PRADITA University

MTI104 - IT Services

Session-12:

Practices to Manage Releases

PRU/SPMI/FR-BM-18/0222

Alfa Yohannis

Introduction to the Cycle

PRADITA University

- Cycle starts with design, followed by build, test, and transition.
- Transition involves moving a built object into production.
- Streamlined process needed for moving packages to higher environments.
- Policies, principles, and processes guide releases.
- Technical aspect of moving packages to minimize risks.
- Releases in the waterfall world happened infrequently.
- Process of development and testing was the focus.

Changes in DevOps Industry

PRADITA University

- In DevOps, releases and deployments happen frequently.
- Automation is key to seamless activities.
- Speed is crucial; blockers are to be eliminated.
- Release management and deployment management are critical practices.
- ITIL 4 emphasizes separate practices for release and deployment.
- Depth of knowledge is essential despite examination requirements.
- Exam tip: Expect questions on release and deployment practices.

Release Management Overview

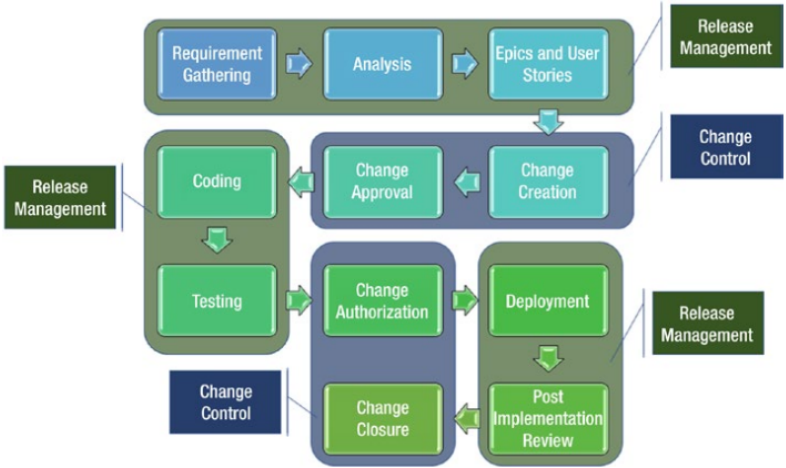
PRADITA University

- Release management spans development and operations.
- ITIL defines a release as a version of a service or configuration item.
- Developed in lower environments and tested in higher environments.
- Testing can be manual or automated.
- Software moved to production as a release.
- Releases may include software, break-fixes, and hardware changes.
- Releases are often packaged to minimize production disruptions.

Change Control vs. Release Management

- Change control: obtaining approvals and controlling changes.
- Release management: managing technical aspects of changes.
- Release management includes requirement gathering, coding, testing, deployment.
- Change control ensures development efforts aren't wasted.
- Change control also modifies solutions if needed.
- Release management handles overall planning and execution.
- Change control provides oversight before production changes.

Change control vs. release management



Release Fundamentals

PRADITA University

- Releases include hardware and software components.
- Training and documentation updates are part of the release.
- Third-party components may be included in releases.
- Release scope is broad, covering all service components.
- Major releases involve significant business impact.
- Minor releases are frequent and less risky.
- Emergency releases are used to fix critical incidents.

