

Figure Verbat’s Devops Workflow

Verbat’s philosophy has been to deliver products that allowed clients to be intimately involved with the development activity. As early as 2005 Verbat had adopted the Agile development practices into its pipeline for software development. This allowed clients to closely observe the product being developed. It created a tight feedback loop that allowed us, along with our clients to better understand the requirements and build a right sized and proportional product.

With the advent of next generation applications and tools that support a wide range of activities related to development, deployment and integration; the feedback loops became tighter and coupled, while at the same time it allowed us to create applications that were decoupled. Thus giving us the capability to have greater flexibility in development and deployment.

While development broke down the barrier between developers and management, Devops broke the barrier between software developers and operations teams. Our development and operations teams work under a single silo. While our Scrum teams are cross functional, in our quest to adopt better integration with operations, we have realized that Devops is not just a set of tools and processes, but a mindset and culture. We have fostered a culture collaboration and communication. Our scrum teams and Devops teams share a symbiotic relationship and they work united towards a common goal.

As a company, we have adopted the following tenets in our Devops philosophy

**Speed:** i.e. Move at a high velocity so as to innovate faster for customers and adapt to changing markets better, and grow more efficient at driving business results.

**Rapid Delivery:** Increase the frequency and pace of releases so we can innovate and improve products faster and thus respond to customers’ needs and build competitive advantage.

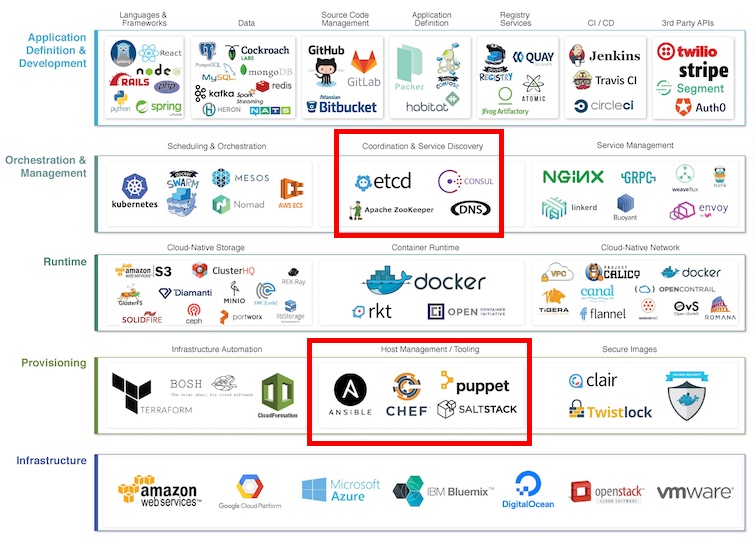
**Reliability:** Ensure the quality of application updates and infrastructure changes so we can reliably deliver at a more rapid pace while maintaining a positive experience for end users.

**Scale:** Operate and manage infrastructure and development processes at scale.

**Improved Collaboration:** Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows.

**Security:** Move quickly while retaining control and preserving compliance. We adopt a DevOps model without sacrificing security by using automated compliance policies, fine-grained controls, and configuration management techniques.

Please see below an illustration of our technology stack. The tools of the trade that allow us to practice an Agile and Devops centric F/W

