

Track Your Game Development with

DevSuite

Integrated Quality Management,
Defect Tracking & Agile Development.





Intro

The interactive entertainment software industry has changed dramatically over the last 25 years, with many of these changes occurring in the last 5 years. Realistic, intelligent and interactive gaming has become more ingrained in the psyche than ever. Just a few years ago, a five-person team could develop and release a best-selling title in just a few months; today's complex projects require millions of dollars and hundreds of team members. Increasing demands from publishers, changing technologies, a combination of licensed and internally developed engines, ever growing webs of art assets, and a myriad of deadlines, milestones, and deliverables add to the barrage of realism that is crucial for success in this highly-competitive field.

With all these moving parts and assets tracking project progress and quality can easily become a daunting and seemingly insurmountable task. Add to this the fact that many gaming companies are forced to release products earlier than desired and it becomes easy to see how user experience can suffer and reputations can be damaged. Quality control has become more important than ever and must be carefully planned and managed for maximum efficiency and cost effectiveness.

The Problem

Quality Assurance for a game needs to begin as quickly as possible. Ideally, game testers should begin working on a title during the alpha phase or earlier. Tests run on these early releases are aimed at detecting major flaws in a game before too much additional code is built on top of them. This is very important since late detection can be extremely costly, especially when a fix requires modifications to deeply embed parts of the game's framework.



It is vital then that game testers be skilled at approaching game testing from a variety of angles to ensure the highest quality product possible. If this isn't done irreparable damage can be done to a company's brand and reputation when a fatal flaw leads to widespread fan criticisms on the Internet and social media.

As if the stakes weren't already high enough, testing of production level (not necessarily release ready) software tends to happen at a frenetic pace. Defects need to be found quickly and early, especially when a big release date is looming ahead. This is often the stage where gaming companies feel the need to cut corners or feel forced to use poorly adapted tools and processes.

Additionally, no standard methodology has been established for game testing. Individual game development studios and publishers typically develop their own testing methods and refine them to meet the needs of their products. Obviously a MMORPG (Massively Multiplayer Online Role-Playing Game) will have different testing needs than an FPS (First Person Shooter) game. However, many of the methods used to test these games are borrowed directly from general software testing principles.

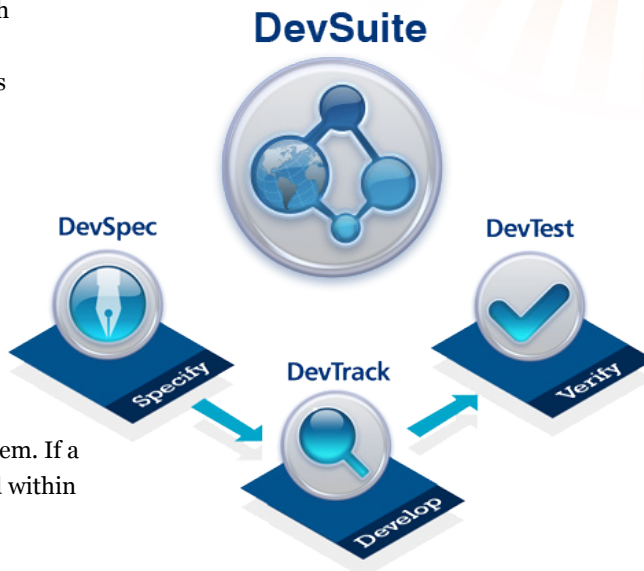


The Solution

Enter quality management that is fully integrated with development and defect tracking. On top of that, integration that works whether a development team is implementing agile, waterfall, or some hybrid methodology.

Manage a Robust Test Library

Quality assurance practices are often perceived as a delay to shipping the product. This doesn't have to be the case though. Creating and managing a library of test cases should be as simple as a few clicks and easily traceable to the requirements that generated them. If a test case fails, defects should automatically be created within the tracking system being use by development teams.



