Programming languages - what to consider.

Programming tool set selection is a crucial decision IT leaders make when embarking on a fresh development project.

The choices you make will have a long-lasting impact on your development team, its productivity and your ability to deliver on your business objectives.

Some tips to get you started and maximise your chances of success:

Get your team's view, not their vote.

Remain cognisant of the the fact that developers love to explore new technologies and enjoy experimenting with new languages. Their views may be based on the excitement of what the new shiny kid on the block does, rather than considering the sustainable operational benefits of less exciting, established languages that satisfy the bigger picture goals, including long-term support and viability.

Common language support.

Keep your business goals forefront of mind when choosing your application development project's language. Gather feedback from a number of technology stakeholders, who will almost definitely have varying opinions.

Access to experienced developers.

The availability of experienced developers is a prerequisite for success. What level of community support a programming language has? Is talent pool of available developers limited to a handful in your local area?
You want to be sure the pipeline of bug fixes, tools and uplifts to core capabilities are well planned and supported into the distant future.

Information security.

Evaluate the track record and practices of any technology as it pertains to security. Establish how security issues are submitted and reported. All technologies have a history of some security issues; understanding the cadence and resolution history of Common Vulnerabilities and Exposures (CVE) is a fundamental element of any technology assessment. Open source development languages, and their associated communities, have very mature security processes and practices due to their widespread use. This is a real plus.

Early adoption.

While it's prudent to use widely adopted programming languages, there are obvious exceptions. In some cases, languages originating for a specific purpose, in a burgeoning market, can be the right choice, especially if you get in early in the development cycle and build your internal capability based on influencing the language's development standards and roadmap.

Risk appetite.

Every organisation has a unique risk appetite and an appropriate budget to support it. More established programming languages are the safe choice, since many of those with longevity, such as C, C++, C#, and Java, have foundational capabilities that make them a sound option, such as their ability to build foundational hardware support, or their portability across execution platforms.

Programming experience.

The established languages do demand a level of programming experience, rigour and discipline. They can also, with less experience, introduce more complex and esoteric bugs that could see our products published with major security failures or performance problems.

Sourcing expert help when you need it.

Leveraging the knowledge and experience of technical experts, when you need it, will deliver robust outcomes for your business. This is particularly true if software development is not your core business. You could spend weeks, months and years making poor technical decisions that deliver sub-optimal results.

For more detail visit: https://catalyst-au.net/blog/top-programming-languages-selection-criteria

If you feel your organisation would benefit from expert assistance or guidance with selecting the languages that best suit your project's requirements, please get in touch.