



Talent Acceleration Program

Workbook



tap.kiitos.tech

Week 0: Preparation

It's not the will to win that matters – everyone has that. It's the will to prepare to win that matters.

Welcome to the *Preparation Module* for the **Talent Acceleration Program (TAP)**. In this module you'll learn all about how to set yourself up for success so that you can get the most out of the program!

It is divided into several sections:

1. Learning How To Learn
2. Web Development Fundamentals
3. TAP Quickstart

Good luck!

The TAP Team

PART 3: TAP QUICKSTART

Now that you have learned about better ways of learning and refreshed your fundamentals of web development, it's finally time to dive deeper into the finer details of the Talent Acceleration Program (TAP).

We'll be going over the following sections:

1. **Educational model** the program is based on
2. **Mindsets** that'll empower you to succeed

Educational Model

The central goal of TAP is to help you

Transition from student to professional software developer

To achieve this the program is based on an educational model, called Self-Directed Learning. It's an approach to learning and facilitating education where *the student takes the lead in the learning process*, instead of the teacher.



This means the student:

1. **Sets learning goals** based on their needs
2. **Develops a plan** to achieve those goals
3. **Learns** all the relevant things they need to know

4. **Shows results** of what they learned
5. **Reflects** on their learning process to make continuous improvements

In this approach you're encouraged to take a more active role in your learning. In education we call this Active Learning: engaging with your topic of study (i.e. through practical exercises or group discussions), because of genuine interest. This proves to be very effective, because it draws on underlying characteristics of how the brain functions during learning.



Whenever you face new information, unconsciously your brain asks 2 questions:

1. Can I understand this?
2. Is this relevant to me?

The first question deals with your *existing knowledge*. For example, you'll only be able to understand the words on this page if you have a mastery of the Latin alphabet and the English language!

The second question deals with *information that will likely be used in real-life*. For example, it's much easier to learn how to write code if you want to do it for a living, rather than memorizing a list of historical facts in order to pass a test (and you'll forget it right after, anyway).

By keeping these 2 approaches to learning in mind, you'll make your learning journey easier and much more enjoyable.

Learning Materials

- [Self Directed Learning](#)
- [The Active Learning Method](#)

The reason why we use this learning approach is because it will help you to learn everything you need to know to become a professional software developer through *simulating the exact behavior you are expected to have in your day-to-day work*. These are practices such as:

- Solve coding problems self-sufficiently
- Research (new) technologies you're using
- Collaborate with other technical and non-technical colleagues
- Interact with clients
- Present your work to colleagues and clients

Getting used to a self-directed and active approach to learning complex subjects will allow you to transition into the role of professional software developer much easier, because you're practicing exactly what you'll be doing day-to-day in the job!

Learning Materials

- [The Art Of Learning For Software Developers](#)

Mindsets

In order to achieve any type of success it's important to get your mindset right first. This is because the way you think determines the reality we see around us. In technical terms this is called the Reticular Activating System, a part of our brain that acts as a *filter* against all of the information that surrounds us.

Learning Materials

- [Reticular Activating System](#)

By adopting the right beliefs we can get our mindset pointing in the right direction, so that we can make the right decisions! The following is a list of beliefs that will help you get there:

1. Growth Mindset
2. Always Think In Solutions
3. All Education Is Incomplete
4. This Is Work, Not School
5. Growth Implies Change

GROWTH MINDSET

One of the fundamental mindset shifts you have to make during TAP is to go from a fixed to a growth mindset. This is a topic we'll go over extensively in Week 1, but let's lay the foundation here.

When we start learning a complex subject we all bump into a fundamental belief about how we learn: can we change and become better, or are we destined to fail and stay the same?

The way we choose to answer that question has a very real impact on how we go about skill development.



LESSON: Start believing in your ability to improve for the better. All you have to do is choose to have faith in yourself and keep on trying until you succeed.

ALWAYS THINK IN SOLUTIONS

In your journey towards becoming a professional you'll face many obstacles. Often, they might not even be of your own making. That said, you are still responsible for the results of your own actions.

That might sound unfair, and it is. However, focusing on the problem will not help anybody. Instead, be solutions-oriented: whenever something undesirable happens, what can you do to improve the situation?

In TAP you should expect things to go wrong; that's just the nature of working with others. Whether it's in a software project, educational session or just a general chit-chat where somebody said something edgy.

By creating solutions immediately you'll not only make the situation better for everyone, but develop your leadership potential!

