Angular Vietnam
5 Sep ‘21

Bill Nguyen

About me

Bill Nguyen

Senior Software Engineer - Tech Lead

Ninja Van · Posts and Telecommunications Institute of Technology

Singapore, Singapore · 500+ connections

About me

2012	StarHub
2016	PayPal
2017	AppliedMesh
2019	traveloka
2020	ninjavan

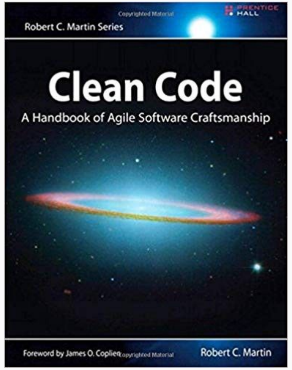
Things I wish I knew earlier

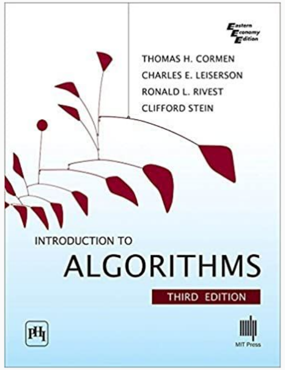
Things I wish I knew earlier

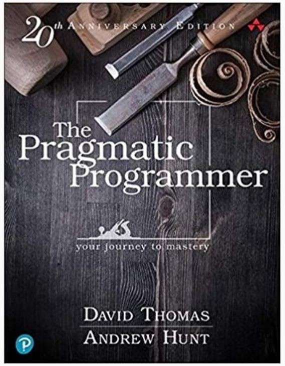
1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

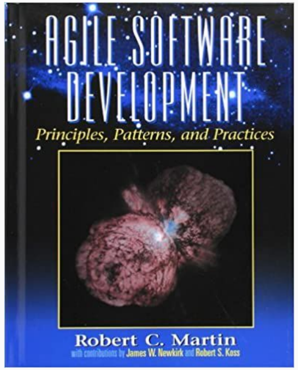
1. It is VERY important to understand the basics

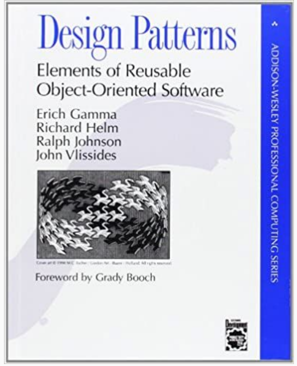
- Tech is moving VERY fast
- Hence you always need to adapt
- The fundamentals are roughly the same
- You should always read the fundamental books again


Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin






AGILE SOFTWARE DEVELOPMENT
Principles, Patterns, and Practices
Robert C. Martin
with contributions by James W. Newkirk and Robert S. Koss


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

1. It is VERY important to understand the basics

CS101 from some top universities:

- https://web.stanford.edu/class/cs101/
- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/
- https://www.classcentral.com/report/best-intro-computer-science-courses/

Things I wish I knew earlier

1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

2. You need to keep learning

You Don’t Need a Degree to Program

- Programming can be self-taught
  - Everything you need is available online, **for free**
  - Even Udemy courses (the most popular, 4.5* and above ones) are $16-$50 each
    - *But you need much more than just 35 seconds*

