#  Обзор рынка IT.

В последнее время наблюдается бурный рост рынка IT: за последние несколько лет количество вакансий на рынке IT-труда увеличилось на 40% (!).

С чем это связано? Рост вакансий на рынке IT-труда связан как с положительными , так и с отрицательными показателями:

* **Рост и развитие самих компаний** (причем, рост и развитие – это разные показатели). Т.е. открываются новые вакансии под новые технологии. IT-компаниям под новый проект либо под замену мигрировавшим специалистам, проще найти новых специалистов на рынке труда, чем переобучать своих. Но в такой сложившейся ситуации виноваты сами программисты этих компаний, которые вскоре пополнят ряды безнадежных безработных. Именно в силу свой высокой квалификации, они не могут пересилить себя, согласиться на добровольное отречение от устаревших знаний, которыми они владеют, и переключиться на более новые и перспективные технологии.
* **Текучка кадров**. В сравнении с другими рынками, на рынке IT – самая большая текучка кадров. Один хороший программист освобождает от работы двух не программистов. Один плохой программист порождает еще две вакансии для программистов. Этот процесс может быть бесконечен. Кроме того, лучшие специалисты уезжают на ПМЖ в Москву, Европу, США. Это утечка продуктивных программистов, или, как раньше было модно говорить, утечка “мозгов”.

Следует помнить, что в IT один год равен пяти. Это значит, что какими бы мы крутыми специалистами на данный момент не были, через пять лет наши знания устареют. Если мы за это время не успеем перестроиться, получить новые навыки, то останемся в неудел. Но для того чтобы перестраиваться, нужно знать куда.

Чтобы определить направление этого рынка нужно обладать развитой интуицией и хорошими аналитическими способностями.

Если раньше требовались специалисты более широкого профиля, то сейчас в вакансиях всегда уточняется, специалисты в какой технологии нужны компании. Связано это с появлением новых и постоянно усложняющихся технологий.

Сами технологии, используемые в программировании, можно разделить на две большие группы: серверные, и клиентские.

Причем, если ранее, на заре зарождения web, ожидалось, что основную нагрузку на себя возьмет серверная часть, т.к. сервера будут мощнее компьютеров пользователей. Оказалось же, что компьютеры пользователей по характеристикам ничем не отличаются от серверов. Поэтому нагрузка решений между сервером и клиентом распределена примерно поровну.

Так же и в рынке IT, выделяется 2 направления: FrontEnd (или клиентская) и BackEnd (или серверная). И 3 группы технологий (для backend, для frondend и общий).

**FrontEnd:**

По сути, в понятие FrontEnd входит всего три технологии: HTML, CSS и JavaScript. Но перечень знаний и умений frontend-разработчика значительно шире. Это:

1. Умение разрабатывать разные типы верстки: табличная и блочная. Резиновая и фиксированная верстка.
2. Гибкая блочная верстка
3. Адаптивная или мультимедийная верстка.
4. Знать и уметь пользоваться такими форматами данных, как JSON (основной формат) и XML.
5. Селекторы: классы, идентификаторы, тэги, вложения, атрибуты и фильтры.
6. Bootstrap 3, API Bootstrap – использование библиотек для адаптивной (или мультимедийной) верстки.
7. Использование в коде архитектурных шаблонов MVC (модель-вид-контроллер), HMVC (иерархические модель-вид-контроллер), MV-VM (модель-вид-пользователь-вид-модель) и MVW (модель-вид-что-то еще).
8. Библиотека запросов jQuery. jQuery позволяет сделать из любого селектора объект.
9. Фрэймворки Angular, React.
10. CoffeeScript
11. Less: переменные, миксины или функции, расширения, импорт, вложенность, соединение в одно свойство несколько свойств.
12. Системы сборки FrontEnd-а. Gulp, grunt, webpack.
13. HTML5 и API HTML5: видео и аудио, холст, перетаскивание, работа с файлами, геолокация (в том числе работа с GOOGLE и YANDEX-картами), web-хранилища, взаимодейтсвие с сервером. Вспомогательный инструментарий.
14. Основы SEO.
15. Schema.org, генератор schema.org микроформат данных, микроданные.

