STX NEXT
python powerhouse

How to Hand Over Your Project *to a* Software Outsourcing Partner

---

10 best practices for handing over your project
to an external software development partner

Solid project documentation

01

Moving from in-house development to outsourcing

As all the necessary knowledge will be on your side, you'll be able to work closely with your external team to brief them on everything and answer all their questions.

Moving development from one partner to another

You'll have to request documentation from your previous software vendor. If documentation is bad or out of date, and your new partner doesn't have the ability to stay in touch with the old one—that's a problem. Your team will have to reverse-engineer the code to figure out why things were done the way they were, which will cost time and money.

Project tools ownership

02

Moving from in-house development to outsourcing

Make sure you own the project tools. This way you will retain exclusive control over everything once the cooperation ends.

Moving development from one partner to another

If you don't own the project tools, switching to a different partner will be a major headache.

Unit test coverage

03

The more unit tests are added to your code, the better.

Code coverage should always be as high as possible, especially during a project handover, ideally nearing 100%.
It's the best way to ensure there are no unpleasant surprises waiting for you and your outsourced team down the line.

Open lines of communication

04

Moving from in-house development to outsourcing

Things to discuss and problems to solve are likely to keep coming up for quite some time, so the software house should have the ability to contact you when they need a consultation.

Moving development from one partner to another

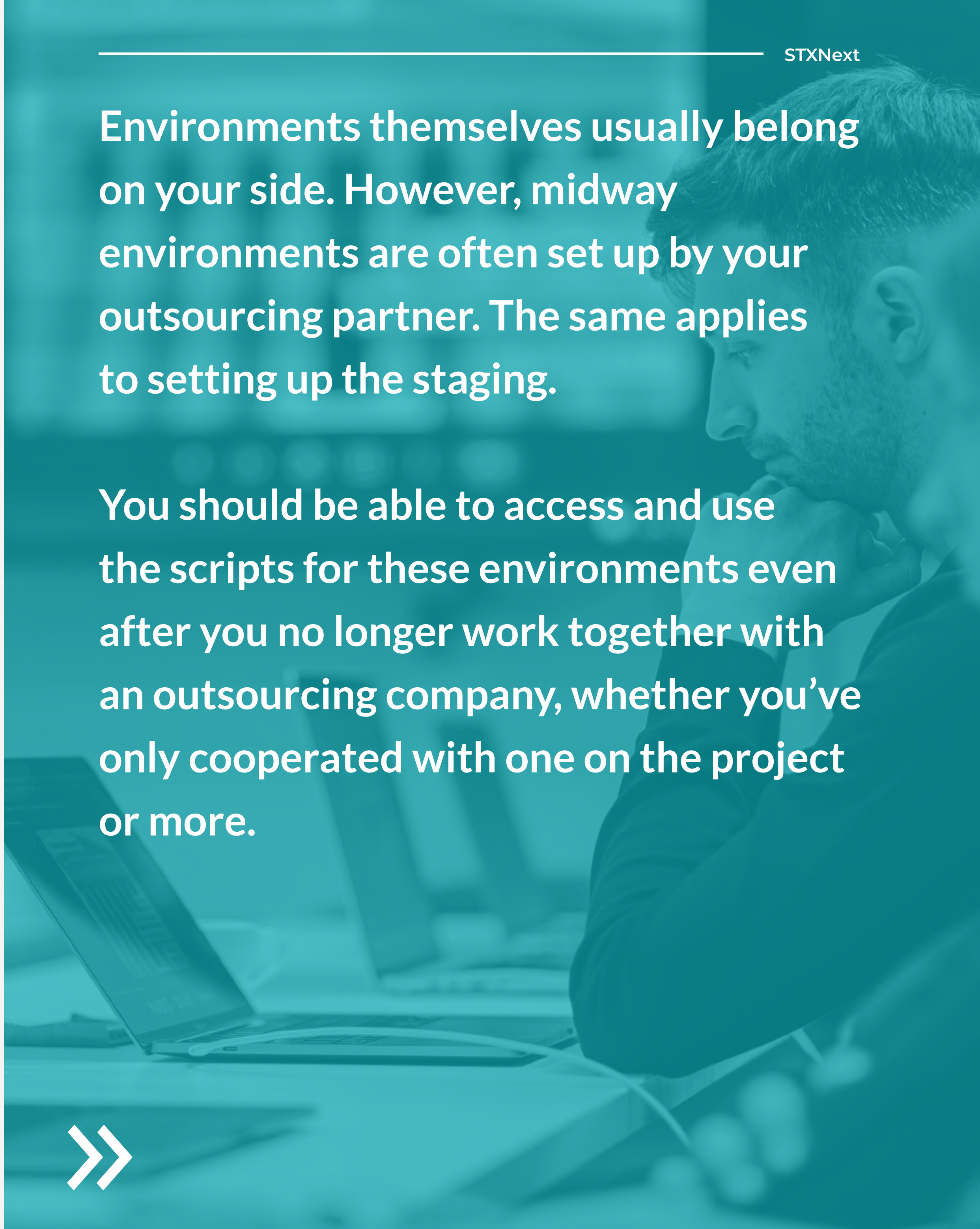
Reaching your previous services provider may be tricky, depending on how your cooperation ended. Consider this kind of support during handover negotiations, or even include it as a condition in the contract termination agreement.

Environment setup scripts

05

Environments themselves usually belong on your side. However, midway environments are often set up by your outsourcing partner. The same applies to setting up the staging.

You should be able to access and use the scripts for these environments even after you no longer work together with an outsourcing company, whether you've only cooperated with one on the project or more.



Multiple Q&A sessions

06

Hold a series of meetings either between yourself and your outsourcing partner, or yourself, your former partner, and the current one.
Such meetings are the perfect opportunity for your new team to ask relevant questions and get the answers they need to take over the development of your software product with a complete set of information.

