

Performance Measure of Project Management Automation Tool based on DevOps Selection Criteria for a General Purpose Software System

Poonam, Pooja Mittal



Abstract: DevOps, a cluster of tools to automate the tasks of building, testing and releasing the software, intends to optimize the end to end delivery lifecycle of software by eliminating the bottlenecks of each and every stage of development. This research paper presents the performance measure of different tools in the project planning and management phase of software development. Diverse tools exist for each and every phase of the development and depending on the outcome of the evaluation or selection criteria, tool with highest score is being chosen. This paper measures the performance of alternative project management tools viz. Jira, Trello, Leankit, AgileCraft based on these evaluation or selection criteria. To execute the underlying research work, a six-point scaling table is defined. The table measures the performance of project management automation tools and realizing their competence level score. This performance measure or the outcome as competence level score will be useful for the researchers or software developers to contemplate the best project management automation tool typically before making a decision about tool selection.

Keywords : Automation, Automation Tools, DevOps, Project Management, Project Planning, Software development

I. INTRODUCTION

DevOps is the effective and efficient communication between Software Development teams and the IT Operations teams for reliable and the faster delivery of software products. DevOps employs various automation tools to bridge the gap of communication between Dev and Ops teams hereby also reducing the software delivery time. These tools include Jira, Bamboo, JMeter, Selenium, Confluence, Bitbucket, Crucible, Splunk etc. DevOps tools come up with automation from the first stage of DevOps development lifecycle. Management of software development and delivery at all the stages is very critical task as lots of alternative automation tools exists to adopt at each and every stage of DevOps. To carefully choose as being the best or most suitable tool, the management include various accustomed

tasks or activities as the selection criteria in routine viz., Continuous Prioritization, Risk Tracking, Workload Management etc. Work being reported herein involves the performance quantification of these DevOps Tools based on evaluation criteria. These criteria also termed as activities when done in collaboration helps to choose the best automation tool from the set of alternative tools and hence accelerating the software development and delivery process considerably. These criteria are also adaptable to any kind of changes occurred during the development of a general-purpose software system.

II. LITERATURE REVIEW

The practices or criteria used by DevOps for tool selection are reviewed separately from different existing research papers. The State of DevOps report (2017) notes that “the ability to develop and deliver software efficiently and accurately is a key differentiator and value driver for all organizations [...] independent of industry or sector.” The authors of the report believe that DevOps is the best way to deliver software as a strategic asset. The report found that high-performing DevOps teams were more than twice as likely to achieve or exceed the following objectives:

- Quantity of products or services
- Operating efficiency
- Customer satisfaction
- Quality of products or services provided [12]

One of the papers applies the well-known Systematic Literature Review (SLR) to compile and analyze DevOps Tools. This review is intended to provide a knowledge base to support the informed, effective and less risky adoption of DevOps for information management systems. [1] Another paper verifies DevOps (a portmanteau of 'development' and 'operations') a software development method that extends the agile philosophy to rapidly produce software products and services and to improve operations performance and quality assurance. This paper explores the usefulness of DevOps concepts to improve the development of software that supports scientific projects. They also establish the similarities and differences between scientific projects and web applications development and discuss where the related methodologies need to be extended.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Pooja Mittal*, Assistant Professor, Department of Computer Science and Applications, MDU, Rohtak, Haryana, India. Email: mpoojamdu@gmail.com

Poonam, Research Scholar, Department of Computer Science and Applications, MDU, Rohtak, Haryana, India. Email: poonam.mehta20@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

[2] In one survey paper on DevOps , the answers to four research questions were found that were - which and how automated procedures support DevOps practices, which automation tools are being used to support DevOps teams, which are the most important service characteristics that affect automatic service synthesis, what is the level of services synthesis automation? This series of research questions clarify many things regarding what DevOps actually is, how it can be automated and, finally, how it can be combined with one of the latest software development approaches, that is, software service composition. [3]

III. ALTERNATIVE AUTOMATION TOOLS

Different sorts of tools exist to render automation at every stage or division of DevOps development lifecycle. Few of these include

Development Phase	Alternative Tools
Requirement Management:	Confluence, Slite, Notion, MediaWiki, BookStack
Project Planning and Tracking:	JIRA, Trello, Leankit, Agilecraft
Code repository:	Bitbucket, GitLab, GitHub, Launchpad,
Build:	Bamboo, Jenkins, Teamcity
Testing :	Selenium, ZAP, TestComplete, UFT, JMeter

DevOps Management elects preminent tool from the set of these alternatives. For this election, management includes various criteria as day to day activities also termed as tasks.

