

Original Article

AI: Claude Code Model Comparative Study in MAPI

Sivamurugan Perumal

Nous infosystems Pvt Ltd, OR, United States of America.

Corresponding Author : sivamurugan.perumal@gmail.com

Received: 18 May 2026

Revised: 27 June 2026

Accepted: 12 July 2026

Published: 29 July 2026

Abstract - Claude Code is one of the agentic tools that can read, modify, executes and integrate with the IDE, development tools, which helps us to create or fix an issue [1]. The work can be accomplished by reading multiple files and processing the statement to get the right output. Claude code model supports various models, a few are free, and a few are paid service either monthly or yearly. In this research, sample .Net C# console simple application was developed, which would deliver MAPISend [3] and MAPISave[4] email options of the E3 instance developed with the following models: Sonnet-4.6[2], Opus3, Opus 4.6, Opus 4.7, Opus 4.8 [4] and Haiku [10]. The primary purpose is to analyze how each model performs for the same prompt on multiple trials to analysis which is best. It evaluates the efficiency, service offered, usage, correctness and architecture and output generated structure. Results: Finally, this research would help developers to opt for the right model with the default option suggested by Cloud Code, in terms of effectiveness, correctness, time effectiveness, and LOC (lines of code). Claude suggested Sonnet-4.6, comparatively Opus 4.8, based on Bench pro percentage Table 1. Based on real-time evaluation, the research concludes OPUS 4.8. high yields the better results compared to sonnet 4.6.

Keywords - AI, Anthropic, Claude Code OPUS 3, OPUS 4.6, OPUS 4.7, Sonnet-4.6, Haiku, OPUS 4.8, MAPI, E3 OUTLOOK, C#, .NET 4.8, MAPISend, MAPISave, PVS Studio.

1. Introduction

Claude Code is an AI-powered assistant that assists developers to generate an application by using either of its available models (free and pro plan). The primary benefits of these models help developers to generate the code, blocks and functionality and enhance productivity and efficiency.

Sample prompt used on the above models to evaluate efficiency by generating a C# application which send email and saves as a draft on Outlook E3 instances from windows app using MAPI messaging [6]. The Models used for evaluations are Sonnet-4.6[2], OPUS 3, OPUS 4.6, OPUS 4.7, OPUS 4.8 and Haiku. Same prompt, how it gets evaluated on these models and output provided and throughput used.

MAPISend[3] sends an ANSI message; in C#, a wrapper class is generated, and it is specific to OS operating systems. MAPISave[4]: This functionality saves the message into the message store. In our case, it would be outlook client or the E3 instance of Office 365 Outlook. It is not advisable to use it any sooner than suggested on the Microsoft portal.

Claude Sonnet-4.6[2] is Anthropic's latest 2026 model, which is available as a free and pro plan. It started at \$3/\$15 per million tokens. Which features 1 million tokens on beta.

2. Literature Review

Claude Code Sonnet-4.6[2] lot improvements on consistency, instructions following. This model is used in coding, computer use, agent planning, knowledge work and design. This model is more widely used in computer skills than the earlier sonnet model. As this is a new model, anthropic have done extensive safety evaluations. Safety researchers conclude there are no major concerns. The computer use percentage rose to 72.5 compared to the previous 61.4%, which is more or less equivalent to Opus 4.6[7]. Safety evaluations are comparatively safer than other models. Smartest model comparatively to OPUS 4.6, which was released in November 2025.

Opus 4.8 [5] High model is used for complex tasks as per Anthropic. They claim it works 2.5 times faster than the previous model, Opus 4.7[8] and is now 3 times cheaper than it was with the previous model. This allows dynamic workflows that tackle the large-scale problems. Evaluation of the model on each criteria sonnet-4.6 benchmark has improved across all Opus of Claude's model based on their metrics [2][4]. The parameters considered are agentic coding (SWE-Bench Pro), agentic terminal coding (Terminal Branch), multidisciplinary reasoning (Humanity's Last Exam HLE), agentic computer use (OSWORD-Verified) and agentic financial analysis (Finance Agent v2 and V1.1). Percentage reflections are shown in table 1.0 with each model. [2][4].



