

Aannotated bibliography example



**ASSIGN
BUSTER**

\n[[toc title="Table of Contents"](#)]\n

\n \t

1. [Sample Annotated Bibliography](#) \n \t
2. [Annotated Bibliography Sample](#) \n \t
3. [Works Cited](#) \n

\n[/toc]\n \n

Sample Annotated Bibliography

Chapter 10: The cost of defects: an illustrated history

The chapter is infinite for our ongoing study concerning software engineering principles. The chapter provides a general background of Boehm's assessment of Software Engineering Requirements. The chapter commences by providing a deeper assessment of the history of the Boehm's curve that provides a vertical scale representation ratio for the statistical section of software engineering. The logarithmic straight being explored is a straight-line assessment that provides an exponentiation of expensive and auto correction of data. The research provides a deeper assessment of the possible changes, amendments, and alterations that are important in software engineering. Each of the three conditions is provided using a straight line that is suggested through the smaller slope of the Boehm suggestion.

A vital section of the research is indicated on (p. 90), where software engineers are questioned where the data of costs of defects is generated from. In generating the data, the research advises the need to defragment projects based on their size. To prove its point, the research quotes findings

<https://assignbuster.com/aannotated-bibliography-example/>

from Bell Labs Safeguard, TRW data or IBM study (Fagan's paper). Bossavit (p. 91) solidifies these results by providing a deeper analysis of the Boehm's curve for more creative ways that provides the right needed in creating the right data trends.

Further research as evidenced in Bossavit (p. 92) changed the line graph to a hierarchical pyramid structure, which was a more simplified way of Boehm's curve. The simplified Boehm's curve provides a graphical reorientation of all relative costs for fixing an error. The Boehm curve was later transgressed into a histogram that provided statisticians with better and deeper data. The logarithmic scales provide a more concise way of managing data that makes it possible for managing and refining data. Also, an elaborated graph as one suggested by Bossavit (96) provided the right assessment of technical evaluation systems that were vital for software engineering principles.

Annotated Bibliography Sample

The changes on the topic provided the correct Extreme Programming that was reignited to embrace change. Bossavit (p. 97) proceeds by offering a deeper assessment on linoleum-floored. Bossavit (p. 97) provides Beck's curve whose primary variables include requirements, analysis, design, implementation, testing, and production. The most critical section as furnished by the Bossavit (p. 98) through reading curves that provides a continuous line while providing the right waterfall cycle. The representation through distorting and leading to the various horizontal axis and the functional cost of defects. While reading, Bossavit (p. 99) quotes the theory-laden diagram for the total costs defects for the phase. Backed by the theory, it is clear that Boehm curve helps in collecting full costs assessment

provides the right and the same data assessment that requires knowing what you mean through each of these terms. The fixed category involves a clearer assessment of the quality and the higher assessment of data proportion. In interpreting, Bossavit (p. 101) provides an alternative to Jones argument through Boehms for the squarely through expecting bad projects for the flatter defects and costs curves that are provided through an endless way for more deliberate enlighten and the projection of fixed expenses. According to Bossavit (p. 102), the fixed assessment for costs of change provides a clearer response that is based on the balancing agility and guiding more explicit representation of changes.

Clearly, a graphical assessment provides a best way to determine the progress of a software project. Graph systems help software engineers to develop an infinite model-driven development. The projects transformation approaches are implemented using various tools that are utilized by various developers at different stages of software developers (Bossavit, p. 101).; The research provides a deeper assessment on the personal experiences while providing an evaluation of the appropriate statistical aspects for the right software development. As suggested by Bossavit (p. 101) using graphical systems, system engineers analyze and verify the right software engineering components. The software engineering process helps in developing the ultimate idea for a complete code through high-level and systems without the need to program, test and debug possible failures. Focus of software engineering process, design, analysis and validation of high-level models. The modeling process brings behavior, requirements, structure for a specific

domain. The meta-model creates a constructive description for obtaining and valid DSML models.

;

Works Cited

Bossavit, Laurent. The Leprechauns of Software Engineering: How Folklore Turns into Fact and What to Do About It. , 2015. Print.