IV. SELECTION CRITERIA FOR OPTIMAL AUTOMATION TOOL ACQUISITION

The selection criteria for acquisition of alternative tools are employed at the very first stage of software development i.e. Project Planning or overall Project Management. These criteria include – *Continuous Prioritization, Dependencies Management, Risk Tracking, E2E traceability, Reporting and Progress Tracking, Resource Workload Management, Ticket Workflow customization* etc.

The rest of this research paper talks briefly of these criteria or activities and find competencies of alternative tools for project management phase of software development.

A. Continuous Prioritization / Project Planning

In the era of Digital Evolution, pace of customers demand for more features is faster than the team velocity in most of the organizations. This gap becomes multifold with the increase in customer base who are using or demanding the product. But with the limited hands to deliver the stuff, it becomes utmost critical to prioritize the requirements and to pick the most valued requirement from the list irrespective of the model you are following either Kanban or Scrum.

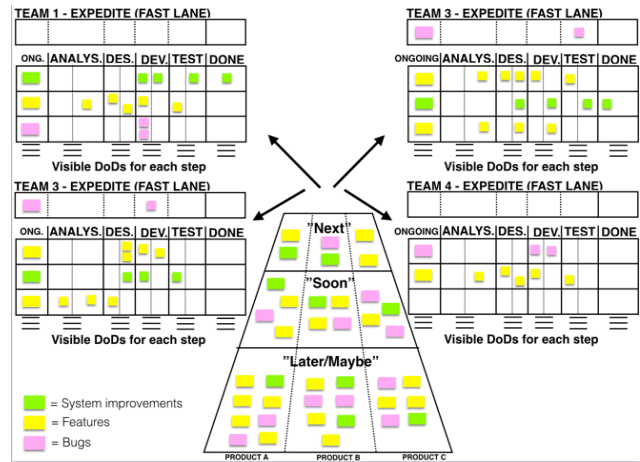


Fig. 1.The Volcano-Prioritize work for multiple teams and products under project planning [4]

For prioritizing the requirements, all the stakeholders should have access to the information like backlog items (i.e. the requirements in pipeline that are not prioritized yet), items in delivery scope, Progress etc.

B. Dependency Management

In elementary scenario where application is not much big in size and there is only one small team handling the delivery then this aspect is not as that valuable. But in realistic world where companies are moving away from monolithic design to micro services world, where many teams are parallelly working on delivery, it is highly possible that task of one member is dependent on others.

Dependency management

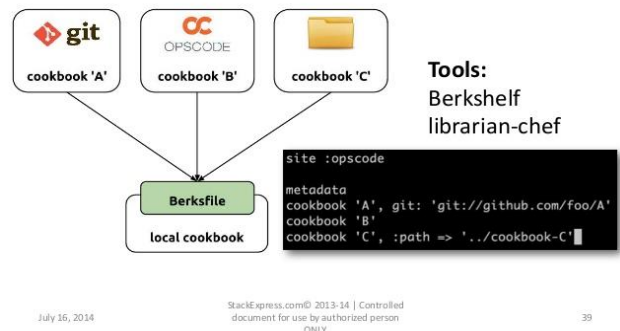


Fig. 2.DevOps and Configuration Management Tools for Dependency Management [5]

One member's code cannot go to production until other member's functionality is ready and depending on the number of teams, interaction with other application. This dependency graph can also be visualized as many parents and many children.

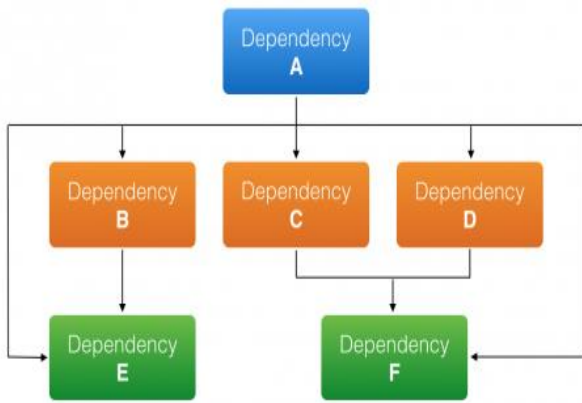


Fig. 3. Dependency Visualization Graph showing Dependencies between Different Applications [6]

This dependency aspect is very critical while making a go-no-go production decision and also for planning a release.

