

CloudTestSoftware.com

Don't be shy! You need a tool to
measure software releases & its
Quality Matrix

August 2015

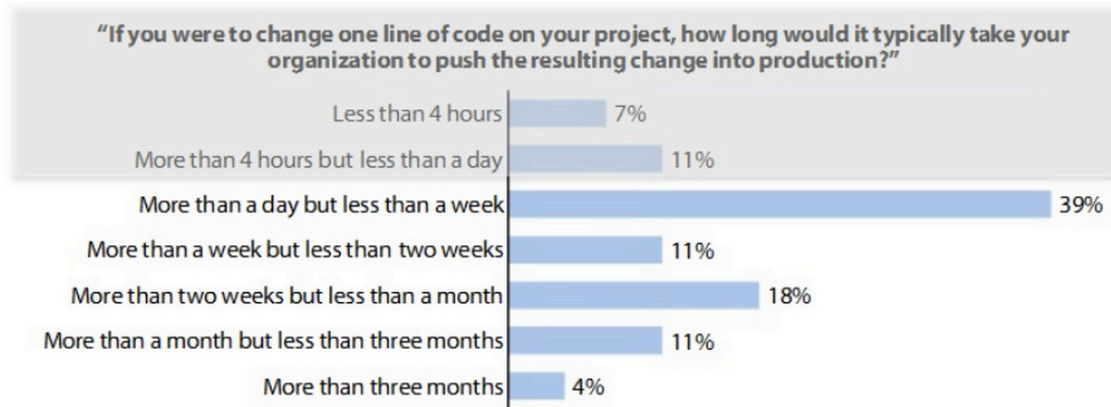
CloudTestSoftware is a young startup company providing Rapid Automation Testing Intelligence in the cloud for Software Product Development and Life Cycle (SDLC) management functions within a single application. Our mission is to provide a single cloud platform for Software SDLC with Test Automation, DevOps, Test & Defect Tracking, Product & Quality Management functions and related services to help our customers to release quality software products sooner to the market with higher quality, traceability, analytics, facts and data that makes them confident with improved profitability

Executive Summary:

Everybody asks the same question, what can I do to release my software timely with highest quality? A [recent survey](#) revealed that 46% of software developers do not perform thorough testing due to lack of time and 36% do not believe that their companies perform enough prerelease testing. Additionally, 88% of software development companies do not use fully automated test systems, opting for less reliable manual testing instead. Last but not least, that 60% of organizations discovering major software errors in production. Research found that 69% of releases planning challenges of software development companies were associated with the priorities and needs of the system stakeholders and the interdependencies of requirements. Additionally, they noted that the release planning process takes approximately 60% of a release manager's time and 30% of the remaining time is spent on re-planning activities due to changes in requirements and stakeholder priorities. It is also found that 97% of software development organizations depend largely on the release manager to ensure successful releases.

As a food for thought, it is clear that a huge amount of software resources and times are spent to manage and plan each software release cycle. But still there is a huge cost of fixing production issues for 60% of the organizations due to the lack of automated tests and missing test coverage against any measurable test matrix in pre-releasing phase of software development lifecycle. There are N number of agile programs and test management tools available in the market but still there is no complete solution to improve release with acceptable quality to the production system.

Cost of change a single line of code to production



82% > 24h!

What is the best solution to address these issues?

Today the software industry needs an integrated SDLC tool that can provide 360 degree view in real time to the complete software development process, to plan, measure and to take timely corrective actions with real-time facts and analytics starting from the day one to day released. It should have intelligence to predict and to allow taking corrective actions even before the occurrence of any such issues. It should also use actual facts and real-time data about projects running in parallel that by knowing their interdependencies in nature and with their current state of development in every phase. This could minimize the amount of time we are spending for planning and re-planning of the release process day after day and also it can improve the productivity of the entire team who could be aware about all facts that likely to surface in ahead of time. This approach automatically distributes the re-

planning actions to multiple members ahead of time by sharing facts with their responsibilities and finding solutions in each unit level rather than taking frequent decisions to alter the release plan as a whole. If you can distribute your challenges and responsibilities into unit level ahead of time, the planning and re-planning process can become very manageable and most likely a manageable solution can be found quickly rather than delaying or changing milestones every now and then. To improve this release process to mitigate its challenges in unit level, you need a process driven SDLC tool that will enable you to drill down to any unit level of issues and provide you facts to likely correct your actions ahead of time such that your complete release milestones remains unaffected.