Before you meet, make sure your partner has already reviewed the code and familiarized themselves with the project.
Also, record the sessions, so that the team always has access to the information.

Testing environment setup

07

Ask your new team to try installing the environment and deploying the code before moving forward with development. Code access alone doesn't necessarily mean they'll have everything they need to use it. A fresh setup often reveals a number of missing pieces.

If the code is deployed automatically, the team will know what to ask about during the Q&A sessions, but with manual deployment, only you or your previous partner may know how to set up the environment, since there won't be any instructions left for your current partner to refer to.

Direct contact information to third-party service providers

08

Moving from in-house development to outsourcing

If you were developing your product yourself in-house or you're starting a cooperation with the first outsourcing partner, contact information to the providers of third-party integrations won't be a problem.

Moving development from one partner to another

If you had a different partner before and they were the ones in touch with the providers of third-party services, they'll also be the ones who have a direct line of communication to them. They need to hand that information over to you before you part ways.

System security audit

09

Moving from in-house development to outsourcing

It's a good practice to have an external company perform a system security audit so that you have absolute certainty they'll be unbiased in their assessment of your code quality and system security, whichever way the handover goes.

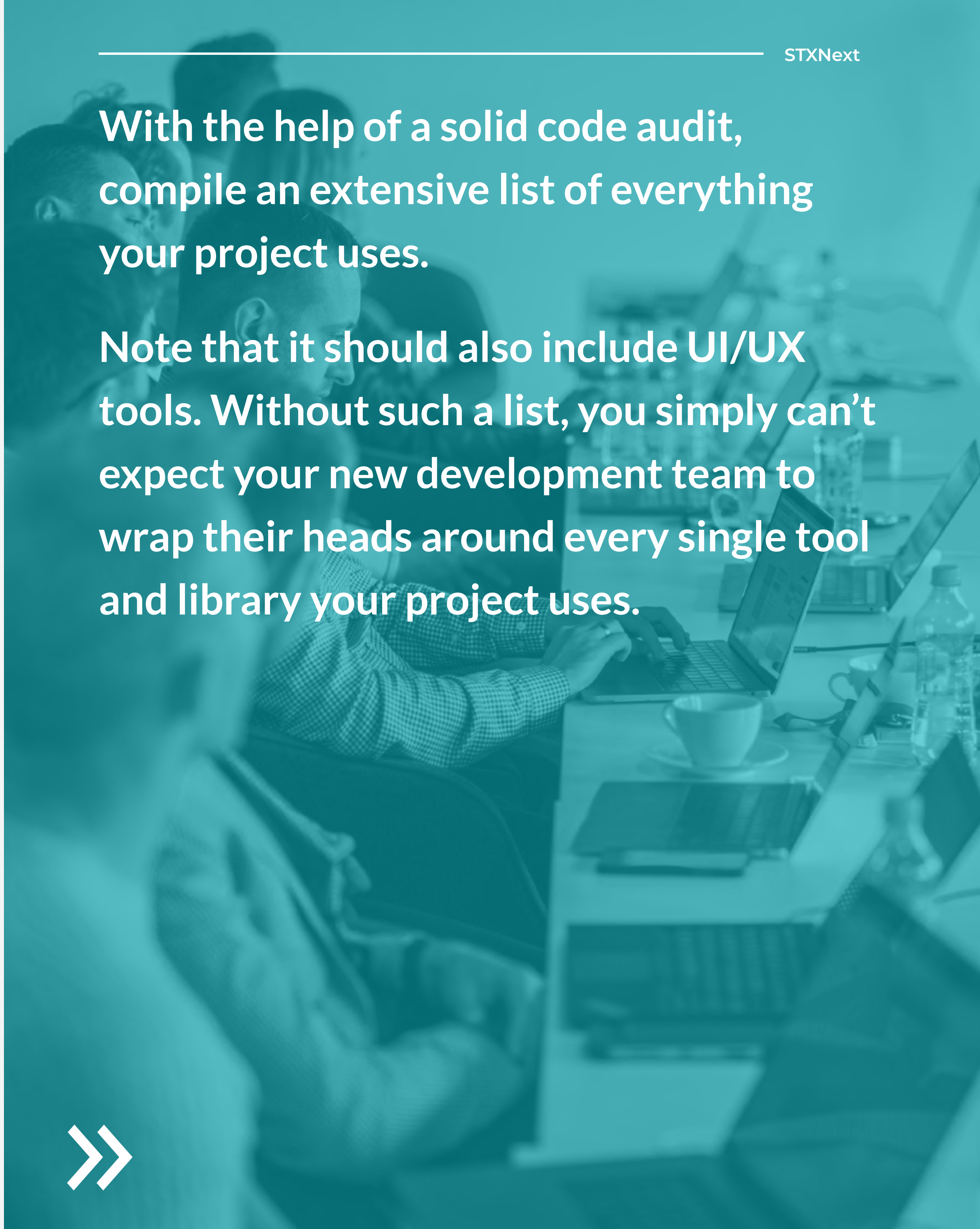
Moving development from one partner to another

A general security audit of the entire system is an especially good idea if you're switching between outsourcing partners. It gives you much-needed clarity, and you'll know what the overall state of your software product is.

The audit ensures there aren't any unwanted surprises left by your previous partner in the code, on purpose or by accident. It also means you won't risk holding your new vendor accountable for any potential mistakes of the old one.

List of all the software tools used

10



With the help of a solid code audit, compile an extensive list of everything your project uses.

Note that it should also include UI/UX tools. Without such a list, you simply can't expect your new development team to wrap their heads around every single tool and library your project uses.

STX NEXT
python powerhouse