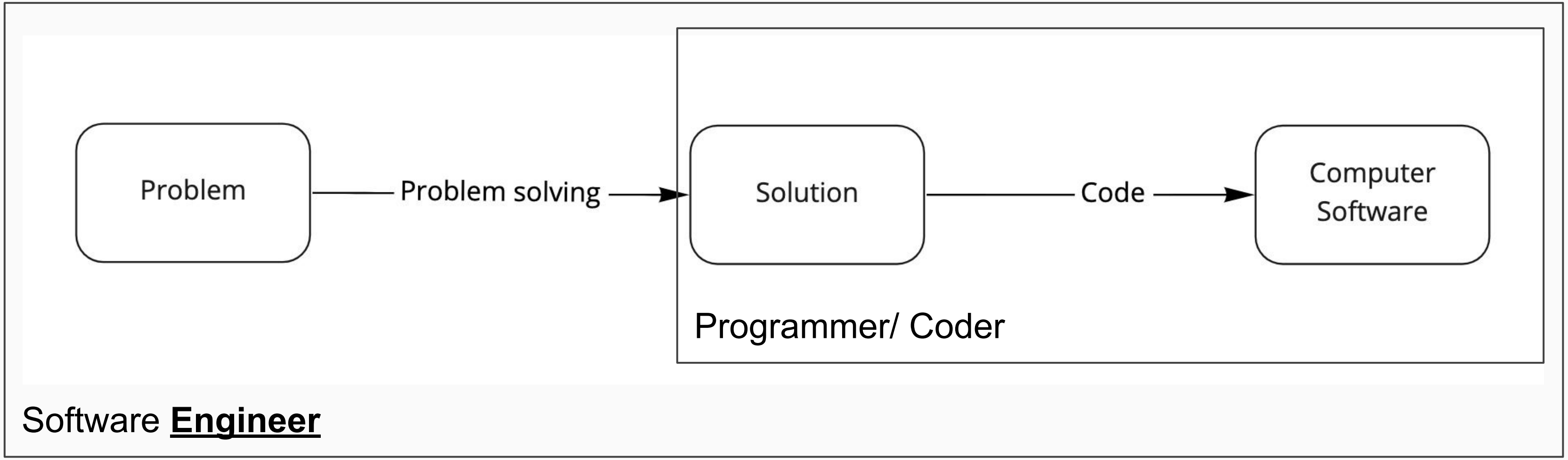
… but You Do Need the Knowledge

- Programming is a knowledge-intensive career
- You need to know a lot of stuffs
- New knowledge is added every day
- As a Software Engineer, you need to learn EVERYDAY

Things I wish I knew earlier

1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

3. You are hired to solve the problems, not just to code



3. You are hired to solve the problems, not just to code

- AI is (almost) powerful enough to be able to code
- Difference between human and computer:
  - Computer is fast (*human is not*)
  - Human is smart (*computer is not*)
  - Current fastest computer: 415,500,000,000,000,000 operations per second (415 millions of billions)
- Software engineer is the one who use the **fast computer** to solve problems in a **smart way**.
  - “Code” is just a language of communication between human and computer.
- There are “developers” who earn $3K a month and there are “developers” who earn $30K a month
  - In the same city
- Problem solver first, coder second.
- Be smart!

3. You are hired to solve the problems, not just to code

Things I wish I knew earlier

1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

4. Learn how to use Google and StackOverFlow

- Everyone uses Google (search engine) nowadays
- Every developer uses StackOverFlow nowadays
- You need to use them smart.
  - Get the correct information/ answer you need
  - In a short enough time
- HOW?
  - Google “How to use StackOverFlow effectively”
  - Read and practice
  - Read guides like this.

4. Learn how to use Google and StackOverFlow

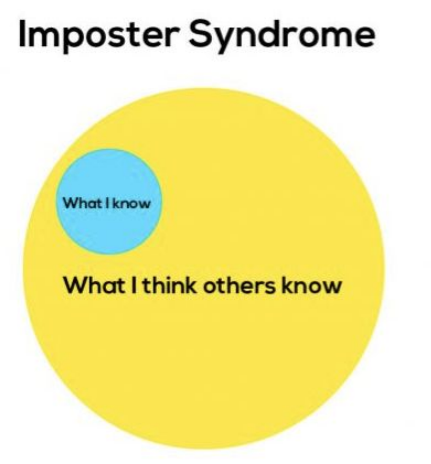
- But DO NOT COPY AND PASTE code from StackOverFlow
- You ALWAYS need to fully understand your code.

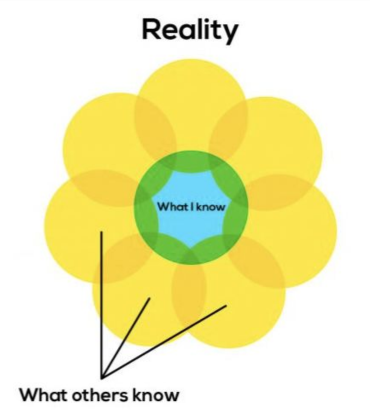
Things I wish I knew earlier

1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

5. The Impostor syndrome

- https://en.wikipedia.org/wiki/Impostor_syndrome
- As a programmer, you will be surrounded by many “geeks”
  - That is so good! Don’t worry about it.
- It is impossible to know everything
- People may do better than you
- What’s important is, tomorrow YOU can do better than yourself
- Use it as motivation to learn new things
- Do not afraid to ask. Do not afraid to show that you don’t know
- Do not afraid to ask for help.