Table 1. Model Evaluation

Claude Model	Sonnet-4.6	Opus 4.8	Opus4.6
Agentic Coding SWE-Bench Pro	79.6%	69.27%	80.8%
Agentic Terminal Coding (Terminal Branch)	59.1%	74.6%	65.4%
Multidisciplinary reasoning: Humanity's Last Exam HLE	33.2% - 49.8%	49.8%- 57.9%	40.0% -53.0%
Agentic Computer Use OSWORD Verified	72.5%	83.8%	72.7%
Agentic Financial Analysis (Finance Agent V2 and V1.1)	63.3%	53.9%	54.5%

MAPI [11] is nothing but Microsoft Outlook Messaging API (MAPI). Transport and application used to maneuver email data, create email messages and store the data in a folder. Concepts apply to Outlook 2013 and 2016. This is a combination of an Application Programming Interface (API) and a Dynamic-Link Library (DLL) component. The intention of the interface is to create and access diverse messaging applications, and the link library contains an interface subsystem that can manage the communication between the front-end messaging application and the back-end messaging system.

MAPI Send[3] targeted platform is windows which uses the MAPI header. MAPI Save[4]. The functionality of this is to save a message.

PVS-studio [9] is a static analyzer used to check the code quality, which would help to identify any dead code and detect any potential vulnerabilities. The tool can be integrated into the Visual Studio IDE for analysis, and metrics are gathered based on that.

3. Methodology

Claude's models are used to generate a sample application that attaches the file selected from the local machine and launch outlook/E3 instance client in draft mode. It uses the MAPI Send and MAPI Save functionality of the MAPI [11] behavior. The application is an executable developed on the C# tech stack, which runs on the Windows OS platform.

The same prompt was used on all models, and the prompt which were used to create the application is "Create .NET 4.8 sample application to send email with MAPI message on E3 instance outlook along with an attachment. Ensure it has the MAPI Send and MAPI Save options. The attachment user can browse and select."

The prompt is executed on the Claude web interface by selecting the respective models and default options, and metrics are collected for evaluations. The approaches and better results are highlighted in green on respective tables for reference. Limits on tokenization utilization are applicable to both the pro / free plans on continuous execution.

1. Time consumed for application generation on each model on 3 trials. Time taken in Table 2.
2. The generation process explains the method used, which is suggested by Anthropic, with better utilization of the token. What are the service plans and output previews stated in Table 2.1?
3. Code quality, the generated code was launched on VSIDE with the help of PVS Studio [9] extensions, detecting bugs and security weaknesses. Table 3
4. Includes a list of files and directories associated with the model. Table 4
5. Test cases: Evaluating the generated code by manually created test cases and output yields. Table 5

Table 2. Application Generation Time Taken

Claude Model / Trials	Trial 1	Trial 2	Trial 3
Sonnet-4.6	3	4	5
Opus 4.8	4	7	5
Opus 4.7	3	4	4
Opus 4.6	5	3	2
Haiku	2	3	6

Table 2.1. Application Generation Process

Claude Model	Sonnet-4.6	Opus 4.8	Opus 4.7	Opus 4.6	Haiku
Default Method	Low	High	Extra	Medium	Extended
Service Plan	Free	Pro	Pro	Pro	Free
Preview	No (Fig 1)	Yes (Main) (Fig 2)	No (Fig 3)	No (Fig 4)	Yes (Read.m d) (Fig 5)

Table 3. Code Quality (PVS Studio)

Claude Model	Sonnet-4.6	Opus 4.8	Opus 4.7	Opus 4.6	Haiku
High	0	0	0	0	0
Medium	0	0	0	0	0
Low	0	0	0	0	0
Warnings	0	0	0	0	0
False Alarm	0	0	0	0	0

Table 4. Includes Application Generated

Claude Model	Sonnet-4.6	Opus 4.8	Opus 4.7	Opus 4.6	Haiku
File Count	4	3	4	4	6
Mark Down	1	1	1	1	4
Downloadable	Yes	Yes	Yes	Yes	Yes
Total # Lines	708	561	458	697	627
Exception Handling	No	No	No	Yes	No

