1. Agility: Fast Service Delivery (Frequent Small Improvements)
   1. Test Automation
      1. Test Case Writing (Even One liner)
      2. Test Script Recording
   2. Deployment Automation (CI/CD pipeline implementation)
      1. Utilize multiple test/staging environments for automated test-execution
      2. Target doable/releasable small improvements for new build every week (if every day is not possible)
      3. Fail Fast: If any functionality in system can fail, it’s better that it get fail as early as possible during development or at-least before production
      4. Execute Unit Tests on each code check-in
      5. Execute Automated UI Tests on each build (if each check-in is not possible)
   3. Microservices: separation of each module as a separate application
      1. Break monolithic application into smallest manageable applications
      2. It will allow Frequent Small Improvements
      3. Quick test would be possible, in case of failure individual service/app would be identifiable
      4. It will allow conversion (into .net core) of each module separately
2. Cost Optimization:
   1. Adopt Serverless: There’s huge difference in pricing either we use AWS or Azure
   2. Move to Linux Containers (.net core can run on Linux): It’s another cost saving parameter and will open doors for us towards Serverless.
   3. Consider opensource: Windows and SQL Server license fees are very high. There are many alternatives available. For example, [Babelfish](https://babelfishpg.org/) (PostgreSQL’s driver for SQL Server compatibility). Blob: [AWS Aurora (PostgreSQL)](https://aws.amazon.com/blogs/aws/goodbye-microsoft-sql-server-hello-babelfish/)