A test library needs to incorporate many types of tests ranging from unit, to load, to regulatory and compliance, to functional tests. A test management tool should be able to handle all of these and more. In the case of functional testing, the quality assurance engineer or game tester needs a simple set of instructions to execute, in game, along with the expected results. TechExcel provides this, as well as a simple, integrated mechanism with which to report test failures to development teams.

Title	Notification subscriptions			Actual Setup Time	0 H 0 M	<input type="button" value="Edit"/>	Actual Test Time	0 H 0 M	<input type="button" value="Edit"/>	<input type="button" value="Save"/>																
Task State	Fail	Task Owner	James Robinson																							
Test Procedure	Users need to be able to subscribe to certain email notification rules. Admins specify the rules to which a team member can subscribe. Users, through the client, subscribe to the rules.																									
Expected Result	<input type="button" value="Edit"/>	<table border="1"> <thead> <tr> <th>ID</th> <th>Test Step</th> <th>Expected Result</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Open notification subscription page</td> <td>confirm page loads correctly</td> <td>Pass</td> </tr> <tr> <td>2</td> <td>Click new notification button</td> <td>Dialog opens and user is able to fill out rules</td> <td>Fail</td> </tr> <tr> <td>3</td> <td>Select notification option: Required. Submit</td> <td>New rule is created with required status</td> <td>In progress</td> </tr> </tbody> </table>									ID	Test Step	Expected Result	Status	1	Open notification subscription page	confirm page loads correctly	Pass	2	Click new notification button	Dialog opens and user is able to fill out rules	Fail	3	Select notification option: Required. Submit	New rule is created with required status	In progress
ID	Test Step	Expected Result	Status																							
1	Open notification subscription page	confirm page loads correctly	Pass																							
2	Click new notification button	Dialog opens and user is able to fill out rules	Fail																							
3	Select notification option: Required. Submit	New rule is created with required status	In progress																							

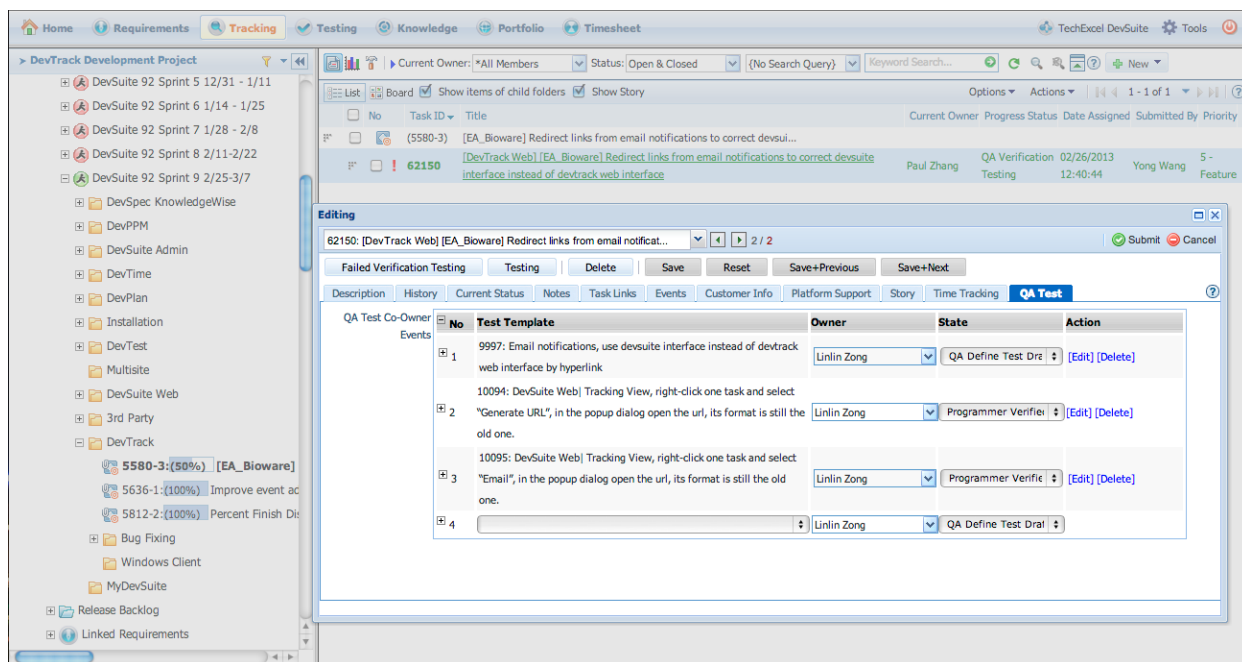
With regards to compliance and regulatory testing, first-party licensors for console platforms have strict technical requirements for titles licensed on their platforms. Sony, for example, publishes a Technical Requirements Checklist (TRC). Microsoft and Nintendo also publish similar guidelines. Among other things, these documents specify standards for error messages, handling of memory card data, and handling of trademarked and copyrighted materials. Ensuring that these standards are met is the responsibility of testers. This is why TechExcel allows linking of all requirements and proprietary documents directly to compliance test cases. These become templates that can be used repeatedly for across multiple releases.