C. Risk Tracking

In organizations, where race is to deliver software faster every time, there are situations when managers take a conscious call (after considering all aspects of course) to release the code even if it contains some defect and track those as Risk that may be a business risk, tech risk, data risk etc.

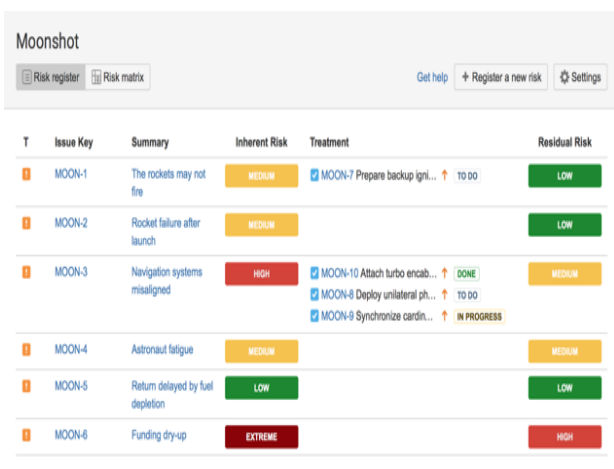


Fig. 4. Risk Tracking during Project Planning and Management [7]

For example, Oracle has stopped support for 11.2.0.4 on 31 Dec 2018, which means Oracle won't provide any support if customer report any issue on this version [8]. Due to delivery commitment if any organization don't upgrade the oracle by EoL date, hence carrying a Tech Risk on their application. Hence it is very important to call out all the risks and track them till closure and provide regular updates about those to all the relevant stakeholders.

D. E2E Traceability/Integration with other DevOps Tools Tables

In larger teams where many developers working together to deliver a great deal of software features, it is utmost important to track the progress specifically when a profusion of tools are being used in software delivery life cycle. For

example, Code has been checked against which requirement, what all requirements has been covered in any artifacts (jar, war, dll), which code snippet has been reviewed etc.



Fig. 5. E2E Traceability with other DevOps Automation Tools [9]

Not tracing under this activity makes maintenance and support difficult for applications having longer lifetime. Specifically fixing an incident/production defect requires prior understanding of why it has been designed/code in such a way, what would be regression impact if developer fix that etc.

E. Reporting and Progress Tracking

Application development is done by a team which not only contains developer or tester but involves a lot of stakeholders like Product owner, BA, Project Manager/Scrum Master, Architect, QA Manager, Tech Director, Client etc. Being a stakeholder in the application delivery everyone wants to track the progress depending on the person.



Fig. 6. Progress Tracking and Reporting for Application Development [10]

For example, Product owner would like to see how requirements/ stories are progressing, how many are In-Progress, Done or yet to be picked for development. Similarly, as a QA manager one is interested in how many stories are ready for testing, how many defects raised, how many fixed etc. Preparing manual report for each person would not only be a nightmare but also would not provide current state.

Rather it would present a state when data for that report is pulled, and it is quite possible that between report creation and consumption data has modified or updated. Hence real time report avoids confusion and communication mishaps.

F. Resource Workload Management

One of the important aspects of effective management is to deploy each member bandwidth wisely and to make a balanced team from workload point of view. If each resource fill time spend in each task assigned to him then it would not only help to track the planned vs actual effort required to complete the task but also to track all the members are completely engaged, no overloaded or not underutilized.

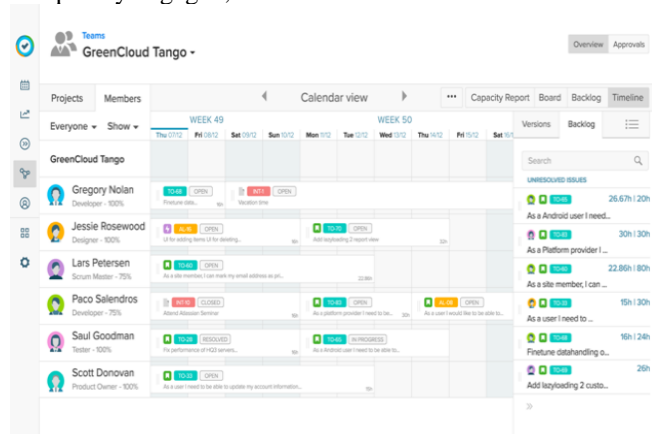


Fig. 7. Workload Management System for different Project Resources [11]

It also helps to track the team velocity in each sprint or in Kanban velocity at some specific intervals like weekly, fortnightly, monthly as per requirement.

