

This page discusses the pros and cons of a software developer choosing to work with EvoluData / WikiSuite. We believe that it is overwhelmingly a fantastic choice for a developer. But nothing is perfect, so you should know the overall picture.

# Code maintenance vs coding new apps

Software engineers are trained to write new software, like architects are trained to design new buildings. However, in reality, a major part of the work in most jobs is software maintenance.

Some developers prefer to use (and learn) the latest available tool which is the most suitable for each project. Whereas for EvoluData, the answer is always WikiSuite. The question is: "what needs to be added/fixed in WikiSuite for the success of the project?"

Repeated usage and extending a solution provides predictability in the projects.

And given the components in WikiSuite, this is even sometimes using technologies that are about the same age as the youngest programmers on the team. 😊 For example, Webmin is written in Perl, and is actively developed since 1997: <https://wikisuite.org/Virtualmin>

That doesn't mean there is not a lot of code to write. There are features to add. There are features to extend. And there are features to be revamped. It means that we have to take into account the past, and make sure data migrates to the revamped code. Most of the software development happens in PHP and it has evolved nicely over the years. Please see [A Process That is Not: A Decade later](#)

Please see:

- <https://blog.codinghorror.com/the-noble-art-of-maintenance-programming/>
- <https://www.joelonsoftware.com/2000/04/06/things-you-should-never-do-part-i/>

## If WikiSuite seems overwhelming

WikiSuite is huge, yes. But that is because it covers so many needs. Doing it in a less integrated fashion would be even more complex

- <https://prezi.com/bemzj02mmypm/how-cms-architecture-affects-dev-communities/>
- <https://tiki.org/Coping-with-Complexity>
- <http://pluginproblems.com/>

We do not expect you to become WikiSuite experts overnight. It will take time. We know.

# End user programming

Content creators used to supply content to webmasters, who used to make web pages with HTML and upload them via FTP. Nowadays, content creators just add/edit content in a CMS.

Today still, a lot of developers around the world maintain code (in PHP or in another language) to do stuff (create forms and reports) that we do via the GUI. Tiki is a web database builder and low code / no code platform, as seen at <http://tikitrackers.org/>

Some prefer to do simple PHP code and not deal with an abstraction. But we want to empower the power user. And this is the value provided by the developer and the tools.

# English skills

You will have no choice to improve your English skills. And this is a very important skill. It maybe be hard at first. But you are smart, and once you reach a certain fluency, you won't worry about it anymore. We can help if you ask.

<http://www.catb.org/esr/faqs/hacker-howto.html#skills4>

# Having diversified experiences.

- Working with WikiSuite will not constrain you to just one programming language though. WikiSuite is a full stack, and there are needs/opportunities in many languages: [Liaison with upstream](#)

This being said, everyone needs to have a certain level of experience with Tiki (in PHP) because it touches pretty much every other component.

# Experience

- You will collaborate with very skilled and experienced developers. The core developers of: <https://wikisuite.org/Software-Components>

# Keeping your future wide open

- [Free Libre Open Source Software Benefits](#)
- <https://dirkriehle.com/publications/2014-selected/the-open-source-software-developer-career-and-its-benefits/>
- <https://wikisuite.org/Why-Free-Libre-Open-Source-software>

# About your future work or business with WikiSuite

By becoming a WikiSuite expert, you gain valuable and tangible experience which increases your capacity to quickly help organizations and projects. You could be hired by an organization and identify some needs they have. You can quickly solve those needs with WikiSuite and create value for the organization.

You could also decide at one point to start your own business and use WikiSuite as a component or a base. Given the Free / Libre / Open source nature of WikiSuite, EvoluData is against the concept of non-compete clauses. Specifically, if a freelancer has a side project or client using WikiSuite technologies, this is encouraged. If one day, a freelancer wishes to start a business and be in co-opetition with EvoluData, this is OK. Let's work together to grow the ecosystem! Notwithstanding the "Absence of a non-compete clause" above, EvoluData does ask for a non-solicitation clause: Freelancer will not approach clients that (s)he has worked with while being with EvoluData. In other words, if a freelancer starts a WikiSuite-centric business, (s)he should find his/her own clients. This clause remains in effect for 12 months after the end of the work with EvoluData.

If a developer works in proprietary software, their options are restricted. Ask [Dele Olajide](#) his story for a real World example.

## See also

- [Why your university should teach WikiSuite](#)