Five reasons

developers hit a productivity wall – and what to do about it

Second-wind strategy

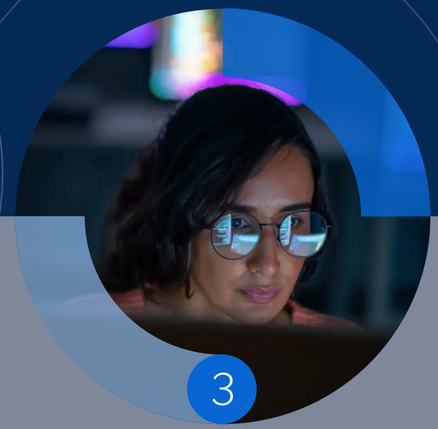
Some marathon runners call it “hitting the wall” – the moment they experience sudden fatigue and a loss of energy in a race. It usually happens near the 20-mile mark. The solution: Brief rest and hydration. Then they can usually find their second wind.

Developers often have a similar experience with data and AI. After a sustained period of early productivity, it flattens out. Productivity gains get a lot harder to come by. The marathon runner’s second wind remains frustratingly out of reach for most developers.

Time after time, the same five obstacles seem to get in the way of productivity gains. In this e-book, we’ll take a closer look at each of them – and show how SAS® Viya® Workbench can help overcome them.

“I don’t have my own space to experiment with models before sharing them.”

“It takes too long to develop a model.”



“The infrastructure we need just costs too much.”

“It’s too hard to keep track of version updates.”

“Everybody’s getting different results from the same data sets.”

“I don’t have my own space to experiment with models before sharing them.”

Think about where some of the world’s most successful engineering projects got their start. A dorm room. A garage. An attic office. These spaces all share some of the same characteristics: They’re “safe” places where smart people were able to create freely, with minimal interruptions, using reliable tools, at their own pace.

The same is true for model development. But if today’s modeling environments were physical spaces, they’d probably look more like Grand Central Station – lots of people bumping into one another in a rush to get to very different places. Developers working in these spaces have more tools and plugins than they’ll probably ever use, working alongside more of their peers than they want.

They need a garage space.

In data and AI modeling, that looks like a streamlined, easy-to-use tool that isn’t weighed down with collaboration-focused plugins and platform requirements. That type of environment gives them everything they need to quickly create a host of models, discover what works well and what doesn’t, then share the most promising, accurate models when they prove viable.

Viya Workbench is designed to allow developers to focus on one job: Creating high-performing models. Deeper collaboration can take place once the models are vetted and published elsewhere.

Everything you need.

Nothing you don’t.

SAS Viya Workbench

- On-demand compute
- Native Python + R libraries
- VS Code & Jupyter Notebook integration
- Data access, prep, quality and integration
- Statistics and machine learning
- Visualizations via code
- Model development

If today’s modeling environments were physical spaces, they’d probably look like Grand Central Station. **Developers need a garage space.**

2

“It takes too long to develop a model.”

Truly great data and AI initiatives that have an impact are the result of constant experimentation and refinement. It’s not enough to create, launch and test a handful of models – you have to be able to do it at scale. That means dozens or even hundreds of models, not a handful. This gives teams the opportunity to observe different model strategies in action, discard those that aren’t cutting it, refine those that show promise, and operationalize those that are clear winners.

Easier said than done. For most, developing a new model today remains too time- and resource-intensive. And after all that work, determining that the model is underperforming can be a crushing blow. There has to be a better way.

Maximum flexibility, minimal IT support

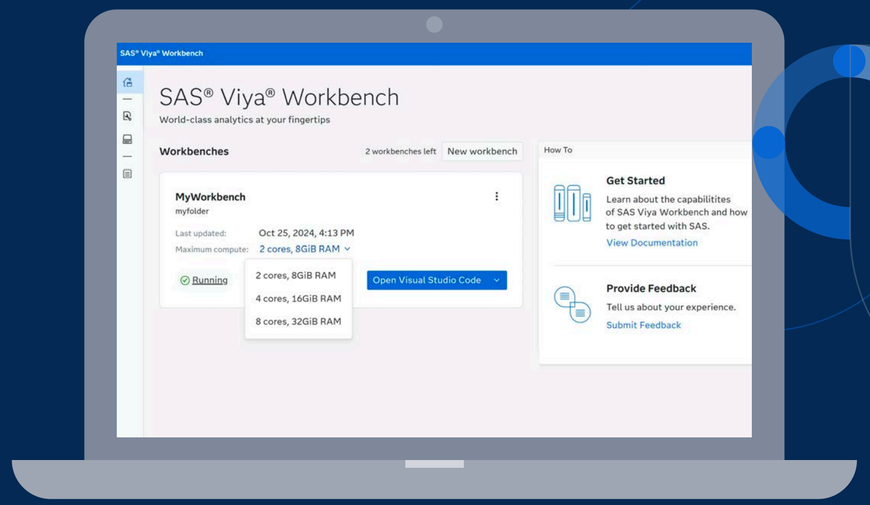
Workbench environments allow developers to create new models quickly, with minimal IT support. Viya Workbench gives developers this ability, with an important twist: The models can be published in a wide range of environments, not just SAS Viya.

