PLUTORA

WHITE PAPER

Achieving IT Delivery Assurance Through Effective Enterprise Release Management

Developers have a number of tools at their disposal and testers take solace in tools that help track test cases and manage defects. However, Program managers, IT delivery managers, and delivery assurance managers have been forced to compromise with a combination of high-level project management tools, low-level spreadsheets, and their own ingenuity to help ensure the integrity of IT delivery.

The main tool used by project management offices is a project portfolio management (PPM) solution. PPM tools help program managers and delivery managers to plan and prioritize business requirements and manage the portfolio of strategic programs and tactical projects. However, PPM tools are largely focused on project costs and the overall success of projects and initiatives from an ROI standpoint; they do not offer much in the way of insight for delivery assurance.

While PPM tools can help with overall planning, cost, and resource allocation, there is often a major gap between what is planned and the fitness-for-purpose of what is delivered by IT. A PPM tool might give delivery and program managers an idea of the overall plan and the human resources allocated to a particular project, but it fails to add any governance, foresight or control to the delivery process.

Delivery Assurance isn't a new concept for companies developing and delivering software. The overarching goal - ensuring that new software is delivered on time and on quality, whilst maintaining the integrity of applications in production - has not changed.

However, due to increases in IT complexity and higher expectations of agility and quality, the need for a more holistic approach to delivery assurance has never been greater.

Role specific tools (such as those used for test management, defect tracking or project management) contain useful data but data that is of little use to delivery assurance when used in isolation.

For IT departments to truly achieve delivery assurance, companies must empower program managers and delivery managers with a dedicated tool for introducing orchestrated, end-to-end control of the delivery process and visibility into project statuses, delivery progress and software quality across the entire organization.

The Typical Approach to Delivery Assurance

For most organizations, the delivery approach generally looks similar:

1. Demand is generated for new business requirements and requests are handed over to IT.
2. The project management office captures all business 2 requirements, applications, and features slated for delivery.
3. A portfolio of releases, projects and systems is generated for monitoring by PPM tools.
4. Individual business requirement projects are dropped into the IT delivery factory for developers to code and build artifacts and QA testers to test and promote through various stages|into production.

The issue with delivery assurance generally emerges in this last layer where the IT delivery factory works to produce software. There is a major gap between those responsible for delivery assurance with their high-level planning tools and the progress data generated in the IT delivery factory. The hundreds of delivery checkpoints and data points generated across projects are typically measured and controlled through spreadsheets, PowerPoint, SharePoint, and other applications not built specifically for delivery management. As new iterations of the code/build/test lifecycle continue to pile up, these options become increasingly cluttered and difficult to manage to the point where there is no way to accurately understand the current status of projects without the need for expensive and expansive meetings or lots of manual consolidation of data.

There is a major gap between those responsible for delivery assurance with their high-level planning tools and the progress data generated in the IT delivery factory.

Manual Delivery Assurance Only Assures a Lack of Visibility

As companies continue to increase the speed and volume of their iterative releases, manual delivery assurance becomes even more unrealistic. Consider even a conservative listing of the release and progress metrics that delivery assurance managers must track:

Speed:
How quickly is the release progressing?

Integration Points:
Which other applications are affected?

Quality:
How many defects have been found and what is the defect density?

Capacity:
What is the demand on and availability of resources?

Compliance:
Has the delivery progressed through the correct process?

Risk:
What is the risk of progressing this release to the next delivery stage?

Escaping Defects:
Defects identified after deployment

Deployments:
Number of deployments delivered between sprints.

Output:
Initiatives delivered as part of this release.

To ensure the quality and pace of delivery, each of these metrics should be measured and tracked for each release of a particular application. In addition, these metrics must be updated constantly (as near to real-time as possible) and accurately in order for delivery managers to have any real impact on the delivery process and take preventative action if the project is at risk.

When the need for updated metrics is multiplied by hundreds of features, iterations, and deliverables, manual delivery assurance quickly becomes unfeasible.

Not only does manual management require that delivery managers spend upwards of 20% of their work week updating spreadsheet information and transferring data from one system to another, it also fails to govern input. There's no way to guarantee that the many people entering data into these spreadsheets are actually providing valid inputs which bring additional risk to the software release process.

Another issue that emerges within software delivery is contention for test environment resources. The majority of organizations have a finite number of test environments that must support an infinite number of projects. The result is a scheduling mess in which many separate projects are vying for space in crowded test environments as project managers attempt to meet strict delivery deadlines.

It's nearly impossible for delivery managers to maintain a wide-enough view of test environment utilization and demand, to realize the impact that each project has on the rest of the portfolio. This lack of visibility leads to inevitable bottlenecks, quality risks and delays in the delivery process that can cause serious problems for businesses that rely on agile development of new business requirements and applications.

It's nearly impossible for delivery managers to maintain a wide-enough view of test environment utilization and demand, to realize the impact that each project has on the rest of the portfolio.

The only way to achieve the visibility necessary to implement real delivery assurance is to provide those responsible for delivery with a dedicated tool that can orchestrate and coordinate every project in the portfolio in terms of IT delivery metrics. By creating layers of control and governance for delivery managers, release management processes can become more transparent and efficient.