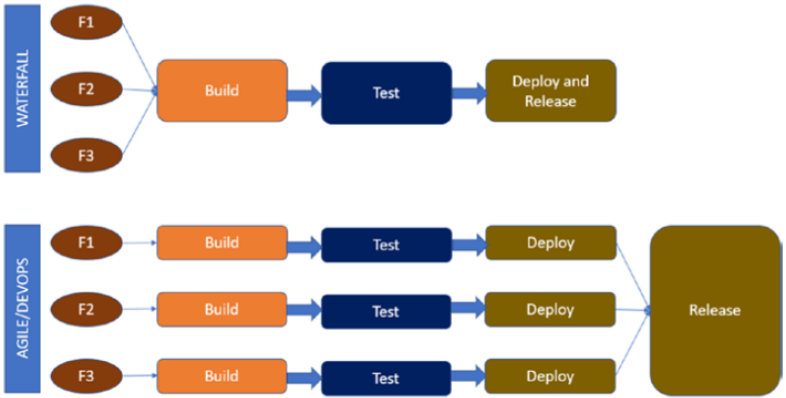
Release Scheduling and Reviews

PRADITA University

- Releases can be scheduled or ad hoc.
- Emergency releases address critical issues.
- Periodic releases provide predictability for businesses.
- DevOps promotes modular, frequent releases.
- Release schedule aligns with change schedule.
- Release reviews identify lessons learned and improve future releases.
- Waterfall vs. Agile/DevOps approaches differ in release management.

Waterfall vs. Agile/DevOps releases

PRADITA University



Release Management in Agile/DevOps

- Agile/DevOps: release management becomes iterative.
- Release planning aligns with Agile Release Trains (ARTs).
- Continuous deployment adapts release management to frequent deployments.
- Continuous delivery combines iteration with traditional deployment.
- Release management ensures stable paths to production.
- Release management in continuous delivery provides control.
- Maturity leads to a preference for continuous deployment.

Release Management Techniques

PRADITA University

- Techniques mitigate risks in stable environments.
- Proof of Concept (POC) tests the feasibility of major changes.
- POC ensures development and testing paths are viable.
- Pilot phase follows POC, testing functionality with real data.
- Pilot releases involve limited, controlled testing.
- Successful pilot further validates the chosen path.

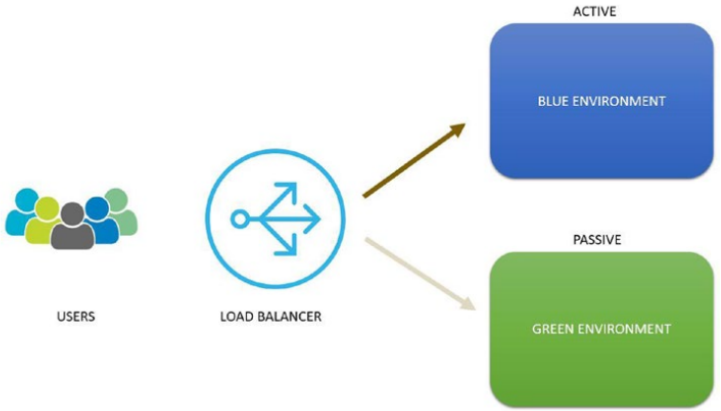
Blue–Green Release Overview

PRADITA University

- DevOps projects deploy without downtime
- Blue-green deployment runs two environments in parallel
- Environments designated as blue or green
- One environment is active, the other passive
- Active environment handles all user requests
- Passive environment deployed first
- Load balancer switches user requests to the updated environment

Blue-green release approach

PRADITA University



Blue–Green Release Example

PRADITA University

- Example: Blue is active, green is passive
- Green environment deployed without affecting users
- Once green is ready, switch user requests from blue to green
- Blue environment is updated next
- Both environments receive updates requiring downtime
- Users experience no downtime
- Load balancer manages user requests between environments

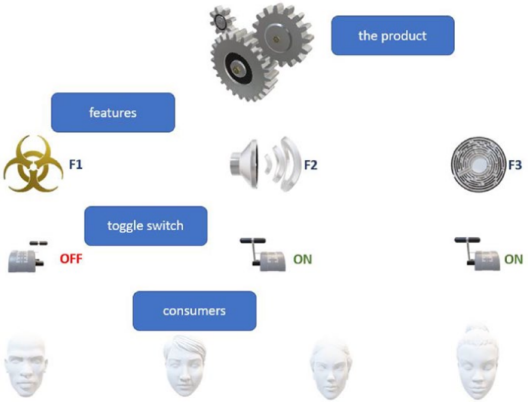
Feature Toggles Overview

PRADITA University

- Software development technique to change features
- No code changes needed
- Branching technique used for feature development
- Features toggled on/off based on criteria
- Example: Features F1, F2, and F3 in a shopping portal
- Only F2 and F3 are live, F1 is toggled off

