

Course: IT Project Management

Lecture 4: Project Integration Management

LECTURER: RAMBU YETTI KALAWAY

Peta Jalan Pertemuan

- Memahami konsep dan pentingnya Project Integration Management.
- Mengetahui proses-proses dalam Project Integration Management menurut PMBOK Guide
- Mampu mengembangkan Project Charter dan Project Management Plan untuk proyek TI.
- Mampu menerapkan Direct and Manage Project Work.
- Memahami teknik Monitoring & Controlling proyek secara integratif.
- Menguasai proses Change Control yang efektif.
- Memahami pentingnya Knowledge Management dan Lessons Learned.
- Mampu menutup proyek (termasuk proyek gagal) dengan baik.

Apa itu Project Integration Management? (Definisi)

- Definisi PMBOK Guide 7: "Kompetensi inti yang menyatukan berbagai elemen manajemen proyek menjadi keseluruhan yang kohesif dan memastikan bahwa proyek disampaikan dengan sukses." (PMI,2021)
- Analoginya adalah "The Conductor of an Orchestra": Seperti konduktor yang memastikan semua bagian orkestra (biola, trompet, drum) bermain harmonis, Project Manager memastikan semua aspek proyek (scope, waktu, biaya, kualitas, risiko, dll) berjalan selaras.
- Fokus: Membuat trade-off yang tepat antara tujuan yang saling bersaing dan mengelola ketergantungan (dependencies) antar bidang pengetahuan.

Mengapa Integrasi Sangat KRITIS di Proyek TI?

- Kompleksitas Teknis: Arsitektur microservices, integrasi API, DevOps, CI/CD. (Kim, *et.al*, 2021; Forsgren, *et.al*, 2018)
- Perubahan Cepat (Volatility): Requirement berubah cepat (misal: karena feedback user atau perubahan teknologi). (Ries, 2021)
- Banyak Pihak Terkait (Stakeholders): Developer, QA, DevOps, Product Owner, End-User, Manajemen. Masing-masing memiliki kepentingan.
- Saling Ketergantungan (Dependencies): Task front-end bergantung pada task back-end, deployment bergantung pada testing.
- Tanpa Integrasi yang Baik: Proyek akan kacau, scope creep, melebihi anggaran, dan tidak deliver value.

Filosofi Baru: Pergeseran dari Processes ke Principles (PMBOK 7)

- PMBOK 6: Berfokus pada 7 proses integration management yang sequential. (PMI, 2017)
- PMBOK 7: Berfokus pada 12 Prinsip dan 8 Domain Kinerja (Performance Domains). (PMI, 2021)
- Integration ada di mana-mana: Integrasi bukan lagi proses terpisah, tetapi merupakan benang merah yang menghubungkan semua prinsip dan domain.
- Domain Kinerja yang Relevan: Project Work, Delivery, Measurement, Uncertainty, dan Teams semuanya membutuhkan integrasi yang kuat.

Tantangan Spesifik di Proyek TI

- Metodologi Hybrid: Menggabungkan Agile (Scrum) dengan pendekatan prediktif (waterfall) untuk bagian tertentu. (Wysocki, 2019)
- Integrasi DevOps & CI/CD: Bagaimana mengintegrasikan pipeline deployment otomatis ke dalam rencana proyek. (Kim, *et.al*, 2021)
- Manajemen Produk vs Proyek: Peran Product Owner dan Project Manager. (Cohn, 2020)
- Teknologi yang Berubah: Mengintegrasikan teknologi baru di tengah jalan proyek.

Konsep Kunci: Tailoring (Penyesuaian)

- Apa itu Tailoring? Menyesuaikan pendekatan, proses, dan template manajemen proyek dengan konteks proyek tertentu. (PMI, 2021; Wysocki, 2019)
- Mengapa Perlu? Tidak ada satu pendekatan yang cocok untuk semua proyek.
- Faktor yang Mempengaruhi Tailoring di Proyek TI:
 - Ukuran dan kompleksitas tim.
 - Tingkat ketidakpastian requirement (menggunakan Agile vs. Predictive).
 - Budaya organisasi dan kematangan tim.
 - Batasan regulasi (contoh: proyek di bidang kesehatan atau finansial).

Peran dan Tanggung Jawab Project Manager dalam Integrasi

- Sebagai Integrator: Tugas utama adalah mengintegrasikan.
- Membuat Keputusan: Berdasarkan informasi dari semua area.
- Memecahkan Konflik: Antara tujuan scope, waktu, dan biaya.
- Melapor dan Berkomunikasi: Menjadi satu titik kontak kebenaran untuk status proyek.
- Memimpin Tim: Memastikan kolaborasi yang efektif.

Develop Project Charter - Memulai dengan "Why" yang Kuat

- Definisi:Project Charter adalah dokumen formal yang mengesahkan proyek dan memberikan wewenang kepada Project Manager. (PMI, 2017)
- Ini adalah "Contract" antara Project Manager dan Sponsorship.
- Tujuan:Memberikan visi, tujuan, dan otoritas untuk memulai proyek.

