

プログラマとしての成長の仕方を考える

株式会社ロングゲート

高橋 晶(Akira Takahashi)

faithandbrave@longgate.co.jp

今日のお題

- この資料では、谷川さん向けに、以下のことを解説します。
 - どのようにすれば一人前のプログラマーになれるか
 - どのようにすれば一人前のプログラマーで居続けられるか
-

まず結論から

- しっかりとした基礎を身につける
 - 自分を成長させる習慣を作る
 - プログラミング技能だけでなく、
開発に必要なあらゆることを学ぶ
-

しっかりとした基礎を身につける

- プログラミングの基礎をしっかりと身につけておくことは非常に大事です。
 - 基礎ができていないプログラマは、場当たりの的であとから苦勞するコードを書きます。
 - プログラミングに関する多くの原理・原則を身につけることで、「最初は学習が大変だけど、学習を進めるほど、開発がどんどんラクになる」という基盤を構築できます。
-

場当たりの学習とは

- 場当たりの学習とは、「自分が実現したいものだけをGoogleで検索し、たまたま見つけた記事の一部分だけを取り込んで使う」というようなやり方です。
 - 基礎ができていないうちから、このような学習方法をとると、あとで痛い目に会います。考え方がわからないのにとりあえず使って、問題が起きたときに対処法がわからないとか、ほかにもいろいろ。
 - また、場当たりの学習は、難易度が低いものを実装するときだけしかできません。考え方が身についていないと、幅は広げられても、深くは進めないのです。
-

成長の習慣を作る

- 職業プログラマの中には、
「忙しい → 勉強できない → 忙しいまま」
の悪循環に陥っている人が大勢います。
(プログラマは、残業で毎日終電、という業界イメージが
できているのは、そのせいです)
 - 「忙しいけど勉強する → 身につけた技術のおかげで、
だんだん忙しくなくなってきた」
というループを構築することが、悪循環から脱出する
効果的な方法です。
-

成長の習慣を作る

- Google社の20%ルールが有名です。
80%の時間は業務に当て、残りの20%は新しいものを創造する時間に使いなさい、という会社から社員に課せられたルールです。
 - このようなものは、ルールがなかったとしても、時間の割り振り次第で、誰でもできます。
 - 毎日の仕事や生活の中に、自分を成長させる時間を取り入れましょう。
-

技術書を読む

- 世に出版されている技術書は、その著者が数年から十数年かけて研究・実践した成果のまとめです。
 - 自分が経験でそれらを学ぶと、同じかそれ以上の時間がかかります。しかし、それらの書籍を読んで取り込むことで、1ヶ月程度でその技術を身につけられます。
 - 技術書は1冊3,000円は超えますし、一般書に比べると高額に思えます。しかし、自分の価値を高めることや、プロジェクトへの恩恵を考えたら、決して高いものではありません。自己投資はどんどんしましょう。
-

「経験＝技術力」ではない

- 「10年の経験があります」という人がいたからといって、その人が優秀とは限りません。
 - 10年間、技術を磨くことなく、今までどおりのやり方で開発してきた人と、常に技術を磨いてきた経験3年のプログラマでは、後者の方が優秀となります。
 - プログラマは経験年数では成長できません。
-

開発に必要なあらゆることを学ぶ

- プログラミングができたならプログラマとして優秀かという、そうとは限りません。
 - 開発には、より多くのことが必要になります。
 - 伝えたいことを必要十分に伝えられるコミュニケーション力、自分の考えをほかの人に提案する力、スケジュール管理力、論理思考力、UI設計、英語力、プロダクトの魅力を高める力、etc…
 - どれかひとつがチームに致命的に欠けていると、プロジェクトは失敗しやすくなります。
-

Betterを目指そう

- 「最高の自分」なんてものには、永遠になれません。
 - 最高とは「これ以上は成長できない」ことを意味します。
 - 「明日の自分は、常に今日の自分を超える」のように考えて、日々の成長を大事にしましょう。
-

成長する人の資質

- 問題意識を持てること
 - 自分に足りないものを見つけられること
 - 問題解決のために行動できること
 - 「できたらいいな」ではなく「やる」
 - 継続の強い意思を持てること
 - 「今日はサボってもいいや」と自分に言い訳を許してしまうと、そこで成長は止まります。
-