GOAT .tax

Automated R&D Tax Credit Software Platform

CASE STUDY

SOFTWARE DEVELOPMENT

This company develops specialized digital signage solutions requiring significant software development.

R&D TAX CREDIT QUALIFICATION FOR SOFTWARE DEVELOPMENT

This company continuously looks to integrate options that will offer scalable and flexible solutions. Research started with a collaborative conceptualization process, which included a high-level feasibility analysis related to functionality, performance, quality and reliability. Developers followed standardized, but situationally adaptive and precise coding guidelines, to generate code. Solutions evolved through a collaborative, cross-functional effort where developers were assigned a task, or user story, to complete in a single development cycle. The development cycle, or sprint, was an iterative time box in which a user story was accomplished in a span of a few weeks. During this time, the developer designed, coded, tested, and fully refined or optimized the user story. Depending on the complexity of the effort, the team met daily or weekly to track the progress of each task and discuss the technical issues and design challenges.

This software developer with annual sales of $2.7 million realized a combined federal and state tax credit of over $101,000.

A simulation environment was also created to be an advanced computing platform with the capability of creating production-like environments for alpha testing. This allowed the company to evaluate the soundness of the software along with any problems that the development team did not foresee during development. Once it passed simulation testing it moved to limited testing in the field or beta testing. Early software iterations typically would not function or perform as designed so the development team analyzed the test results to determine where the code needed

to be optimized, refined, or completely redesigned. As the software failed, the testing process repeated from the beginning and the software cycled through revisions before it was deemed ready for deployment.

Analysis took place throughout the development life cycle. Each iteration involved the entire team and included full software development cycles (planning, designs, requirement analysis, coding, and testing). Because the applications were unpredictable, these incremental cadences allowed a demonstration of the functionality. The requirements of a user story map to real-world scenarios, framed the acceptance criteria of the project. The goal was to try out different conditions so programmers could write unit tests, one-by-one, to obtain insight at each stage so that the company can review the code base in small modular units. Static code analysis tools were often used while the stories are sized and prioritized so that weaknesses, security vulnerabilities, or potential concurrency issues could be identified. This methodology fosters a no-time-delay strategy between coding and testing so that results can be optimized, refined, or completely redesigned quickly.

Once all iterations of the systematic testing process were completed, the product was released.

THE RESULTS SPEAK FOR THEMSELVES

This software developer with annual sales of $2.7 million realized a combined federal and state tax credit of over $101,000.