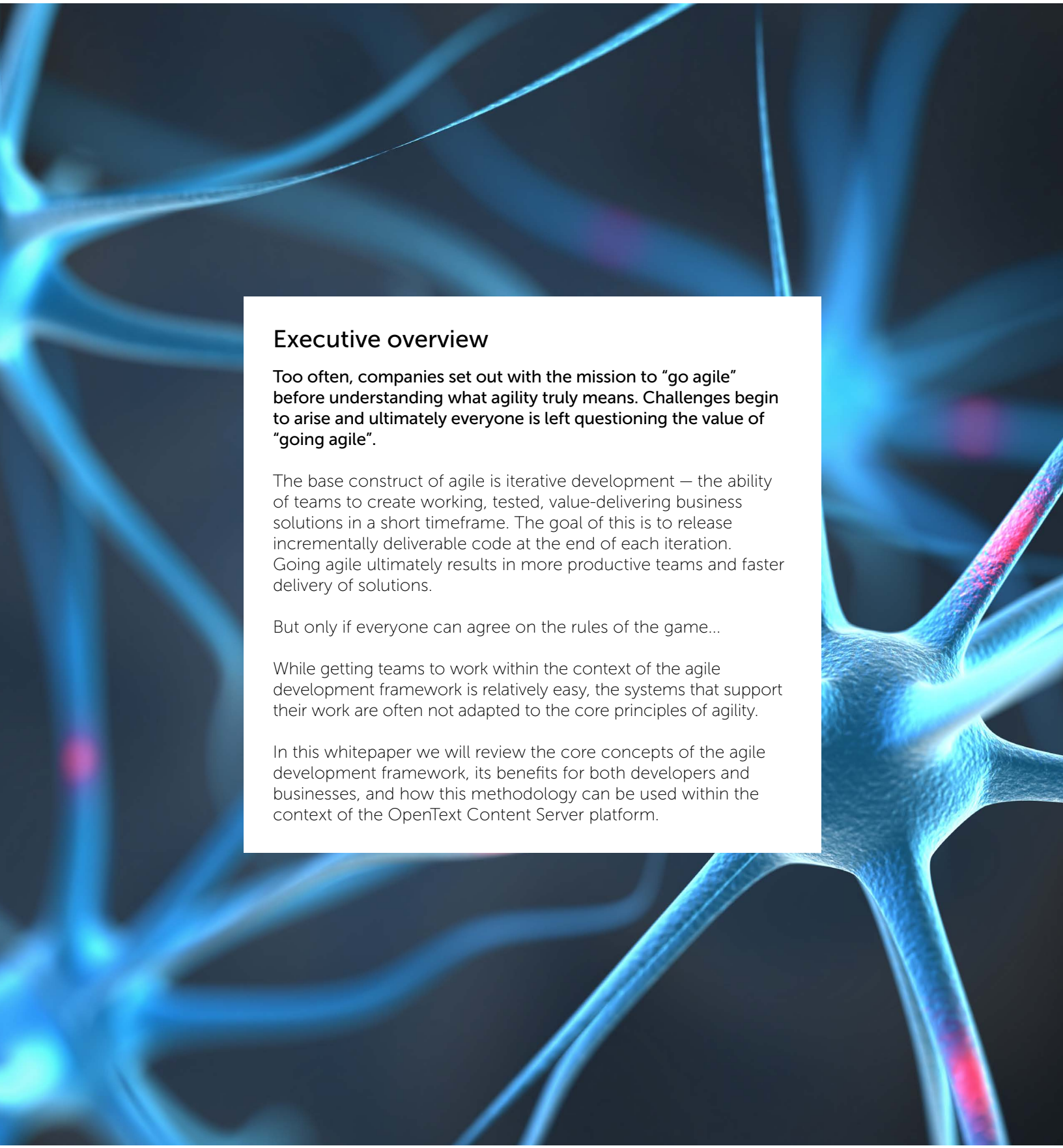
GLOBALCENTS

opentext™ | SolEx Partner Platinum

WHITE PAPER

Delivering Enterprise Agility within OpenText Content Server



Executive overview

Too often, companies set out with the mission to "go agile" before understanding what agility truly means. Challenges begin to arise and ultimately everyone is left questioning the value of "going agile".