Implement Testing Best Practices

Virtually any team taking on the task of producing a game will have existing processes in place for development testing. Although no team is perfect, they all rely on speed and efficiency to meet the demands placed on them.

Having a flexible test management tool that is fully integrated with defect and development tracking allows teams to continue using their existing processes while adding in a layer of security brought about by traceability and a variety of automated checks and balances. A team may have a great process in theory but fail during execution. With the right workflows in place, however, correct execution can be automatically enforced until it is habitual.



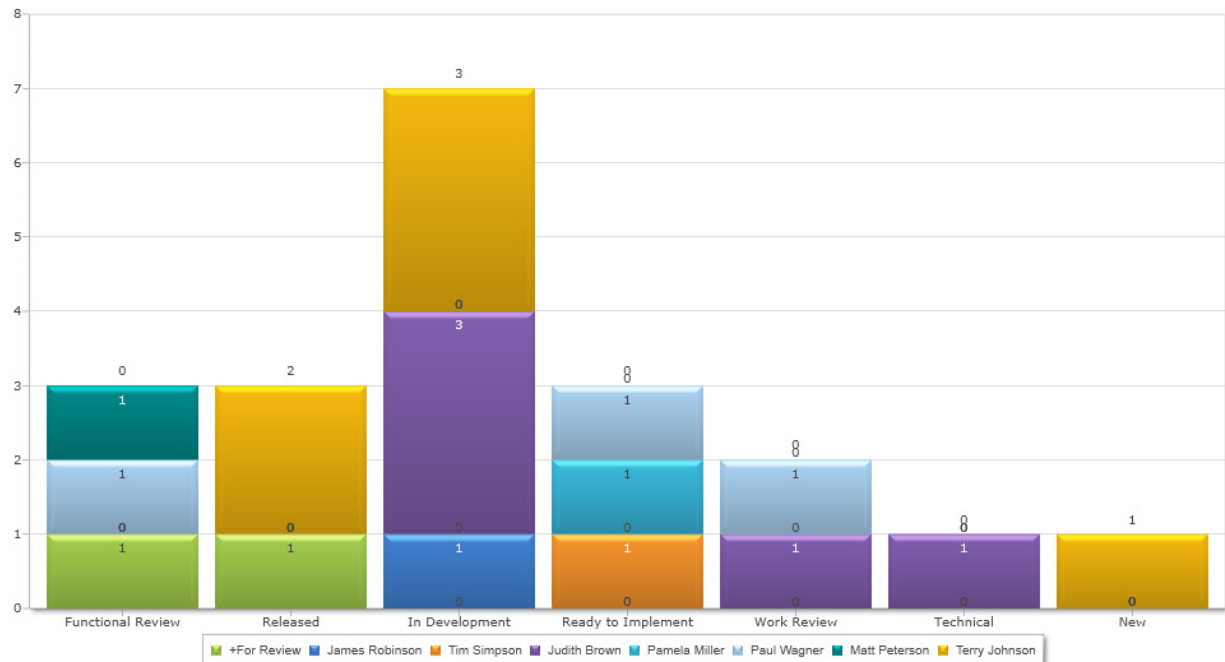
If your team isn't sure about their processes then they can build on the success of TechExcel's many other game development customers by selecting to start from a preconfigured template.



