Nibbles

OPENTURF
TECHNOLOGIES PVT.LTD.

#36 – Best of Three years of curation
"Dedicated to the silent souls who toil behind the scenes and before the humble screen, illuminating our world.
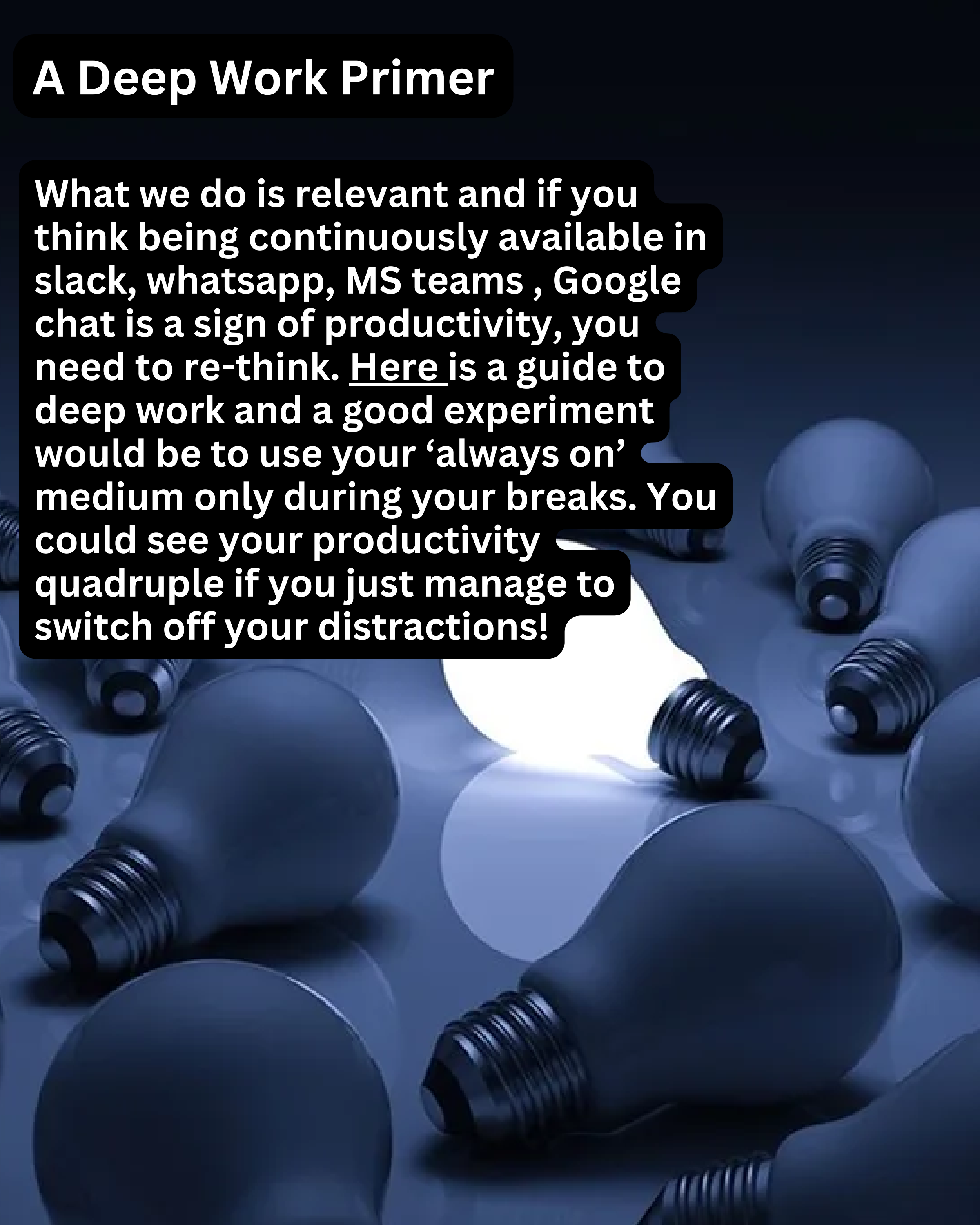
CS50

A fascinating introduction to computer science and the art of programming. We are sure that novices and experts will enjoy this: https://www.edx.org/course/cs50sintroduction-to-computer-science

The 30 best bits of advice

Firstround is a place we visit regularly and recommend every aspiring engineering leader to follow. They came out recently with the 30 best bits of advice they have heard on their podcast. Our favorite is Level-up your leadership by getting off the floor. What is yours?

A Deep Work Primer

What we do is relevant and if you think being continuously available in slack, whatsapp, MS teams , Google chat is a sign of productivity, you need to re-think. <u>Here </u>is a guide to deep work and a good experiment would be to use your 'always on' medium only during your breaks. You could see your productivity quadruple if you just manage to switch off your distractions!

**Focused and Diffuse:
Two Modes of Thinking**

While creating we are focused,
but while fixing we need to
diffuse. Check this article out
and decide for
yourself

Chesterton's Fence: A Lesson in Second-Order Thinking

At first glance, this article may seem irrelevant to the software industry, but if you go through it and have programmed for some time, you will realize that there is great merit in this article. We have a saying 'Respect working code'. So before you start to refactor take the effort to understand why certain things were done the way it was done. Do not reject anything outright

It is ok to have an ordinary mind

"I think people who keep an ordinary mind are more relaxed, have no internal distortions, observe things with a more nuanced perspective, are practical and have more patience" - Zhang Yiming, Byte Dance CEO. Read the full transcript <u>here</u>. The counter perspective is refreshing.

The Best Engineers Think Like Investors, Not Builders

If you are the one who calculates the opportunity cost of technical debt, probably you should think like an investor, not an engineer. Head, here for an interesting take.

The forty year programmer

You wish to be a 'coder for life'. <u>This</u> article gives an insight. We love them.

Things they didn't teach you about Software Engineering

- Lesser known aspects of being a craftsman. This can be classified as a classic article in our view.
- How many of you have realized the importance of understanding business value!!. Save the article, memorize it if possible and go back whenever you want to.

Motivation vs. Discipline: Which Helps You Go Further?

Our vote is for Discipline. What do you think? Take a look here

THANK YOU