How CloudTestSoftware can address these issues?

Unit level drill down with measurable facts is one of the approaches to solve many release and software development issues. CloudTestSoftware provides an SDLC tool that enables release manager and product owner to define any deliverables into a measurable unit with respect to individual roles and responsibilities in the SDLC process. Using CloudTestSoftware, the project administrator can define roles and can manage privileges to access different level of information to make their decision making process faster. For example a VP of Engineering or VP of Product Management are mainly interested in knowing facts related to the product readiness level. However a Release Manager or Program Manager focuses on each release level. Similarly business owners and product managers mainly focus to each individual project on which they are currently working on. In the same way Development Manager, QE

Managers, Developers, Testers or Scrum Masters need information related to their respective projects, Epics or Features on which they are working on. There are also cases that much information related to projects or stories, which are not yet ready, may not be published to the entire team until those items are groomed and reviewed. Sometimes product manager needs additional clarifications from the business owner and also needs a round of discussion to finalize the feature details and/or its use cases. Until those facts are understood and defined clearly such information may not be published to the whole agile team who are currently working on the project. In that case product managers or business owners need to protect their current working features and even though the features or user-stories are sharable, they are to be shared when the product manager see it fit or under certain conditions in order to avoid confusions.

CloudTestSoftware provides SDLC tools that can help define unit level privacy using privilege group and sprint planning board. It also provides facts and analytics in each level such that the end users know the facts to the level they are supposed to know and could access.

Release Fundamentals & Planning

In the software release process, the basic fundamental things are establishing certain key milestones with dates to develop, test, measure and release. The key features those are built within a release are priority based. Depending on the business need, the release manager can negotiate with the stack-holders and includes certain deliverables to build and releases within a given time. To optimize the business need

and profits, certain numbers of deliverables need to be completed and should be released as a viable quality product. There could be other deliverables, that may not so critical for release decision but they may improve customer's satisfaction and hence these are also important. As a release manager, the challenges are handling multiple projects at the same time and combining them together into one release. Omission or addition of features are not so important for release manager to release software although it is very important for product or business owner of any particular project unit. Every business owner may try to include all high priority deliverables as early as possible and hence they want to include them in earliest release. The fact is, a release manager looks at each project from various angles, whereas a product or business owner has different priorities. CloudTestSoftware provides the necessary tool to prioritize each of their needs. Release managers can focus to the milestones and how each project is approaching to that timeline, when and with how much progress in every sprint. Whereas product managers looks at how deliverables like epics or stories are getting developed in each sprint and whether they include all necessary features or product behaviors that business owners are looking at.

Priority driven approach and measuring unit to bottom level

When talking about business priority in unit level and how can it be measured, the most important thing about any project delivery is how can you quantify the deliverables with their number of tests against different test matrices that are aligned with different timelines or milestones. For example, for a single sprint the timeline is usually two weeks but for a project it's spread to many sprints. Again for the release

the timeline is more than each project. So depending on the unit size, the timeline changes. Based on priority and stakeholder's need, the system should provide all necessary facts with a measurable timeline against its measurable units. In CloudTestSoftware, we call this measurement a Test Matrix. Depending on the size of measurable units, your test matrix can be defined and assigned to the unit. To make it flexible enough, it also has a quantity of measure that is called test environments associated with a matrix in terms of number of tests. If you want to shorten your timeline against the matrix, you just need to change the unit of measure for a matrix that is number of test environments to be executed for that matrix. For the granular unit, which is normally known as feature or its use cases like scenarios, your test matrix can be very short and it will only last for one sprint cycle. In that case to measure a matrix within one sprint for a unit of feature or scenario, you have less test environments compared to its epics or project that are spread across multiple sprints. So depending of the size of a unit and its timeline, you can assign a longer test matrix, which may have more test environments compared to the shorter timeline. For example, if developers need to define a unit test matrix, he may use only one or two browsers with a very short timeline in the sprint. Whereas if QE Tester does functional testing within the same sprint or in couple of sprints, he/she may test using multiple test environments, browsers or devices for the same functional unit. So the facts or test results that will be produced from different matrices will not be the same to the stakeholders who are responsible to test these matrices and its timeline of testing. Based on this concept, the re-planning or re-alignment to the deliverables are different with respect to the member depending on