The Benefit of Improving IT Delivery Factory Visibility

The goal of delivery assurance should be to identify quality related risks to a project and provide insight and recommendations for corrective action so that projects can respond in time without impacting delivery timeframes or compromising the integrity of production.

Delivery assurance tools provide delivery managers with a real-time, holistic view of delivery cadence, quality and potential risk both across the software delivery process and also across the whole portfolio of initiatives being delivered. This visibility helps mitigate delivery risks for projects and across the portfolio as a whole.

Delivery assurance tools provide delivery managers with a real-time, holistic view of delivery cadence, quality and potential risk both across the software delivery process and also across the whole portfolio of initiatives being delivered.

The key to this visibility is to enforce a delivery pipeline with associated quality gates at each step of the delivery process. As the delivery assurance tool tracks gate completion, delivery assurance managers can see a real-time stream of project statuses that are informed by reliable input from integrated tools, developers, testers and managers at each stage of the IT delivery process.

The Five Tenets of Delivery Assurance

The Plutora platform is the only dedicated delivery assurance tool that can ensure visibility for delivery managers and enable real-time tracking of all aspects of the delivery process. To achieve delivery assurance, the Plutora platform supports the five tenets that create delivery assurance—the release plan, governance gates, the environment plan, quality assurance and deployment management.

The Release Plan

The Release Plan function is supported by the Release Manager module of the Plutora platform. Before any developer starts coding or creating builds, the Release Manager creates a plan for the course of the project. The Release Manager bridges the gap between what business stakeholders need and what delivery teams are going to produce. The module compiles data for initial requests and release plans to help delivery assurance managers view all business requirement changes, see all impacted systems, and prepare to track the wide range of releases across the delivery portfolio.

The release plan includes the following capabilities:

- Release Schedule
- Release Scope
- Release Phases
- Impacted Application Matrix

Governance Gates

When delivery and DevOps teams aren't disciplined, they can speed through continuous delivery and continuous integration processes by stepping over quality requirements. This might help meet deadlines, but eliminates any chance at delivery assurance. Governance gates help to protect the integrity of the production estate by ensuring that the correct path to production has been followed and that quality criteria have been met at each phase of the release.

The Environment Plan

As explained above, delivery managers with manual approaches often falter when it comes to understanding the availability and utilization of finite test environment resources. The Test Environment Manager module helps delivery teams to keep track of the quality, availability, and efficiency of pre-production environments across the entire portfolio of releases.

The Test Environment Manager module introduces capabilities that dictate environment readiness and integrity, including:

- Booking and Reserving Test Environments
- Tracking Environment Change Requests Managing Test Environment Contention Environment Scheduling
- Test Environment Configuration Tracking

Quality Assurance

This is where the Plutora platform is able to overcome the disorganization and confusion inherent to manual spreadsheet management of releases. Regardless of the test management tools and defect tracking solutions in use, Plutora can soak up the information to inform the real-time view of project statuses.

Once coded and built, artifacts must pass certain quality gates before moving across delivery phases. For example, prior to user acceptance testing, artifacts must run through 100% of test cases and exhibit zero outstanding severity 1 and 2 defects.

Deployment Management

Deployment management encompasses deployment planning and coordination and provides the last stage of governance for delivery assurance. In enterprise environments, deployments are much broader and more involved than simply installing and configuring software on a set of servers. Typically a deployment includes many prerequisite steps.

They may include documentation, training, planning as well as many executions and post validation actions that include both automated and manual steps that need to be coordinated. Deployment management helps ensure that every step of the deployment is planned and executed according to a defined schedule, resources, and dependencies By providing this level of rigor and by providing collaboration and real-time visibility during deployment, delivery assurance is provided across the whole delivery pipeline.

Delivery Assurance Boils Down to Orchestration and Coordination of All Five Functions

Delivery managers will only be able to keep up with the rapid pace of project delivery if they can achieve total visibility over their portfolio of projects. While PPM tools help manage costs and detail ROI success, they can't orchestrate and coordinate all of the detailed stages involved at the IT delivery factory level.

Plutora is able to compile data from across the delivery portfolio and pipeline to help delivery managers see the progress clearly and accurately in order to preempt any problems and ultimately increase delivery success.

If you want to learn more about how the Plutora platform can optimize your manual delivery assurance approach, contact us now for a demo of the solution and see how lightweight governance and total visibility enables true delivery assurance.

About Plutora

Plutora, the market leader of value stream management solutions for enterprise IT, improves the speed and quality of software creation by capturing, visualizing and analyzing critical indicators of every aspect of the delivery process. Plutora orchestrates release pipelines across a diverse ecosystem of development methodologies, manages hybrid test environments, correlates data from existing toolchains, and incorporates test metrics gathered at every step. The Plutora Platform ensures organizational alignment of software development with business strategy and provides visibility, analytics and a system of insights into the entire value stream, guiding continuous improvement through the measured outcomes of each effort.

PLUTORA®

Learn more: www.plutora.com
Email: contact@plutora.com