Distributed Teams - Making sure everyone is working from the same page

The business world is more remote than it has ever been, and while we will likely see a shift back toward in-office work later this year, it is equally likely remote work will continue to be a common practice. In the world of software development, there are steps that can be taken to improve the productivity of a remote workplace. Here are three tips to best support distributed software development.

Here are three tips to best support distributed software development:

1. Develop a modular architecture.

One of the most critical first steps in supporting distributed development is ensuring the software itself supports this type of development. Research shows that modular codebases lend themselves better to distributed development, a model the open source community has taken advantage of for years. Take the time to collect files in the codebase into their respective modules, define and enforce API usage between modules, and document how modules are intended to interact. Even better, add a layer of software governance to identify unintended usages or API circumvention automatically during continuous integration.

2. Create a robust unit and integration test framework.

Another key step in supporting distributed development is ensuring all developers understand the construction and behavior of their codebase and are capable of making changes confidently to even the most complex code. Particularly in cases where a new developer is beginning to make changes to

an unfamiliar codebase, unit and integration tests ensure that remote developers are capable of updating code confidently without degrading the function of the codebase or creating bugs that surprise your customers. A great place to start is to understand where your code complexity is highest and shore up areas where changes are riskiest in your codebase.

3. Make use of tools to encourage remote collaboration.

Finally, be sure to take advantage of advances in collaboration technology to ensure your teams continue to code effectively. Communication tools such as Slack enable developers to quickly share their thoughts, ask questions, and participate in conversations both within their team and cross-functionally. Additionally, screen-sharing via video tools such as Zoom has made Agile techniques such as pair programming easier and more convenient than they ever have been previously. Take advantage of these tools to establish mentor/mentee relationships, set up internal workshops, and share knowledge from the first two points above.

There are many other techniques for making the most of remote working conditions. However, these are a great start for any distributed software organization. We here at Silverthread use CodeMRI Care internally to define our own software architecture, ensure everyone shares the same understanding of its construction, and identify key areas of our own codebase for testing and validation. This enables our developers to confidently work together in today's distributed environment, and we would be glad to do the same for you.

Contact us to learn more

Know someone who would benefit from this newsletter? Share it with them!