



Six Drivers of Productivity in 2022

Determine the Best Practises Driving Software Development Productivity

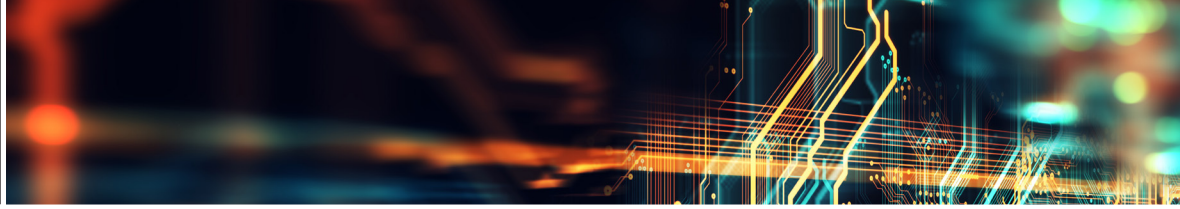
In every business, senior engineering leaders look at processes to increase the productivity of their development teams. In this report, Six Drivers of Productivity, we dive into the factors and recommendations contributing to the efficiency of software development organisations. Our aim is to support businesses in establishing these processes and improving efficiencies as they maximise on the increasing demand for software.

Our findings and recommendations are based on the adoption of an advanced machine learning statistical

modelling approach to analyse the drivers behind productivity within large financial service businesses.

The analysed data and engineered features were collected from a wide array of tools used by the clients to measure their correlation and impact on productivity and maintainability metrics. These included bug tracking systems (Jira, TFS Task, Rally etc.), code quality (BlueOptima and non-BlueOptima), build and deployment tools (Jenkins, TeamCity etc.). On analysing the activity across approximately 2,000 applications, we identified the most influential contributors to a developers productivity, which were objectively quantified to aid the clients in establishing or improving their software developer best practises. The six drivers that we found to have the greatest impact on a software estates productivity were:

- Difference in time zones between team members
- Longevity of developers on a project
- Sprint issues planned vs completed deviation
- Balancing code maintainability
- Continuous integration maturity
- Application size and straight-line coding effort



1. Difference in time zones between team members:

Finding

Our study found that the weighted time-zone difference between the location of less productive developers was four times higher than the top performing teams of that organisation. Essentially, the greater the difference in time zones between developer teams who were working on the same project, the less productive a team was overall. Whereas, teams that had a smaller difference between the time zones of their working locations, demonstrated higher levels of productivity.

Solution

Where possible, we recommend bringing teams together and hiring project teams in similar locations or at least similar time zones to increase productivity. This will enable greater collaboration on projects as well as fewer delays. These actions, supported by the adoption of collaboration tools and best practises to ensure team members are not isolated, will place the company in a strong position for the future.

2. Longevity of developers on a project:

Findings

BlueOptima classifies long-term developers based on their codebase activity. Developers who have been active in the codebase for more than six months and have recently delivered meaningful change, having developed expertise in certain areas, are grouped as long-term developers. We have found that applications where long-term developers deliver a significantly higher proportion of Coding Effort (CE) are more productive compared to those with an average or lower contribution from long-term developers.

Solution

We highly recommend that long-term developers are given enough time to remain active in the codebase. Increased activity means that they build a greater familiarity with the code and are able to make more meaningful contributions. Furthermore, if you pair your long-term developers with new developers, this enables the short-term developers to gain knowledge and expertise into becoming long-term developers. This helps new developers become confident with the codebase, know

and learn to manage the expectations. Sample representation of the feature scoring model to help application managers understand where their application is, in comparison to the top and bottom performers. Thresholds are established based on the applications that are performing better in terms of BCE/Day than the application being evaluated.

3. Sprint issues planned vs completed deviation (per assignee):

Finding

This is a relative measure of the work completed in a sprint by a developer, compared to what was planned to be delivered. Higher deviations between the delivery and the set targets suggest lower productivity levels. This is one of the top predictors of performance and shows that overburdening developers with tasks will have a direct impact on their productivity.

Solution

Software development teams should ensure that the developers are only allocated with tasks that can be realistically delivered within the sprints. Large tasks should be broken down into smaller pieces of work, by proper estimation in sprint planning.

4. Balancing code maintainability:

Finding

Through analysing the objective metrics across the maintainability of developers code, we found that the applications with the highest productivity are those that do not have extremely high or low maintainability. Interestingly it was about finding a healthy balance. Radical changes in approaches to keeping source code maintainable can lead to more variance in the codebase which can reduce the overall maintainability of code.

Solution

We recommend keeping consistent and evenly applied policies regarding the style and structure of your codebase to ensure its continued maintainability. Establish accurate thresholds based on industry and company best practises to assure the quality of your code and keep a healthy balance within your projects. This will help to reduce technical debt build-up and achieve faster times to market by reducing the time spent on maintenance.

5. Continuous integration maturity:

Finding

We have found that applications with a better defined, and well-adhered to continuous integration processes, are more productive than those that integrate less frequently. Developers who commit frequently (which results in automated testing etc.) and integrate the changes faster will tend to receive their reviews faster which in turn increases their releases to the end-user. Continuous Integration is a best practice in software development and we saw that teams adhering to this best practice, tended to be more productive. Our studies found that three times more builds were generated by top performing applications compared to applications with lower productivity.

Solution

It is worth investing in building continuous integration processes as they reduce repetitive and low-value-add release preparations for software development teams, leaving more time to deliver valuable functionality to users.

6. Application size and straight-line coding effort:

Finding

Straight-Line Coding Effort is a measure of the Coding Effort required to build an application from scratch to its current state. During our study, we found that teams working on larger applications were likely to achieve higher productivity. There could be several factors contributing to these findings but we saw that a notable contributor to larger well-managed teams were by having better tools and processes in place as well as experienced developers. In essence, we saw that it is highly beneficial to vary the seniority within teams to include both junior, mid-level and senior developers on projects.

Solution

We recommend consolidating the long-tail of small applications which are assuming similar operational logistics and identical domains into larger applications. This enables the smaller applications to benefit from the better processes that the larger teams are utilising. If merging these applications is not possible, try to mimic the best practises and processes of larger applications within the smaller applications where possible. Additionally, varying talent within a team is also highly recommended as it enables a greater optimisation of a team's productivity, hence why we suggest differing levels of seniority. This enables teams to be pushed, providing an opportunity for everyone to contribute different skills as well as compensate for one another so that the higher achievers can boost the more junior staff.

Who are BlueOptima?

We provide a SaaS technology that objectively measures software development efficiency. Our core metrics for productivity and code maintainability allow executives to make data driven decisions related to talent optimization, vendor management, location strategy and more.

Contact the Data Science Team

To discover powerful insights and determine areas of improvement specific to your organisation, reach out to our Data Science team and explore our custom analytics solutions at:

consulting@blueoptima.com

[Email Data Science](mailto:consulting@blueoptima.com)