5. The Impostor syndrome

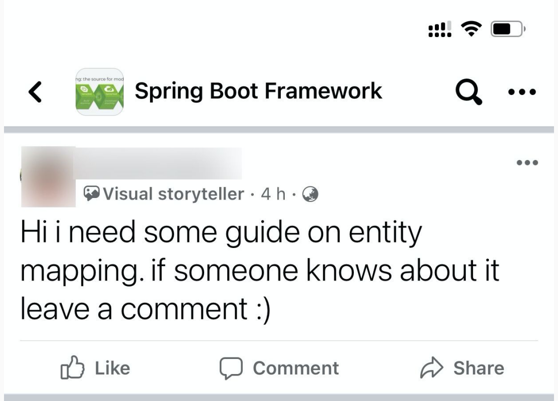
- “三人行必有我师” (孔子)
- “When I walk along with two others, they may serve me as my teachers”
  (Analects of Confucius)

Things I wish I knew earlier

1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

6. Learn how to ask for help/ ask questions

How NOT TO ask:

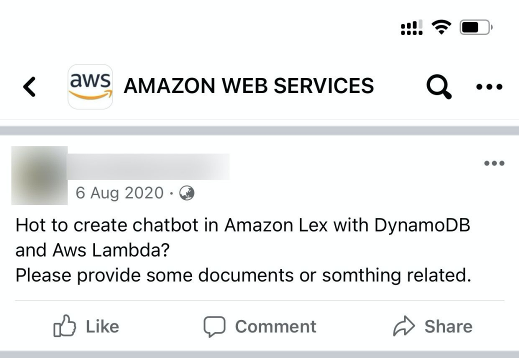


Spring Boot Framework

[redacted]

Visual storyteller · 4 h ·

Hi i need some guide on entity mapping. if someone knows about it leave a comment :)

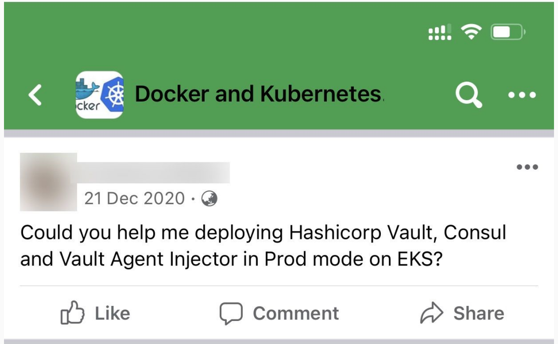


AMAZON WEB SERVICES

[redacted]

6 Aug 2020 ·

Hot to create chatbot in Amazon Lex with DynamoDB and Aws Lambda?
Please provide some documents or somthing related.

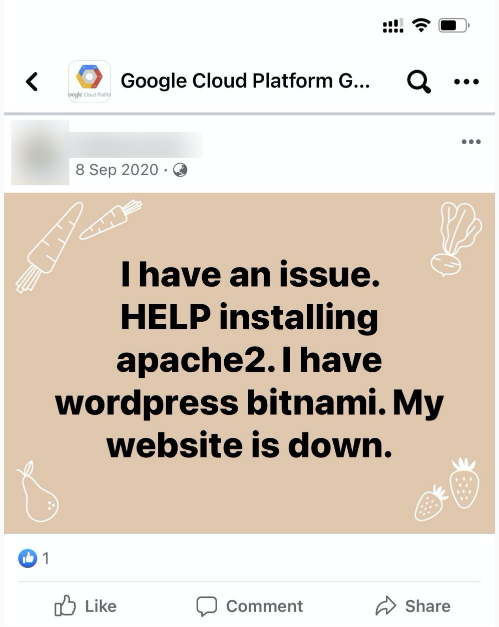


Docker and Kubernetes

[redacted]

21 Dec 2020 ·

Could you help me deploying Hashicorp Vault, Consul and Vault Agent Injector in Prod mode on EKS?



Google Cloud Platform G...