But Enterprise Level adoption of any tool requires to access tool from other critical aspects. For example, Ticket Workflow Customization, Field Customization, Ticket/Issue type customization, Interface (Web/APIs/Mobile App), deployment (In-Premise, Public Cloud, Vendor Cloud/SaaS), Email Notification, Single Sign-On, Monitoring, Support, Data Import/Export, Active Directory Integration, Custom Development, Third Party Integrations/Plug-ins, Customer base/Market share etc.

Let’s discuss why we include these parameters in evaluation criteria.

G. Ticket Workflow Customization

There is no any single defined thumb rule to follow or fixed definition of how a ticket/work item should flow from inception to closure. Various teams in organization practices agile in different ways so does life cycle of work item. There is also a possibility of having different workflow for each ticket type rather than single workflow adopted by all ticket type. Story workflow would be different from Defect workflow. Story lifecycle could be: In Backlog > Prioritized> In Progress > Dev Complete>In Review>Done. Defect lifecycle could be: Open>In Progress >Fixed>In Testing > Closed. Similarly, teams may include various other status On-Hold, In Analysis, Awaiting production etc. as per need. It helps them to track the exact progress on the work item.

H. Customizable Fields

Each deliverable entity/ticket/work item must also contain some attributes which defines what exactly is required, which goal that item is satisfying, if multiple team working in parallel (scrum/feature teams) then item belongs to which team, what is the impact, deadline, benefits, Story Points and many more. Teams should be able to define how many free

text fields, select list, checkbox, parent-Child field they would like to have in their set up.

I. Customizable Ticket Title / Labels

Being no standard definition helps team to use ticket label same as teams uses in day to day conversation about delivery. One team can call a ticket as Bug while other team can refer the same ticket as Defect. It’s all depends on terminology used in their definition of agile. So, at enterprise level it is always good to have flexibility that allows implementation/labels of tickets same as team terminology.

J. Customizable Look / Screens

While having custom tickets/fields/workflow is important, equally important need to have flexibility to design the screen that should be visible to team, better to have flexibility of having project/team level customization. What is the order of ticket type, in which order fields should be displayed.

These selection criteria acts as the bases for analyzing different existing tools for project management. For this purpose, Six-point scaling table is also defined that indicates the competence level of project automation tools. Following which another table measures the performance of different automation tools viz. Jira, Trello, Leankit, Agilecraft based on this scale.

Table- I: Six- point Scaling Table for measuring Competence Level of different Project Management Automation Tools

Scale	Indication
5	High Level of Satisfaction
4	Average
3	Below Average
2	Low Level of Satisfaction
1	Weak or Slight Support
0	No Support

Below table assign figures to the tools as per the scaling table.

Table- II: Competence Level Score of different project management automation tools on a Six-Point Scale [13]-[20]

Criteria	Project Management Automation Tools			
	Jira	Trello	Leankit	Agilecraft
Interface				
Web	5	5	5	5
Mobile App	5	5	5	5
Rest API	5	5	5	1
Deployment				
In-Premise	5	5	0	0
Public Cloud	5	5	0	0
Vendor Cloud (SaaS)	5	5	5	5
Support				
Community	5	5	4	4
Vendor	5	5	4	5



Criteria	Project Management Automation Tools			
	Jira	Trello	Lean kit	AgileCrack
Knowledgebase	5	5	4	5
Generic Factors				
Customizable Workflow	5	5	5	5
Customizable Fields	5	5	5	5
Customizable Tickets/Issues	5	5	4	5
Screens	5	5	5	5
Deadline Tracking	4	5	4	5
Notification (E-mail)	5	5	4	5
Integration with Other DevOps Tools	5	5	4	4
Project Planning	4	5	5	5
Reporting	4	5	5	5
Resource/ Team Management	4	5	5	5
Single Sign On	4	5	3	3
Monitoring	5	5	5	5
Data Import/Export	5	5	4	5
Custom Development	5	4	3	3
Third Party Integrations /Plug-ins	5	3	3	3

V. CONCLUSION AND FUTURE WORK

This paper discussed many tasks or activities as the selection criteria that are efficacious for persona of a Delivery Manager. Opting the best of these selection criteria, a right set of toolchains from alternative automation tools can also be proposed that escalate the development process of a general-purpose software system.

