТ-специалисты, но и аналитики.**

Вопрос №1. Разработка программного обеспечения как бизнес-процесс



Современная **разработка ПО** — это своеобразный комплекс мероприятий, проводимый над реализуемым решением, все задачи которого в большинстве случаев просто невозможно реализовать интеллектуальным потенциалом одного специалиста.

В виду этого в разработку ПО вводится такое понятие как **«команда»**.

В коллектив ИТ-команды входят не только ИТ-специалисты, но такие специалисты как дизайнеры, экономисты, аналитики.

В дополнение ко всему, временно в коллектив команды может вливаться специалист, владеющими знаниями предметной области решения.

В зависимости от типа ПО, которое разрабатывает ИТ-команда, ее состав типов специалистов может меняться. К примеру, разработка ПО с применением технологий пользовательского графического интерфейса GUI (graphical user interface) он же UI (user interface) требует наличия в команде графического дизайнера, разработка ПО для межсетевого взаимодействия требует в команде сетевого инженера, разработка web-приложения требует в команде помимо дизайнера еще и верстальщика.

Любая серьезная организация, занимающихся разработкой ПО имеет в своем составе самый большой отдел — **ИТ-отдел**, который в свою очередь уже делится на подотделы. К примеру: отдел web-разработок, отдел мобильных разработок, отдел сетевых систем и т.д. Отделы уже делятся на команды. В основном одна команда работает над одним проектом, но бывают и случаи, когда одна проект делится на части и одна команда реализует одну часть, а другая - вторую.



У каждой команды есть **тимлид (Team Lead)**, он же **техлид** (технический лидер), он же руководитель проекта. За весь проект абсолютно и полностью отвечает тимлид. Тимлид также отвечает за свою команду: за отпуск сотрудников, за организацию обучения состава команды и т.д.



Разработчики в составе команды по уровню навыков делятся на:

- **senior developer,**
- **middle developer,**
- **junior developer.**



Junior developer — младший разработчик, который зачастую допускает больше ошибок остальных и задает больше всех вопросов. В позиции junior в основном находятся начинающие разработчики, имеющие мало опыта и только что устроившиеся на работу после ВУЗа. Как правило заработная плата у junior-а самая низкая из разработчиков. Junior не разрабатывает основную логику ядра и не проектирует архитектуру системы, а в основном работает с отдельными модулями или UI-прослойкой системы. На позиции junior в основном стоят не долго, переход от позиции junior к позиции middle происходит по усмотрению тимлида, который решил что вы уже достаточно опытни и не допускаете критических ошибок.



Middle developer — разработчик среднего уровня опыта, знаний и навыков. Middle достаточными знаниями платформ разработки и информационных систем. В позиции middle реализовывают логику среднего уровня, в некоторых случаях работают с ядром и архитектурой системы. Middle не задает «странных» вопросов тимлиду и допускает допустимое количество ошибок при разработке. Переход от позиции middle к позиции senior обычно происходит через длительный промежуток времени (от 2 лет), это обосновано достаточно высоким требованиям к разработчикам в позиции senior.

