



Let's Solve

# Whitepaper

**DevOps – A cultural change**

**Author: Deepak VS**



A Larsen & Toubro  
Group Company

## Introduction

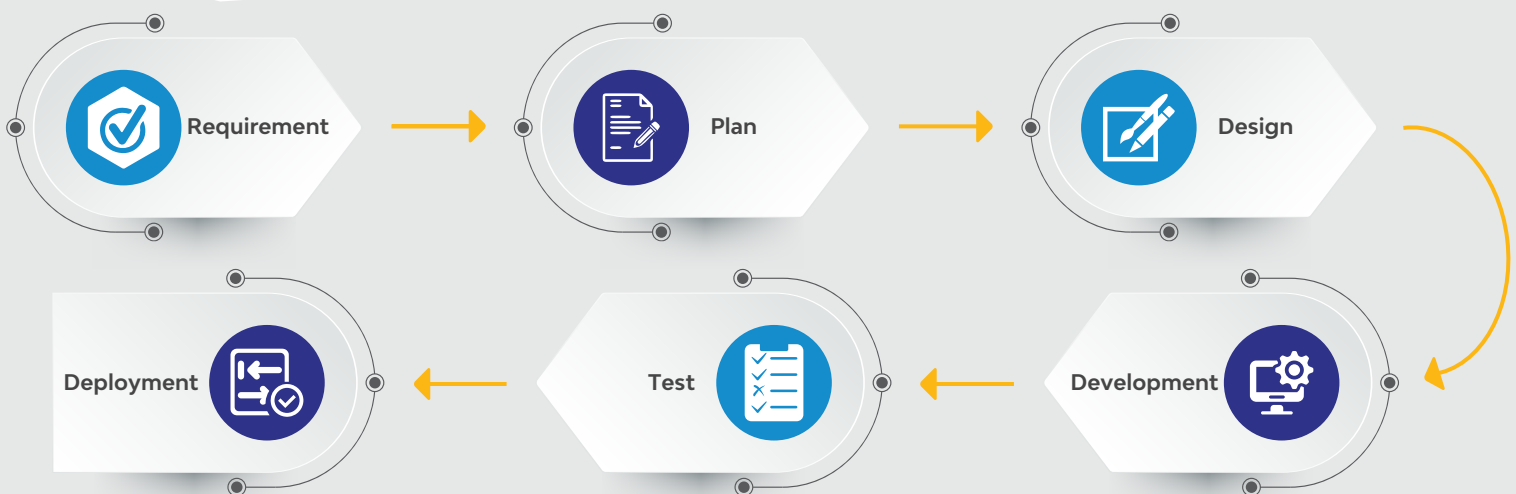
The world is thriving on digitization and innovation, which is causing service providers to expedite their offerings with a more predictable approach. Traditional planning and delivery takes months or years to complete and could result in the company's products and solutions lagging behind in an ever-evolving marketplace. Many organizations run legacy systems and processes which fail to meet current and future demands, thereby creating competitive disadvantage. Against this backdrop, DevOps (Development and Operations) is gaining popularity, and being considered a paradigm change in the IT philosophy. It describes the process of reducing IT operating costs, while enhancing software delivery quality and shortening the turnaround time.

DevOps enables departments responsible for software development and operations to collaborate effectively, in order to produce high-quality deliverables quickly. The underlying lifecycle includes Continuous Development, Integration, Testing (automated and manual), Deployment, Monitoring, and other processes that are typically followed in software development and operations.

## DevOps Journey

### Waterfall

For several years we employed a typical 'Waterfall' development process, which involved a linear sequential approach spanning requirement gathering to production deployment. However, since a 100% sign-off was expected at every stage, such an approach proved to be time consuming, inflexible and monolithic.



## Agile

Later, we transitioned to a new process that leveraged an 'Agile' model. This approach was more effective than the 'Waterfall' one as it facilitated increased collaboration between the IT and Business teams. Here, projects would be broken down into multiple logical iterations, enabling flexible rework, re-engineering and business changes. This helped businesses change product features quickly, fulfill customer requests as per market demand, and helped IT teams adapt to changes in the early stages. Few processes like unit test, code review, integration and regression were introduced in the early stages, resulting in improved efficiencies.