Table 5. Test Case Execution

Claude Model	Sonnet-4.6	Opus 4.8	Opus 4.7	Opus 4.6	Haiku
Cases Count	22	22	22	22	22
Applicable	14	14	14	14	14
Pass Count	12	14	12	12	12
Fail Count	2	0	2	2	2

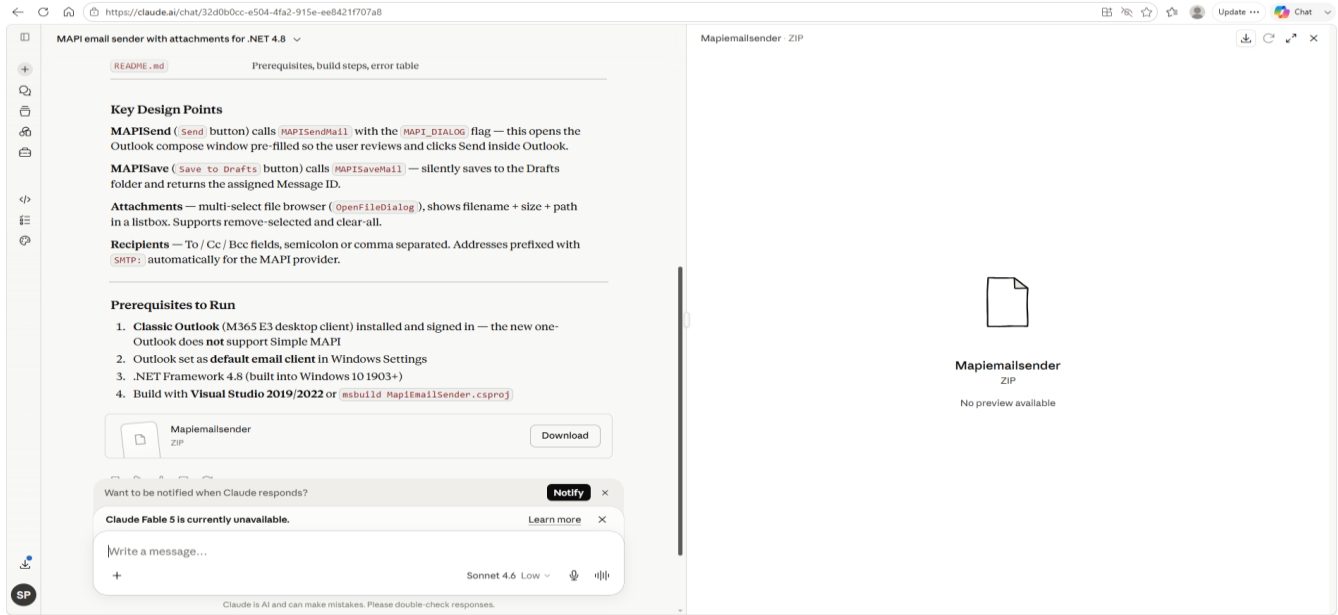


Fig. 1 Sonnet-4.6 Low

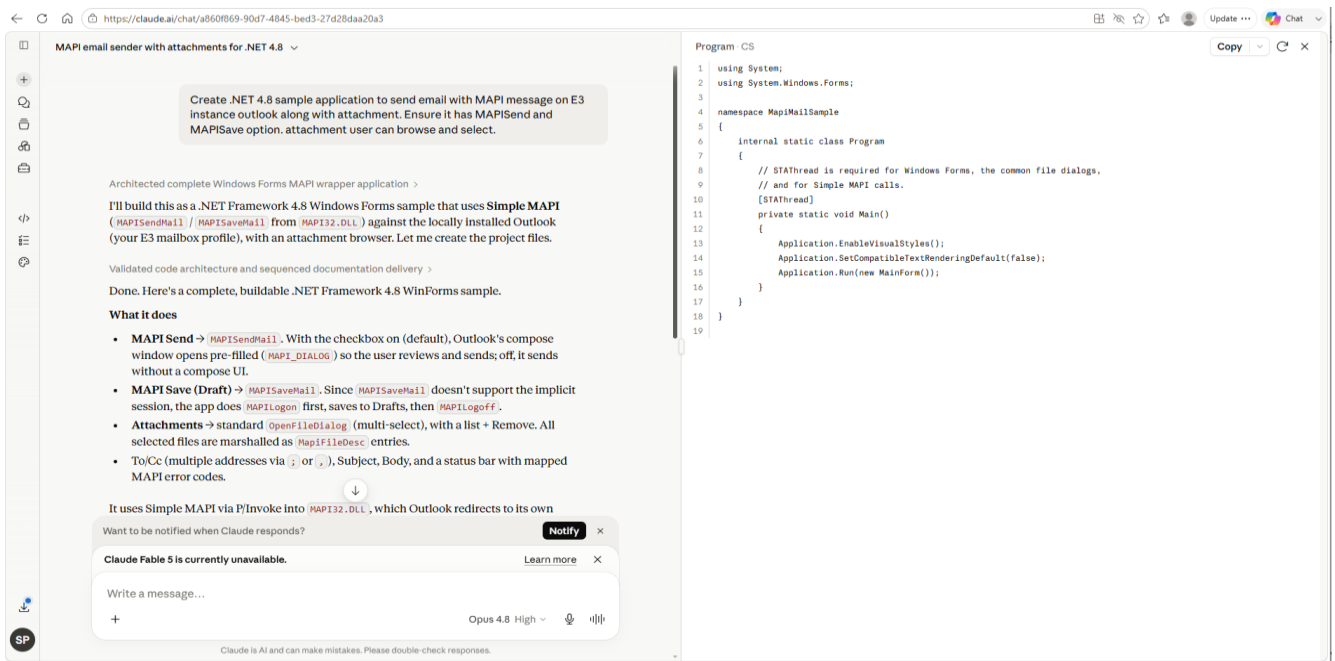


Fig. 2 Opus 4.8 High

