



CI for BI

How the Business Intelligence Industry
Can Benefit from Continuous Integration

by Lance Hankins
CTO, Motio, Inc.

In industry terms, BI is still a relatively new field. Like many technology based industries, BI has progressed through its early stages with implementations subject to ad hoc processes and widely varying success. In the past, it has been commonplace for multiple BI projects implemented by the same organization to take wildly different approaches en route to very similar goals. In recent years, however, forward thinking organizations have increased their BI capabilities through the centralization of BI knowledge and expertise. With models such as the “BI Competency Center” (BICC) and the “BI Center of Excellence” becoming increasingly prevalent, these organizations are now defining BI technology stacks, toolsets, processes and techniques for the entire organization in order to ensure success and maximize ROI on new BI initiatives. They are also taking cues from best practices in flanking categories, in this case, the software industry.

One best practice that has not yet been recognized by the BI community is that of Continuous Integration (CI). In the field of software development, Continuous Integration is the process by which a software codebase is automatically built and smoke-tested at frequent intervals – in the development environment. On a typical CI-enabled software project, a “build server” monitors the project’s source code repository and, when changes are detected, pulls a clean copy of the source, does a full rebuild, runs all regression tests, and proactively notifies the development team of any failures. Each fully successful cycle¹ produces an installable set of binaries for the software product.

This frequent, automated integration quickly catches any errors that are introduced into the system (often within minutes of their introduction), and makes it much easier to see who introduced the error and when. Defects and incompatibilities are invariably cheaper to correct when they’re caught within minutes of their introduction (especially if they never make it out of the development environment).

The practice of CI has enjoyed a significant uptake in adoption rate by the software development industry since the early 2000s, thanks in large part to the pioneering efforts of individuals such as Martin Fowler² and Kent Beck. The practice of CI is at the heart of most agile development methodologies:

The Main Principles of Continuous Integration (CI):

1. Repeatable, automated build and test processes.
2. These automated build and test processes are executed frequently so that integration problems are detected very early.
3. The frequent, automated cycles provide early warnings for broken / incompatible artifacts.
4. Near immediate validation and testing of all changes to the system.

There is little dispute that the practice of CI has become an invaluable tool in the arsenal of the modern software development organization. CI improves both the quality and momentum of software development teams. Experienced development teams who have embraced the concept of CI cannot imagine undertaking any sizable software project without it.

¹ A successful cycle is one in which no tests fail.

² Martin Fowler’s original paper describing [Continuous Integration](#) was published in September of 2000.

Could the BI industry also benefit from the practice of Continuous Integration? Absolutely. In the coming years, the practice of Continuous Integration will be recognized for its huge potential when applied to modern BI development environments.

BI ecosystems are inherently complex (see figure 1). They're often made up of a large number of moving parts, with many interdependencies. For example, a typical BI ecosystem may contain:

- Multiple upstream data sources.
- ETL processes periodically extract, clean and load data from each of these upstream sources into data marts or data warehouses.
- Many BI products add a “model” layer on top of these marts of warehouses.
- Professional BI authors build out BI content against this model layer (e.g. reports).