the stake-holder who is responsible for, compare to the release manager, product manager, developers or testers in that same timeline but with different roles. Thus the re-alignment of time and responsibilities to any affected issue on the project delivery lifecycle for any functional unit to a developer or tester is faster and quickly adjustable than a product manager or business owner who is responsible for the entire project. In the same way, re-alignment of deliverables in project level is faster than entire release level and so on. The CloudTestSoftware platform is built on this very concept and produces facts related to test execution, planning and development process based on its timeline related to the unit of deliverable with respect to the responsible stake-holder's responsibility such that any realignment to the software delivery process should start from the bottom with its priority to meet its own timeline rather than re-planning the whole delivery process from the top which normally falls down to a new milestone that is decided by the release management and becomes very costly to the entire release and delivery process.

Faster rollup to project within a release cycle

All measurable units and its facts roll up to the release level much faster with improved test coverage matrix and quality by using CloudTestSoftware. If any alignment is needed, release managers can focus on realigning resources, environments, and how to adjust or plan test coverage ahead of time, rather than changing the timeline of software delivery lifecycle. Without re-adjusting the timeline and by adjusting resources within a release, it increases productivity and profit with a timely delivery of software products. CloudTestSoftware produces

such facts that can help release manager to make a faster decision and allows them to align all necessary resources ahead of time. Instead of shifting milestones to new dates, they can focus to add additional resources brought-in from the faster-progressing project or asking developers and testers to realign their priorities to meet the timeline if necessary. Automation is also another remedy and CloudTestSoftware provides hybrid automation platform where browser based testing can be created 10 X faster than any other scripting tool in the market. It is also possible to build automated tests using CloudTestSoftware automation platform just by writing few XML tags and by reducing or eliminating the need for any Selenium, Java or Python code. In today's business world, companies are investing very heavily by hiring automation engineers who can write browser test scripts using Selenium page object model and lots of automation code needs to be written and maintained on an ongoing basis with heavy investment. But CloudTestSoftware's automation platform eliminates writing of script using any such programming language just by writing few lines of XML code; automation engineers can save 1000s of lines of code. Thus CloudTestSoftware can help you to improve your productivity, quality, release cycles and test coverage with a lower cost and resource by stream lining your project delivery process into measurable granular unit with multiple matrices to measure and monitor within single cloud application that can provide all key facts in granular level to make faster decision about your software release cycle.

Release decision matrix and timelines

In every software release cycle there are few key dates associated with milestone. Once the release is planned, normally in a large corporate release, couples of projects are merged together into a single release branch and finally the product is rolled out to production after completing all testing in that branch. To ensure a smooth release, release manager sets some timeline with key dates like, Code-Freeze, Test Pass1, Test Pass2, Staging, and Production Release etc. Depending on the release process, every corporation puts some key milestones around these dates and defines certain acceptance criteria to move from one stage to another in the complete timeline during the release process. For example, if multiple projects are going together into a single release cycle, then all the projects should checking their code into release branch within code freeze date. Similarly, after merging the codes from all project branches, regression testing should be completed by the Test Pass1 date. If there are any defects raised in Test Pass1 cycle then it must be fixed and retested within Test Pass2. In the same way, once the Test Pass2 is completed, code will be deployed in staging and the final round of testing should be completed by Staging date. There could be multiple stage of testing and finally releasing of the software to the production but every stage has its own acceptance criteria to measure its test coverage against its own matrix. For example, Test Pass1 matrix should cover all supported browsers and mobile devices, where as Test Pass2 may be some selected browsers and mobile devices.

