**Waterfall** follows a continuous flow of development

Goes through 8 stages (conception, initiation, analysis, design, construction, testing, implementation, and maintenance). Once a previous stage has been completed, the developers move onto the next

Once a step is completed developers cannot revisit a step without scrapping the whole project and starting from the beginning.

Because of this the project must follow a rigid and almost fool proof plan which is put in place prior to beginning the project. Must be followed carefully as there is no room for error

**Advantages:**

-Stresses meticulous record keeping. Having such records allows you to improve upon existing program in the future.

-Client has a definite idea of the project and they know what to expect. Size, cost and timeline are all clearly documented and explained. They know what they will be getting in the end

-In case of employee turnover, strong documentation means it has minimal project impact

**Disadvantages:**

-Once a step has been completed, devs cannot go back to the previous step to make changes

-Relies heavily on initial requirements, if these are wrong in any way it can be catastrophic for the project

-If a requirement error is found or a change needs to be made, project must start from beginning with all new code

-Does not take into account a client’s ever evolving needs.

-Testing is only done at the end of the project. If bugs are written early but discovered late, they may influence how the rest of the code is written

**Should be used when:**

-When there is a clear picture of what the final project should be

-When clients do not have the ability to change scope once the project has begun (still does not account for errors that may be made by the developers)

-When definition is more important than speed

**Possible brief intro:**

Not sure what to write here yet

**Agile** was created in response to the apparent disadvantages of waterfall. This follows an incremental approach as opposed to a sequential one

The developers start with a simplistic design approach and begin work on small modules. Work on said modules is done in weekly or monthly sprints. At the end of each sprint project priorities are evaluated and test are run. These allow for bugs to be discovered or customer feedback to be incorporated before the next sprint is run

The process is with its lack of proper planning is often criticized for its collaborative nature that focuses on principals rather than process

**Advantages:**

-Allows for changed to be made after initial planning. Rewriting code as the client makes changes to their initial requirements are expected

-Because this allows for changes mid-development you can make changes to the program which keeps you up to date with the latest industry standards

-Because clients provide feedback of the end of each sprint, they are fully able to see the project as it develops before them and make changes as they see fit. Client will end up getting the product they desire.

-Testing at the end of each sprint ensures that bugs are found and fixed mid-development cycle and are not left to the end of the project.

-Due to the thorough testing of agile, projects are fit to be released at the end of many different cycles. More likely to reach it’s intended release date

**Disadvantages:**

-Can become disorganised with less successful project managers. This may lead to the product being late or over budget.

-As it does not have an initial plan before development, the project can differ greatly from what was originally intended (This can be seen as negative or positive)

**Should be used when:**

-Rapid production is more important than the overall quality of the product

-Clients are freely able to change the scope of the project

-There is no clear picture of what the final product should look like

-When your team has skilled developers who are able to adapt and think independently

-The product is being developed for an industry with rapidly changing standards

**Possible conclusion:**

Both methodologies offer a unique approach to project design and it’s clear there is no ‘one size fits all’ solution to project design. In more rigorous or controlled environments it’s evident that the favourable methodology would be the waterfall. However, when a project allows for continuous feedback or exists in an ever-changing industry, agile methodology is clearly the superior choice. That said it is possible to mix the two methodologies if you project does not strictly conform to any one of these archetypes. These are certainly useful guides on how to approach a whole number of different projects that could be created.