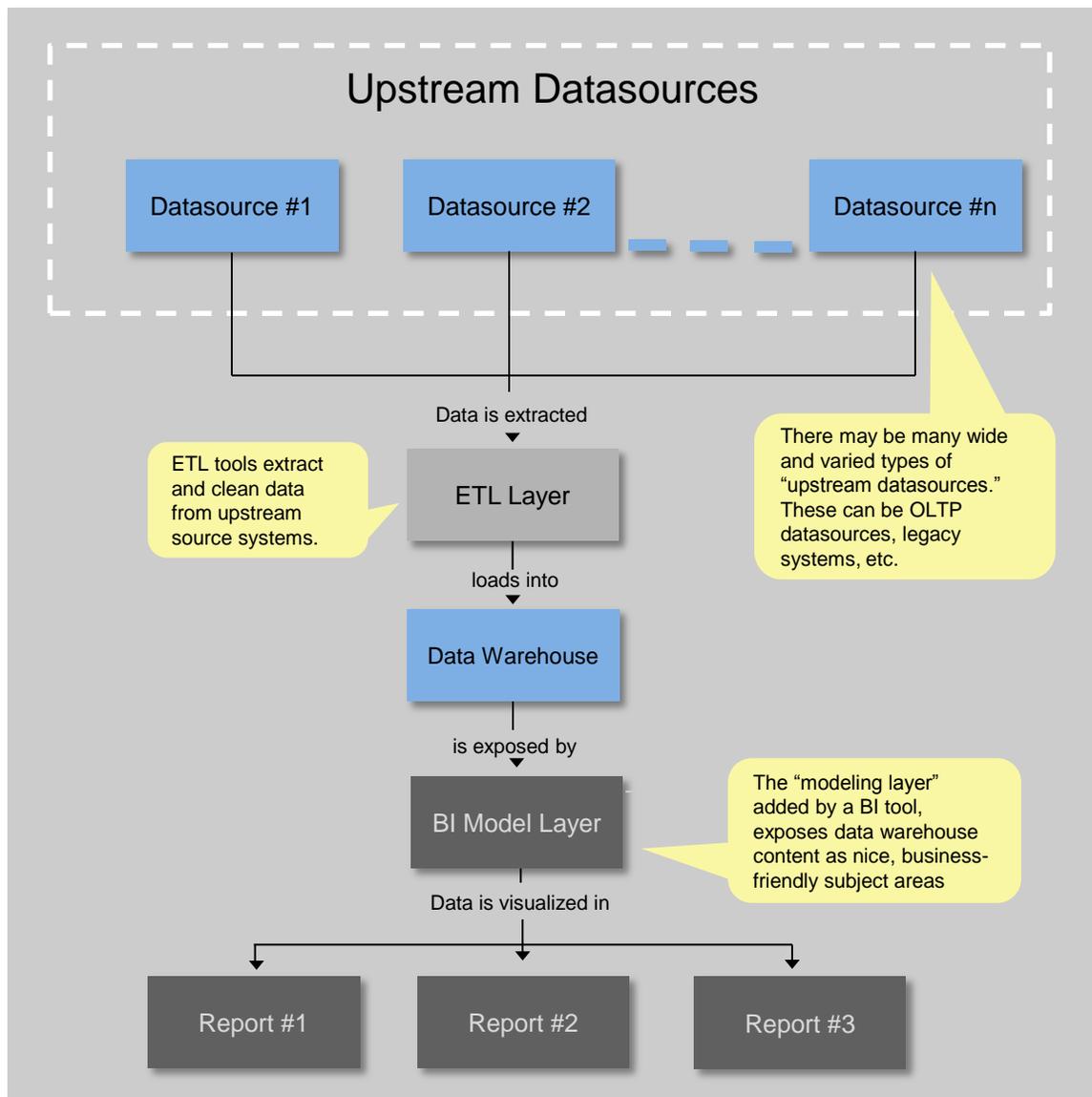


Figure 1: Typical BI Ecosystem

As experienced BI practitioners can attest to – minor changes in any of these layers can ripple throughout the overall system – creating errors or inefficiencies in the resultant BI outputs. Depending on where the BI team is in a release cycle, these errors or inefficiencies may go unnoticed for days, weeks or even months.

Here are a few real-world examples:

- A seemingly innocuous modification at the model layer causes unexpected changes to the numbers for a report that hasn't been edited in months. These changes also degrade the performance of the same report (a condition that's even harder to quantify and detect manually).
- A change in a view in a DB causes a dramatic increase in report runtimes.
- A modeler renames or deletes a column that a report depends on.
- A report author attempts to optimize a report, but the new report does not produce correct results when optional parameters are set.