CloudTestSoftware facilitates to create such matrix and allows to define acceptance criteria related to matrix with its test environments,

measures and provides require facts to monitor progress and helps to take remedial actions quickly to align resource or addressing issues in the granular level. Once the facts are available to the granular level, a quick turn out can be possible by identifying the level of priority and responsible key member who can work on that priority.

What could be the quick turn around for this issue?

Imagine, you are very close to the last sprint before codes freeze of your release milestone. Now suddenly it comes to the notice, there are some major issues to hit the milestone. There are four different concerns seams to be appeared from four different stakeholders of this release. These are as follows

- 1) The release manager is concerned about the code freeze milestone date and likely not able to hit this milestone because of one of the major project in release is getting delayed. He is thinking to move this milestone to a new date.
- 2) One of the project managers is highly concerned that few of their key features are not yet completed in coding. Developers are asking more time and resource. So he needs to request release manager to change the target date of this code-freezing milestone.
- 3) One of the tester of this project raised concern that he cannot complete his tests because the features are not yet ready. If the feature is not completed in first week of the last sprint, he may not have sufficient

times to complete all testing. So he is requesting to change the milestone date and likely one more sprint needed.

4) The developer of this project is concerned that although they can complete the code but they do not have sufficient time to do unit testing. Since the unit testing is very important, the QE may not accept their code or it can break some of the existing features already tested. So he is also requesting the product owner to delay and re-plan the milestone.

If you look at this above issue, you can see that everybody is concerned about his or her own unit level and they are most like not able to hit the milestone since there are lots of concerns in each level. So what could be the best turn around to this issue and how release manager can tackle this issue without changing the key milestone date?

CloudTestSoftware can provide you a real time fact to resolve this issue and help you to drill down to the actual issue and get it resolved. Only through a click you can navigate to the bottom of this issue and find out what are the amount of unit tests needs to be completed and what is the priority of these tasks based on the priority to these features. It may happen that the feature that is in concern may be priority-2 feature in the release matrix. Or it could be possible that the unit testing environment can be changed and adjusted with the attached matrix to complete the task earliest or can be disassociate to the functional unit since it is priority-2 feature. Or even you can see there are certain percentage of unit testing already completed for some use cases associate with this matrix for this feature and remaining few use cases

may not needed to be unit tested or can be done in next phase of the release cycle. In worst case, you may see there are few resources available from other project that can help to complete the unit testing. Since you can drill down so quickly to the bottom of this issue, there is likely a chance to find a quick turn around about this issue and you can still hit your target milestone.