Developers aren’t locked into a SAS environment just because they created their models in Viya Workbench.

Plus, it’s easy to configure new projects in Viya Workbench. How big does the environment need to be in terms of CPUs or GPUs? How much memory should it require? Developers can specify the compute size and power needed based on the size of their data and complexity of their use case. Create as many project instances as you want, all in one place.

Already working on an existing Python or SAS project? Just pull it into Viya Workbench – **minimal or no code modifications required.**

Plus, you can choose either Visual Studio Code or Jupyter Notebook as your IDE – so there’s no learning curve for Python users, and no need to learn SAS code.

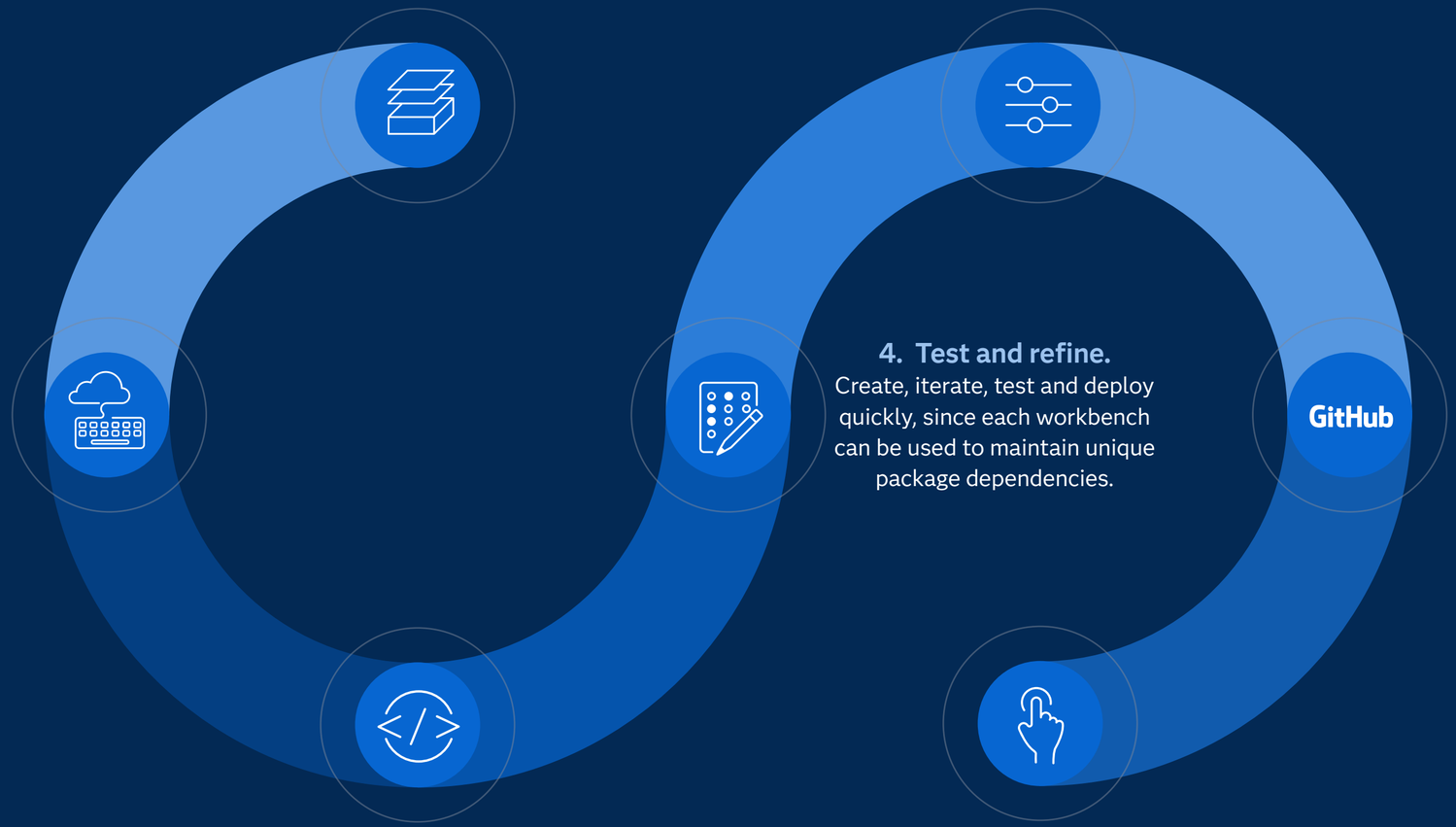


Viya Workbench makes it easy to configure new projects, with a user-friendly interface.

How to complete a project in Viya Workbench

1. Set your preferences.
Easily modify your settings and create more workbenches from the Viya Workbench home screen. Tailor your environment to fit your projects.

2. Set the parameters.
Specify the compute size and power needed based on the size of your data and complexity of your use case.



3. Start building.
Choose your preferred code editor, then build a project from scratch or streamline your work by pulling in existing Python or SAS projects – with minimal to no code modification needed.

4. Test and refine.
Create, iterate, test and deploy quickly, since each workbench can be used to maintain unique package dependencies.