The base construct of agile is iterative development — the ability of teams to create working, tested, value-delivering business solutions in a short timeframe. The goal of this is to release incrementally deliverable code at the end of each iteration. Going agile ultimately results in more productive teams and faster delivery of solutions.

But only if everyone can agree on the rules of the game...

While getting teams to work within the context of the agile development framework is relatively easy, the systems that support their work are often not adapted to the core principles of agility.

In this whitepaper we will review the core concepts of the agile development framework, its benefits for both developers and businesses, and how this methodology can be used within the context of the OpenText Content Server platform.

Going Agile with OpenText Content Server

What is agile development?

Agility is about simplifying the solutions to complex problems. At its core, agile is an iterative approach to software development performed in a highly collaborative manner by autonomous, self-organized teams within an effective governance framework. It produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

> "The solutions need to be simpler than the problems."
>
> - Nassim Nicolas Taleb

Every agile methodology engages in an iterative workflow and incremental delivery of working software in short, time-limited iterations (think of it as periodic, small software releases). Generally, during an iteration, many activities occur in parallel. These include requirements, coding and testing. Typically, iterations have a fixed length and are referred to as time-boxed.

While the Agile framework is a singular idea, there are a variety of methodologies to accomplish the same result. There's Scrum, LeSS (Large Scale Scrum), Scrum@Scale, SAFe (Scaled Agile Framework for Enterprise) and CI/CD (Continuous Integration / Continuous Delivery or Deployment). Since we have already focused on CI/CD on our blog, we'll be using this agile methodology as a reference.

How does the agile development framework relate to OpenText Content Server?

Agile is all about breaking down a complex problem into small, incrementally deliverable solutions. In the case of Content Server, as organizations scale their use of OpenText products, the complexity of the business applications they produce significantly increases.

The simple use of documents, categories and attributes is quickly superseded by ever more complex workflows, forms and WebReports. While agility focuses on simplicity, speed and quality, the typical process of integration, troubleshooting and deployment of new modules and functionality in OpenText Content Server requires labor-intensive procedures with a high margin for error.

For example, the standard method of performing migrations between Content Server environments as part of the development process entails countless hours recreating taxonomies and reconfiguring categories, permissions, forms, web reports, and workflow maps.

Producing business applications built around workflows and WebReports involves creating highly complex sets of relationships between one object and others. At a small scale, these can be migrated by manually recreating links. However, when more sophisticated applications are built, even by small teams of developers, this approach quickly fails to scale; mistakes creep in and timescales rapidly explode. Thus the need to be able to quickly integrate changes and automate the deployment of applications within Content Server becomes business critical.

The benefits of CI/CD

The agile framework has proven its numerous benefits across a variety of industries and spanning from the smallest startups to the world's biggest enterprises. Praised by developers, team leaders, managers and all the way up to stakeholders and C-level executives. Below we outline some of the most crucial benefits for each Agile practice:

PRACTICE	REQUIREMENTS	BENEFITS
Continuous Integration	Automated tests for each new feature, improvement or bug fix. Continuous integration server that can monitor the main repository and run the tests automatically for every new commit. Developers need to merge their changes as often as possible.	Less bugs get shipped to production as errors are detected early through automated testing. Building the release is easy as all integration issues have been solved early. Testing costs are reduced drastically – your CI server can run hundreds of tests in a matter of seconds. QA teams spend less time testing and more time improving application quality.
Continuous Delivery	A strong foundation in continuous integration and your test suite needs to cover enough of your codebase. Deployments need to be automated. The trigger is still manual but once a deployment is started there should be minimal need for human intervention.	The complexity of deploying software has been taken away. Your team doesn't have to spend days preparing for a release anymore. You can release more often, thus accelerating the feedback loop with your customers. There is much less pressure on decisions for small changes, hence encouraging iterating faster.
Continuous Deployment	Testing culture needs to be at its best. The quality of your test suite will determine the quality of your releases. Documentation process will need to keep up with the pace of deployments.	You can develop faster as there's no need to pause development for releases. Deployment pipelines are triggered automatically for every change. Releases are less risky and easier to fix in case of a problem as you deploy small batches of changes. Customers see a continuous stream of improvements, and quality increases every day, instead of every month, quarter or year. You speed delivery of business value to your customers.