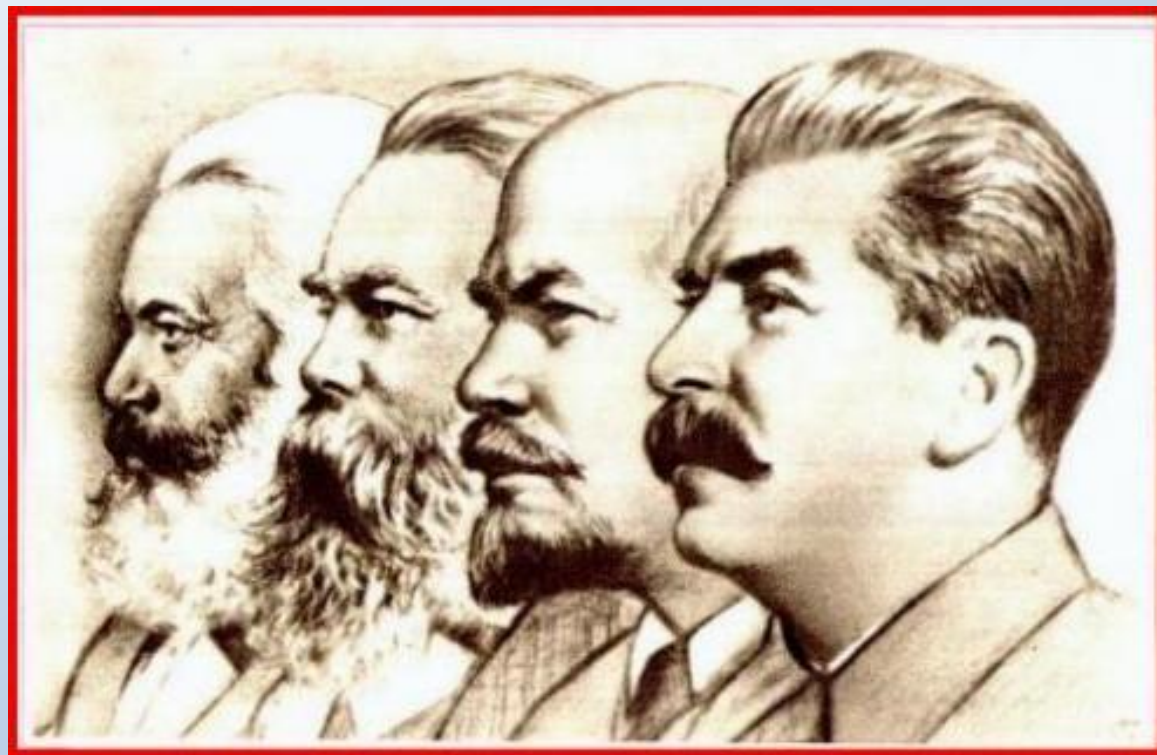


Senior developer — разработчик высшего уровня, который практически не допускает ошибок, имеет сильные знания о процессах разработки, платформах, языках программирования, технологиях и их средствах.

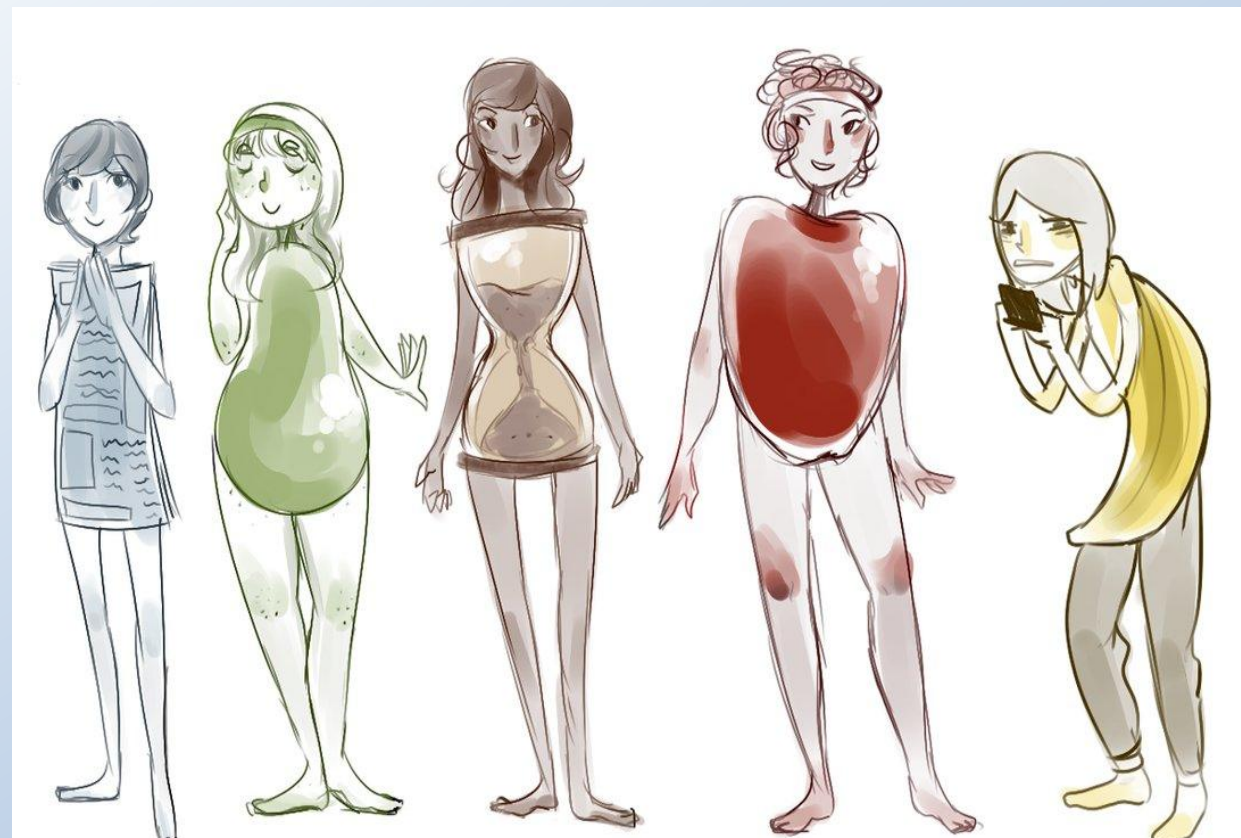
Senior работает с ядром и архитектурой, реализовывает программные абстракции, проектирует решение совместно с тимлидом. Senior-у не нужно указывать как разрабатывать ПО, так как он уже обладает достаточным уровнем для самостоятельного принятия решений. Следующим шагом от Senior-а является позиция тимлида, переход к которой не имеет общего промежутка времени. В некоторых случаях, senior и является тимлидом.