## DevOps

Over the last few years, the 'Agile' process has extended to 'DevOps', which involves collaboration between all key streams including Development, Testing (Automation & Manual), Performance, Security, System and DB Administration, Network teams, Infrastructure and business teams.

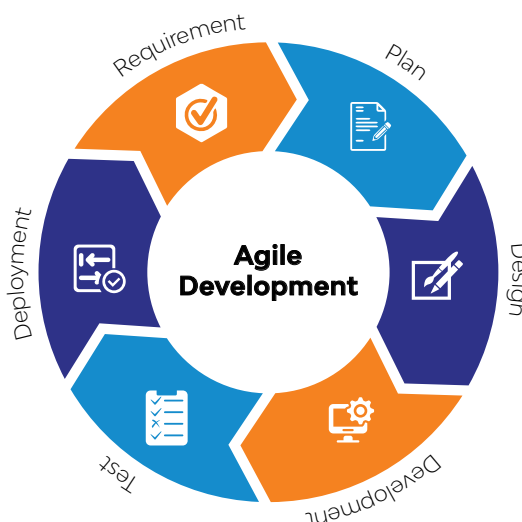
Patrick Debois, the creator of DevOps, a portmanteau of "Development" and "Operations", describes the methodology as a "movement for people who think it's the change in the IT Industry". DevOps can help "stop money wastage, deliver great software, and build systems that scale and last longer," Debois argues.

### Key points about DevOps







- It is a philosophy for better collaboration and relationship across teams
- Creates new, innovative customer experience, keeping pace with customer demand
- Automates most processes and eliminates all redundant and manual activities
- Lean with increased focus on process, purpose and people, and minimized waste

### Core activities in DevOps process

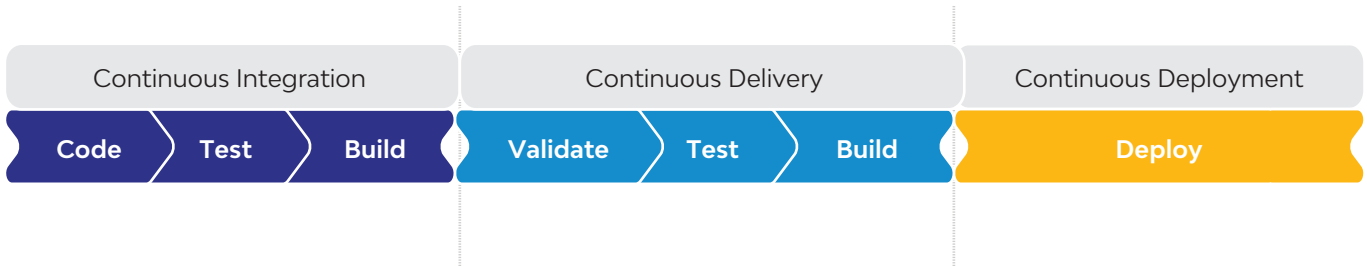
Continuous Integration (CI) - CI is the process of compiling the source code across teams into a single platform, along with running automated test scripts. This is executed continuously through a tool.



### Some of the key activities in DevOps process

|   |   |
|---|---|
|  <p>Automation</p>               | <p>Automation is the backbone of DevOps chain in all streams, which entails multiple tools that minimize manual intervention at each stage.</p>   |
|  <p>Continuous Testing</p>       | <p>A vital component in DevOps, continuous Testing leverages automation to conduct unit test, system test, performance test and business test, and reduce overall effort</p>                            |
|  <p>Continuous Delivery (CD)</p> | <p>CD is the process of making production-ready code by compiling, testing and packaging</p>  |
|  <p>Continuous Deployment</p>    | <p>Execution of continuous delivery and rollback processes enables continuous deployment of applications</p>  |
|  <p>Continuous Monitoring</p>  | <p>Continuous delivery and number of deployments require scrutiny, failures need to be fixed and streamlined as part of the monitoring process to avoid repetitive mistakes and delays in the chain</p> |
|  <p>Delivery Pipeline</p>      | <p>All stages of production deployment for an application, including Build, Test, Validation and Deployment</p>   |

