**Training Software Leaders**

# **Part 1: Proposal**

***Value***

We are writing to advocate for the implementation of comprehensive software analytics training within our organization. As a leading web application company, staying competitive in today's fast-paced market requires us to make data-driven decisions and continuously improve our development processes. Software analytics provides invaluable insights into our development lifecycle, enabling us to identify areas for optimization, mitigate risks, and deliver higher-quality products to our clients.

By investing in software analytics training, we can unlock numerous benefits for our organization. We can streamline our development processes, reduce inefficiencies, and ultimately enhance product quality and customer satisfaction. Furthermore, having a standardized set of software engineering metrics based on accurate data will allow us to make informed decisions, allocate resources effectively, and prioritize initiatives that most impact our business goals. We observed this case in Fannie Mae that took most of their applications from a yellow and red Total Quality Index state to green. [1] Additionally, Analytics Driven Testing allows Analysts to predict errors. [2] Our following approach follows industry recommendations of tying our metrics with real tasks to make sure we are helping our team’s productivity. [3]

***Training Objectives***

1. **Understanding on the importance of Software Analytics and Quality Measurements.**

* Leaders will understand the significance of having a standardized set of software engineering metrics based on accurate data, serving as vital indicators of development progress, efficiency, and quality.

1. **Develop a standardized set of software engineering metrics based on accurate data.**

* Train leaders on implementing measurement tooling within each team to gather accurate metrics data, facilitating data collection, analysis, and reporting for informed decision-making.

1. **Create a culture of transparency and data-driven decisions.**

* Leaders will know the importance of fostering a culture of transparency and collaboration.

***Best Reference for Further Investigation***

While all the selected references offer valuable insights, "Software analytics in practice” [3] emerges as a foundational resource for further investigation. This article provides a comprehensive overview of software analytics and the practical challenges associated with its implementation. It lays the groundwork for understanding the complexities of implementing metrics and addressing the general challenges discussed in our context.

Additionally, the reason for using these specific metrics is laid out in "Simpler questions can lead to better insights" [4]. They are simple to explain, implement, and measure but provide a high-level quality and customer satisfaction metric.

Moreover, it sets the stage for integrating metrics into the broader Agile–DevOps transformation context, as discussed in "Using analytics to guide improvement during an Agile–DevOps transformation” as part of the *Analyzing Structural Quality* section [1] . It also resonates with the principles of "Lean software development: An agile toolkit" [5], emphasizing the importance of structured processes, such as contract structures, in enforcing metrics per team, bug, project, or feature.

# **Part 2: Training Outline**

***Training Objective Selection***

Develop a standardized set of software engineering metrics based on accurate data.

***Training Outline***

1. Introduction to Software Engineering Metrics

* Definition and importance of standardized software engineering metrics in optimizing development processes.
* Overview of critical metrics: lead time of new endpoints, Apdex measurement, time between defects, and defect counts.

1. Identifying Team-Specific Metrics

* Understanding the lead time of new endpoints and their relevance to backend development.
* Introduction to Apdex measurement and its importance in assessing user satisfaction.
* Exploring the significance of time between defects and defect counts in overall product quality.

1. Implementing Measurement Tooling

* Guidance on selecting and implementing tools to measure the lead time of new endpoints.
* Demonstrate tooling integration into backend development workflows.
* Recommendations for choosing suitable tools for Apdex measurement and usability testing.
* Hands-on training sessions for implementing and integrating defect-tracking tools into the development lifecycle.

1. Establishing Traceability and Accountability

* Establishing a process for associating metrics with specific project components, such as user stories or features.
* Demonstrating how to maintain accurate documentation of metric data for each requirement or reported issue.

1. Conclusion and Next Steps

* Recap of key concepts and techniques covered in the training.
* Discussion on the importance of ongoing metric monitoring and refinement.
* Encouragement for teams to implement learned strategies and continuously improve metric measurement practices.

# **References**

|  |  |
| --- | --- |
| [1] | B. Snyder and B. Curtis, "Using analytics to guide improvement during an Agile–DevOps transformation," *IEEE Software,* vol. 35, p. 78–83, 2017. |
| [2] | F. A. Batarseh and A. J. Gonzalez, "Predicting failures in agile software development through data analytics," *Software Quality Journal,* vol. 26, p. 49–66, 2018. |
| [3] | D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang and T. Xie, "Software analytics in practice," *IEEE software,* vol. 30, p. 30–37, 2013. |
| [4] | B. Turhan and K. Kuutti, "Simpler questions can lead to better insights," in *Perspectives on Data Science for Software Engineering*, Elsevier, 2016, p. 359–363. |
| [5] | M. Poppendieck and T. Poppendieck, Lean software development: An agile toolkit: An agile toolkit, Boston [u.a.]: Addison-Wesley, 2003. |
| [6] | H. Gall, T. Menzies, L. Williams and T. Zimmermann, "Software development analytics (dagstuhl seminar 14261)," in *Dagstuhl Reports*, 2014. |