5. Modify the parameters.
Did the scope change? No problem. Easily change the number of cores, RAM and storage size yourself without IT support.

6. Wrap it up.
Button up and commit your projects with easy source version control through GitHub.

7. Press pause.
When you're done, your session will automatically stop or you can manually shut it down. Just pick up where you left off when you're ready – without worrying that your projects have been running up costs on the cloud.

3

“The infrastructure we need just costs too much.”

From servers and cloud hosting to software and staffing, data and AI initiatives can be costly – everybody knows that. Which is why so many good ideas never make it out of the list of good ideas. Decision makers are slow to commit the infrastructure resources they know are needed to make them work, and who can blame them?

Cloud services make it easier for teams to get new initiatives up and running without spending months and big chunks of their overall budgets on infrastructure or waiting for the annual budget cycle to refresh. That’s one reason why Viya Workbench costs so little to use – no big new infrastructure purchases, no complex configurations to meet the demands of a larger platform, minimal ramp-up costs and no training required. Just click a button to create a workbench and start creating models, taking advantage of powerful, accelerated SAS procedures built into Viya Workbench.

No surprises

Experienced developers know to look out for another, less-obvious cost: cloud hosting. There are two ways Viya Workbench helps developers avoid cloud hosting surprises. First, as soon as they create a new project to process data and train models, they can configure it immediately – how big should the environment be in terms of CPUs and/or GPUs? How much memory should it require? Over-provisioning is wasting money, and under-provisioning is wasting time. Scaling up or down as needed is easy.

Second, Viya Workbench is self-terminating, so users won’t incur unnecessary compute costs on the cloud.

Over-provisioning is wasting money, and under-provisioning is wasting time. **Scaling up or down as needed should be easy.**

Once work is stopped, cloud hosting costs stop immediately. Your session will automatically stop when you’re done, or you can shut it down manually. Pick up where you left off whenever you want without worrying that you’ve been running up the cloud hosting bill.

4

“It’s too hard to keep track of version updates.”

Today, most data and AI models rely heavily on open-source tools, for reasons that have been well documented elsewhere. These tools have an important place in the mix – and will for the foreseeable future. But development teams that rely too heavily on open-source capabilities run into some familiar challenges, time after time, starting with the simple issue of version updates. It’s anything but simple to keep up with constant open-source version updates. Multiply that problem across different open-source languages and packages/libraries, across many different data and AI jobs, and suddenly you have a very complex problem.

In Viya Workbench, version control is made easy with GitHub. Build your models from scratch using either Python or the SAS programming language, working in your preferred IDE. Since all development in Viya Workbench can be coordinated through GitHub, source code updates are accounted for and integrated into the model development process. Share through GitHub and manage, update and back up projects with Git version control and tracking. Easy.

When multiple developers in your organization are all using Viya Workbench, they’re all using the same set of libraries and utilities, all with the same updates. So when it’s time for everyone to share their models and collaborate, everyone is on the same page. That’s the power of a common, consistent Viya Workbench developer environment.

When multiple developers are all using Viya Workbench, **they’re all using the same set of libraries and utilities, all with the same updates.** So when it’s time for everyone to share their models and collaborate, everyone is on the same page.

5

“Everybody’s getting different results from the same data sets.”

It’s a simple rule: **If you have the same data, you should get the same results** … regardless of which language you’re using, the compute resources you’re using on the back end, and other variables.

The same data should lead to the same results. But in so many data and AI initiatives, that’s not happening – different environments and programming languages using the same data are generating different results. That’s a problem.

While there are many possible culprits behind this problem, the solution is simple enough.

Development teams can achieve data and AI consistency by using a common platform – one that is able to integrate a wide range of different solution inputs and data types. Regardless of the programming language your developers use, the same underlying analytical tools driving Viya Workbench are consistent – and will deliver consistent results when using the same data.

Start now. Go big.

It's never been more important to be able to crank out a wide range of data processing and AI models at high volume. Which explains why there are so many tools – many of them free – designed for doing just that. But many are so lightweight that they can't be relied on to create models that can be successfully operationalized at scale. Others are so overstuffed with features and broader platform tie-ins that they're too difficult and time-consuming to use.

But what about the middle path: a flexible, easily deployed, easy-to-use, standalone platform that also packs real analytics heft? That's Viya Workbench's sweet spot. If you're looking for a developer tool that allows you to begin creating a lot of viable, high-performing models immediately, with minimal supporting infrastructure and virtually no IT support requirements, look no further. It's all here.

For more on SAS Viya Workbench on AWS, start by visiting this link.

Learn more ▷

Get going with:

Developer canvas
Easy-to-use, lightweight and scalable data science modeling and analytics workspace.

Best-in-class analytics
Latest SAS analytical Procedures (PROCs) and Python APIs.

Open-source coding
VS Code and Jupyter Notebook interfaces. SAS and Python compute runtimes.

Cloud native
Elastic cloud compute, on AWS.

Efficient compute engine
Self-service provisioning and termination with scalable GPU to meet workload.

Industry-specific security compliance
Managed deployment with update and backup capabilities.