[redacted]

8 Sep 2020 ·

I have an issue. HELP installing apache2. I have wordpress bitnami. My website is down.

6. Learn how to ask for help/ ask questions

- How NOT to ask:
  - “Hi there”
  - Too trivial problems
  - Expect other people to do the work for you
  - Showing that you are lazy

6. Learn how to ask for help/ ask questions

- Do some research first
  - I won't be happy if you ask me questions that have answer on the first Google result page
  - Try to solve the problem yourself/ find the answer yourself
  - Keep track of what you have tried.
  - Make sure that the problem is not caused by common causes (missing NOT operator, missing semicolon, etc)
- Ask in a way that people are willing to answer
  - Show that you have done some research
  - Provide enough background information
  - Be specific.
  - Ask for advice/ comment/ suggestions
  - DO NOT expect other people to do the work for you

6. Learn how to ask for help/ ask questions

- Be specific & Providing enough background information
  - What are you trying to do?
  - What is the expected result?
  - What is the actual result?
  - How to replicate the error?
  - What have you tried, and what happened?
  - Provide all necessary information so that the helper can start looking into the problem without having to ask you for further details
    - Env setup: Which OS, compiler, IDE, etc. with version
    - The source code (a github URL would be the best)
    - Different setups/ source code of different scenarios that you have tried

6. Learn how to ask for help/ ask questions

Example of good questions:

- https://stackoverflow.com/questions/39152071/cant-bind-to-formgroup-since-it-isnt-a-known-property-of-form

Things I wish I knew earlier

1. It is VERY important to understand the basics
2. You need to keep learning
3. You are hired to solve the problems, not just to code
4. Learn how to use Google and StackOverFlow
5. The Impostor syndrome
6. Learn how to ask for help/ ask questions
7. Last but not least

Last but not least

TEST YOUR CODE

TEST YOUR CODE

- Remember, you're hired to solve the problems
- Your code is the representation of your solution
- It SHOULD work exactly like what you want it to work
- It SHOULD not produce unexpected behaviour/ result
- How to ensure? Write tests.

TEST YOUR CODE

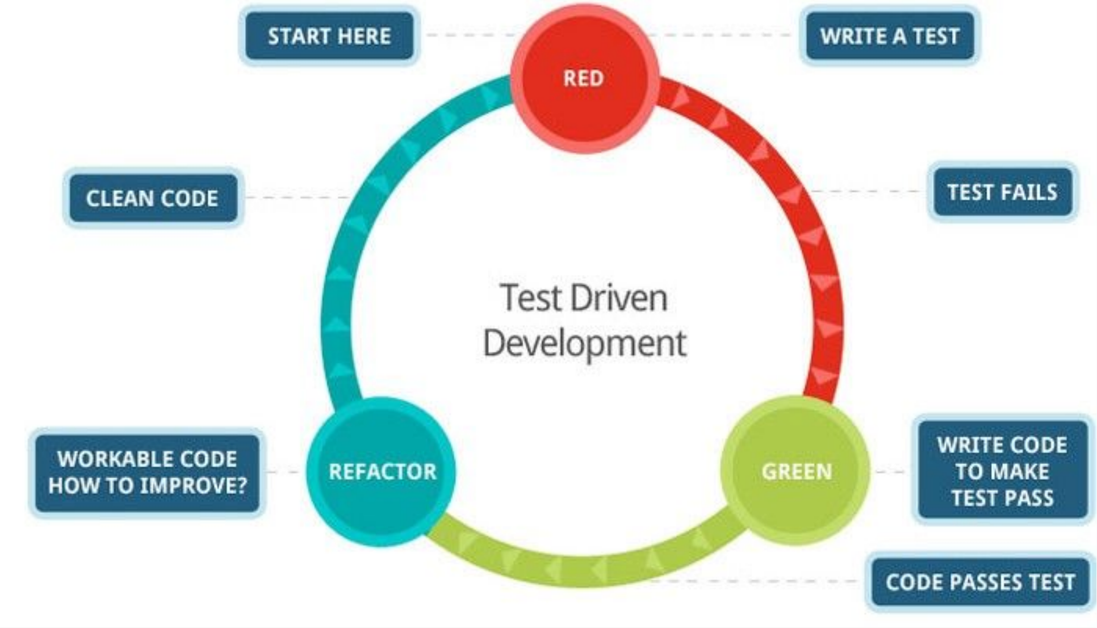
- You should test your code manually
- More importantly, you should write Unit Test and Integration Tests
- Your test should cover all possible cases
- ***Everytime you fix a bug, write a test case to ensure that the same bug will never happen again***
- It is normal that the effort that a developer use for writing tests is more than the effort used for writing the code.
- Tests is an ***indispensable*** part of our code.

