

Об этом выпуске



Джулия Эванс
Привет Я — Джулия. Воть Твиттер: @bOrk
А вот блог: jvnc.ca

На самом деле, я не всегда ощущаю себя волщебником. Я - не самый опытный участник моей команды и иногда я, как и большинство других людей, нахожу свою работу сложной, осознавая, что мне ещё учиться и учиться.

Однако за последние пять лет я научилась некоторым трюкам, которые помогли мне. Мы поговорим:

- О том, как умение задавать глупые вопросы может стать супер-способностью.
- Об инструментах дебага, которые позволят тебе ощутить себя волшебником.
- О том, как мне помогло обучение составлению проектировочной документации.
- О моем подходе к изучению сложных систем.
- О чтении исходного кода, относящегося к зависимостям и том, почему это полезно.

Конечно же, этот выпуск сам по себе не сделает из вас волшебника. Однако, я надеюсь, что один или парочка лайфхаков, описанных в нём, окажутся для вас полезными.

Значительная часть из описанного тут имела отношение к моей работе в недалёком прошлом 😃

Содержание

Итак, вот что мы будем учиться делать $^{
m V}_{\circ}$

 Задавать хорошие вопросы Я выучила молько мо, чмо мне нужно знамь

- Читать исходный код

Этот код не документирован, но я справлюсь

- Дебажить

Хитрый баг? Интересненько... Я его исправлю!

- Проектировать

Большая и плохо описанная проблема? Приступим!

- Развивать компетенции Как мне изучить предмет, тредмет, требующий многих лет для того, чтобы стать мастером?

- Строить стратегии для обучения Вом эмо да! За эмом год на рабоме я узнала мак много нового!

Как стать Волшебником-программистом 3

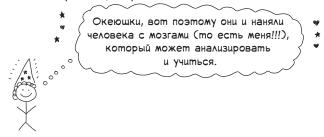
который может <u>всё, что угодно</u> (только потребуется куча времени)

- Задавать вопросы. Вокруг тебя всегда есть люди, которые знают что-то, чего не знаешь ты – спращивай их как правильно сделать это самое "что-то". Глупые вопросы, вопросы-которые-сложно-задать. Твои вопросы будут становиться менее глупыми очень быстро.
- Подписывайся под проблемы, которые не могут решить твои коллеги.
- ПОВЕРЬ, ЧТО ТЫ В ЛЮБОМ СЛУЧАЕ РАЗБЕРЁШЬСЯ, КАК РЕШИТЬ ПРОБЛЕМУ (это очень сложно, но иногда это работает:)

Чем больше я занимаюсь программированием, тем чаще я сталкиваюсь с проблемами, которые:

- -Я не могу решить.
- -Мои коллеги не могут решить.
- -Даже Гугл-Всемогущий не знает, как их решать.
- -Мы всё равно решим тем или иным способом.

Когда такое случается, я думаю:



Этом выпуск рассказывает о том, что из себя представляет навык "решить проблему так или иначе".

Когда следует вложиться в понимание?

Мы работаем с большим количеством абстракций. Вовсе не обязательно тратить время на то, чтобы понять как на самом деле работает та или иная штуковина.



Однако, огромная часть становления волшебником — это понимание того, как эта магическая компьютерная система работает. Так как же понять, когда стоит потратить время на понимание принципов работы новой загадочной штуки?

- Когда ты дебажищь хитрую проблему:
 - иногда библиотеки, от которых ты зависищь, сами содержат баги,
 - часто библиотеки и системы (CSS/Linux) содержат сложные абстракции, которые требуют времени, чтобы понять их (например, блоковая модель или еро! в Линуксе).
- (2) Когда ты <u>пытаешься преодолеть границы</u> / оптимизировать производительность

Я не часто думаю о железе, которое обрабатывает мой код. Но следует помнить, что когда ты записываещь данные в файл, ты ограничен скоростью твоих дисков.

③ Когда ты пытаешься сделать что-то принципиально новое

Если ты создаёшь новую абстракцию (например, асинхронную библиотеку), ты должен понимать, как устроен предыдущий пласт технологии (epol, select и другие подобные штуки)!

Задаём правильные вопросы

Один из моих любимых приемчиков для обучения — это приставание κ крутым специалистам вокруг меня с хорошими вопросами.

Что же такое хороший вопрос?

Признаки хорошего вопроса:

- 🖈 Человеку легко на него ответить.
- 🖈 Он даёт тебе инфу, которую ты ищещь.

Вот, к примеру, некоторые из моих способов формулирования таких вопросов:



Это полезно потому, что:

- Заставляет меня перебирать в голове уже усвоенные знания.
- Сильно снижает шанс получения слишком поверхностного или же слишком заумного ответа.

Когда я пытаюсь предугадать ответ на свой вопрос, это заставляет меня напрячь извилины и иногда помогает собеседнику понять, какой именно ответ мне нужен.





Это удачный выбор для ответа на твой вопрос + у него гораздо больше свободного времени :)

Я уважаю чужое время, особенно, если вопросов у меня много :)



Я \P задавать вопросы — на да/нет, вроде таких.

На них легко ответить, если я постаралась правильно формулировать вопрос.

 Тот, кто знает больше всех – не всегда лучший вариант.

Очень часто спрашивать лучше человека, который усвоил нужную информацию торошо помнит, каково это, не понимать что-то.



 Если я потрачу совсем немного времени на предварительное изучение вопроса, я смогу смогу сформулировать более точный вопрос :)



Читаем исходный код

Окей, вот только нельзя ПОСТОЯННО задавать вопросы.

В некоторых случаях:

- → Нет документации
- → Коллеги заняты
- → Или они не знают ответа
- Ну или тебе нужно узнать больше аеталей, настолько больше, что спрашивать об этом даже неуместно.

К счастью, у нас есть опенсорс!!!



Однажды я хотела узнать, можно ли настроить сокет на Linux так, чтобы он не убирал соединения в очередь. Я погуглила и получила несколько противоречивых ответов. Однако один из ответов на Stack Overflow содержал ссылку прямиком на $KO\Delta$ SAPA!

В общем, это выглядело так:

Таким образом, невозможно сделать бэклог равным нулю. Его значение всегда будет минимум 8:

Советы по чтению кода



Вот пара вещей, которые помогают, когда работаещь с незнакомым кодом:



Когда я вижу ошибку, которую не понимаю, поиск ее причины обычно не составляет сложности и часто помогает.

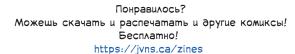
Если код, который я использую содержит всего пару тысяч строк, обычно я читаю его весь, чтобы понять базовые принципы его работы.





Не бойся запачкать руки:

- Воспользуйся дебаггером.
- Добавь тестов.
- Добавь текущее состояние!
- Обсуждай баги в команде!
- Экспериментируй!
- Не доверяй слепо комментариям :)



Комиксы перевели и оформили команда FirstVDS

<u>Ещё комиксы с переводом</u> <u>читайте в нашем блоге</u>

CC-BY-NC-SA

Julia Evans, wizard wow fun industries 2017