"I pledge my honor that I have abided by the Stevens Honor System."

1. We discussed in-process metrics for both traditional (waterfall) and Agile methodologies.
A) Detail three in-process metrics for traditional development models. You can use methods discussed in class. Detail the pro's and con's for each method. Provide insight on what type of development projects the method would be beneficial. (Max 300 words)
B) Detail three in-process metrics for Agile development models. You can use methods discussed in class. Detail the pro's and con's for each method. Provide insight on what type of development projects the method would be beneficial. (Max 300 words)
Every sentence must be precise. Every sentence must have data to back up your point. Must have a bibliography for your references used in answering these questions.

A) In-process metric for Traditional development…

1. Lines of code: the number of lines of code written per unit of time, such as per week or per sprint. This can provide insight into the productivity of the development team and is useful identifying potential bottlenecks or areas for improvement. However, this metric has some limitations, as it does not account for the quality or complexity of the code being written. It may encourage developers to write more LOC rather than focusing on writing good/maintainable code and therefore this metric is useful for projects with a focus on quantity over quality, such as projects with tight deadlines or large-scale codebases.

2. Defects: the number of defects identified during development, such as during code review or testing. This metric can provide insight into the quality of the code being written and can help identify areas where the development team may need to improve their processes or practices. However, this metric can be subjective and may not accurately reflect the quality of the code, as different reviewers/testers may have different standards for what is a defect. It is useful for projects with a focus on quality and reliability, such as projects with strict compliance or safety.

3. Requirements traceability: the extent to which the code being written can be traced back to the original requirements/specifications. This metric helps ensure the development team is building the right product, and the final product meets the needs/expectations of the stakeholders. However, this metric can be difficult to measure accurately, as it requires a detailed and well-documented set of requirements, as well as a disciplined and consistent approach to linking the code to the requirements. This metric is useful for projects with a high complexity or uncertainty, like projects with multiple stakeholders or changing requirements.

B) In-process metrics for Agile development…

1. Velocity: the velocity of the development team, which is typically measured in units of work, such as story points or features completed. This can provide insight into the team's ability to deliver value and can be used to predict the team's future sprint performances. However, this metric is limited, as it does not account for quality or complexity of the work being done, and may encourage the team to prioritize quantity over quality. This metric is useful for projects with a focus on delivering value quickly, such as projects with tight deadlines or very competitive markets.

2. Cycle time: an in-process metric for Agile development models, cycle time, is the amount of time it takes for a piece of work to be completed from start to finish. This metric can provide insight into the efficiency and effectiveness of the team's processes. It can help identify areas for improvement. However, this metric can be affected by external factors, like interruptions or dependencies, and may not accurately reflect the team's ability to deliver value. This metric is useful for projects with a focus on continuous improvement and optimization, such as projects with a large number of small/independent tasks.

3. Burn-down chart: A third in-process metric for Agile development is the burn-down chart, which is a graph representing the amount of work remaining in the sprint. It can provide visibility into the team's progress and can help identify potential problems or risks. However, this metric can be affected by changes in scope / priorities, and it may not accurately reflect the team's capacity or ability to deliver value. This metric is useful for projects with a focus on transparency and collaboration, like projects with distributed or cross-functional teams.

Bibliography:

In-Process Metrics and Reports | Quality Management Models. (n.d.). Flylib.com. Retrieved December 11, 2022, from https://flylib.com/books/en/1.428.1/in_process_metrics_and_reports.html

In-Process Metrics and Quality Management | In-Process Metrics for Software Testing. (n.d.). Flylib.com. Retrieved December 11, 2022, from https://flylib.com/books/en/1.428.1/in_process_metrics_and_quality_management.html

Lines of Code | Complexity Metrics and Models. (n.d.). Flylib.com. Retrieved December 11, 2022, from https://flylib.com/books/en/1.428.1/lines_of_code.html

Choras, M., Springer, T., Kozik, R., Lopez, L., Martinez-Fernandez, S., Ram, P., Rodriguez, P., & Franch, X. (2020). Measuring and Improving Agile Processes in a Small-Size Software Development Company. *IEEE Access*, *8*, 78452–78466. https://doi.org/10.1109/access.2020.2990117

Gustafsson, J. (2011). *Model of Agile Software Measurement: A Case Study*. https://publications.lib.chalmers.se/records/fulltext/143815.pdf

Staff, E. (2012, June 11). *Using requirements traceability with model-driven development*. Embedded.com. https://www.embedded.com/using-requirements-traceability-with-model-driven-development/

Julia Nelson

"I pledge my honor that I have abided
by the Stevens Honor System."