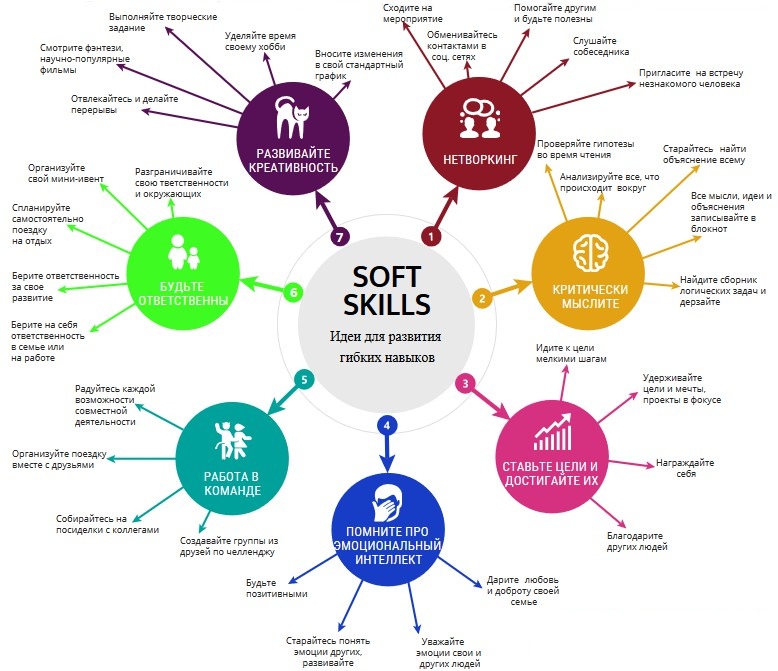


Если говорить простыми словами, то Agile — это такой процесс, процесс работы над проектом, который делится на конкретные итерации — иначе говоря, короткие временные отрезки. В среднем такой отрезок составляет 2–3 недели. Каждая итерация должна решать ряд задач: анализ требований, проектирование, программирование, тестирование и документирование. Эти задачи подходят для IT-сферы, где впервые и был придуман и применён agile. Они, конечно, могут быть изменены, но суть примерно во всех проектах будет одинакова: анализ, планирование, разработка, реализация и оценка.

В России тоже применяют Agile в разных сферах: гипермаркете электроники «М.Видео», службе доставки «Dostаевский», онлайн-кинотеатре IVI, магазине одежды 12 Storeez, металлургическом предприятии НМЛК и др.

Но мастодонтами Agile до сих пор остаются крупные цифровые корпорации: Google, Microsoft, Spotify, Netflix, WordPress, Adobe, Riot Games и др.

Как мне построить, своё обучение на таком проекте? В первую очередь я бы уделил внимание идее для развития гибких навыков Soft Scills и мне так кажется, что для работы программистом эти идеи очень подходят. Они даже укладываются в смысл 12 принципов нашего Agile манифеста. Потому, что Agile это прежде всего команда. В своей жизни я ещё с таким подходом не сталкивался, поэтому для меня здесь всё в новинку, но интересно. Я никогда не вступаю в спор, если я ничего не знаю о теме этого спора, и считаю, что, прежде чем об этом говорить или спорить нужно понимать эту тему, хотя бы на процентах 80-90. Если ты знаешь эту тему на процентах 20-30, то поговорить можно, но спорить не советовал бы, может быть твой собеседник знает эту тему гораздо лучше. И тогда, в каком цвете буду выглядеть я? Я не знаю, конечно, может быть это и неправильный подход, это лично моё мнение?



В этой сфере программирования я новичок, поэтому для меня важно знать всё что касаемо этой профессии, может быть эти знания мне уже и не пригодятся в жизни? Я военный пенсионер, поэтому воздушная подушка от государства у меня есть. Но я считаю, что для своего же развития это нужно делать, да иногда через силу. Считаю, что учиться, никогда и никому не помешает, это только лишь движение вперёд. Уже даже с этих первых занятий, при разговорах с сослуживцами могу им задавать разные тупиковые вопросы. Допустим, что такое язык си-шарп(C#)? У многих была ассоциация, почему-то с продукцией компании Sharp. Но я их не осуждаю, а только наоборот рассказываю, чтобы люди знали, что есть такой язык программирования. Многим интересна работа с Git, создание первых своих страниц в интернете. Мы ещё даже не проходили этой темы, но лично меня она уже заинтересовала, а заметил я её в ролике на YouTube, на страничке GeekBrains. Также много дополнительной информации берёшь из книг, но и ролики в youtube помогают с практическими навыками. Вот по гиту нам тоже дали практически первые навыки на уроках, остальное я уже добирал из книг и интернета.