Facts and number can make your decision making faster

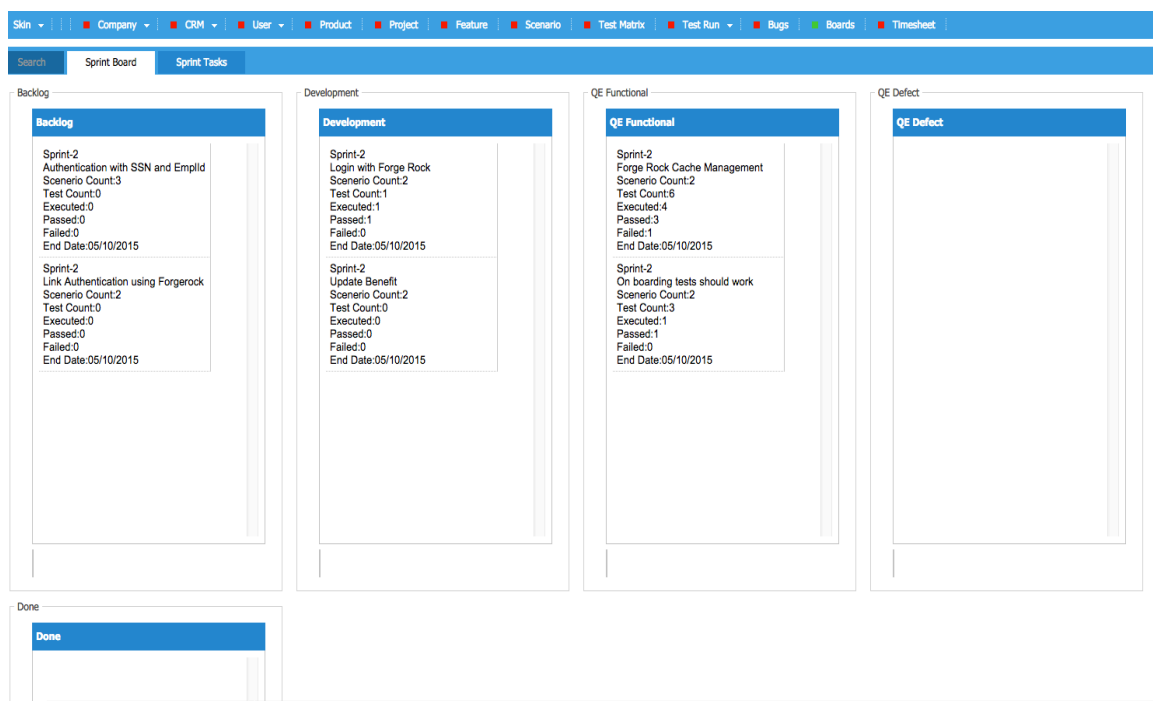
To make your decision-making process faster, you need established facts. Without facts, it is prejudice and you may not be confident to take proper decision. Your decision can be found in fault if the real facts are not available to you. That's why moving milestones or re-planning of release schedule may not be always a correct decision. The release that is getting delayed or re-planned it loses lots of resource, time, productivity, and profits and increases cost. To improve decision making process and increasing ROI, industries needed a complete integrated SDLC tool like what CloudTestSoftware is offering today. You need a tool that not only provides you facts but measures, monitors and helps you to resolve issues by re-adjusting any matrix against its functional unit based on the previous release's established facts that necessary to make a decision change. Another important challenge is the resource plan and its projections for the remaining tasks to complete based on the facts that previously occurred. If you know the amount of resource you may need to complete the functional unit within an established milestone, you can figure out the way to adjust resource or what could be alternative plan to address this issue. If you find out a cost effective way to complete a test matrix in alternative to manual testing

combining with automation, you may likely to choose best possible alternative path and may be able to increase the coverage by using few development resource in the automation. CloudTestSoftware provides automation platform that can build rapid automated tests using XML configuration and may be another fastest way around to address your test coverage issues.

CloudTestSoftware functional units to drill down

Using CloudTestSoftware you can drill down to the facts in multiple levels. You can drill down to the facts related to Product, Release, Project, Epics, Feature, Scenario, Matrix, Matrix mapping to its Test Environments. Also you can drill down all facts related to Project Board, Sprint Board and Test Run. You can also create multiple Test Run using same or different test matrix depending on your need. For example, you can define Test Matrix for Unit Test, Functional Test, Smoke Test, Code Freeze, Test Pass1, Test Pass2, Staging, UAT or Production. Each of these Test Matrix again can be grouped with software sprint planning phases like Development, Testing or Done. If you define a Test Matrix related to Development group, then this matrix will automatically be available for Testing when the features are residing on Development board in that sprint cycle. In similar way, based on your current sprint planning when the feature moves to the testing phase on the sprint board from the development phase, the system automatically shows the test Matrix to run for that feature to test during the period the feature residing on that board. In a nutshell, your test run for each functional unit is completely automated and controlled by the Scrum Master or project owner by moving each item from one board to another. The

