International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Optimizing Software Quality through Internal Developer Portals

Srividhya Chandrasekaran

Senior Product Manager Bedford, MA, United States
Email: *srividhya.chandrasekar[at]gmail.com*

Abstract: ***The focus of this paper is exploring methodologies for enhancing software quality through the utilization of Internal Developer Portals (IDPs). In today's dynamic software development landscape, the significance of IDPs in streamlining workflows, fostering collaboration, and optimizing software quality is paramount. This article explores the multifaceted role of IDPs in elevating software quality by enabling efficient resource utilization, facilitating standardized processes, and promoting enhanced communication among development teams. Analyzing the impact of IDPs on development cycles, quality assurance measures, and overall software performance, this review underscores the pivotal role of these platforms in driving continuous improvement and delivering superior software products.***

Keywords: Software Quality, Platform engineering, Test Automation tools, Internal Developer Platform, Internal Developer Portal

1. Introduction

Quality engineering is a broader field that encompasses software complete software development life cycle (SDLC). It strives to prevent problems proactively and enhance the overall quality of a software product by embedding quality assurance practices testing and centers on the at every stage of development, starting from design and coding through deployment and maintenance.

2. Literature Review

By the IEEE Standard Appendix of Software Engineering terminology quality is evaluated as, “The degree to which a system, component, or process meets specified requirements”, IEEE further explained quality as “The degree to which a system, component, or process meets customer or user needs or expectations”. It is urged by" a product's quality is a function of how much it changes the world for the better". Quality is conformance to predefined specifications that meet the customers’ needs.it is urged by quality refers to the extent or degree to which a customer’s requirement is met.

In a study “Developer Productivity for Humans, Part 7: Software Quality” conducted by the Engineering Research Team at Google, the authors classified Software quality into four components. They classified Software Quality into 4 different component types namely Process Quality, code Quality, System Quality and Product Quality. They concluded that the four types of quality are dependent on each other. While System Quality and Product Quality are focused upon by Engineering leaders and Product managers, the individual developer focuses on Process and Code Quality. [1]

1) Evolutionary Insights into Advancements in Software Quality

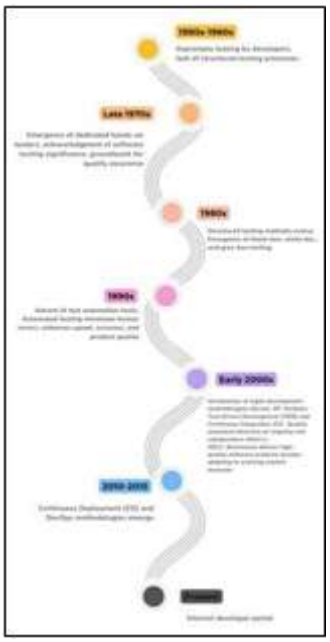

Figure 1: A timeline and insights from software quality evolution over decades

The way organizations approach Software Quality has evolved over the years. It began with manual testing to improve product quality and today it has evolved into using Developer portals and plugins to automate and enhance an individual developer experience throughout their journey thereby leading to building a high-quality software product. This is built on the premise that Happy Developers build better Code [2].

2) The Rise of Platform Engineering

Firstly, what is Platform Engineering? As modern software architectures increased in complexity, forward-thinking companies began to support the application teams by building and maintaining core systems and workflows for delivering infrastructure. A Platform engineering team is composed of Data scientists, Engineers, Analysts along with the platform organization’s leaders.

An overview of how platforms look is referred to in Fig 2.

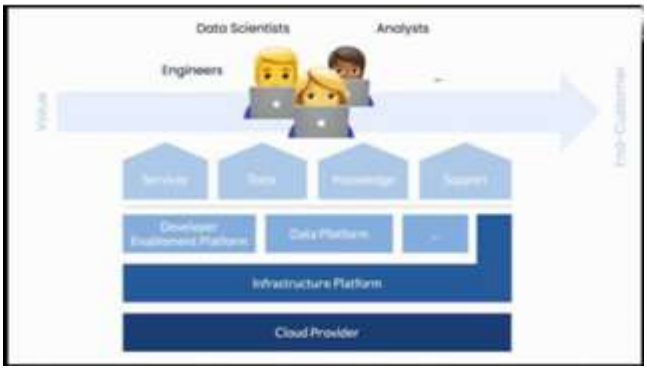

Figure 2: How Platforms look

By 2026, Gartner predicts [3] that 80% of large software engineering organizations will establish platform engineering teams as internal providers of reusable services, components, and tools for application delivery. Platform engineering will ultimately solve the central problem of cooperation between software developers and operators.