Но это маленькое отступление всё-таки от темы и цели нашего разговора. Мне лично для того, чтобы попасть в такую команду разработчиков нужно много чего сделать:

1. Не останавливаться на начальных знаниях только C#, иметь также представление о HTML, CSS, JS, PHP, Python. Swift: разработка мобильного приложение для iOS. Чтобы хоть понимать о чём будем разговаривать?
2. Учёба, ещё раз это хорошо, это просто отлично. Но важны для человека ещё и практические навыки, это работа в какой-то фирме, компании и т.д. по данному профилю. Даже для того, чтобы на тебя обратили внимание!
3. Если чуть более подробно углубиться начнём изучать тему лидерства, то в ней я могу узнать, что вместо высокого уровня IQ настоящие лидеры обладают высоким уровнем EQ - эмоциональным интеллектом. Эти люди могут не уметь выполнять работу исполнителя, но умеют делегировать, брать ответственность за результат и принимать решения. Что будет не маловажным, для команды разработчика. Разумеется, главная задача быть профессионалом в своей области.
4. Составить свой индивидуальный план развития, на год, два. В GeekBrains это помогает понять подталкиванием к твоему развитию, заполнение анкет, опросных листов. Что тебе нравиться, что не нравится, чтобы ты хотел поменять и т.д.
5. Разговаривать с людьми, интересоваться их мнениями, позициями, идеями, делиться своими мыслями по этим вопросам.
6. Также знакомимся со всем новым словарным запасом, разработчиков:

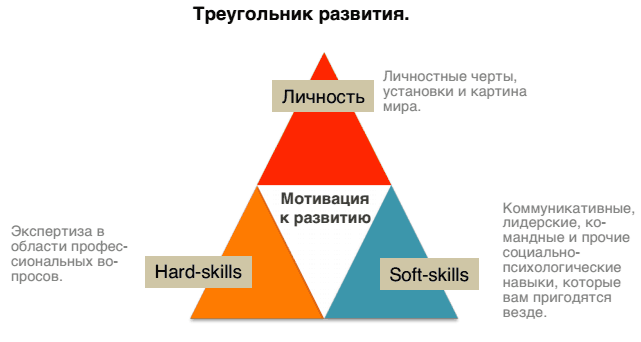
Frontend – внешний интерфейс (это разработчик сайтов);

Backend – базовая разработка;

Fullstack – сложные технические разработки.

Я уже просто молчу про то, что мы ещё не проходили Scrum, Kanban и т.д. Поэтому, знание английского языка не помешает, хотя бы на 70-80%.

Можно ещё очень много об этом говорить, но тут я вроде перечислил, главные факторы: коммуникация, управление собой, мышление, управленческие навыки. Методы развития всех этих навыков: самообучение, поиск обратной связи, обучение на опыте других и нетворкинг, специальные задания (фоновые тренинги), развитие в процессе работы.



И вот когда я пройду, через все эти ступени, тогда уже можно говорить о своём стремлении к повышению.

1. Помимо этого, всего перечисленного как руководителю проекта нужно владеть управленческими навыками, которые требуются людям на этапе, когда они становятся руководителями любых бизнес-процессов и предпринимателями.
2. Коммуникабельность. Менеджер может найти подход к любому человеку: разработчику, дизайнеру или заказчику. Он хорошо разбирается в людях и понимает причины их поступков.
3. Аналитическое мышление. Умеет анализировать происходящее, сопоставлять факты и делать правильные выводы.
4. Требовательность. Умеет не только распределять задачи в команде, но и следить за сроками и качеством их выполнения. Не боится наказывать и не стесняется хвалить сотрудников.
5. Внимательность к деталям. Видит больше, чем другие. Например, первым заметит, если что-то в проекте пошло не так.
6. Смелость. Не боится принимать сложные решения, брать на себя ответственность и нести её до конца.
7. Честность. Честен по отношению к себе, к команде, к заказчику. Если видит, что не получается, — команда не успевает в срок, например, — не пытается скрыть это и сделать вид, что все под контролем. А принимает проблему и ищет способ изменить подход, чтобы найти решение.
8. Искать проблемы, которые требуют решения. Чем раньше обнаружен баг в коде, тем меньше времени и денег команда тратит на его исправление.
9. Внимательность — важное качество для менеджера. Любой документ нужно читать так, как будто у менеджера не два глаза, а целых четыре. Одна неточность или пропущенное слово могут стоить денег.

**З А К Л Ю Ч Е Н И Е**

Чем-то мне это напоминает многими фрагментами Agile методологию. Если я хорошо знаю своё дело, постоянно получаю новые знания и прокачиваю уже сложившиеся навыки. То почему бы не попробовать развиваться дальше? Но только уже, наверное, не в моём возрасте! И всё равно я очень доволен подобными курсами, огромный респект всем преподавателям.

**В Ы В О Д**

С опытом мы понимаем, что и как правильно делать. Поэтому мне сейчас трудно сказать, какие процессы и инструменты Agile работают, а какие — нет? А пока, чтобы наша жизнь была чуточку проще, вот такой простой чек-лист по управлению проектом от меня лично, спонтанно пришедший в голову (я его назвал «By Agile»):

* Определить цель.
* Посмотреть, какие есть ресурсы.
* Спланировать свои действия и действия своей команды.
* Действовать. Следить за ходом работы. Сделать все возможное, чтобы план сбылся.
* Проверить получившийся результат.
* Осмыслить. Сделать выводы. Повторить.