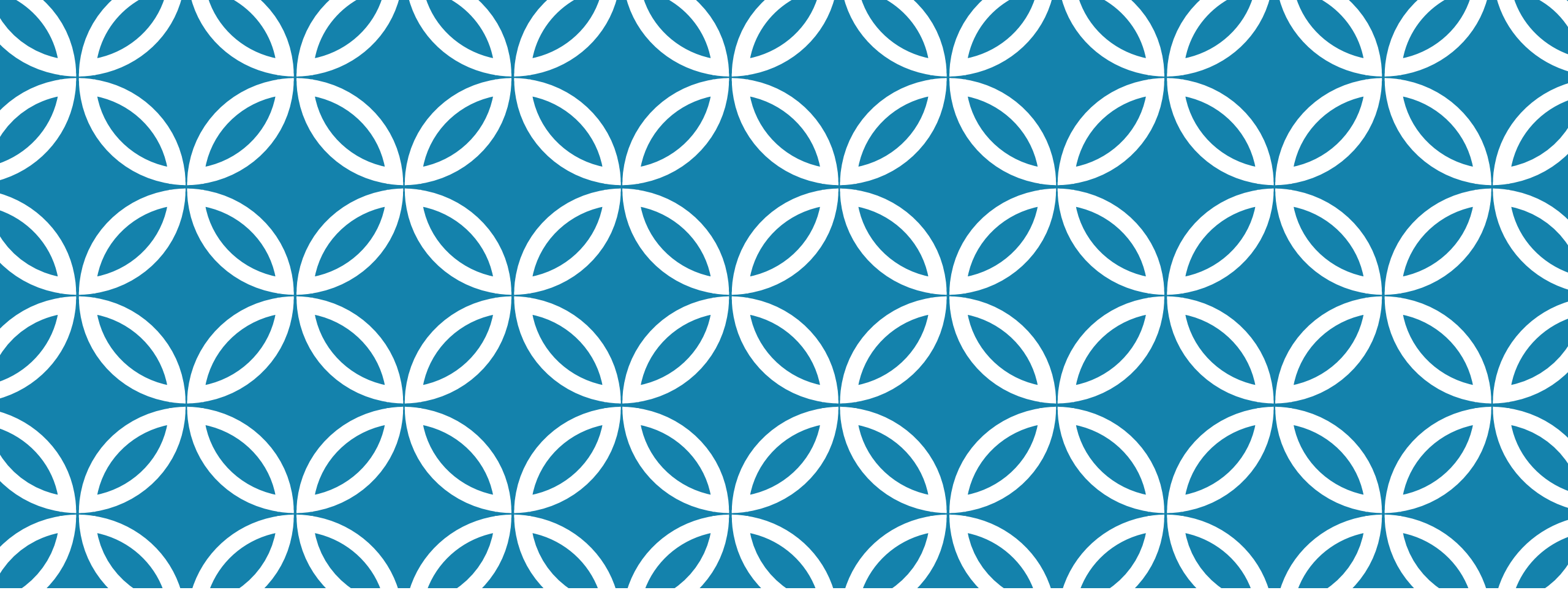


UDAPEOPLE – CONTINUOUS DELIVERY

Continuous Integration + Continuous Deployment = Continuous Delivery

UDAPEOPLE – THE REVOLUTION WITH CHALLENGES

- UdaPeople product, a revolutionary concept in Human Resources which promises to help small businesses care better for their most valuable resource: their people.
- Developing and releasing Udapeople software will be a complicated process, especially as applications, teams, and deployment infrastructure grow in complexity.
- Releasing to production will be easier in initial phase as there would not be any customers however bringing down the product during customer use would be difficult.
- Development and release product in Big Bang is very risky proposition.
- Repetitive development is prone to Human Errors.

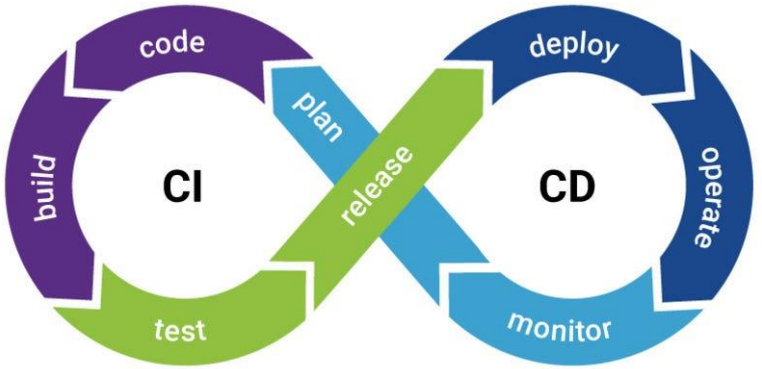
SOLUTION -

Introduction of Continuous Delivery – An engineering practice in which teams produce and release value in short cycles.

CONTINUOUS DELIVERY – A PARADIGM

Introduction of Agile methodologies and adoption of Devops practices. – A step towards modern trend of continunity.
Continuous Delivery = Continuous Integration (CI) + Continuous Deployment (CD)

- Continuous Delivery is an overarching paradigm or mindset that informs and enhances the practices of Continuous Integration and Continuous Delivery.
- Continuous Integration - The practice of merging all developers' working copies to a shared mainline several times a day.
- Continuous Deployment - A software engineering approach in which the value is delivered frequently through automated deployments.



Continuous Delivery is a True North
- Production will be available for users
- Features will be built correctly

HOW TO ADOPT CONTINUOUS DELIVERY

STEPS

- Collaborative, comprehensive grooming of features that include team and stakeholders
- Ruthless slicing of features to smallest valuable increments
- Build team-wide, deep understanding of each feature's requirements and characteristics before coding starts
- Write comprehensive automated unit tests in front-end and back-end layers
- Shoot for high coverage from automated back-end integration tests
- Shoot for high feature critical-path coverage from end-to-end UI tests
- Include automated smoke tests that can be run on production-candidates
- Ensure all post-commit tasks and hand-offs must be automated in CI/CD
- Strive for quick, reliable rollback if smoke tests fail

TOOLS

Cloud based Tools for High Availability –

- Code Check IN Tool – GIT HUB 
- Code Editor – Visual Studio 
- Pipelines – Circle CI 
- Cloud Deployment – Amazon AWS 

STAKEHOLDERS LENS

Reduce Cost

Avoid Cost

Reduce Cost	Avoid Cost	Increase Revenue	Protect Revenue
• Less developer time on issues from new developer code • Less infrastructure costs from unused resources	• Less bugs in production and less time in testing • Prevent embarrassing or costly security holes • Less human error, Faster deployments	• New value-generating features released more quickly • Less time to market	• Reduced downtime from a deploy-related crash or major bug • Quick undo to return production to working state

Increase Revenue

Protect Revenue

CONCLUSION

Lets Give Flying colours to Udapeople by inculcating CI/CD culture and Make the difference

THANKS