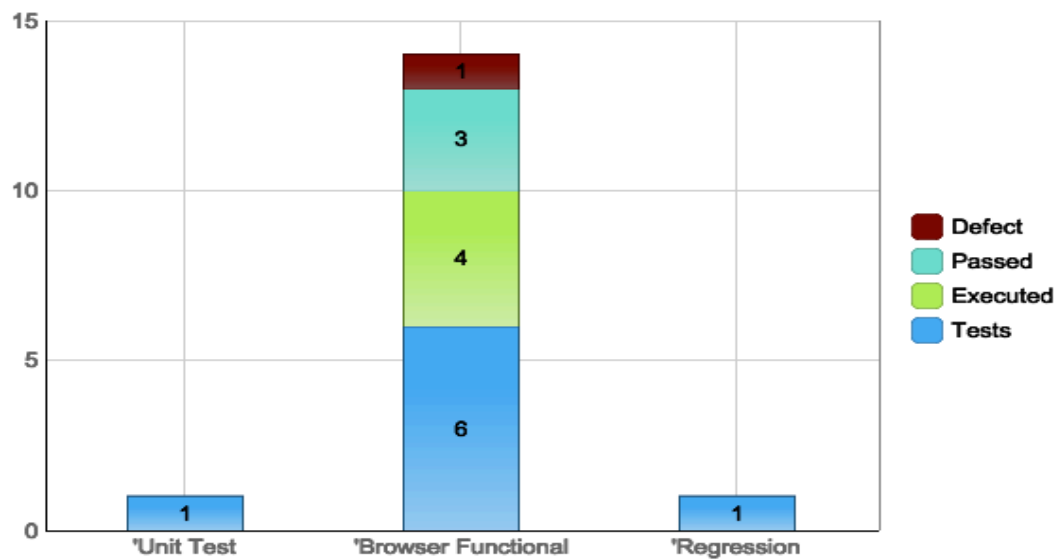
project owner also can see the Test execution status and it counts automatically on the board such that he can make decision to move the item from one board to another when the complete test matrix is executed and all tests passed. It also reduces time during scrum standup meetings since all facts are available on the sprint board and the Scrum team knows each feature or story's test execution status related to the board they are responsible for.



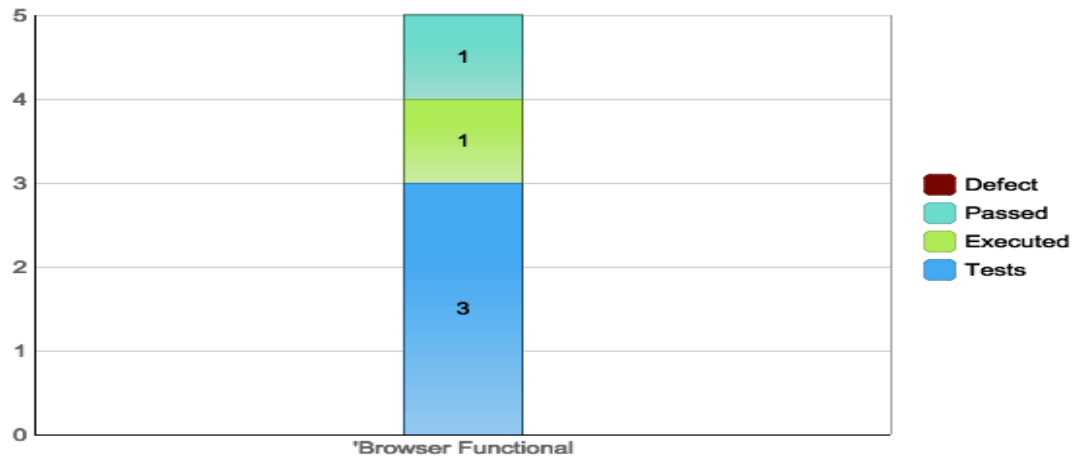
The complete facts on the board automatically rolls up to the Feature, Epics, Projects, Release, Product, Matrix and Board level based on any changes happening on the board for each functional unit. Using CloudTestSoftware, you can even see facts and real time execution of matrix for each feature level based on the functional priority for each use cases or scenario related to Feature and supported Matrices. Below is an example for three Features associated with an Epic but having

different Test Matrix associated for each of them and needs to be tested to move these Features from one board to another during the sprint cycle.

