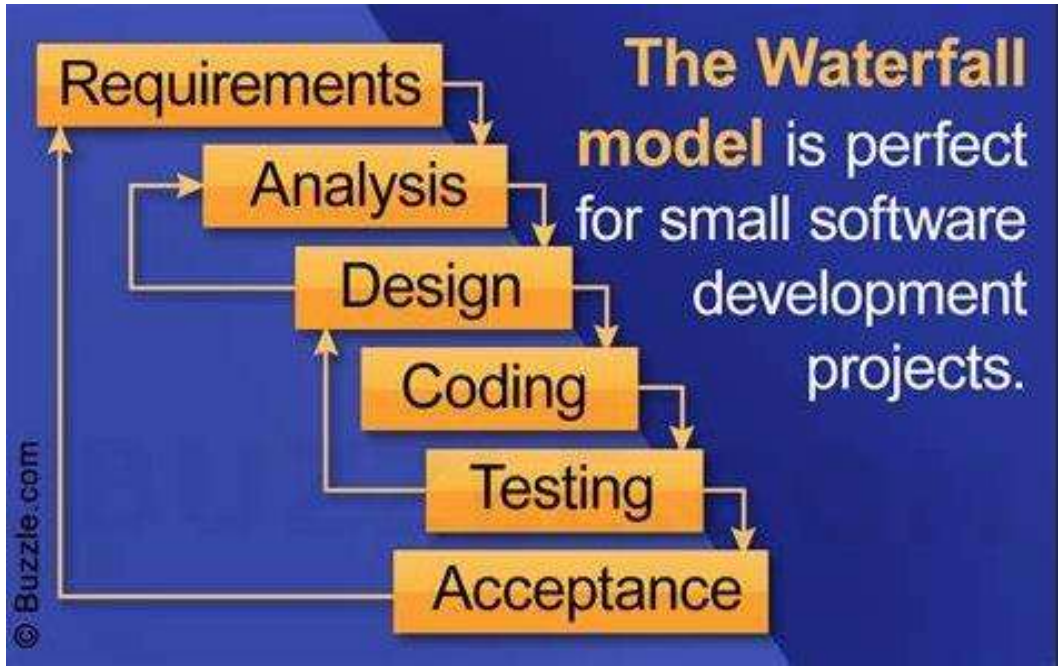
- AGILE AND WATERFALL
- GIT COMMANDS

- VISHAL KADAM

WATERFALL METHOD

- Waterfall is a traditional project management method that follows a **step-by-step** approach.
- **Waterfall** is a traditional project management methodology that follows a **linear** and **sequential** approach.
- This means that the project is divided into distinct, non-overlapping phases, with each phase needing to be completed before moving on to the next one.
- In Waterfall, the project flows **downwards**, much like a waterfall. The phases of the project follow one another in a clear, defined order.

- Each phase must be completed before the next one can start. For example:
- Planning → Design → Development → Testing → Deployment → Maintenance



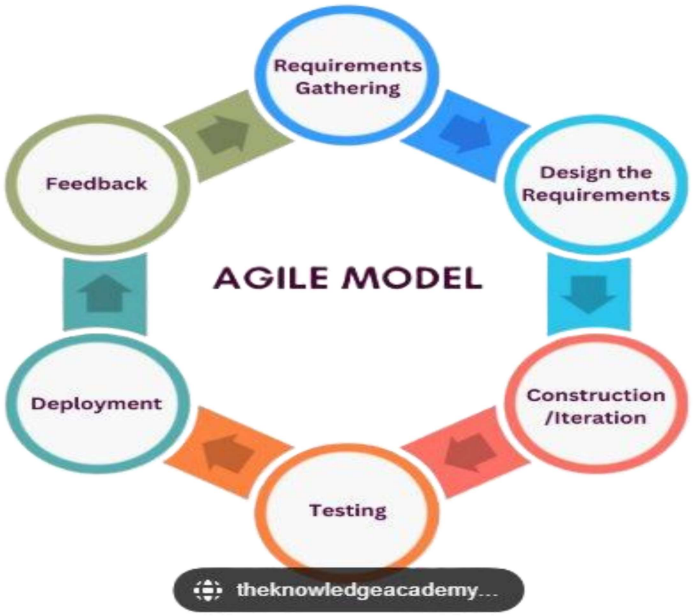
EXAMPLE OF WATERFALL IN PRACTICE:

- Let's say a company is building a **new building**. The requirements (e.g., floor plans, materials, regulations) are **clearly defined** from the start. The process proceeds step-by-step:
- **Planning**: The design and blueprints are created.
- **Design**: Engineers and architects finalize the design.
- **Construction**: The building is constructed as per the design.
- **Testing**: After construction, safety inspections and tests are done.
- **Deployment**: The building is handed over to the owners.
- **Maintenance**: The building undergoes periodic maintenance.
- Throughout this process, each phase has a defined scope, and it's difficult to change the design once construction has begun, which is typical for Waterfall.

AGILE METHODOLOGY

- **Agile** is a way of managing projects by breaking them down into **small, manageable parts** (called sprints) and improving the product step by step.
- Focuses on flexibility, quick changes, and delivering value to customers early and often
- allowing teams to get **real-time feedback** and make adjustments quickly. The goal is to create products in **small, iterative cycles** and make continuous improvements based on feedback.

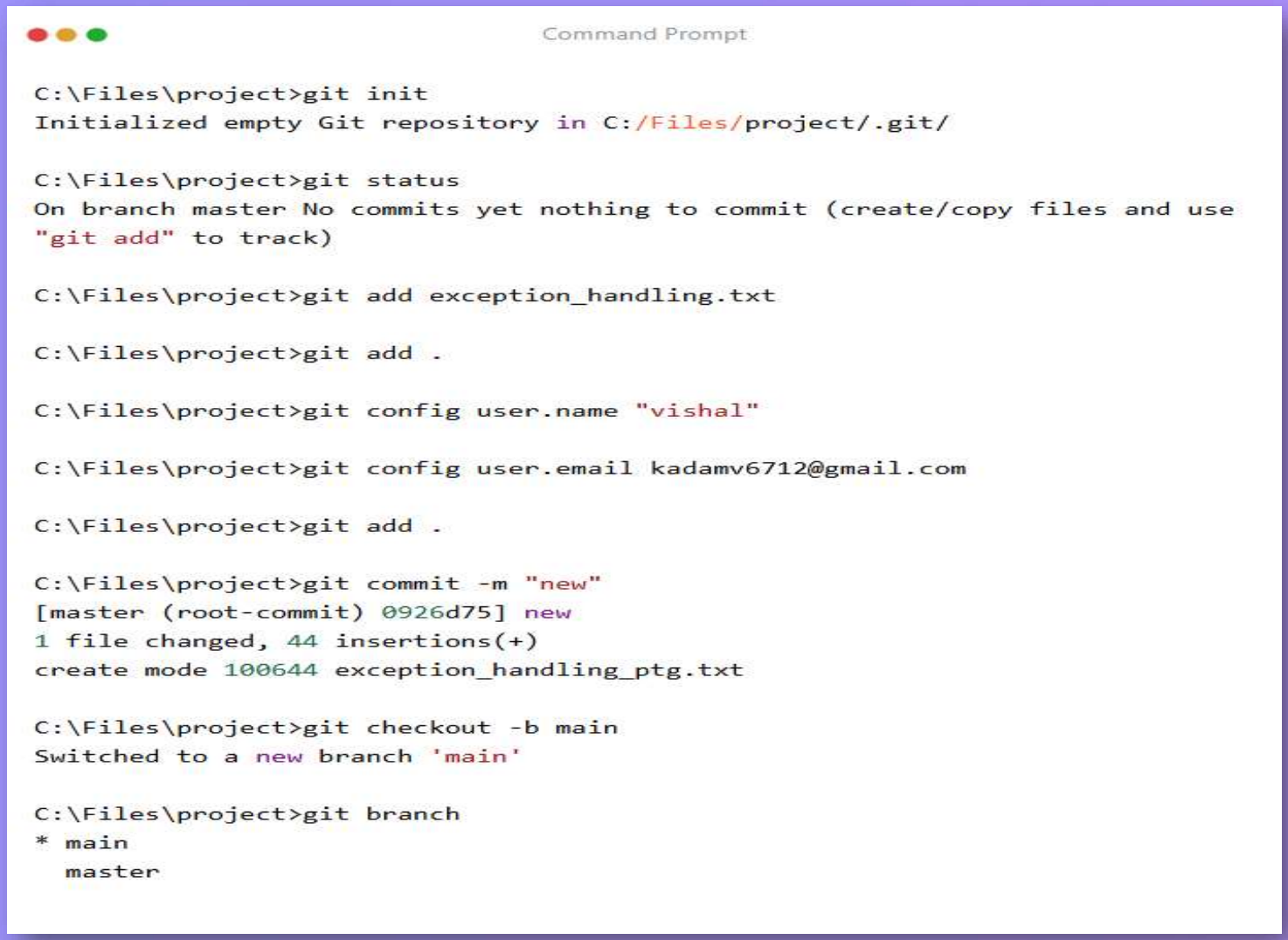
- The main focus of Agile is delivering a **working product** that adds value to the customer at the end of each sprint, rather than just focusing on documentation or lengthy planning.
- Regular meetings or **reviews** allow customers to see the progress, provide feedback, and make changes to the product.