TEST YOUR CODE

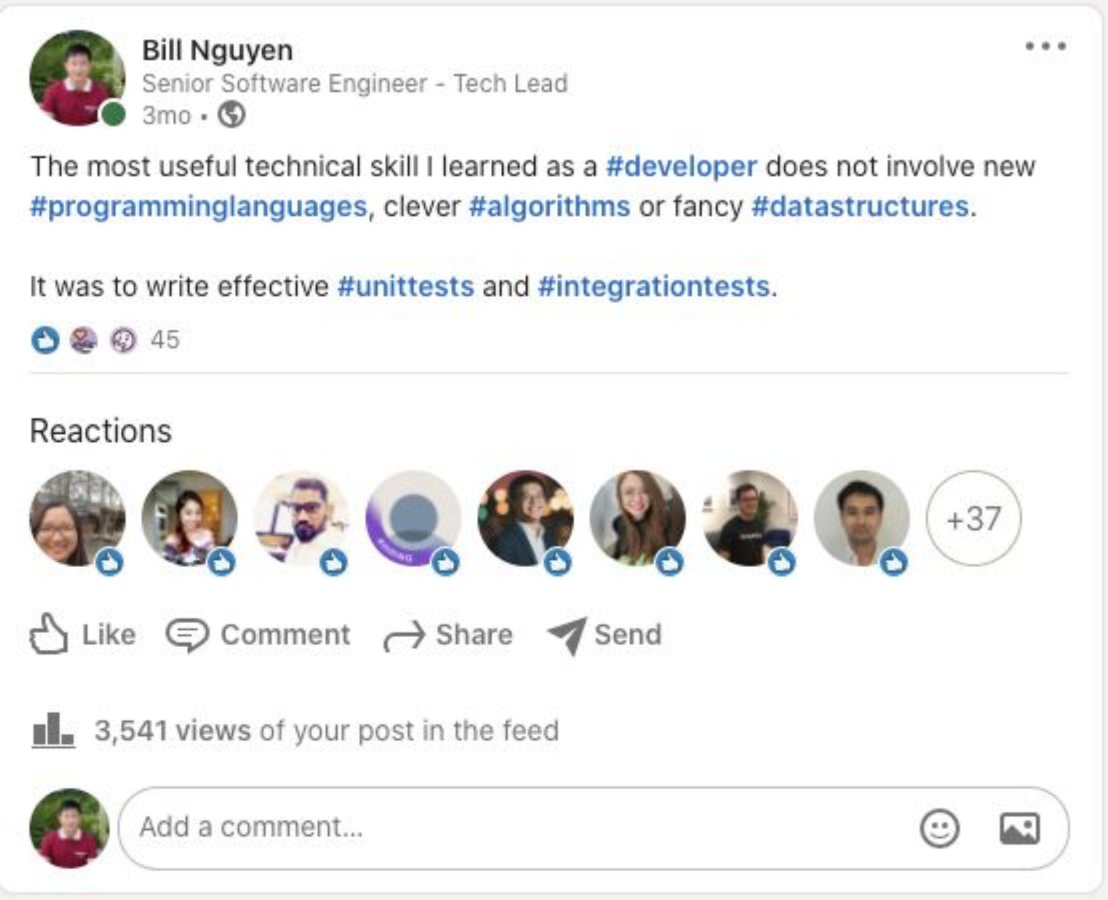
- Writing test benefits YOU **before** benefiting the company

TEST YOUR CODE

Practice TDD



TEST YOUR CODE



Bill Nguyen
Senior Software Engineer - Tech Lead
3mo •

The most useful technical skill I learned as a #developer does not involve new #programminglanguages, clever #algorithms or fancy #datastructures.

It was to write effective #unittests and #integrationtests.

Q&A