3) How can Platform Engineering tools influence Software Quality?

Platform engineers play a crucial role in establishing standardized tools, frameworks, and development practices, ensuring consistency and minimizing errors in software development while enhancing overall quality. By integrating automated testing suites and CI/CD pipelines, they rigorously test code changes, reducing the likelihood of bugs in production environments.

These engineers implement robust security protocols, mitigating vulnerabilities and enforcing adherence to security standards. Additionally, well-designed platforms offer scalability and performance optimization, contributing to better user experiences and overall software quality. Streamlined processes and developer-friendly tools foster increased productivity, allowing developers to allocate more time to ensuring code quality and implementing best practices. Ultimately, through robust foundations, best practices, and necessary resources, platform engineering significantly influences and improves software quality across the entire development lifecycle.

A recent advancement within platform engineering involves the emergence of Internal Developer Portals (IDPs). An Internal Developer Portal is built by a platform team to build golden paths and enable developer self-service. An IDP consists of many different techs and tools, glued together in a way that lowers cognitive load on developers without abstracting away context and underlying technologies.

IDPs automate repetitive tasks, such as building, testing, and deploying code, thereby reducing manual effort and enabling developers to focus on writing high-quality code. This efficiency leads to increased productivity and faster development cycles.

Some examples of Internal Developer Portals include Backstage by Spotify [4], Cortex by Atlassian [5], Port [6] and Opslevel [7].

4) Quick Exploration of the different Internal Developer Portal available

a) Backstage by Spotify

Backstage is an open-source framework for building developer portals, created at Spotify, donated to the CNCF, and adopted by thousands of companies. For a developer, it can be the fastest way to create a new software component. For platform teams, it can be the easiest way to encourage best practices. For your entire engineering org, it can be the central nervous system for all your software, tooling, and documentation. Since it can do so many things, Backstage can be almost anything.

Soundcheck [8] is a commercial backstage plugin within backstage that ensures quality, reliability, and alignment of software development with codified checks and guidance. Soundcheck uses automation to verify that all of your software is built to the engineering standards that you expect at your organization while making it easy for all of your engineers to know about and follow those standards. At Spotify, Soundcheck started as an effort to draw a straight line between an individual developer's decision-making to our topline goals for engineering excellence and quality.

Soundcheck's impact on the tech health of software components has been highlighted in multiple blog posts on the backstage website [9].

b) Cortex by Atlassian

Cortex is an internal developer portal by Atlassian that cuts noise for developers with paved paths to production. Catalog, score, and drive action to improve software.
Scorecards within Cortex can be used to define requirements for any standard like production or migration readiness, development quality, security compliance, and more.

c) Port

Port creates compelling developer experiences and delivers them in a portal, using the building blocks you need, according to your stack, developer personas, and culture. Scorecards within port help Achieve security and quality at scale, by defining custom standards for quality, maturity, and readiness.

d) OpsLevel

The OpsLevel internal developer portal helps your team ship fast without risking your software standards.

5) Ways to use Internal Developer Portals [10] (IDP)]to improve software quality

Dashboards built within these plugins can be used by individual developers to assess the quality of your software components against standards defined by the organization. When development teams see a dashboard of passing and failing checks related to security, reliability, and Code quality, it helps them understand the tech health of their components and fix those failures.

Viewing a dashboard that illustrates their team's software quality fosters a positive competitive spirit among team,

driving them to attain these objectives through the gamification process.

Technical documentation within an IDP provides a document-like code solution enabling engineers to write their documentation along with code and to find all related documentation within one place without needing to go searching for it. Kubernetes plugins within IDPs Easily oversee all your service's deployments with a single glance. Software catalog helps oversee all your services and software components effectively.

Though not exhaustive, this overview encompasses the essential opportunities available within internal developer portals. It addresses the crucial options necessary for platform engineering teams to enhance developer productivity and elevate the quality of their software components.