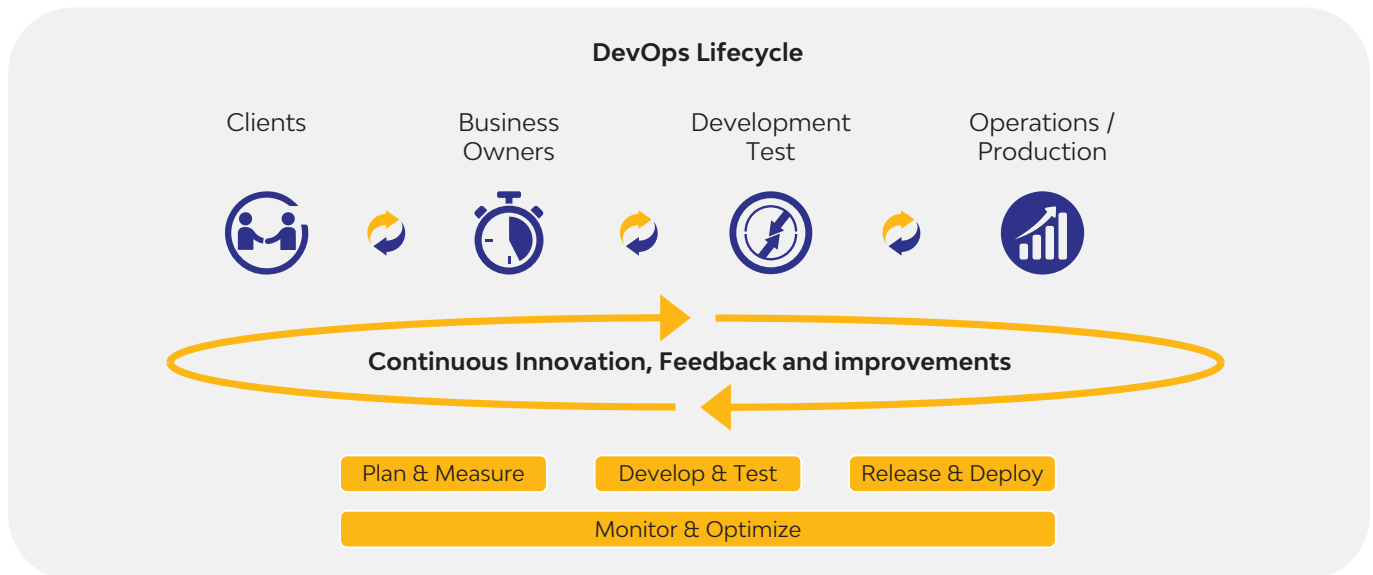
### Typical DevOps chain



### Why DevOps

DevOps facilitates expedited feature creation and outcomes, enhances employee productivity, boosts customer satisfaction, and increases the flexibility for changes. It provides IT agility, which paves the way for reliable, predictable and efficient software development and rollout. The core focus of DevOps is code building, coverage, unit testing, system testing (automation and manual), performance, business validation and deployment.

### One Team One Goal



### DevOps Goals

- |                          |                                |                                       |                              |                                       |
|--------------------------|--------------------------------|---------------------------------------|------------------------------|---------------------------------------|
| Faster innovation cycles | Increased automation           | Standardized process                  | Collaborative teams          | Shorten lead time between deployments |
| IBetter product quality  | Reduced failures and rollbacks | Shorter , automated deployment cycles | Improved customer experience |                                       |

Modifying even one line of code makes pre-production and deployment processes and associated steps very cumbersome. Typically, these activities are implemented manually and shared between multiple teams, and the entire chain could take days or even weeks. Moreover, sometimes numerous small changes will not be deployed to production until major deployment is completed or incorporates all small changes

into one major release.

Continuous Integration (CI) and Continuous Delivery (CD) address these challenges. Implementing automation across the software delivery pipeline including check-in, build, compile, deployment, validation and sign-off operates like an efficient factory model with repeatable concepts.