Implementing an Agile Development Approach Within OpenText Content Server

The four pillars of Agile & how to implement them within the context of OpenText Content Server

1. **Interactions over processes:**
Process and tools have limitations, while the problem-solving capabilities of collaborative teams are limitless. If processes and tools are the primary driver of software development, the teams are likely to be less responsive to change and will likely fail to meet your customers' needs more often. By placing more value on the interactions between team-members and clients, both the processes and tools are likely to be refined to accommodate flexible interactions. If your existing Content Server capabilities do not match the dynamic of your teams, viable solutions ought to be implemented to ensure that the tools and processes are aligned with the development teams and not vice versa.

2. **Working software over comprehensive documentation:**
Historically, enormous amounts of time were spent on documenting the solution or application for development and ultimate delivery. Technical specifications, technical requirements, technical prospectus, interface design documents, test plans, documentation plans, and approvals required for each. The list was extensive and was a cause for long delays in development. Agile does not eliminate documentation, but it streamlines it in a form that gives the developer what is needed to do the work without wasting valuable time, energy and talent in bottlenecks. Agile documents requirements as user stories, which are sufficient for a software developer to begin the task of building a new function.

3. **Customer collaboration over contract negotiation:**
Negotiation is the period when the customer and the product manager work out the details of a delivery, with points along the way where the details may be renegotiated. Collaboration is a different creature entirely. With development models such as Waterfall, customers negotiate the requirements for the product, often in great detail, prior to any work starting. This meant the customer was involved in the process of development before development began and after it was completed, but not during the process.

   Agile takes a different direction - it describes a customer who is engaged and collaborates throughout the development process. This makes it far easier for developers to meet the needs of the customer. The constant feedback-loop between development teams and customers ensures that product delivery is continually refined throughout the development process, thus significantly improving its quality and alignment with customer needs.

4. **Responding to change over following a plan:** Traditional software development regarded change as an expense to be avoided. Contrary to this belief, Agile focuses on the shortness of an iteration, which means that priorities can be shifted from iteration to iteration and new features can be added into the next iteration. Agile's view is that changes always improve a product and provides additional value.

Existing Content Server limitations

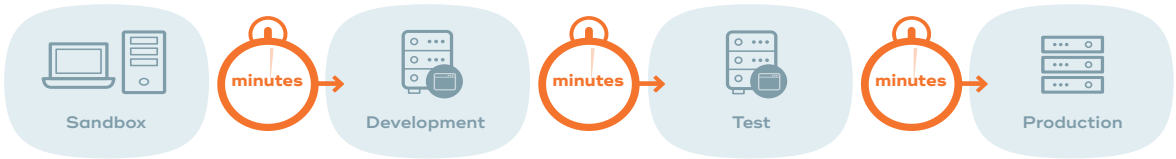
Software development is tightly aligned with the systems and tools implemented in any given organization. As mentioned above, contrary to the infinite human ability to create and innovate, systems and tools have limitations. And OpenText Content Server is no exception to this rule.

a. **Scalability for more complex agile processes:** OpenText has a long history of developing excellent ECM products - some of their core solutions are older than the agile framework, therefore many OpenText products are not easily adaptable to complex agile processes and require expensive customization or third party applications to help handle agile practices at scale.

b. **Lack of automation for deployments between environments:** One of the biggest limitations of Content Server is the inability for automated transfer between different environments. Development – Test – Production is a highly recommended IT best practice, which requires automation especially for more complex applications.

c. **Manual & error-prone approach:** Following up on the argument above - the lack of automated deployments requires users to manually transfer the application from a development environment to the testing environment and then to the production environment. The current method for accommodating this migration entails countless hours recreating taxonomies and reconfiguring categories, forms, web reports, and workflow maps.

