OPEN SOURCE AND WORKLOAD AUTOMATION

The success of open source

"Every company is a software company" says Satya Nadella, the CEO of Microsoft. This is evident when you look at the size of organizations' software development teams. Amadeus IT Group, a global leader in the travel industry, has a development team of 5,000+ developers. Goldman Sachs has an army of 11,000 engineers and programmers. It's no longer just about purchasing and deploying software. Digital organizations today are aiming at differentiating themselves through software that meets company-specific requirements.

In this development-driven market, open source is emerging as the new normal in enterprise software. After some initial skepticism, open source is now widespread, and it is adopted to support mission-critical business processes and is largely embedded within commercial products. Right now, you are likely running more open source than you might think.

Reasons for adoption include:

- **Control –** open source can be modified for any specific organizational need, enhanced and shared in a collaborative community.
- **Agility –** open source code promotes the principles of agile methodology: open collaboration, transparency, streamlined process, frequent delivery, immediate feedback. Therefore, it can be easily changed at the speed that the business requires.
- **Collaboration / Longevity –** unlike custom applications, whose life cycle depends entirely on

internal skill-set and terminates when the skill-set departs the organization, open source life cycle is determined by the ecosystem around it, and when the ecosystem is successful, it can last very long and evolve quickly.

However, cost is still cited as one of the most significant benefits to adopting open source. But the open source software cost, exactly like the commercial software cost, is much more than just the initial licensing cost. When considering costs, you should consider the total cost of ownership, which is the long-term cost, and include any additional costs associated with supporting and maintaining the software.

Open source total cost of ownership may be influenced by many elements, such as:

- Software Development Life Cycle costs – any skill and training costs, hardware and software costs associated with open source SDLC
- Additional license costs – open source enhancements may be charged by the community
- Additional maintenance costs – open source may come with periodic payments or a one-time payment
- Additional support costs – open source may be charged for consulting and technical support

What software categories are best suited for open source?

There are no better or worse suited categories. Any software category can be potentially well suited for open source. If you look around, you can find open source solutions in operating systems, i.e. Linux; infrastructures and servers, i.e. Apache, OpenStack, Ken; databases, i.e. MySQL, PostgreSQL; ERP, i.e. ERPNext and MixERP, and many more.

But the best-served software categories, where you can find a vast choice of open source implementations, are: development languages and environments, big data and devops. In those categories there are the most popular open source cloud technologies. Some examples below:

- **Development languages and environments:** PHP, Rails, Perl, Node.js, Eclipse, etc.
- **Big data:** Apache Hadoop and its sub-projects Hadoop Distributed File System (HDFS), YARN, MapReduce, Spark, Hive, Impala, Kafka, Pig, etc.
- **DevOps:** Subversion, Jenkins, Docker, Cucumber, Jasmine, Ansible, Chef, Puppet, OpenShift, Cloud Foundry, Kubernetes, etc.

How does open source play into workload automation?

How does open source play into workload automation and its related job scheduling and application workflow orchestration disciplines? First, as these three terms are often used interchangeably, I'm going to try and clarify what I believe are the major differences between them.

- **Job scheduling** tools, typically platform-based, automate and monitor the execution of job flows (i.e. concatenated Linux scripts, Windows commands, mainframe JCLs) triggered by calendar rules.
- **Workload automation** tools elevate job scheduling to the highest level where business applications are automated across multiple platforms and systems, in a dynamic, event-driven and SLA-driven approach.
- **Application workflow orchestration** (or digital business automation) represents the last evolution of automation. It brings the automation into new frameworks and scopes, simplifying

the complexity of data analytics, and delivering application agility, across on-premises as well as elastic and ephemeral multi-cloud and containerized infrastructures.

While job scheduling tools are essential to drive IT efficiency, application workflow orchestration tools are a must-have for the business to simplify the complexity produced by the [digital transformation](#) and drive the success of digital initiatives.

For this reason, the three disciplines are widely populated by open source implementations – as well as commercial tools. Below are some examples of open source implementations:

- Airflow, Activeeon, Azkaban, CloudSlang, Luigi, ARM Workload Automation, SOS JobScheduler, Schedulix, Fenzo, OpenLava, Google Cloud Composer

The list is not exhaustive and continues to grow fast.

Build versus buy

Choosing among open source and commercial solutions becomes a build versus buy type of question, for which there is not a single answer or a scientific formula available for selection process.

However, you can apply some logic to the equation if you consider three key points and decide where you stand on them. Without pretending to be exhaustive, here are the three points you should consider when making the decision:

Understand your business need

What problems are you trying to solve? What business scenarios are you envisioning that require job scheduling, workload automation or application workflow orchestration to succeed? How important are those scenarios for your business? Which capabilities are key to implement those scenarios and comply with regulatory requirements? How urgent are your business scenarios and how fast you are projecting to grow?

Consider all the costs

The above analysis will form the basis to evaluate costs and risks of adopting an open source solution. If your open source solution does not cover all of your business needs, then you need to conduct a feasibility and investment assessment of implementing and maintaining extra coding.

For example, Airflow, Luigi, Azkaban open source have been conceived to manage Hadoop dependencies and data pipelines, respectively in AirBNB, Spotify and LinkedIn, and they are still focusing on data analytics scenarios. If you plan to use them to manage core business functions, like supply chain management, billing, and invoices, you may need to add new capabilities and integrate additional applications and platforms such as SAP, Oracle, and mainframe.

As another example, Schedulix and OpenLava deliver capabilities that fall in the job scheduling category more than the application workflow orchestration category. If you intend to expand their application into big data or devops scenarios, you may need a lot of extra coding.

Extra coding is not a problem in se, in fact it enables us to shape products on business needs and drives innovation and competitiveness. However, its cost must not be under evaluated, as already discussed in the beginning of this article. In addition, for workload automation open source solutions,

consider the following aspects:

- Technical proficiency in workload automation may be difficult to find and costly to develop
- Health, growth and longevity are key requirements, so look at how active the community is
- Open source solutions are time consuming and timing can be decisive for competitiveness

What's your risk tolerance

Risks arise when the investment cost is underestimated, and the open source is not properly managed. If you don't exploit the potential value of an open source investment, all possible advantages may turn into problems, as you can run into bugs, security risks, and business services outages. For workload automation open source solutions, consider the additional aspects:

- Unlike commercial products that have been working successfully for twenty or thirty years, open source solutions are new and introduce an unknown level of risk
- If not supported by a successful ecosystem, your open source solution can barely keep up with the evolving technology
- Maturity may be a critical requirement for solutions that run your business applications in production and therefore have the highest impact on business value

The bottom lines

The open source option for application workflow orchestration must be evaluated on a case-by-case basis. This article provides a few tips to take in consideration in your selection process.

In this "always-on" era of big data and agility, you should look for application workflow orchestration, rather than workload automation or job scheduling.

If you pursue a commercial solution as a result of your analysis, consider Control-M. Its operational maturity has proven its market share. The three categories outlined above, of job scheduling, workload automation and application workflow orchestration, which are individually populated by some vendors, represent a journey for Control-M. Along its journey, it continues to be a Value Leader of Enterprise Management Associates (EMA) Radar [Report](#) for Workload Automation, with superior grades in most of the evaluation categories. It confidently runs core business processes mainframe to multi-cloud. For example, it's used by [airlines to process critical transactions across the globe and by other industries for](#) its continuous innovation helping customers [achieve agility](#), [orchestrate data pipeline](#), [train machine learning models](#), and apply preventive maintenance.