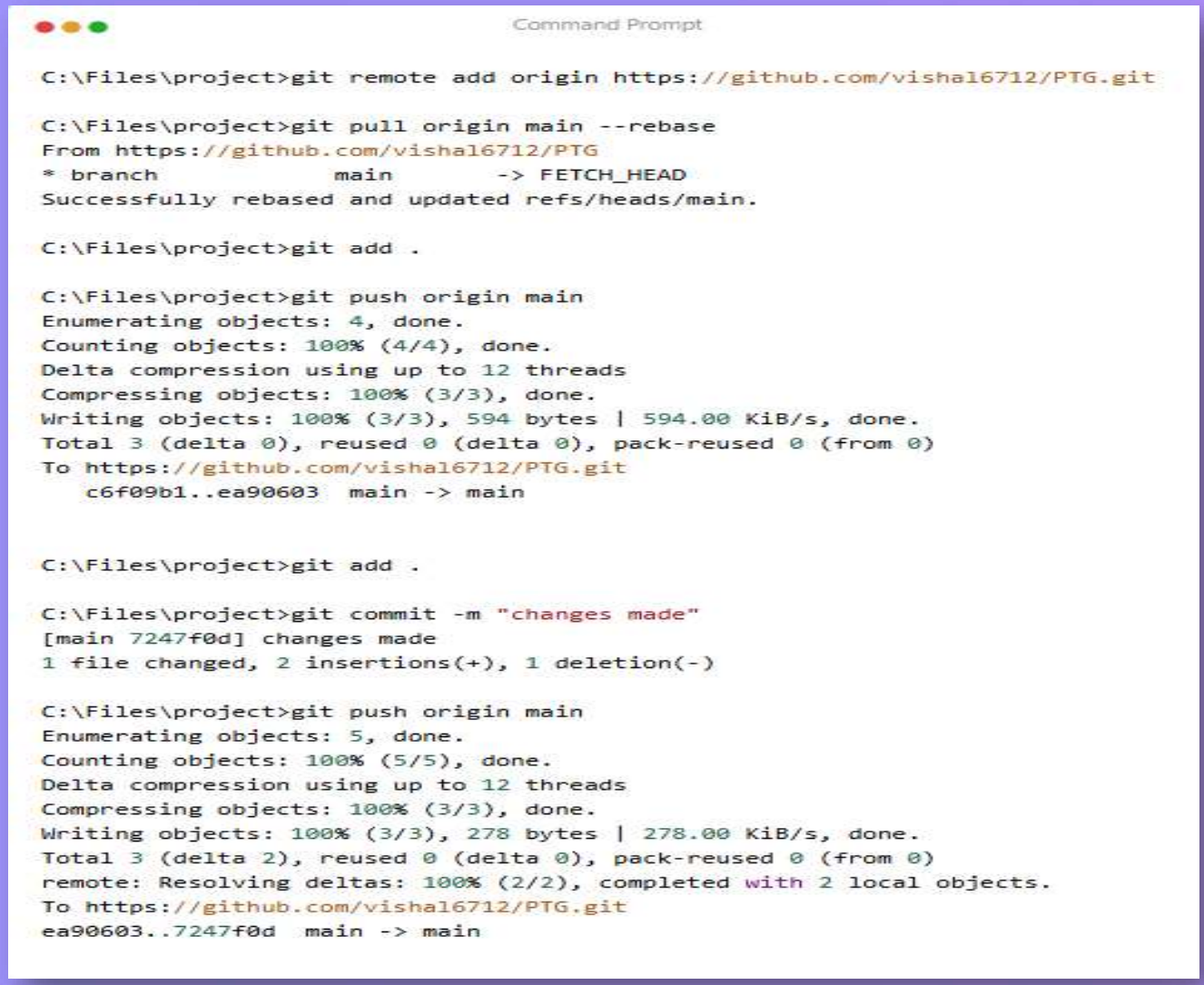


Agile vs Waterfall

Differences	Waterfall	Agile
Structure	A sequential process that follows a linear, step-by-step approach	A iterative and flexible process, with a focus on continuous improvement and adaptation
Project planning	Have a detailed plan with a fixed scope and timeline	Have a flexible plan that is continuously adapted based on feedback and changing requirements
Testing	After development is complete	During each development sprint
Documentation	Require a high level of documentation	Focus more on face-to-face communication and collaboration
Project type	Best suited for projects with well-defined requirements	Best suited for projects with rapidly changing requirements

GIT COMMANDS USED


Command Prompt
C:\Files\project>git init
Initialized empty Git repository in C:/Files/project/.git/
C:\Files\project>git status
On branch master No commits yet nothing to commit (create/copy files and use "git add" to track)
C:\Files\project>git add exception_handling.txt
C:\Files\project>git add .
C:\Files\project>git config user.name "vishal"
C:\Files\project>git config user.email kadamv6712@gmail.com
C:\Files\project>git add .
C:\Files\project>git commit -m "new"
[master (root-commit) 0926d75] new
1 file changed, 44 insertions(+)
create mode 100644 exception_handling_ptg.txt
C:\Files\project>git checkout -b main
Switched to a new branch 'main'
C:\Files\project>git branch
* main
master


Command Prompt
C:\Files\project>git remote add origin https://github.com/vishal6712/PTG.git
C:\Files\project>git pull origin main --rebase
From https://github.com/vishal6712/PTG
* branch main -> FETCH_HEAD
Successfully rebased and updated refs/heads/main.
C:\Files\project>git add .
C:\Files\project>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 594 bytes | 594.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/vishal6712/PTG.git
c6f09b1..ea90603 main -> main
C:\Files\project>git add .
C:\Files\project>git commit -m "changes made"
[main 7247f0d] changes made
1 file changed, 2 insertions(+), 1 deletion(-)
C:\Files\project>git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 278 bytes | 278.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/vishal6712/PTG.git
ea90603..7247f0d main -> main

https://github.com/vishal6712/PTG