**IITB Standard definition of a DevOps team**

**DRAFT**

# Document Purpose

To have a common understanding and expectations across IITB as to the description, roles, and minimum responsibilities for a team labelled as “DevOps”.

# Definition

## Description

A [**DevOps team**](http://everything.explained.today/DevOps/) is a cross functional, multidisciplinary team that emphasize the collaboration and communication of both software developers and information technology (IT) professionals while automating the process of software delivery and infrastructure changes.

The team comprises various sets of skills amongst its members necessary for software development, delivery, reliability, security, operation, and maintenance.

## Roles

A DevOps team comprises of, at minimum, the following roles:

1. Product Manager
2. Product Owner
3. Scrum Master
4. Software Development
5. Operation Expert
6. Security Champion
7. User Experience (UX) Champion

## Responsibilities

DevOps team members perform the following specific tasks according to their role(s):

1. Coding software
2. Configuring software
3. Testing software
4. Securing software
5. Architecting software
6. Deploying software in production
7. Operating software in production
8. Maintaining software
9. Managing backlog of features for a software
10. Prioritizing work of team members
11. Documenting software requirements and specifications
12. Interacting with end-users
13. Managing the rollout of software features and its coordination with end-users
14. Managing software change requests
15. Monitoring [software performance metrics](http://everything.explained.today/Application_performance_management/)
16. Engaging with other IT professionals, outside of the DevOps team, to fulfill software requirements (e.g. IT Continuity team)

# Checklist

The following checklist can be used to evaluate whether an IT team is recognized by IITB as a “DevOps team”:

1. It has the roles listed in section 2.2
2. It performs the tasks listed in section 2.3
3. It uses [build automation](http://everything.explained.today/Build_automation/)
4. It uses, at minimum, API-driven [testing automation](http://everything.explained.today/Test_automation/)
5. It uses a [decentralized version control system (DVCS)](https://www.atlassian.com/blog/software-teams/version-control-centralized-dvcs)
6. It uses [Application Release Automation](https://www.techopedia.com/definition/31061/application-release-automation-ara) across various environments
7. It is able to receive direct feedback from end-users
8. Most of its infrastructure is defined as code
9. Most of its [software performance monitoring](http://everything.explained.today/Application_performance_management/) practice uses automation