This performance measure reveals JIRA as the best project management tool depending upon the priority of the evaluation criteria. A performance evaluation or quantification of the impact of using other tools at next phases of development lifecycle may also be carried out as a part of further research.

This research work will also be useful for researchers/students to understand the modus operandi of DevOps along with its different tool selection criteria for alternative tools accessible for automation.

REFERENCES

1. Asif Qumer Gill, Abhishek Loumish, Isha Riyat, Sungyoun Han "DevOps for Information Management Systems," in *Vine Journal of Information and Knowledge Management Systems*. ISSN: 2059-5891, February 2018.
2. Maximilien de Bayser, Leonardo G. Azevedo, Renato Cerqueira "ResearchOps: The Case for DevOps in Scientific Applications," in *IBM Research*.
3. Andreas Andreou, Andreas Christoforou, Luciano Baresi, Mike Papazoglou, Damian Tamburri, " *DOSSIER-Cloud DEVOPS-BASED*

SOFTWARE ENGINEERING FOR THE CLOUD", Deliverable D4.5 Survey Paper / Technical Report on DevOps Automation and Software Service Composition. February 2019.

4. Tomas Rybing (2016, May 21). The Volcano – Prioritize Work for Multiple Teams & Products. [Online]. Available: <https://www.infoq.com/articles/volcano-prioritize-work/>
5. Sonu Meena (2014, July 16). DevOps & Configuration Management Tools (Slide-39/45). [Online]. Available: <https://www.slideshare.net/sahilsk/configuration-management-stackexchange-20140610>
6. [Online]. Available: <https://salesforce.stackexchange.com/questions/184346/dependency-visualisation-javascript-library-compatible-with-locker-service>
7. Risk Management for an Agile World [Online]. Available: <https://marketplace.atlassian.com/apps/1213146/riskregister?hosting=cloud&tab=overview>
8. Oracle FAQ's. Oracle 11gR2 . [Online]. Available: http://www.orafaq.com/wiki/Oracle_11gR2
9. Tibor Kiss (2019, Mar 08). E2E Python Solutions in DevOps – Part 4. IT Blog for Sharing. [Online]. Available: <https://the1bit.com/e2e-python-solution-in-devops-part-4-create-azure-devops-pipeline-staging/>
10. Rich Filters Add-ons for Jira. [online]. Available: <https://marketplace.atlassian.com/apps/1214789/rich-filters-for-jira-da-shboards?hosting=server&tab=overview>
11. Transparent Resource Planner Management Software for Jira. [online]. Available: <https://www.tempo.io/tempo-planner-jira-resource-planning-and-management?hsCtaTracking=288155ca-e621-45dc-b9a0-81a6f421e6ee%7Cc9373262-4023-4896-bbbd-62daaf8bf3eb>
12. The States of DevOps Report(2017). Benefits of the Right DevOps Tool. [Online]. Available: <https://www.smartsheet.com/devops-tools>
13. Melissa Jackson, "New Look Same Leankit", February 2018. [Online]. Available: <https://leankit.com/blog/2018/02/new-look-same-leankit/>
14. View Leankit's Full Feature List. [Online]. Available: <https://leankit.com/compare-editions/>
15. Trello. [Online]. Available: <https://trello.com/tour>
16. Trello_Blog. [Online]. Available: <https://blog.trello.com>
17. Trello_Blog Productivity. [Online]. Available: <https://blog.trello.com/topic/productivity>
18. Atlassian Jira Software. [Online]. Available: <https://www.atlassian.com/software/jira/guides/getting-started/overview>
19. Atlassian Jira Software Features. Scrum Boards. [Online]. Available: <https://www.atlassian.com/software/jira/features>
20. Atlassian Jira Software. Product Guide. [Online]. Available: <https://www.atlassian.com/software/jira/guides>

AUTHORS PROFILE



Poonam, Research Scholar, pursuing PhD from the Department of Computer Science and Applications, Maharishi Dayanand University, Rohtak, Haryana under the supervision of Respected Dr. Pooja Mittal (Research Guide and Second Author). Author's Qualification is M.Phil.(CS), MCA. She had attended National Conferences and also published research papers.



Dr. Pooja Mittal obtained her Ph.D. degree from Maharshi Dayanand University. Her area of research and specialization include Data Mining, Data Warehousing, and Computer Science. She had published more than 50 research papers in renowned International and National Journals and attended more than 30 Conferences. Currently she is working as Assistant Professor in the Department of Computer Science & Applications, Maharishi Dayanand University, Rohtak, Haryana.

