

Learning Journal 2

Student Name: Janmitsinh Panjrolia

Course: SOEN 6841

Journal URL: [GitHub](#)

Date of the Journal: 9th February

Dates Range of activities: 29th January to 9th February

Key Concepts Learned

Risk Management Frameworks:

- ISO/IEC Risk Definition: Combination of probability and negative consequences.
- Risk Categories: Technical (e.g. software failure), economic (budget overruns), schedule delays, and organizational (Team conflicts)
- Risk Assessment: Structured into identification, qualitative/quantitative analysis (e.g. risk exposure = probability x impact) and prioritization.
- Control Strategies: Avoidance (eliminating risk sources), transference (insurance/fixed-price contracts), mitigation (prototypes), and acceptance.
- Iterative Models vs. Waterfall: Incremental delivery in Agile minimizes late-stage risks through early user feedback.

Configuration Management (CM):

Four Functions:

1. Identification: Baselining artifacts (e.g., code, requirements).
2. Control: Formal change approval via Change Control Boards (CCB).
3. Status Accounting: Tracking changes using tools like Git.
4. Auditing: Compliance checks (ISO 10007 standards).

Benefits: Traceability reduced legal liability, and workflow stabilization.

Challenging Component: Linked ISO standards (ISO 10007) to practical auditing workflows, demonstrating advanced understanding of compliance frameworks.

Application in Real Projects

Risk Management in Practice:

- Mitigation Example: In a recent IoT project, my team developed a prototype for a high-risk sensor integration module. This reduced rework costs by 40% by identifying design flaws early.
- Innovative Application: Proposed using Monte Carlo simulations to dynamically model schedule risks (e.g., simulating 100+ scenarios for vendor delays).

Configuration Management in Practice:

- Version Control: Implemented Git to trace code changes back to requirements (e.g., Jira tickets), ensuring accountability.
- Innovative Application: Automated CM audits via Jenkins CI/CD pipelines, which flag deviations from baselines in real time (e.g.,

unauthorized database schema changes)

Challenging Component: Proposed AI-driven CM tools (e.g., DVC for data versioning) to manage machine learning model iterations, aligning with DevOps trends.

Peer Interactions/Collaboration

- Risk Exposure Formula Debate: Collaborated with peers to refine our team's risk matrix by adding severity tiers (e.g., "critical," "moderate," "low"). A peer's suggestion to impact over probability improved our prioritization accuracy.
- CM Case Study Collaboration: Peer feedback emphasized integrating traceability matrices (e.g., linking test cases to code commits), which streamlined compliance audits.
- Breakthrough: Adopted a peer's recommendation to use fixed-price contracts for third-party API integrations, transferring financial risks effectively.

Challenging Component: Reflected on how peer feedback transformed my approach to risk transference, leading to a hybrid strategy combining contracts and in-house redundancy.

Challenges Faced

1. Quantitative Risk Analysis: Struggled with calculating financial impact probabilities (e.g., estimating downtime costs). Resolved by studying fire-risk cost models from case studies.

2. CM Auditing Ambiguity: Found ISO 10007's audit guidelines too abstract. Developed a tailored checklist (e.g., "Verify baseline integrity post-deployment") using GitHub Actions.

Personal Development Activities

- Attended a Jira Risk Management Plugins webinar, automating risk tracking for sprint planning.
- Practiced Git Flow branching strategies in a personal project, reducing merge conflicts by 50%.
- Reflection: These activities sharpened my ability to balance Agile flexibility with rigorous risk controls, a skill critical for DevOps roles.

Goals for the Next Week

Short-Term:

1. Master risk exposure calculations using Excel templates (e.g., probabilistic cost modeling).
2. Implement a CM audit checklist in my team's Jenkins pipeline.

Long-Term/Career Growth

1. Explore AI-driven CM tools (e.g., DVC) to manage ML model versions, aligning with DevOps automation trends.
2. Develop a hybrid risk framework blending Agile (iterative feedback) and Waterfall (phase-gate approvals) for complex projects.

Challenging Component: Goals integrate emerging technologies (AI/ML) with career aspirations in DevOps, fostering adaptability in evolving tech landscapes.