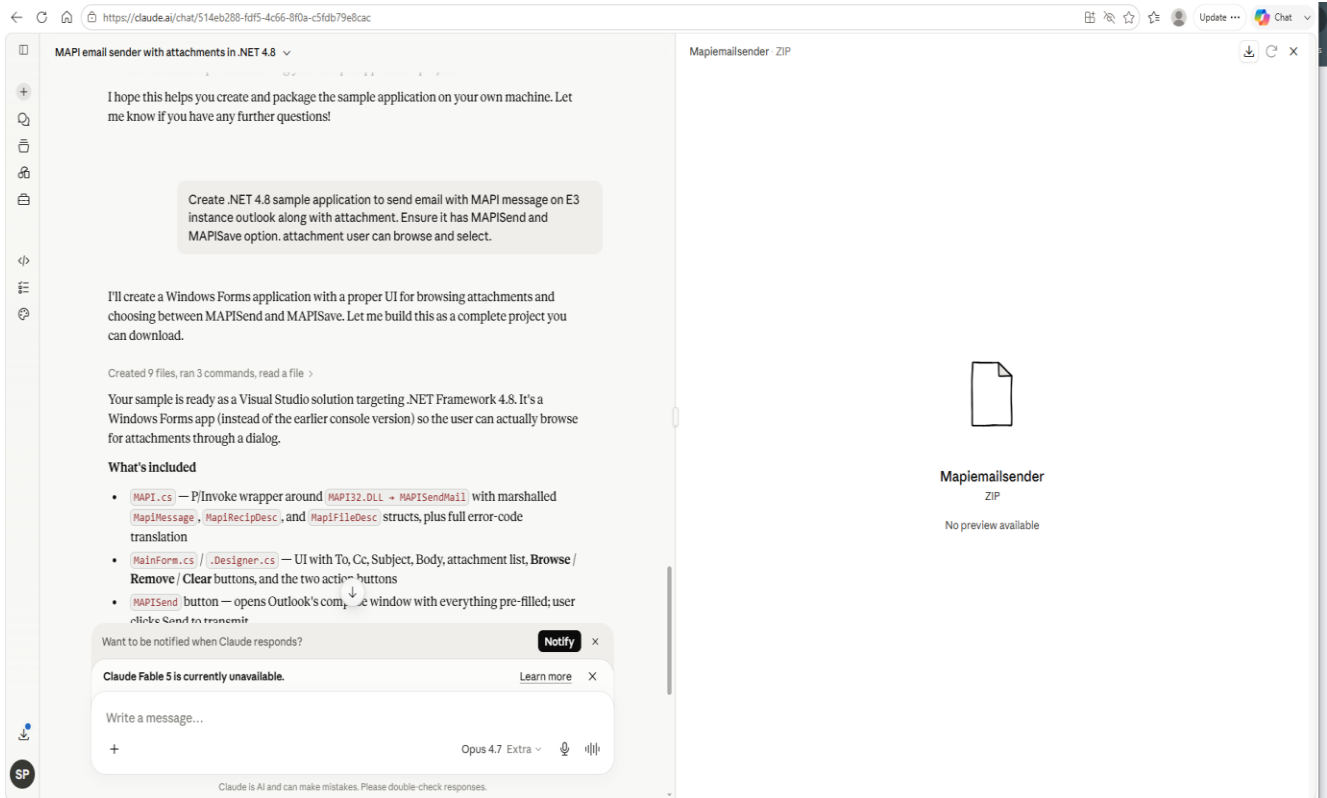


Fig. 3 Opus-4.7 Extra

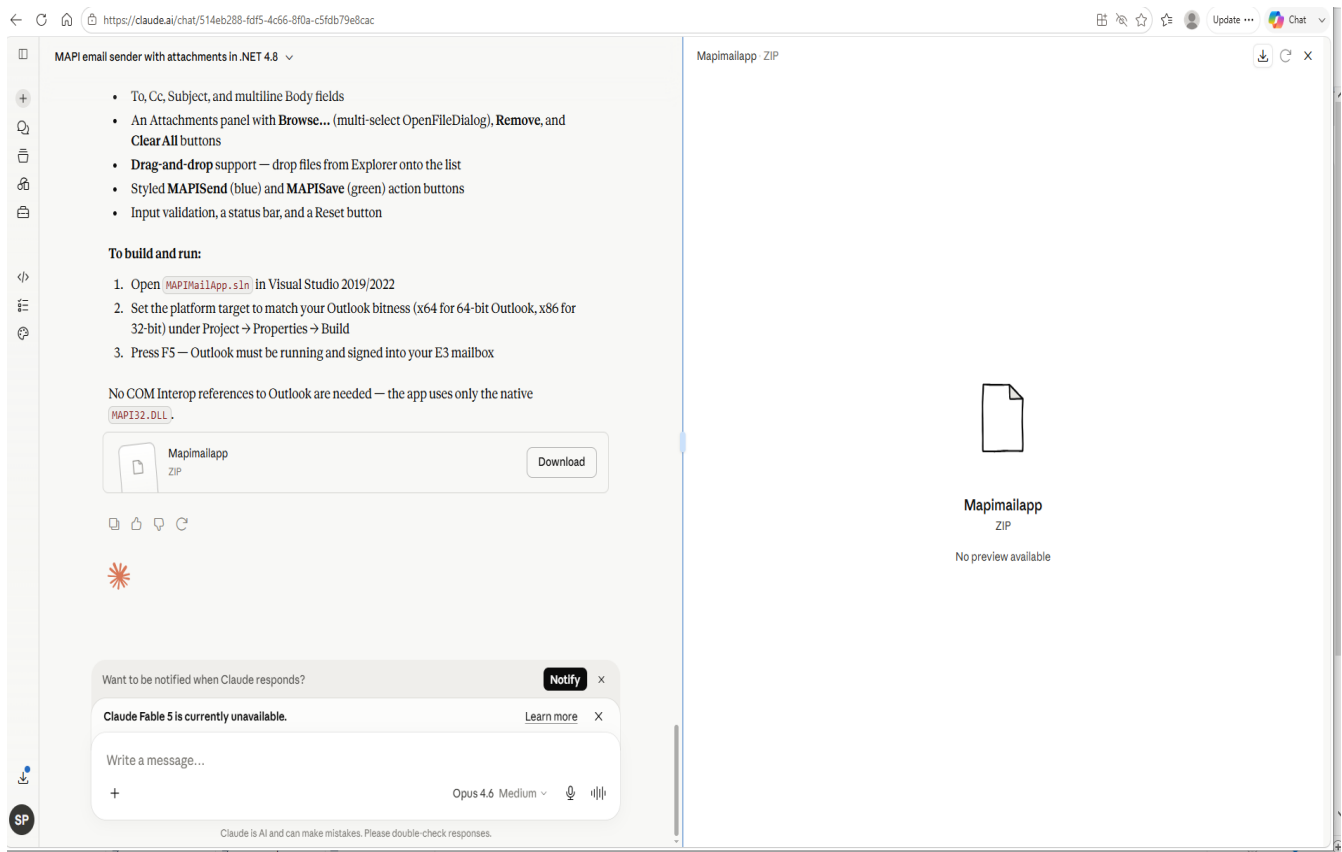


Fig. 4 Opus 4.6 Medium

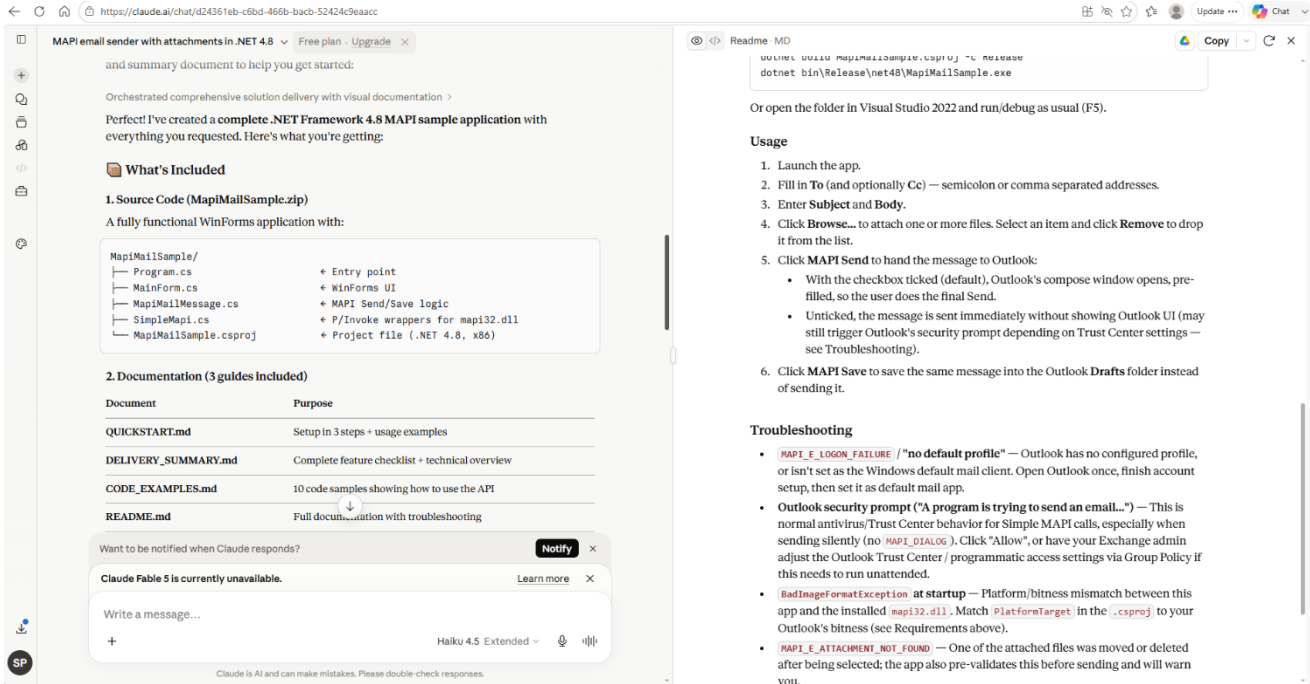


Fig. 5 Haiku Extend

4. Results and Analysis

This section contains two vital parts, mainly the execution procedures and the comparison between the various Claude code models.