## Implementing DevOps

### Does DevOps suit every project?

The cost of implementing DevOps is less for new projects, as compared to existing ones.

This is because significant investment is required to increase the automation coverage across streams for existing projects. Cost reduction can

be easily measured by comparing the costs incurred before and after implementation of DevOps. The following four points should be considered necessary for ensuring a successful DevOps implementation:

- Identify & start with small projects
- Identify right individuals with required skills
  - Analyst
  - Engineer
  - Test Engineer
  - Release Manager
- Cultural attributes required for individuals
  - Trust
  - Collaboration
  - Sense of ownership
  - Improvement & Empowerment
  - Continuous learning attitude
  - Responsible
  - Customer-centricity
- Process
  - Project methodologies
  - Quality gates
  - Delivery discipline
  - Compliance
- Tools selection
  - Right tools
  - End-to-end collaboration between tools

Existing projects also can be transformed to DevOps culture, but require significant assessment and change in the mindset across the process.

## Key stages for successful DevOps for existing projects



### Assessment

This stage is to assess the capabilities, processes, technologies and skills required for implementing the project. Study the current challenges and limitations which pose hurdles for DevOps execution.



### Implementation

This step is to identify a module or area for a Proof of concept (PoC) and perform end-to-end checks. Based on the outcome of the PoC, this process is continued for the next module, while making improvements at every stage. At the end of each module, organizations have a comprehensive DevOps scorecard to monitor various activities and evaluate the condition of the project, based on which they can plan for an improvement in the next module.



### Maintenance

This is an ongoing exercise based on the scorecard results captured during the implementation stage.

## Common tools & Technologies

Most popular tools, combined with a stack of open-source tools, are helping organizations move toward DevOps models. Prominent examples include:

| Main Stages        | Sub Stage                        | Tools                            |
|--------------------|----------------------------------|----------------------------------|
| Planning           | Stories, Issue                   | Jira, Cucumber, Jasmine, Redmine |
| Development        | Unit testing                     | Junit                            |
|                    | Control management               | Git, SVN                         |
|                    | Code Coverage                    | SonarQube, JaCoCo, Codeclimate   |
|                    | Continuous integration           | Jenkins, Hudson                  |
|                    | Performance                      | JMeter,                          |
| Testing            | Automation                       | Selenium, Protractor             |
|                    | Service testing                  | SoapUI                           |
|                    | Cross browser                    | CBT, SauceLabs                   |
|                    | Defect management                | Jira, HP ALM,                    |
| Build & Deployment | Build & Release management Maven | Maven, ANT                       |
|                    | Deployment (CI/CD)               | Jenkins                          |
|                    | Analysis                         | Splunk                           |
|                    | Configuration management         | Chef, Puppet                     |

## Conclusion

Overall, DevOps has the potential to add significant value to business stakeholders due to its agility, flexibility, high scalability and increased quality of outcome. Moreover, it can help minimize feedback time, risks and costs. It is, therefore, evident that DevOps is not just another process or set of tools but represents a major cultural change for the organisation.



## About the Author



**Deepak VS**

Deliver Manager – Assurance Practice – LTI

Deepak VS has over 18 years of industry experience. His areas of experience include automation, performance and end-to-end test management. He has successfully led several large-scale enterprise engagements for Fortune 500 clients.

LTI (NSE: LTI, BSE: 540005) is a global technology consulting and digital solutions Company helping more than 300 clients succeed in a converging world. With operations in 30 countries, we go the extra mile for our clients and accelerate their digital transformation with LTI's Mosaic platform enabling their mobile, social, analytics, IoT and cloud journeys. Founded in 1997 as a subsidiary of Larsen & Toubro Limited, our unique heritage gives us unrivaled real-world expertise to solve the most complex challenges of enterprises across all industries. Each day, our team of more than 25,000 LTItes enable our clients to improve the effectiveness of their business and technology operations, and deliver value to their customers, employees and shareholders. Find more at [www.Lntinfotech.com](http://www.Lntinfotech.com) or follow us at @LTI\_Global