**Question 1**

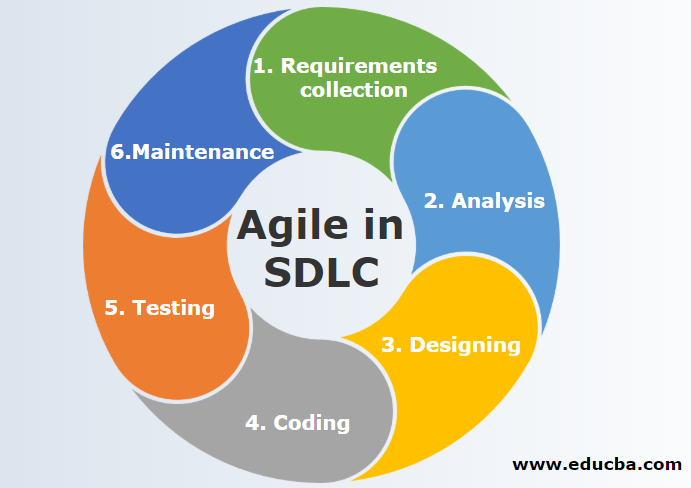
Create a 2-column multi-line table. In the left-hand column, include the software development stages of the Scrum agile life cycle approach to project management. In the right-hand column, describe the processes which you recommend are applied at each stage to ensure that secure software is produced at the end of the development. To support the preparation of your response, you can refer to the following literature:

Sharma, A. & Bawa, R. K. (2020) Identification and Integration of Security Activities for Secure Agile Development. *International Journal of Information Technology*.

Agile software development is considered the way to go in today's software development and project planning world. Before the agile methodology, project managers used the waterfall method to plan, execute, deliver and maintain a software project. The biggest problem with the waterfall methodology is that the project manager does not go back to the previous stage once a stage is completed. This is a big mistake. The reason for this is typically with the waterfall methodology, the client does not see the software until delivery. At that point, the client might have changed their mind or want to add a new requirement to the project. In addition, after the client tests the final product, they might realise that they prefer some other feature to the one they agreed upon in the requirement stage and would request the software to be changed. This is an issue because making a simple fix or changing a feature at the end of the product life cycle is much harder than early on.

This is why the agile methodology is much superior. The agile methodology collects the product requirements, and a single feature is analysed, designed, developed, and tested. The client is then shown the software up to that point and would be able to see and test this feature. If they do not like it, changing it is simple since the software is still early in its life cycle. However, if the client is satisfied, the following feature requirement is collected, and the whole agile cycle happens again. Unlike the waterfall methodology, the client is always involved in the development and testing of the software.

|  |  |
| --- | --- |
| Agile software life cycle stages | Description |
| Requirements collection | Collect the requirements of the software from the client going over every single small detail |
| Analysis | Plan the software architecture, and create a timeline and the project sprints, GANT chart, scrum meetings, budget and more |
| Designing | Design the Software user interface |
| Coding | Start programming an MVP or a single feature |
| Testing | Test the MVP or the single feature and make sure everything works |
| Maintenance | Deploy the feature and show it to the client |

Part of the agile methodology is sprints and sprint points. Typically a sprint is one week's worth of work. However, every organisation does it slightly differently. Some organisations make each sprint two weeks or four weeks. Once all the sprints are set up, each task or user story is assigned a sprint point. The higher the sprint point is, the more difficult to accomplish the task is. Typically tasks are created to be as small as possible to estimate the project length and deadline better.

Part of planning the sprints is creating velocity, burn-down and burn-up charts. These charts are essential to figuring out how productive the team is on any given sprint and how much of the project is done.

References:

*Agile in SDLC: Characteristics and functionality of agile in SDLC* (2021) *EDUCBA*. Available at: https://www.educba.com/agile-in-sdlc/

*The Agile Software Development Life Cycle: What is agile SDLC and how to use it?* (no date) *The Agile Software Development Life Cycle: What is Agile SDLC and How To Use It? | Brocoders blog about software development*. Available at: https://brocoders.com/blog/agile-software-development-life-cycle/

Taylor, H. (2022) *How to use Burndown and burnup charts in Agile*, *AI Project Management Software For Professional Services*. Forecast. Available at: https://www.forecast.app/blog/burn-down-burn-up-charts