6) Backstage's API Spectral Linter Plugin

One of the Open-source backstage plugins, the API Spectral Linter is a quality assurance tool that checks the compliance of API's specifications Spectral rule sets. Developers want consistency while designing API's. API linting is the process of making sure that APIs are not just technically correct, but that they also comply with a set of additional constraints that often are documented in the form of API guidelines. The utilization of a style guide facilitates decreased decision-making requirements and fosters enhanced consistency across all team members. The API Spectral Linter plugin allows you to validate API's with rule sets.

7. Unlocking the Benefits of API Spectral Linter Plugin for Software Developers within Backstage

With the rise of OpenAPI [12]as a standard API specification, developers using OpenAPI must ensure their API definitions follow the specification. However, OpenAPI and similar API description formats can often lack specificity, causing confusion for novice API designers and occasional oversights for experienced ones. This leads to the creation of valid but inadequately informative documents that may appear dysfunctional in documentation tools or lack essential validation details for contract testing.

Using an API Linter assures consistent formatting and adherence to best practices, enhancing API comprehensibility for developers while minimizing errors. Additionally, it identifies maintainability concerns, such as deprecated features, ensuring future ease of maintenance. Ensuring optimal API performance, it addresses reliability by pinpointing and rectifying performance issues. Lastly, by detecting security flaws like authentication lapses, linting safeguards the API and its data from potential risks.

3. Conclusion

In the present software engineering world, developers face significant cognitive load and context switching in day-to-day work. Applying platform engineering practices to software development helps elevate software quality.

Organizations should assess their need for a platform engineering team to support their mainstream applications and increase efficiency and productivity across development teams by providing reusable components, automating repetitive tasks, and optimizing workflows. Similarly, there exists a need for organizations to consider investing in Internal Developer Portals to Improve onboarding, training, and promote consistency across development processes by providing frameworks, standards and guidelines that these portals provide.

Making use of the easily accessible collection of plugins and tools within Internal Developer Portals diminishes barriers, fosters innovation and teamwork, enhances the potential for reuse, and decreases the time taken to resolve incidents thereby elevating software quality for Software organizations

References

[1] Green, C. Jaspan, M. Hodges and J. Lin, "Developer Productivity for Humans, Part 7: Software Quality," in IEEE Software, vol. 41, no. 1, pp. 25-30, Jan.-Feb. 2024, doi: 10.1109/MS.2023.3324830.

[2] M. Kuś, "Do happy software developers write better code?" *Better Programming*, Jun.

[3] Available: https://betterprogramming.pub/do-happy-software-developers-write- better-code-5eee72b4b1c2

[4] "What Is Platform Engineering?" *Gartner*. https://www.gartner.com/en/articles/what-is-platform-engineering

[5] "Backstage by Spotify," *Supercharged devel oper porta ls*. https://backstage.spotify.com/ (accessed Jan. 08, 2024).

[6] "Cortex," *Internal Developer Portal*. https://www.cortex.io/ (accessed Jan. 08, 2024).

[7] "Internal Developer Platform & Portal," *Port*. https://www.getport.io/ (accessed Jan. 08, 2024).

[8] "OpsLevel," *The Developer Portal for High-Performing Teams*. https://www.opslevel.com/ (accessed Jan. 08, 2024).

[9] "Soundcheck," *a Spotify plugin for Backstage*. https://backstage.spotify.com/marketplace/spotify/plugin/soundcheck/ (accessed Jan. 08, 2024).

[10] S. Chandrasekaran, "Spotify Backstage," *How Soundcheck improves tech health and developer productivity at Spotify*, Sep. 19, 2023. https://backstage.spotify.com/blog/how-soundcheck-improves-tech- health-pt-1 (accessed Jan. 08, 2024).

[11] "Internal Developer Platform vs. Internal Developer Portal: What's The Difference?" https://www.qovery.com/blog/internal-developer-platform-vs-internal-developer-portal-what-is-the-difference/ (accessed Jan. 08, 2024).
[12] dweber019, "backstage-plugin-api-docs-spectral-linter/plugins/api- docs-spectral-linter at main · dweber019/backstage- plugin-api-docs- spectral-linter," *GitHub*. https://github.com/dweber019/backstage-plugin-api-docs-spectral- linter/blob/main/plugins/api-docs-spectral- linter (accessed Jan. 08, 2024).
[13] T. Charboneau, "What's New in OpenAPI 3.1.0?," *Nordic APIs*, Apr. 07, 2021. https://nordicapis.com/whats-new-in- openapi-3·