Следующие позиции, находящиеся выше тимлида — руководитель ИТ-отдела и технический директор. Исходя из этого можно сказать, что инфраструктура ИТ-отделов имеет подчиненную иерархическую структуру.



ПО решает разные задачи, начиная от включения электрочайника и заканчивая искусственным интеллектом. То есть специфика разработки ПО тоже абсолютно разная. По характеру разработки и требованиям к разработчикам, можно выделить основные типы ПО: *системное, прикладное, web-приложения, приложения хранилищ данных, ПО для big data, сетевое, мобильное, облачное.*



Системное ПО — решения, основная задача которых обеспечить интерфейсы взаимодействия между аппаратной частью устройства с прикладной. Зачастую такая реализация выглядит в виде API (application programming interface, интерфейс программирования приложений). Для разработки ПО этого типа, разработчику необходимо знать низкоуровневые языки программирования, (например assembler), архитектуру и принцип работы аппаратной части.

Прикладное ПО — решения, использующие API операционной системы(при необходимости). Прикладное ПО не реализует механизмы доступа к аппаратной части и лишь использует готовые решения для реализации поставленных задач. Разработчика прикладного ПО не нужно знать низкоуровневые языки программирования, а достаточно всего лишь(почти во всех случаях) разрабатывать решения с использованием языков программирования высокого уровня (к примеру java, c#). Прикладное ПО в большинстве случаев не ограничивает себя лишь работой с прикладным механизмом операционной система, а так же использует при разработке сетевые сервисы, базы данных и др.. технологии.

Web-приложения — тип программного обеспечения, разворачиваемый на web-серверах. Web-приложения реализовывают решения в режиме «real time». Web-приложения имеют две основные части — серверная и клиентская. Серверная (она же back-end) реализуется на абсолютно различных языках программирования, единственным условием серверного языка является поддержка его web-сервером, на котором будет разворачиваться приложение. Клиентская (она же front-end) реализуется на языках программирования, исполняемыми в браузере пользователя. На сегодняшний день основным языком программирования front-end является JavaScript (он же EcmaScript и JScript). Разработчикам web-приложений необходимо знать дополнительно язык html, а так же каскадные таблицы стилей css.