LESSON: You may not always get to pick your circumstances, but you can always do something positive with what you are given.

ALL EDUCATION IS INCOMPLETE

If you've ever taken a course, it's easy to assume that the teacher has designed it in such a way that you'll learn exactly what you need to be able to apply your knowledge perfectly. In truth this is not how educational design works.

Firstly, any educational program is restricted by time, subject and the creator's limited knowledge of what they want to communicate. More importantly, the way it's designed takes in mind *a certain type of student with a certain type of background*. What if you don't fit the mold? Then you might need extra support to truly understand what's being taught.

Secondly, education is mostly theory. In order for it to have effect you have to actually apply it to get the lesson. This is not something you can learn by reading a book or watching a video.

TAP, by design, is incomplete. It helps you make a headstart into the factors that make a great professional software developer, but it doesn't cover everything you need to know in order to actually get there. Everybody's different, so you have to figure out for yourself what it is you need to learn to get to the next level.



LESSON: Always look for the gaps in your knowledge. Then go out and find what you need to know to improve your understanding.

THIS IS WORK, NOT SCHOOL

In school it's very easy to shove responsibility for your learning over to the teacher, the study material or your parents. Simultaneously, you might believe that the exams you do and the grades you receive are an indication of what you actually know or can do.

That's not how it works in the real world of business. You might have gotten the highest grade for passing an exam, but if you can't build working software you still failed.

TAP is not a school program; it's your first contact with working life. You're not a student anymore, but a working professional. In the real world of work, the only things that matter are results.

While you might be learning how things work in business, a customer doesn't care about any of that. They just want what they paid for to work. If you or your company can't deliver, then they take their money elsewhere.

LESSON: Treat TAP like a job. If you don't deliver, you're fired!

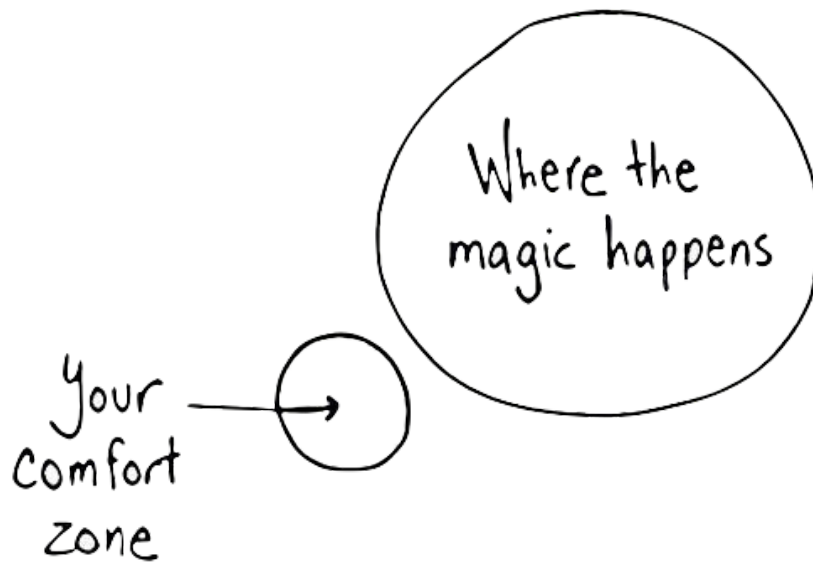
GROWTH IMPLIES CHANGE

You can't expect to grow in your skillset *without actually changing the way you think and act*. What separates a senior developer from a junior? They think and act differently.

TAP encourages you to keep challenging yourself. You'll work with people you might not necessarily like. You'll learn topics that will ask you to change how you think about yourself or others. You'll work on features you've perhaps never have done before.

The trick is to keep putting yourself in these new, albeit uncomfortable situations anyway. The result is that you'll learn a lot and come out stronger every time.

LESSON: Change is inevitable, growth is optional. Be conscious and proactive in developing yourself in the right direction, by getting out of your comfort zone.



Final Words

You're about to embark on a transformational journey. While you will be exposed to various ideas, it's important to realize that nothing is set in stone. The field of software development, as well as business, is ever-evolving.

In order for you to make the most out of your time here, you should always focus on understanding *why things are the way they are*. Only when you can recognise the problems that came before can you think of creative solutions for tomorrow.

If you do this right you'll truly make your learning journey yours.

Good luck,

The TAP Team