Week 9: ITIL Management Practice – Change

A. Learning Objectives

By the end of this lecture, you will be able to:

1. **Define Change Enablement** and explain its role in minimizing risks when modifying IT services or components.
2. **Differentiate Standard, Normal, and Emergency Changes**, including their approval paths and risk profiles.
3. **Describe the Change Advisory Board (CAB)**, explain change models, and outline evaluation criteria used for decision-making.
4. **Define Release Management**, its lifecycle stages, and how it works hand-in-hand with Change Enablement.
5. **Explain automation and tool-driven orchestration** (e.g., CI/CD pipelines, release pipelines) and how they support fast, reliable service deployments.

B. Detailed Content Discussion

---

1. Change Enablement

Purpose

Change Enablement ensures that modifications to infrastructure, applications, processes, or documentation are performed in a **controlled, risk-aware manner**, protecting service stability and business operations.

Changes may involve:

- Hardware upgrades
- Software patches or feature deployments
- Configuration modifications
- Process/documentation updates

The goal is **not to prevent change**, but to **manage risk, increase success rates, and ensure business alignment**.

Types of Changes

1. Standard Change

- Pre-authorized and low risk.
- Well-documented, frequently occurring, with proven success history.
- Example:
    - Routine patch installation
    - Password policy update
    - Predefined server restart maintenance

2. Normal Change

- Requires **risk assessment**, approval, and scheduling.

- May require CAB review depending on impact.
- Can be categorized as **low-risk normal**, **medium-risk normal**, or **high-risk normal**.
- Examples:
    - Database configuration change
    - Deployment of a new feature requiring downtime

3. Emergency Change

- Implemented urgently to resolve critical incidents or prevent major disruptions.
- Requires expedited approval — sometimes via **Emergency CAB (eCAB)**.
- Must undergo **Post-Implementation Review (PIR)** to verify necessity and prevent unnecessary emergency use.
- Examples:
    - Security patch for active vulnerability exploitation
    - Reconfiguring a firewall blocking production traffic

Key Activities in Change Enablement

1. Record & Categorize

- Logging the Request for Change (RFC)
- Documenting risk levels, business impact, proposed back-out plans
- Classifying type (Standard / Normal / Emergency)

2. Assess & Authorize

- Use **Change Models** for recurring, predictable changes
- Review by Change Manager or CAB for non-standard changes
- Evaluate risk, resources, business impact, and scheduling constraints

3. Plan & Schedule

- Collaborate with stakeholders
- Identify dependency conflicts
- Publish and maintain the organizational **Change Schedule**
- Ensure alignment with maintenance windows

4. Implement & Review

- Execute the approved change
- Log results and deviations
- Conduct **Post-Implementation Review (PIR)**
- Update documentation and knowledge bases
- Improve future Change Models using insights from PIRs

Change Advisory Board (CAB) and Evaluation Criteria

Purpose of CAB

- Provide governance and structured decision-making
- Assess high-impact or high-risk changes
- Support prioritization and business alignment

Evaluation Criteria Used by CAB

- Risk and business impact
- Readiness of back-out/rollback plan

- Technical feasibility
- Resource availability (people, tools, downtime window)
- Alignment with business schedules and releases
- Compliance with policies and security controls

CAB also reviews:

- PIR reports
- Improvements to change policies
- Trends in failed or emergency changes

---

2. Release Management

Purpose

Release Management is responsible for **planning, scheduling, building, testing, and deploying releases** to production with minimal disruption.

A **release** is a bundle of authorized changes deployed together to achieve a specific business outcome.

Examples of releases:

- Major application upgrades
- New features deployed via sprint cycles
- Infrastructure rollout or network upgrades

Stages of the Release Management Lifecycle

1. Release Planning

- Define the release scope
- Identify included RFCs (linked to Change Enablement)
- Assemble the **Release Package**, including:
    - Build artifacts
    - Deployment scripts
    - Version documentation
    - Configuration updates
    - Test results

2. Build & Test

- Use automated build pipelines and testing frameworks
- Types of testing:
    - Unit Testing
    - Integration Testing
    - Functional Testing
    - User Acceptance Testing (UAT)
- Validate compatibility with existing services
- Ensure the release meets quality standards before deployment

3. Deployment

- Coordinate cross-team activities (DevOps, Network, DBAs, Service Desk)
- Execute deployment scripts, automation playbooks, or CI/CD pipeline actions
- Manage communication to stakeholders (downtime, success notifications)
- Monitor initial performance after release

4. Review & Close

- Verify release success against KPIs and service acceptance criteria
- Update CMDB entries with new versions and relationships
- Conduct a **release retrospective** for continuous improvement
- Document findings and close the release record

Integration with Change Enablement

- Each release contains **multiple RFCs**, all of which must follow the Change Enablement workflow.
- Change Enablement ensures controlled approval; Release Management ensures synchronized, safe deployment.
- Automation tools such as:
  - CI/CD pipelines (Jenkins, GitLab CI, GitHub Actions)
  - Configuration management (Ansible, Puppet, Chef)
  - Infrastructure-as-Code (Terraform, CloudFormation)
  - Release orchestration platforms

These tools apply the ITIL guiding principle:
"Optimize and Automate."

They eliminate manual steps, reduce deployment failures, and accelerate delivery.

C. Discussion Questions

1. How do predefined Change Models reduce risk and accelerate routine changes?

- Students should explain that Change Models provide **repeatable, tested workflows** that minimize uncertainty, reduce approval cycles, and ensure consistent outcomes.

2. In what scenarios is an Emergency Change justified, and how can an organization prevent abuse?

- Appropriate when:
  - A critical incident is disrupting business
  - A vulnerability requires immediate patching
  - Service restoration cannot wait for a normal CAB cycle
- Abuse prevention measures:
  - PIR reviews
  - Tracking emergency change frequency
  - Strict criteria and Change Manager oversight

3. Describe how automated CI/CD pipelines support the Release Management lifecycle and align with the "Optimize and Automate" guiding principle.

- Expected answers:
  - Automation reduces human error
  - Pipelines improve speed and reliability
  - Supports continuous integration/testing
  - Enables rapid, repeatable deployments
  - Frees staff from manual repetitive tasks, letting them focus on higher-value work