Communicate

Communication is critical to the success of any project. Changes must be tracked so that all team members know their impact. If a requirement changes, developers need to know so the change can be implemented; testers need to know the impact of code changes so they can adjust their testing scenarios; and managers need an overall perspective of the entire project.

Specification State Tabular Report



TechExcel gives teams the ability to thoroughly link all related items. So managers, designers, developers, and testers all know exactly what is expected and are instantly made aware of any changes. Even defects can be linked back to the original design documents. This means that if correcting the defect requires a design change an impact analysis can quickly be performed on all related items.

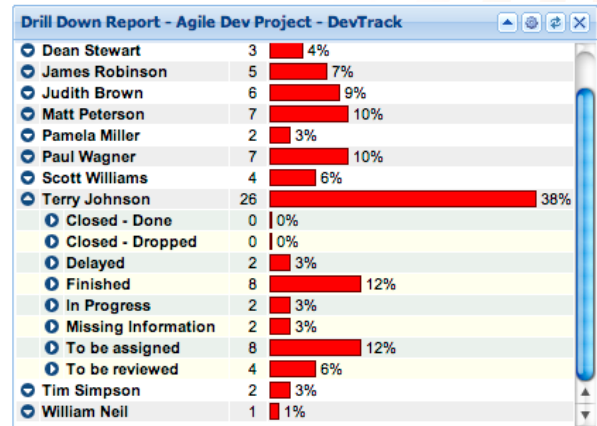
Additionally, automated triggers, events and notifications can be put into place for when a defect is found or an item reaches a review state. These features help keep the entire team in sync even when they may not be interacting directly. This is especially useful when teams are distributed across multiple locations.



Conclusion

Launching a successful game into the market can be an expensive task that takes countless hours of planning and execution. Thorough testing and progress tracking are just two of the ways to mitigate risk during this process.

Shockingly though, many gaming companies do not use commercial grade development tracking and test management software. Instead they try to force cheaper solutions to accomplish what is needed. Essentially building their games without a solid foundation.



Fortunately, when the time is taken to implement a solution, like TechExcel's DevSuite, the rise in efficiency and quality is almost instant. Thanks in part to the ease of communication and total traceability provided.

Just because games are fun doesn't mean they should be taken lightly. Building a successful title is a serious undertaking that requires the right tools. Games have become exponentially more powerful and complex and your team needs a scalable flexible tool that will continue to evolve with your team and its products.

About TechExcel

TechExcel unifies the enterprise by bridging the gap between product development and service / support. This end-to-end environment empowers companies to optimize the relationship between these important organizations without sacrificing autonomy or business processes and provides a collaborative environment that significantly saves time and resources.

Other TechExcel Products						
DevSpec	DevTrack	DevTest	Portfolio	KnowledgeWise	ServiceWise	CustomerWise
Requirements Management	Issue and Task Tracking	Complete Quality Management	Project Planning & Risk Management	Enterprise Wiki & Knowledge Management	IT Service Management	Customer Support Tracking

System Requirements

Operating System: Windows 2000 / XP / Vista / 7
Database: Microsoft SQL Server 2008 and above, Microsoft SQL Express 2000 and above, Oracle 9i and above, MySQL 5.0 and above
Application Server: Microsoft IIS 5.0 and above
Framework: .Net 2.0 and above

Supported Web Browsers: Internet Explorer 6.0 and above, Google Chrome 5.0 and above, Firefox 3.0 and above, Safari 4.0 and above
Supported Virtual Servers: VMware Server, VMware ESX