4.1. Execution Procedure

4.1.1. Application Generation Time Taken

To evaluate the application generation, consider the following models: Sonnet-4.6, OPUS 4.8, OPUS 4.7, OPUS 4.6 and Haiku with their default method suggested by the Claude interface. The facts considered for the evaluations are the time taken to generate the application for the same prompt on multiple trials, as referenced in Table 2.

4.1.2. Code Quality

Code quality was evaluated with a tool, PVS studio which integrates with the Visual Studio IDE. PVS studio which helps to find errors and vulnerability on the code reference Table 3.

4.1.3. Includes

Includes evaluating the files generated as part of the application and any additional files, the number of lines generated, and markdowns to understand the steps in each mode. The data collected are shown in Table 4.

4.1.4. Test Case Execution

It covers the total functionality of MAPI, not just Send and Save functionality. Overall cases, actual cases executed associated to generated application. Pass and fail counts data are associated with Table 5.

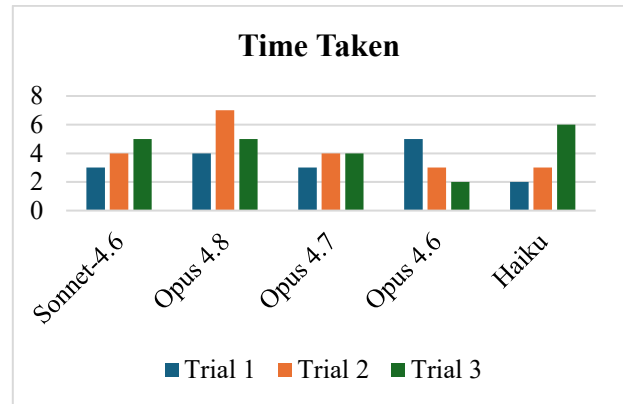


Fig. 6 Application Generation Time Taken (3 Trials)