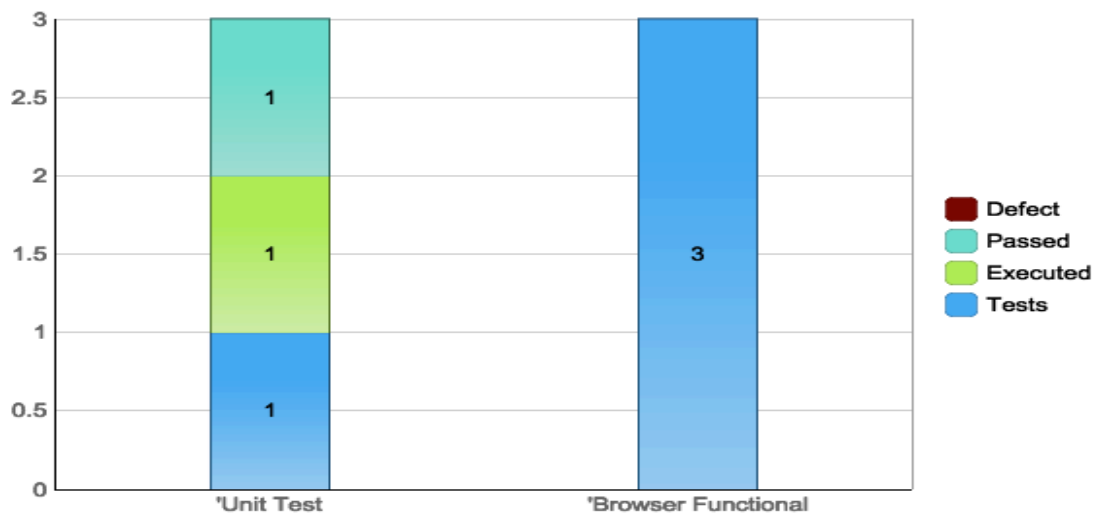
Feature -1: This feature is associated with 3 test matrices and below diagram shows their test execution status against each matrices



Feature 2: This feature is associated only functional test and no unit or regression test matrices are needed for this feature as per the decision of agile planning during the sprint.



Feature -3: This feature is associated with unit and browser functional test matrices. No regression tests are marked for this feature as per agile team decision during planning



Point to Note: Feature -1 looks something to be unusual since although there is one unit test associated with Feature -1, but there is no execution history from the above diagram. It looks, developer did not

complete the unit test and QA started testing of this feature which violates the development process approved by the agile team and CloudTestSoftware is able to catch this unusual process, which needs to be avoided. It also could be work around that, the scrum master moved this feature to the Testing Board on request of developer who may not have time to complete his unit testing or the use case is low priority and risk can be taken to achieve the release milestone target dates through approval of higher authority.

This helps to make decision making faster since all stake holders based on their current roles in the release cycle can access and monitor all kind of facts and matrices appropriate for their level of responsibilities for project they are currently working on.

CloudTestSoftware.com

Freemium

- Project Management
- Sprint Planning
- Test Management
- Defect Tracking
- Time Management
- CRM

Enterprise

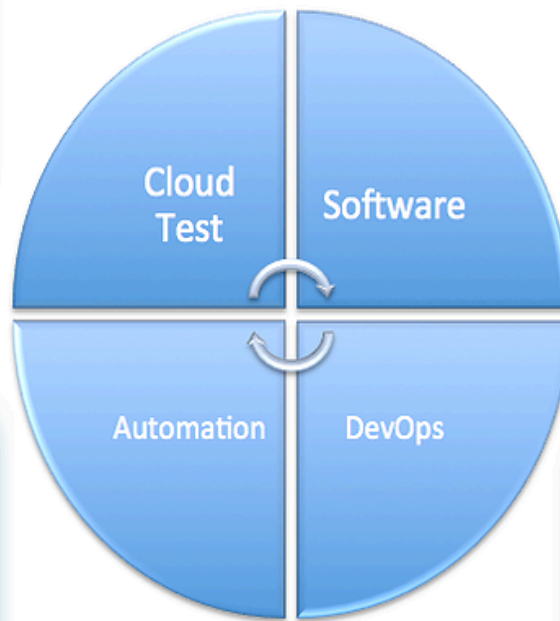
- Freemium +
- Automation Platform
- CI Integration
- Cloud Infrastructure
- DevOps

Services

- Live Go-to-Project
- Automation Scripting
- On-Demand Consultants
- DevOps Scripting
- On-Site Support

Training

- Our Cloud Platform
- Automation Platform
- DevOps Platform
- Live Automation Script Execution & Debugging



Contact Us

Srimanta Jana, President
Artitelly Solutions, Inc
sjana@CloudTestSoftware.com
408-390-9131