ActiveState®

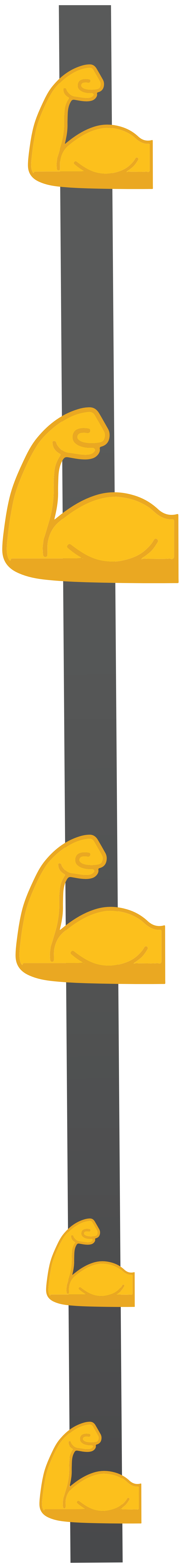
Checklist: Adding a Language

ActiveState has spent 20+ years building, supporting and maintaining open source languages for enterprise. And has learned organizations struggle with questions like:

1. Which language best fits our polyglot environment?
2. What resources will we need to support the selected language?
3. Should we DIY support or get commercial support to decrease risk?

Set yourself up for success. Use this checklist to answer questions like these.

Effort



Before You Start - 10-20% of Effort

Do your research:

- ☐ **Pick a language** with strong community support
- ☐ **Study & learn** with a good book on the language
- ☐ **Determine goal/application** for the language
- ☐ **Find allies on your team** willing to back the initiative
- ☐ **Calculate opportunity** cost of building vs using a commercial distribution
- ☐ **Decide between 3rd party** indemnification or internal legal resources review language and library licenses
- ☐ **Assess risk** to the project/organization
- ☐ **Get sign-off** from required stakeholders
- ☐ **Secure an education budget** (in a "train the trainer" model, at least 500 hrs/trainer)
- ☐ **Plan to set aside regular** blocks of time (2-4 hrs/day) for implementation

Learning the Language - 50-70% of Effort

Reduce time to language fluency by completing the following tasks:

- ☐ **Learn the syntax** of the language
- ☐ **Install a code** linting tool
- ☐ **Study examples** of good code, and then tinker with them
- ☐ **Solve some basic programming problems**
- ☐ **Find a mentor/trainer** to review your code and answer questions
- ☐ **Do pair programming** (preferably with your mentor) to understand how/when/which third-party libraries/frameworks to use
- ☐ **Write a shared library** to exercise what you've learned
- ☐ **Create a small project** as a proof of concept
- ☐ **Educate your whole team** on what you've learned

Finding the Right Tools - 20-25% of Effort

Eliminate the ramp up of new tools & decrease "language adoption pain" by reusing your existing tooling as follows:

- ☐ **IDE -** check to see whether it can accommodate the new language
- ☐ **Code Repository -** ensure it can accommodate the new language
- ☐ **Binary Repository -** ensure it can accommodate the new language
- ☐ **Pen Test/Security Test Tools -** ensure they can accommodate the new language
- ☐ **Packaging Tools -** ensure they can accommodate the new language
- ☐ **Performance Test Tools -** ensure they can accommodate the new language
- ☐ **Governance Tools -** ensure they can white/blacklist the new language's packages
- ☐ **Code Quality Tools -** look for plugins that support the new language
- ☐ **Automated Test Tools -** look for plugins that support the new language

Source for & learn about tooling you will need as 'net new', including:

- ☐ **Build Tools -** ask the community for recommendations
- ☐ **Dependency Management Tools -** ask the community for recommendations
- ☐ **Unit Testing Tools -** ask the community for recommendations
- ☐ **Integration Testing Tools -** ask the community for recommendations
- ☐ **Functional Testing Tools -** ask the community for recommendations

Updating Processes - 5-10% of Effort

Flexibility varies between languages so you'll need to assess how extensive changes will be for two key processes:

- ☐ **Build Process -** create new or modify existing Continuous Integration (CI) scripts
- ☐ **Deployment Process -** create new or modify existing Continuous Deployment (CD) scripts

Empowering Operations - 5-15% of Effort

Don't forget to consider what's required to put a new language into production:

- ☐ **Verify License Use -** does the language distro require a production license?
- ☐ **Consider Commercial Support -** do company policies require commercial support of applications running the language before deploying to production?
- ☐ **Optimize Server Settings -** for the specific runtime (.Net, JVM, etc)
- ☐ **Determine Hardware Resources -** memory, disk space, I/O, etc requirements
- ☐ **Vet Monitoring Tools -** do they work with the new language?
- ☐ **Test Existing Security Tools -** do they work with the new language?

Read More

For more information on adopting a new language in your enterprise, refer to the ActiveState white paper, "The Polyglot Enterprise: Best Practices for Adding a New Language"

www.activestate.com/resources/white-papers/polyglot-enterprise-best-practices-adding-new-language/

website: www.activestate.com
Toll-free in NA: 1.866.631.4581
email: solutions@activestate.com

ABOUT ACTIVESTATE

ActiveState helps enterprises scale securely with open source languages and gives developers the kinds of tools they love to use.
More than two million developers and 97% of Fortune 1000 companies use ActiveState open source language builds including CA, Cisco, Pepsi, Lockheed Martin and NASA. To learn more, visit activestate.com