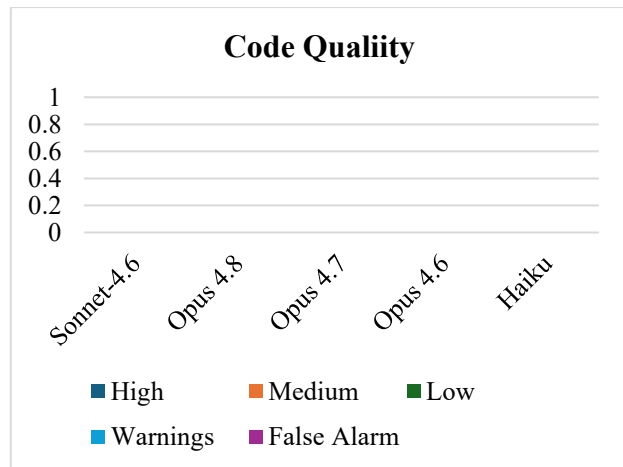


Fig. 7 Code Quality

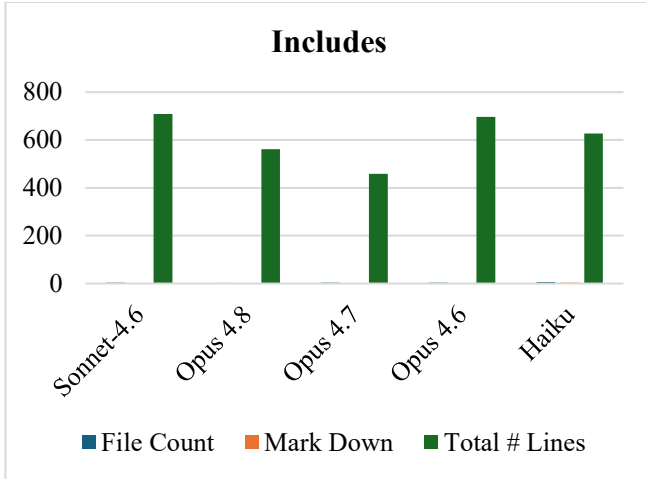


Fig. 8 Includes File Generation

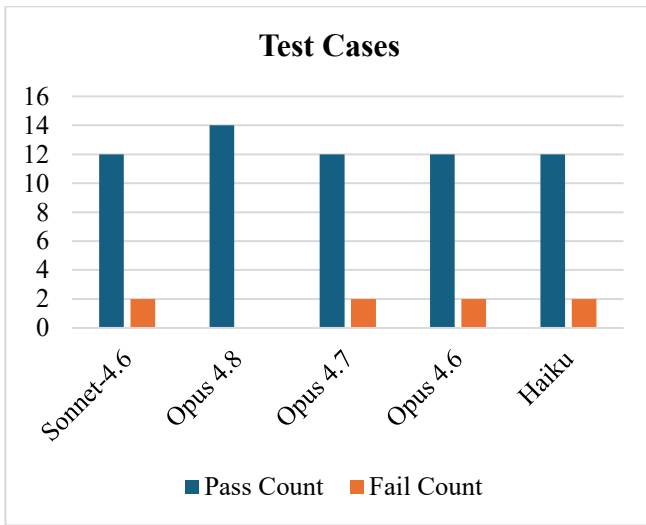


Fig. 9 Test Case Execution

4.2. Comparison

Based on metric evaluations Opus 4.8 method performed well compared to the other Claude Code models. In some cases not much difference, but Opus 4.8 performed well [5].

5. Limitations and Discussions

Not all Claude models are available for free, and the free version has limitations on a daily basis, as continuous chains of operations or large executions will get a timeout. The Pro version too has limits, but compared to the free it is few hours will be available. Higher utilization of the token happens when the default mode is not opted for. Code quality is the same on all evaluated models. Total LOC (Lines of Code) on application Sonnet-4.6 [2] perform well compared to their previous model of Sonnet-4.5 and equal to Opus 4.6.[7]. Exception handling does not error code is captured and shown as a message, mostly on all except for Opus 4.8.

6. Conclusion

Based on evaluations and metrics, Opus 4.8 performs well compared to other models on the default method. Opus 4.8 also has a high option as the default method and is available on the Pro plan, which could yield better results compared to other options. Same functionality with fewer code and the same results. Future work on various other method options on the same model can be evaluated, which will provide a gateway for research to choose the right option for execution. Same evaluations can be made with complex problems, which can lead to different yields on the code quality and case verifications, and time consumption for generating the application. Performance verification on generated applications with various models can also be considered for future research. Conclusion overall OPUS 4.8 High method performed well as per statistic data referred to in the above tables.

References

- [1] Claude, Claude Code, Claude, 2026. [Online]. Available: <https://claude.com/product/claude-code>
- [2] Anthropic, Introducing Claude Sonnet-4.6, 2026. [Online]. Available: <https://www.anthropic.com/news/claude-sonnet-4-6>
- [3] Windows App Development, MAPISENDMAIL Callback Function (mapi.h), Windows App Development, 2024. [Online]. Available: <https://learn.microsoft.com/en-us/windows/win32/api/mapi/nc-mapi-mapisendmail>
- [4] Windows App Development, MAPISAVEMAIL Callback Function (mapi.h), Windows App Development, 2022. [Online]. Available: <https://learn.microsoft.com/en-us/windows/win32/api/mapi/nc-mapi-mapisavemail>
- [5] Anthropic, Introducing Claude Opus 4.8, 2026. [Online]. Available: <https://www.anthropic.com/news/claude-opus-4-8>
- [6] Windows App Development, Windows Messaging (MAPI), Windows App Development, 2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/_mapi/
- [7] Anthropic, Introducing Claude Opus 4.6, 2026. [Online]. Available: <https://www.anthropic.com/news/claude-opus-4-6>
- [8] Anthropic, Introducing Claude Opus 4.7, 2026. [Online]. Available: <https://www.anthropic.com/news/claude-opus-4-7>
- [9] PVS Studio. [Online]. Available: <https://pvs-studio.com/>
- [10] Anthropic, Introducing Claude Haiku 4.5, 2025. [Online]. Available: <https://www.anthropic.com/news/claude-haiku-4-5>
- [11] Office Add-ins, Outlook MAPI Reference, Office Add-ins, 2016. [Online]. Available: <https://learn.microsoft.com/en-us/office/client-developer/outlook/mapi/outlook-mapi-reference>