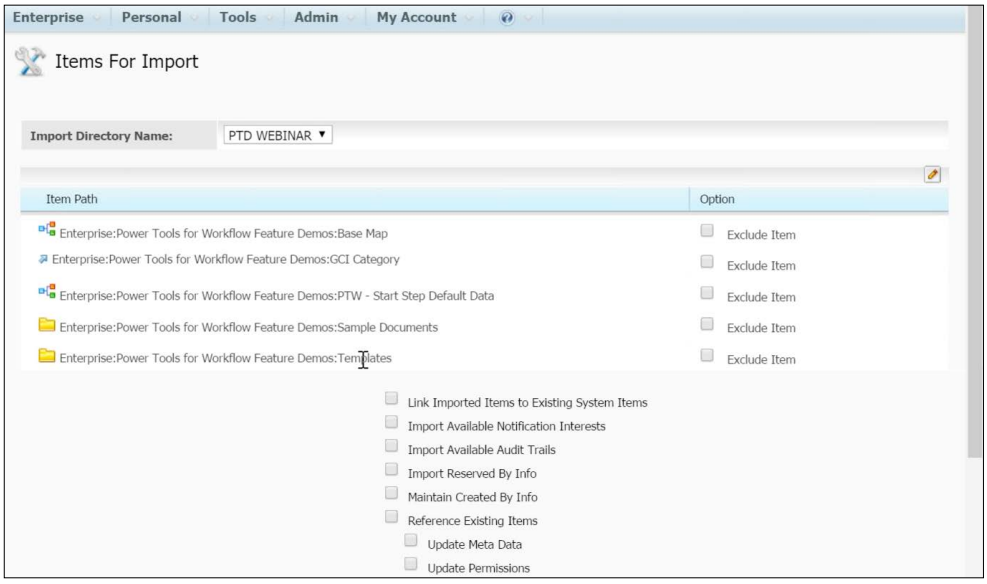
MANUAL CI/CD IN OPENTEXT CONTENT SERVER (>2+ WEEKS)



PTD CI/CD IN OPENTEXT CONTENT SERVER (<1 DAY)

Overcoming existing Content Server limitations with GCI PowerTools

GCI PowerTools for Deployments is the perfect solution for automating the deployment of Content Server applications between server environments. Its simple, web interface enables developers to select the small sections of an application they are working on and deploy them within seconds from a local development instance to a central build validation server.



The powerful migration engine within PowerTools for Deployments takes care of any background mapping processes, ensuring Content Server objects such as users, groups, forms, templates, workflow attachments and other relationships are maintained as the application is moved to the new platform.

As the process continues, PowerTools for Deployments also functions within the deployment phase. Using automated background tasks, an entire application can be selected and migrated from a central build server through to a testing machine overnight, with minimal manual intervention. Using this approach, CI/CD within OpenText becomes not only possible, but a very practical approach to developing WebReports and workflow based applications.

Handling complex agile processes & effective continuous deployments with GCI PowerTools

GCI PowerTools for Deployments enables you to move documents, metadata and taxonomies between environments, while maintaining object relationships as well as original permissions and audit data in a safe, efficient and error-free manner.

It is the most comprehensive solution for all Content Server information transfer problems. GCI PowerTools for Deployments solves these problems by exporting the entire object structure into a set of XML files, and then supports moving them into the new environment together, while recreating the entire structure and updating references and links in the new Content Server instance.

With PowerTools for Deployments you can move workflow maps between instances, while maintaining attachments, performer data, categories, form links, data stored and related database tables. You can transfer physical objects retaining all customer attribute data, as well as records and classification data. Or take a complex hierarchy of related objects from an old version of Content Server and easily recreate it within the latest version. Here are some of the more crucial capabilities of the tool:

- Maintain all references between imported objects as well as references to users following import;
- Maintain circular category dependencies;
- Forms and form templates can be exported and imported maintaining not only the template links, but also the view links;
- Form data and form data stored in tables can be exported across systems;
- Workflow maps are updated to re-link all forms, categories, objects, and attachments after a successful deployment;
- PowerTools for Deployments also supports exporting user and group items created during development and recreates them on the new system, as well as remapping the user IDs;
- Ability to merge and split Content Server instances;
- Supports audit and creation information

If you'd like to learn more about using CI/CD or any agile-based approach for deploying your Content Server applications, then **get in touch** and we'll show you how easy it can be.

To learn more visit our website globalcents.com or contact us at info@globalcents.com

GLOBALCENTS

Global Cents (GCI Live LLC) Tel: 1-847-348-9121 Email: info@globalcents.com Web: www.globalcents.com