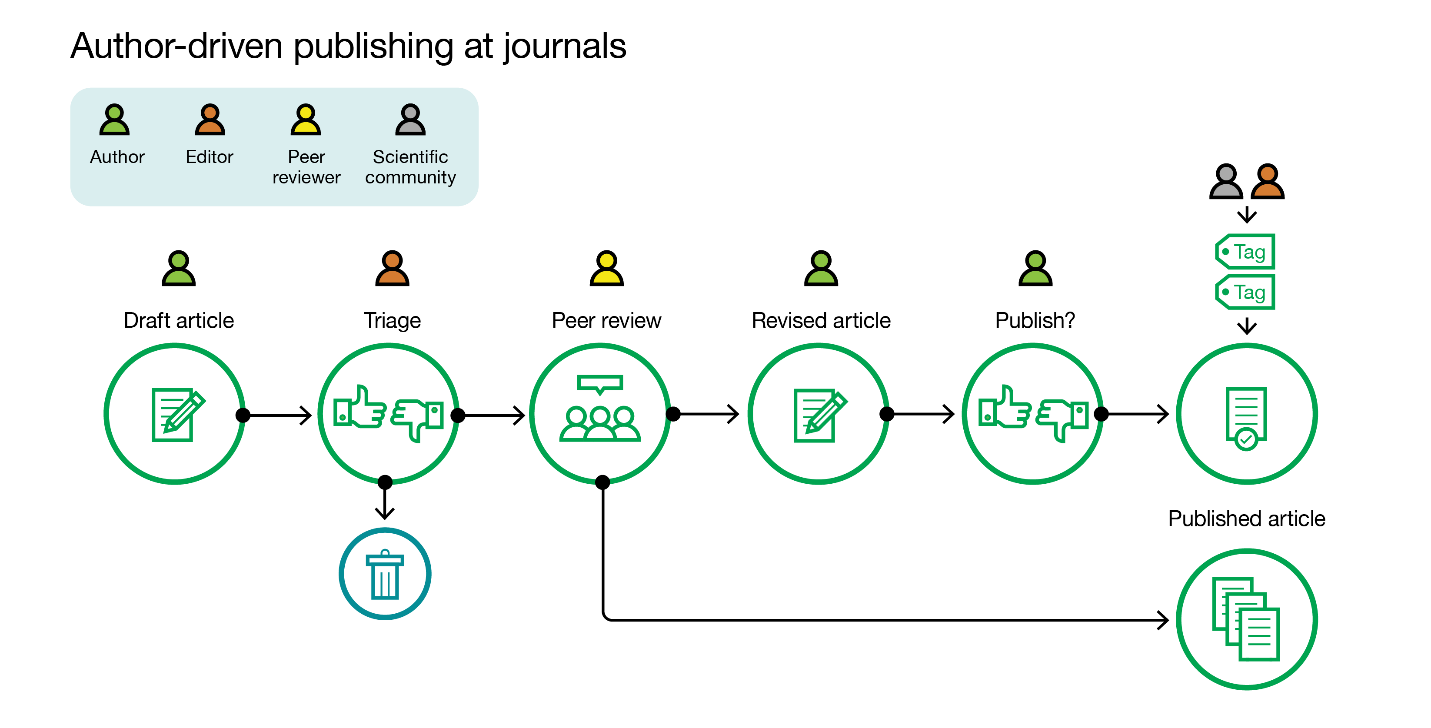


**Course Content Management**

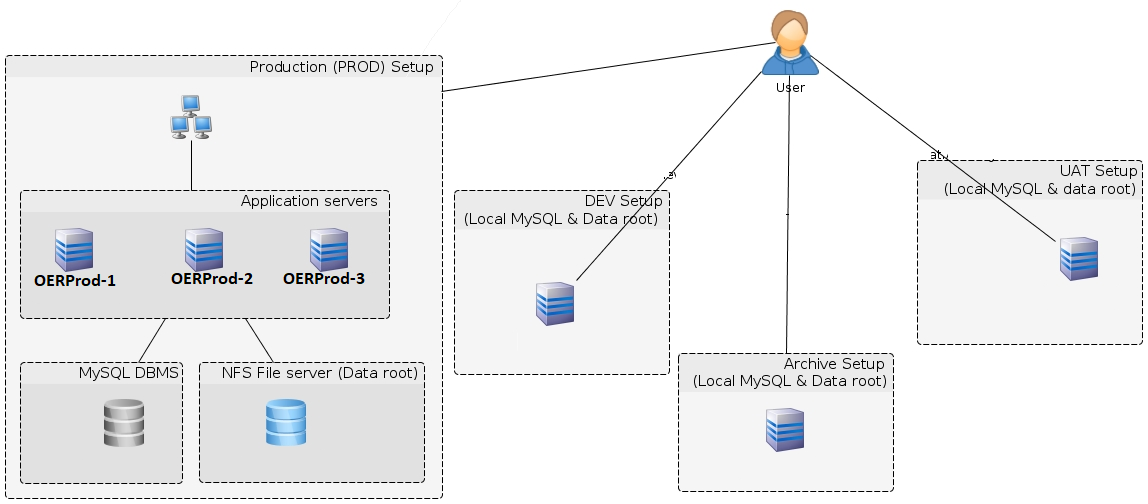
Content creation follows the life cycle of typical online content management systems (CMS). Users can be enrolled as authors, publishers, editors etc. Course/Book creators are responsible for the content and copyright clearances. The system will guide the authors to manage copyright restrictions of the content that they have included along with reminders to include acknowledgments.

Where possible all content is released under a Creative Commons license, e.g. cc BY-NC-SA 4.0 UK (there shall be a license chooser when the course/study is being setup and there shall also be a license chooser when images are uploaded).

The material can be kept in draft so that it can be developed with other collaborators, after which it can be sent to the editors. Once the editors approve the article it shall be send to the publisher who shall publish the book. Content authoring can be accomplished in the app using a rich text pane. The system shall also maintain a workflow that supports content authoring and publishing. Meta tags can be added to describe the content. Content can be categorized as well. Please see below a trivial workflow that explains the process. The system shall authors to create/build lessons and modules.



**Local Server Instances (On Premise)**



**Load Testing**

Load testing is based on the effort expended. Charges vary based on the amount of testing and the type (scenarios used) of testing that is being conducted. Additional charges may be incurred if the tool set has to be procured. Typically load test that use branded tools from IBM etc. tend to be excessive. Additionally these tools come with licenses for the no. of concurrent virtual users that shall be used for testing. Hence a license with a large number of users will cost more than for a few users.

When load test are conducted, Test architects interpret the results to recommend the server architecture that should be used to deploy the product.

To sum it up below are the factors that will contribute to the total cost

1. Test planning and execution effort
2. Not all tools cover all scenarios esp. when it relates to security, hence additional tools such as load runner will need to be used. These tools can be quite expensive
3. Most of the branded tools require additional licenses to generate virtual users. These are also not cheap.

(Note: Information gained from Essementor)