In most BI development environments, testing of the BI content under development is often done in a very manual way (e.g. "run a report, check the numbers, verify they are correct"). BI teams tend to focus this manual testing on the artifacts³ they are actively changing, rather than those which haven't been modified recently. This tendency lends itself to undetected problems when changes to a lower level of the system begin to ripple upwards and affect many BI artifacts.

Most organizations will periodically deliver baselines of BI content from a development environment into a testing or quality assurance (QA) environment, where they will undergo more formal testing by QA professionals. Depending on the thoroughness of the QA team, defects or degradations in performance may be caught here, but at this point, the cost of correcting these issues has increased considerably. Once a defect has made it out of the development environment (e.g. into a QA environment), it becomes much more expensive to correct. Typical workflow for correction includes creation of a problem ticket describing how to reproduce the defect (by the QA team), BI team triage of all pending problem tickets (to decide which ones get priority), reproduction of the problem in development, implementation of a fix, and then redeployment of another baseline to QA. Likewise, defects discovered in production environments are even more expensive to fix than those discovered in QA.

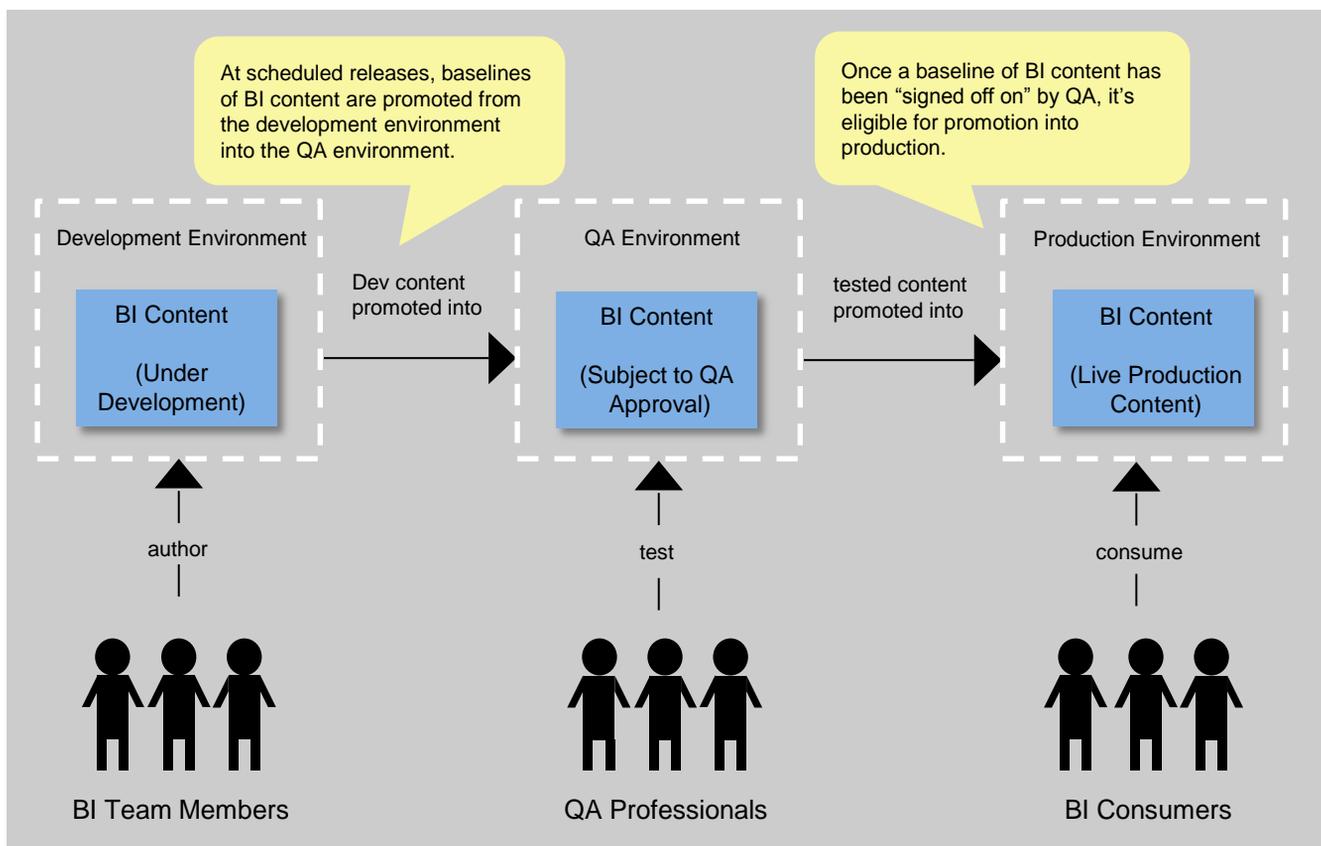


Figure 2: Typical Staged Environments

3 In BI terms - a Report, a Query, a Cube, etc.

Using the principles of Continuous Integration, a BI development team is able to proactively detect issues such as these (often within minutes of the change which caused them), and take corrective action while the BI content is still in the development environment. This means the overall cost of correction is much less expensive.

So how can the principles of Continuous Integration be applied to a typical Business Intelligence project? For some concrete examples, we'll consider MotioCI™, a commercial tool enabling Continuous Integration for Business Intelligence development environments. MotioCI provides BI teams with the following features:

CI for BI:

1. Automated validation of all BI artifacts against their corresponding model. This ensures that any model or database changes don't "break" existing BI artifacts.
2. Automated execution of test cases for each artifact. These test cases can be used to ensure such things as:
 - The execution of the artifact produced accurate data
 - The execution of the artifact produced the expected amount of data
 - The performance of the artifact is acceptable (execution completes in the expected amount of time)
3. Automated consistency checking. For each artifact:
 - Verify that it adheres to the established project or corporate standards for things such as colors, fonts, styles, embedded Images, etc.
 - Verify that parameter names are consistent across artifacts
 - Verify that drill relationships between artifacts are still valid
4. Tracking of BI ecosystem changes so that when a test starts to fail, project stakeholders have a clear view of "who's changed what" since the last cycle. For example:
 - What models have been changed (and by whom?)
 - What artifacts have been changed (and by whom?)
 - Have there been schema changes to the relevant data sources?
 - Have there been drastic changes to the amounts of data in the relevant data sources?

By automating the above process and having it run at frequent intervals, the BI content produced by a team will be continually verified for accuracy, consistency and performance while still in the development environment. If the CI process detects a failure, it will proactively notify the BI team of the issue, as well as catalog the changes to the BI ecosystem which have occurred since the last successful cycle. This method enables the BI team to quickly notice issues created by recent changes, take corrective action and minimize costs.

Net Results of Implementing CI for BI:

1. Errors, inefficiencies and standards violations are caught very early (usually within minutes or hours of their introduction).
2. The BI team gains back countless hours otherwise spent manually testing all artifacts to make sure something hasn't broken, saving time, but also maintaining momentum (it allows BI authors to focus on real development tasks).
3. The BI team gains increased visibility into "who's changing what" in their BI ecosystem.
4. The outputs produced by the BI team are of much higher quality.
5. Upstream QA organizations are able to focus their energies on more high-level testing (all of the "low hanging fruit" is automatically filtered out before the BI content was promoted into QA).

In summary, as the BI Industry matures and establishes best practices in the consolidation, management and application of business intelligence, emerging BICCs should examine and leverage the lessons learned in flanking categories, specifically the software industry. Continuous Integration is not only a software industry best practice, but it is also evolving into a standard operating procedure. As proven practices such as Continuous Integration are adopted, BICCs will continue to mature as a core business discipline by not only improving the throughput of a BI team (critical to scalability), but also by increasing the quality of its outputs. This twofold impact represents a leap in BICC performance and will soon be a mainstay for modern BI environments.



Motio, Inc.
18333 Preston Road, Ste. 475
Dallas, Texas 75252

www.motio.com