Book: The Passionate Programmer

Author: Chad Furman

Claps: 81

Date: Sep 28, 2021

Do you want a strategy for planning and creating a radically successful life in software development? Thatâ€™s the catch phrase from a book I just read by Chad Fowler titled â€œThe Passionate Programmerâ€. Thereâ€™s great little side stories in it by people including co-founder of Github and more. Each chapter ends with a set of action items, and the chapters are grouped by several parts. Iâ€™ll quickly summarize each part, but I highly suggest you read it yourself :)

Choose your market carefully
Love what you do, or do something else. Pick a tech stack (and subsequently a market) to invest yourself in. Youâ€™ll want a market that ideally has an imbalance â€” more demand than supply.

Go deep into your chosen tech stack â€” know the ins and outs of it. What makes the compiler choke? What are some ways you can optimize when things get hairy?

Surround yourself with people who are already in the market and who are doing great things. Be the worst person, not because youâ€™re not trying but because other people around you are simply better than you. This will lift you up and give you more opportunities to excel.

Invest in your product
You are your product. Take time, at least a few hours a week but as many as perhaps 20 hours a week to hone your craft. Just like a musician has to practice outside of the show, you also have to practice. Try code katas, or read open source projects like a book. Try adding unit tests to an open source project.

Learn about software methodologies like TDD. See about improving your productivity. Can automate a routine task you do to give yourself a sort of robotic arm and increase your productivity? Can you find a mentor? Can you be a mentor?

Learn the language of the business that youâ€™re in â€” what does your business consider valuable? What makes you valuable in your role?

Execute
Every day, try to have a daily win. Donâ€™t let this stress you out â€” just take a deep breath and do what you can. Enjoy the ride, enjoy the process. Donâ€™t be afraid, and just do it right now â€” whatever it is.

Know that your goal is to support your team and that the success of your team is your success â€” youâ€™re not trying to climb a corporate ladder by usurping the more senior people on your team or your boss, youâ€™re trying to help them reach their goals as well. This, in turn, helps you because theyâ€™ll know youâ€™re a valuable team member. Make sure you deliver results. Say what youâ€™re going to do â€” make a commitment â€” and do it. Once youâ€™re done, report back. This builds trust.

And one of the most important things you can do to increase your productivity and increase your overall success is to get 8 hours of sleep a night. 8 hours of sleep will improve your willpower,

your concentration, your creativity, and your mood. All of these will make you a better software engineer and a better human.

Market yourself
Some general tips here are to look at yourself as a brand and to know your customers. You want to be the kind of person that your clients and your co-workers love to talk about. â€œDid you see that Jonny did that great thing? How cool!â€œ.

Get to know people, get to know the business, what are the problems theyâ€™re facing? How can you help? Hang out with the smartest people you can find, especially if it makes you uncomfortable.

Make sure you deliver value. Your salary is half your cost. The business needs to make 3x your salary in order for you to be a valuable employee. Did you deliver 3x of your salary in value today? If not, why not? If so, how? Note it down and try to improve for tomorrow. 1% incremental improvements every day add up.

Know how people see you. What are people thinking when they see you? What are they looking for? Make sure your conversations are professional and reflect the best you that you want other people to see.

Keep your edge
That latest tech you just learned is already starting to become obsolete. Put a couple hours a week (at least) into learning something new. Know that you are not your job â€” you could be a QA engineer this week, a software engineer next week, a manager, a designer, a sales person â€” it doesnâ€™t matter.

Assume youâ€™ve already lost your job and focus simply on the path to delivering the highest quality and the best you possible every day. Measure your results, evaluate them, and come up with a path to improve yourself.

Summary
Thereâ€™s a lot in this book, and itâ€™s hard to break it down in such a short post. Iâ€™ve read this book a few times now and every time I read it I get more out of it. Each chapter ends with an â€œAct On Itâ€ section with ideas you can apply to your daily life right now so that your career is more successful and your life is more fulfilling. The â€œAct On Itâ€ that Iâ€™ll leave you with now is to buy the book and give it a solid read through! Itâ€™s worth your time :)

Some quick thoughts:

- Talks about how you should love your job and what that means
- Learn your tech stack deeply
- Be familiar with the DB, Redis, Celery, Rabbit, Django, Grafana, and more. Have breadth
- Practice your craft. Work on open source projects â€” add some tests to one for example. Study how they do things. What works? What doesnâ€™t?
- Know methodologies and help implement some
- Have a mission and share it with the world. What are you passionate about? Tests? GraphQL? What are others passionate about? How is their mission represented in their actions? What works? What doesnâ€™t?
- Talk with people you admire. Get to know them. â€œMake the hangâ€ and surround yourself with great people. Be the worst person on your team, not because youâ€™re not trying to be great but because the other people are better than you and lift you up.
- Invest in yourself. Make your brand and your product cutting edge. Learn the latest tech, practice it. Learn and practice tech you donâ€™t normally care about, simply to reaffirm **why** you donâ€™t care about it (or to find that itâ€™s actually pretty cool). Spend atleast a

few hours a week working in a new tech. Stranger the better. Try functional programming. Check out graphics programming. Try something completely new.

- Definitely more in the book. Highly recommend. Go check it out.