Feature toggles illustration

PRADITA University



Feature Toggles Use Cases

PRADITA University

- Activate F1 during Christmas shopping for promotions
- Feature changes introduced without code changes
- Minimizes risks from system changes
- Feature toggles prevent downtime and loss of revenue
- Different features can be toggled for different regions
- Example: UK gets F1 and F2, US gets F2 and F3
- India receives all features (F1, F2, and F3)

Engagement with Service Value Chain

ADITA iversity

- Release management involvement across SVC activities
- Medium involvement in planning, building, and improving
- High involvement in design and transition
- Low involvement in engagement and support
- Release plans define schedule, types, and techniques
- Coordination with Design and Transition activity
- Building and testing in Obtain/Build activity

Deployment Management Overview

PRADITA University

- Deployment management in ITIL practices
- Deployment: moving software, hardware, or processes to live environments
- Not limited to production environments
- Includes moving packages to testing and staging environments
- Infrastructure deployments included in ITIL 4 scope
- Infrastructure as Code (IaC) for cloud infrastructure
- Deployments are critical for service delivery

Deployment Approaches Overview

PRADITA University

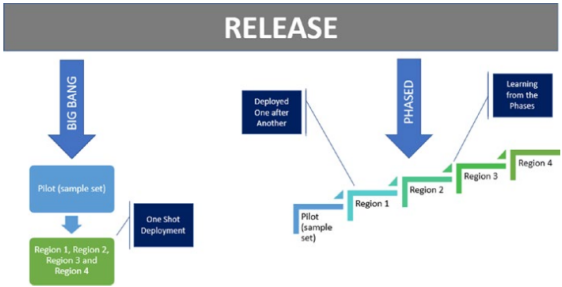
- Multiple deployment approaches based on context
- Big bang deployment: all users experience the change simultaneously
- Phased deployment: gradual release across regions or features
- Continuous delivery/deployment: automated deployment
- Pull deployment: users choose when to deploy updates
- Each approach has pros and cons
- Choice depends on the organization’s needs and risks

Big Bang and Phased Deployment

PRADITA University

- Big bang: all users get updates at the same time
- Pilot deployment precedes big bang to ensure readiness
- High risk due to simultaneous update
- Phased deployment: updates rolled out over time
- Allows for learning and correction before full deployment
- Geographical or feature-based phased deployment
- Phased deployment mitigates risks and manages user impact

deployment approaches



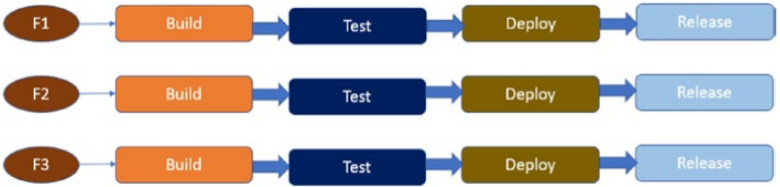
Continuous Delivery/Deployment

PRADITA University

- Continuous delivery: manual trigger for production deployment
- Continuous deployment: automated deployment to production
- Simplicity in no waiting for other parameters
- Risk of exposing flaws to end users quickly
- Automation plays a key role in deployment pipelines
- Requires organizational maturity in DevOps practices
- Ensures quick delivery of features to users

Continuous delivery/deployment

PRADITA University



Pull Deployment Overview

PRADITA University

- Pull deployment: users choose when to install updates
- Software packages available in a repository (DML)
- Users notified of available updates
- User-triggered installation at their convenience
- Effective for minimizing disruption to user productivity
- Less common compared to push deployment approaches
- Not suitable for urgent security updates

Multiple Choice Question

PRADITA University

Which of the following is not a valid type of deployment approach?

A. Phased deployment
B. Continuous deployment
C. Continuous delivery
D. Emergency deployment