1.Contrast "waterfall" development model with "agile". What applications are best suited for continuous development? What applications are less suited for it and why?

Waterfall development model is usually a linear approach to software development, like it follows a phase by phase and you cannot move forward without completing the current phase completely, any changes to the scope has to be considered as change control after the main project delivery. It took a lot of time to deliver each project, because product defects are identified very late in the stage.

Agile is an iterative approach to software development. Deliver the projects frequently and iteratively so business can get the customer feedback sooner to improve the products.

Projects that require frequent changes better opt for Agile for the reasons mentioned above. The projects that require less changes/ the requirements are pretty clear and straightforward or if the application is in decommission state and not worthy of an investment to implement a process that can support agile are best suited for waterfall.

2.How would you convert a legacy product with periodic releases to use continuous development?

These are some activities that have to happen to convert a legacy product to CI/CD.

As we convert or upskill the team to adopt Agile methodology, we have to implement devops practices to ensure the product can be shipped iteratively as expected in Agile. Ensure the team is using a tech stack that has community support for devops. Have the code in a common version control repository for collaboration. Automate the build/test and deploy trigger as soon as code is either pushed to github or merged into the branch via pull request.

3.What are some prometheus metrics you could imagine that would indicate the health of a code base? In terms of its susceptibility to bitrot, the quality of its evolution, etc.

Bit rot happens to all software when the dependencies and tooling required to build, test, and deploy it change over time. This can lead to all types of issues in the application such as latency, broken CI/CD, increase in memory usage. CPU and memory usage are some of the metrics that we can use for it.

Reference -> <https://blog.tidelift.com/bit-rot-the-silent-killer>

4.What are the major stages and jobs of a pipeline in a continuous integration process?

Test, Build, Install and Deploy are some major stages in continuous integration process

5. You have a new feature you want to introduce, but do not want everyone to use it all at once. Describe ways you could introduce it. How can you minimize risk?

We can do canary deployment which is supported by Istio service mesh. This can handle the configuration of splitting the traffic between the current and the new release. There is no risk as if there is any issue with the new feature, the users can be routed back to current version immediately.

6. The reading's examples are from big tech companies. Discuss how they are applicable to smaller companies. When and why?

These are applicable to the companies that are in small/ early stages where they can introduce this process to adopt immediately. This will also give flexibility in iterative delivery, being competitive in business, etc..

7. Scrum can be a lightweight process where developers merely inform each other on a regular basis what they are doing, or a heavyweight process where program managers monitor the velocity by which developers achieve goals. Which implementation is better suited to your workplace? Why?

The 2nd option is best suited for my workplace as it is a big company and it has established the agile process for everyone to follow so they can pull data for compliance. Team has to be managed by a scrum master to make sure team velocity is good. The team is also a little big so it needs some kind of coordination between all the stakeholders.

8. Choose 4 items from table 1 In the "Adages" paper and discuss how we utilized them in our homework assignments this semester. Describe how these items help developers deliver software faster to customers.

| Item | Description |
| --- | --- |
| Continuous deployment | We used a CD with the calculator program to deploy in kubernetes and tested it on various cases. It helps to automate testing |
| Branching or branch deployments | We mostly worked on the main branch however got an opportunity to work on other branches. It helps to maintain the code base based on version. |
| Change ownership | As a developer we implemented the steps for software changes for all phases, including development, testing, deployment, and fixing problems. |
| Deployment pipeline | We used jenkins pipeline to deploy our changes in server ( was used aws). This will help automate the deployment process. |