Приложения хранилищ данных — ПО для обслуживания хранения и обработки данных. К ним относятся СУБД и хранилища NoSQL. Для разработки ПО этого типа необходимо владеть знаниями основных алгоритмов и структур данных, языка SQL и BSON. ПО для bigdata — ПО реализуемое на компилируемых языках программирования, основной задачей которого является обработка больших объемов данных с большой скоростью. Требования к специалистам, разрабатывающих ПО данного типа — это отличные знания математических концепций и алгоритмов, проводящих быструю обработку.

Сетевое ПО — телекоммуникационные решения, обеспечивающие и обрабатывающие взаимодействия между узлами в сети. Основные требования к разработчикам следующие: знание сетевых протоколов и служб, умение эффективного управления ресурсами и возможностями оборудования.

Мобильное ПО - решения реализовываемые для встраивания, разворачиваемые, на платформах для мобильных устройств. Язык реализации зависит от платформы. От разработчика требует знаний SDK (software development kit) платформы для которой разрабатывается ПО.

Облачное ПО — реализации решений для выполнения обработки данных. Из-за широко спектра функции облачных платформ и самих платформ, язык реализации может выступать абсолютно разный(зависит от поддержки облачной платформы). Разработчикам облачного ПО необходимо знать инфраструктуру сетевых протоколов, принципы и механизмы хранения данных.

Стоимость разработок каждого типа ПО зависит от сложности реализации, применяемых технологий и требований согласно технического задания, применения математических алгоритмов. Оценка стоимости разработки можно представить следующим списком(сортировка по убыванию):

- **системное,**
- **ПО для big data,**
- **сетевое,**
- **приложения хранилищ данных,**
- **облачное,**
- **мобильное,**
- **прикладное,**
- **web-приложения.**



Ежедневно разработчики стараются упростить процесс разработки, в следствии чего появляются новые продукты и сервисы. Основным требованием при работе в команде является **командная работа**.

Зачатую нескольким специалистам приходится работать над одной частью проекта, но это становится неудобным при хранении данных проекта на одной из машин разработчиков. Для работы команды необходим доступ к проекту в режиме реального времени. Так в разработку ПО пришли системы контроля версий, основной механизм которых основывается на хранении проекта на облачных сервисах, доступ к которому имеет любой разработчик проекта. Помимо хранения проекта, эти сервисы позволяют отслеживать изменения в проектах каждого разработчика, получать определенную версию проекта, отлить бэкап (восстановить) и сделать пуш (заливка на сервис).

Системы контроля версий (далее СКВ) различаются по принципу работы над проектом, но несут одну цель разделение работы на них. На сегодняшний день популярны следующие системы контроля версий: git, mercurial, tfs. Сервисы же это всего лишь звено, которые обеспечивают обслуживание СКВ. На сегодняшний день большой популярностью пользуется СКВ Git и ее сервис Git Hub.



Помимо СКВ немаловажную роль в разработке играют **case-средства** (далее СС). СС автоматизируют сам процесс разработки и моделирования системы. Если разделить разработку на несколько этапов, то немалую часть занимает проектирование. Именно проектирование и обеспечивают СС. СС позволяют построить схему проекта и по ней сгенерировать шаблонный код, после чего разработчикам останется лишь допилить его. Одним из таких продуктов СС является пакет от компании Intel — Rational Rose.



ИТ-аутсорсинг — обслуживание ИТ-компанией процессов производства другой компании. На сегодняшний день перед компаниями стоит широкий ряд задач, которые зачастую не пересекаются с основной ее профобластью, а создание ИТ-отделов или даже дополнительный штат сотрудников не позволиммы. В связи с этим на рынке появляется услуга называемая ИТ-аутсорсингом, основная функция которой обеспечение, поддержка и модернизация жизненного цикла процессов сторонней организации. ИТ-аутсорсинг пользуется большой популярностью среди бизнеса больших и средних масштабов, так как в состоянии позволить качественное обслуживание процессов использующих ИТ-



Параллельно ИТ-аутсорсингу, такую же нишу занимает и **ИТ-консалтинг**, основной задачей которого является экспертная оценка системы, позволяющая оптимизировать процессы производства. На сегодняшний момент большой популярностью пользуются услуги ИТ-консалтинга в информационной безопасности.



