Planning for DevOps

Eng Teong Cheah

Contents

Linear Regression Implementation from Scratch

DevOps

DevOps Role

DevOps professionals combine people, process, and technologies to continuously deliver valuable products and services that meet end user needs and business objectives.

DevOps professionals streamline delivery by optimization practices, improving communications and collaboration, and creating automation. They design and implement strategies for application code and infrastructure that allow for continuous integration, continuous testing, continuous delivery, and continuous monitoring and feedback.

Azure DevOps professionals must be able to design and implement DevOps practices for version control, compliance, infrastructure as code, configuration management, build, release, and testing by using Azure technologies.

Transformation Planning

Separating Transformation Teams

There are several challenges when creating teams:
1. Availability of staff
2. Disruption of current procedures and procedures

To overcome the challenges, create a team that is:
1. Focused on the transformation
2. Well respected in their subject areas
3. Internal and external to the business

A transformation project can conflict with on-going business needs.

Defining Shared Goals

Project must have a clearly-defined set of measurable outcomes, like:

1. Reduce the time spent on fixing bug by 60%
2. Reduce the time spent on unplanned work by 70%
3. Reduce the out-of-hours work required by staff to no more than 10% of total working time.
4. Remove all direct patching of production systems.

One of the key aims of DevOps is to provide greater customer value, so outcomes should have a customer value focus.

Setting Timelines for Goals

Measurable goals should have timelines that challenging yet achievable

Timelines should be a constant series of short-term goal – each clear and measurable

Shorter timelines have advantages:
1. Easier to change plans or priorities when necessary
2. Reduce delay between doing the work and getting feedback
3. Easier to keep organizational support when positive outcomes are apparent.

Project Selection

Greenfield and Brownfield Projects Defined

Greenfield software projects develop in a totally new environment

Brownfield software project develop in the immediate presence of existing software applications/systems

Choosing Greenfield and Brownfield Projects

Greenfield projects
1. Appears to be an easier starting point
2. A blank slate offers the chance to implement everything the way you want

Brownfield projects
1. Comes with the baggage of existing code bases, existing teams, and often a great amount of technical debt
2. Spending time maintaining existing Brownfield applications, limits the ability to work on new code.

There is a common misconception that DevOps suits Greenfield projects better than Brownfield projects, but this is not the case.

Choosing Systems of Record vs Systems of Engagement

Systems of Record
1. Emphasize accuracy and security
2. Provides the truth about data elements
3. Historically evolves slowly and carefully

Systems of Engagement
1. More exploratory
2. Uses experimentation to solve new problems
3. Modified regularly
4. Making changes quickly is prioritized over ensuring that the changes are correct.

Both types of systems are important.

Selecting Group to Minimize Initial Resistance

Different types of staff members

1. **Canaries** voluntarily test bleeding edge features
2. **Early adopters** voluntarily preview releases
3. **Users** consume the products after canaries and early adopters

Ideal DevOps team members

1. They think there is a need to change and have shown an ability to innovate
2. They are well-respected and have broad knowledge of the organization and how it operates
3. Ideally, they already believe that DevOps practices are what is needed

Ideal Target Improvements

1. Can be used to gain early wins
2. Is small enough to be achievable in a reasonable time-frame
3. Has benefits that are significant enough to be obvious to the organization

Identifying Project Metrics and KPIs

Faster Outcomes
Deployment frequency, deployment speed, deployment size and lead time

Efficiency
Server to admin ration, Staff member to customers ratio, application usage, and application performance

Quality and Security
Deployment Failure Rates, Application Failure Rates, Mean Time to Recover, Bug Report Rates, Test Pass Rates, Defect Escape Rate, Availability, SLA Achievement, and Mean Time to Detection

Culture
Employee morale and retention rates

Goals must be specific, measurable m and time-bound

Team Structures

Agile Development Practices Defined

Waterfall approach

1. Define, analyze, build and test, and deliver
2. Hard to accurately define requirements, which can change over time
3. After delivery requires change requests and additional cost

Agile approach

1. Emphasizes constantly adaptive planning, and early delivery with continual improvement
2. Development methods are based on releases and iterations
3. At the end of each iteration, there should be testing working code
4. Focused on these short-term outcomes.

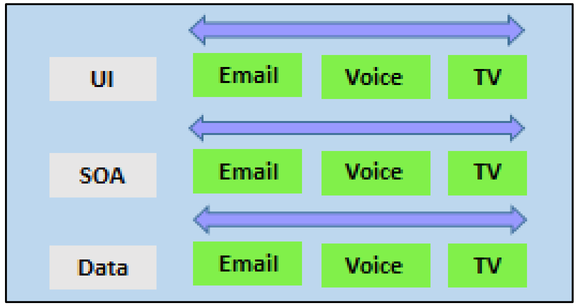
Vertical teams have been shown to provide stronger outcomes in Agile projects

Principles of Agile Development

1. Satisfy the customer
2. Welcome changing requirements
3. Deliver working software frequently
4. Work together throughout the project
5. Build projects around motivated individuals
6. Use face-to-face conversation
7. Measure progress through working software
8. Agile process promote sustainable development
9. Continuous attention to technical excellence and good design
10. Simplicity– the art of maximizing the amount of work not done
11. Use self-organizing teams
12. Reflect on how to become more effective

Creating Organizational Structures for Agile Practices

Horizontal Team structures divide teams according to the software architecture

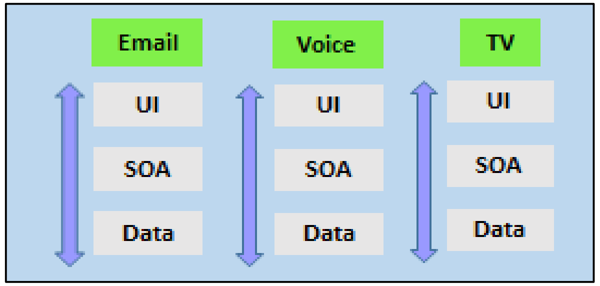


UI Email Voice TV

SOA Email Voice TV

Data Email Voice TV

Vertical Teams span the architecture and are aligned with product outcomes and scaling can occur by adding teams



Email Voice TV

UI UI UI

SOA SOA SOA

Data Data Data

Vertical team have been shown to provide stronger outcomes in Agile projects

Mentoring Team Members on Agile Practices

1. Many teams hire external Agile coaches or mentors
2. Agile coaches have teaching and mentoring skills
3. Agile coaches ten tend to be both trainers and consultants
4. Some coaches are technical experts
5. Some coachers are focused on Agile processes

Team members must learn as they work, and acquire skills from each other

Enabling In-Team and Cross-Team Collaboration

Cultural changes

more open work spaces, meeting etiquette, outsourcing, better communication

Cross-functional teams

collaboration with others, diversity of opinion, rewarding collective behavior

Collaboration tooling

Skype, Slack, Teams, Google Hangouts

Selecting Tools and Process for Agile Practices

Tools can often enhance the outcomes achieved

Physical tools such as white boards, index cards, sticky notes

Project Management tools for planning, monitoring, and visualization

Screen Recording tools for recording bugs, building walk-throughs, and demonstrations.

Thanks!

Twitter: @walkercet
Blog: https://ceteongvanness.wordpress.com