**BackEnd**

1. Linux-подобные операционные системы. Знать, уметь с ними работать, настраивать и администрировать.
2. Docker
3. Языки программирования.
* PHP +
* Java +
* Python +
* Perl -
* Ruby -
* Asp.net -
1. СУБД (это ПО предназначенное для работы с базами данных):
* MySQL (PHPMyAdmin)
* NoSQL, SQL-lite
* Oracle
* MS-SQL
1. Один, лучше 2-3 фрэймворка. Для PHP – это Laravel, YII, ZendFramework Symfony. На сегодняшний день лучшим PHP фрэймворком является Laravel.
2. Системы управления контентом CMS: Drupal и 1Cбитрикс MoDex, Joomla, WordPress, Magenta, OpenCart.
3. Менеджер зависимостей для PHP: Composer.
4. Web-интерфейсы для серверных решений: PHPMyAdmin, Webmin, cloud, media-center.
5. WebRTC
6. Платежные системы. Подключение платежных систем.
7. Парсинг. Библиотеки и классы для парсинга внешних страниц. CURL. Имитация действий пользователя.
8. Умение делать тесты, профилировать, дебагировать приложение.
9. Умение делать аудит, анализ и консультирование проектируемых и уже существующих сайтов. Ajile-технологии разработки проектов. Умение разрабатывать проекты без технического задания.
10. Основы хакинга. PHP-include (вставка исполняемых хакерских php-скриптов на взломанном сайте/сервере, бывает локальный и удаленный), SQL-injection (вставка исполняемого sql-кода в строку запроса, url), XSS (это вставка html/JavaScript/VBScript-кода в сгенерированные сервером страницы; пассивный — срабатывает только при передаче пользователю сформированной хакером ссылке; aктивный — непосредственная вставка html-кода в сгенерированную страницу).

**Инструментарий**

1. IDE:NetBeans, PHPShtorm, WebShtorm, VS – или любая другая интегрированная среда разработки.
2. Postmen
3. OpenServer – Локальный сервер (Содержит встроенные технологии:PHP, Apache, MySQL и системы управления MySQL).
4. Git – система контроля версий. Кроме git сущестуют и другие системы контроля версий, такие как Mercurial, SVN, Subversion. github.com использует git, bitbucket.org – git и mercurial. Основной и наиболее востребованной системой контроля версий является GIT.
5. Composer – локальный менеджер зависимостей.
6. Firefox, Chrome – Браузер.
7. Firebug – Отладчик кода на стороне клиента
8. FireFTP – или любой другой FTP-клиент.
9. Node.js – программная платформа
10. Npm — менеджер зависимостей для Node

**\*\*\***

Frontend и backend программисты постоянно сталкиваются с новыми технологиями, которые приходится изучать.

На что в первую очередь следует обрадить внимание:

* Возможности технологии. Для чего и в каких случаях рационально использовать данную технологию.
* Особенности синтаксиса.
* Документация.
* Платформенная настройка. Установка, обновление и удаление неоходимых инструментов.
* Hello World. Если вы разобрались как вывести hello world на изучаемой технологии, значит вы уже изучили половину этой технологии. На изучение оставшихся, хотя бы 30%, могут уйти годы практики.
* Работа с выводом (шаблонами)
* Базы данных, хранение данных
* Регистрация авторизация
* Обработка изображений
* Формы, обработка данных из форм.
* Безопасность и защита отдельных страниц.
* События
* Клиент-сервер

Чтобы упростить изучение новых технологий, можно отметить, что существует только один программный язык: язык логики, все вышеперечисленные технологии являются частными случаями. В ходе развития IT что-то может появляться новое, что-то устаревает. Сами потребности отсеивают ненужное технологическое нагромождения и приводят к новому. Нужно быть вовлеченным в разработку, чтобы остановить свой выбор на правильных технологиях, которые действительно упрощают решение задач.

Важно, для самообразования использовать не только Интернет, а задействовать как можно больше органов восприятия (ведение конспекта, схематизация, пересказ и т.д.).