Komponen Project Charter untuk Proyek TI

- Judul Proyek & Deskripsi Singkat
- Project Manager & Otoritasnya
- Tujuan Bisnis (Business Case): Mengapa proyek ini dilakukan? (Contoh: Meningkatkan retensi user sebesar 15%, mengotomasi proses bisnis untuk menghemat 200 jam kerja/bulan).
- Tujuan Proyek (Project Objectives) - SMART:
 - Specific: Membangun aplikasi mobile e-commerce.
 - Measurable: Dengan fitur checkout, push notification, dan integrasi payment gateway.
 - Achievable: Dengan tim 5 developer dalam 6 bulan.
 - Relevant: Sesuai dengan strategi digital perusahaan.
 - Time-bound: Diluncurkan pada 30 November 2024.

Komponen Project Charter untuk Proyek TI

- Deliverables Utama High-Level:APK/IPA, Dokumentasi API, Database Schema, User Manual.
- Stakeholder Kunci:Sponsor, Product Owner, Head of IT.
- Risiko Tingkat Tinggi:Ketergantungan pada API pihak ketiga, perubahan regulasi data.
- Anggaran Awal (Budget Summary):Rp 500 juta untuk tools, cloud, dan overtime.
- Success Criteria:Aplikasi digunakan oleh 10.000 user aktif dalam 3 bulan pertama, rating >4.5 di App Store.

Input untuk Develop Project Charter

- Business Case:Dokumen yang membenarkan proyek dari segi finansial dan strategis.
- Agreement/SOW (Statement of Work):Jika proyek untuk klien eksternal.
- Enterprise Environmental Factors (EEFs):Budaya organisasi, infrastruktur TI yang ada, regulasi (seperti PDP/UU Pelindungan Data Pribadi).
- Organizational Process Assets (OPAs):Template charter dari proyek sebelumnya, kebijakan perusahaan.

Teknik: Expert Judgment & Facilitation

- Expert Judgment: Berkonsultasi dengan Senior Developer, Architect, Product Manager, Legal. (PMI, 2021)
- Facilitation Techniques (Brainstorming, Workshop): Mengadakan charter workshop dengan semua stakeholder kunci untuk mendapatkan konsensus. (Atlassian, 2023)

Develop Project Management Plan - Peta Menuju Sukses

- Definisi: Dokumen yang menjelaskan bagaimana proyek akan dijalankan, dimonitor, dikendalikan, dan ditutup. (PMI,2017)
- Ini adalah "Sumber Kebenaran" (Single Source of Truth) untuk proyek.
- Bersifat Dinamis: Diperbarui seiring berjalannya proyek (progressive elaboration).

Komponen Project Management Plan (PMP)

- Daftar komponen utama PMP, yang merupakan konsolidasi dari semua rencana subsidiary:
 1. Rencana Pengelolaan Scope (Scope Management Plan)
 2. Rencana Pengelolaan Waktu & Jadwal (Schedule Management Plan)
 3. Rencana Pengelolaan Biaya (Cost Management Plan)
 4. Rencana Pengelolaan Kualitas (Quality Management Plan)
 5. Rencana Pengelolaan Sumber Daya (Resource Management Plan)

Komponen Project Management Plan (PMP)

6. Rencana Komunikasi (Communications Management Plan)
7. Rencana Pengelolaan Risiko (Risk Management Plan)
8. Rencana Pengelolaan Pengadaan (Procurement Management Plan)
9. Rencana Pengelolaan Stakeholder (Stakeholder Engagement Plan)
10. Rencana Pengelolaan Perubahan (Change Management Plan)
11. Rencana Pengelolaan Konfigurasi (Configuration Management Plan) – Khusus TI

Rencana Pengelolaan Perubahan (Change Management Plan)

- Sangat Penting di Proyek TI!
- Menjelaskan: Bagaimana perubahan (misal: perubahan fitur) akan diajukan, dianalisis, disetujui, dan diimplementasikan.
- Mendefinisikan peran Change Control Board (CCB) untuk proyek besar.
- Template formulir perubahan.

Rencana Pengelolaan Konfigurasi (Configuration Management Plan) - Kunci di TI

- Apa itu? Proses untuk mengelola versi dari semua produk proyek (code, dokumentasi, database schema). (Kim, *et.al*, 2021; Kerzner,2022)
- Mencakup:
 - Version Control System (Git):Strategi branching (GitFlow, Trunk-Based Development).
 - Naming Convention: Untuk branch, commit, tag.
 - CI/CD Pipeline: Kapan dan bagaimana automated build, test, dan deploy dijalankan.
 - Database Change Management: Cara mengelola migrasi database (menggunakan tools seperti Flyway, Liquibase).

Contoh: Struktur PMP untuk Proyek Agile-Scrum

- Bagian 1: Pendahuluan & Charter
- Bagian 2: Rencana Pengembangan (Development Plan):
 - Timeline Release & Sprint (misal: 12 sprint @ 2 minggu). (Cohn, 2020; Kerzner, 2022)
 - Definition of Ready (DoR) & Definition of Done (DoD).
 - Ceremony: Perencanaan Sprint, Daily Stand-up, Review, Retrospective.

Contoh: Struktur PMP untuk Proyek Agile-Scrum

- Bagian 3: Rencana Teknis:
 - Arsitektur Teknologi Stack (MERN, LAMP, dll).
 - Configuration Management Plan (Git Strategy, CI/CD).
 - Quality Plan (Automated Testing, Code Review).
- Bagian 4: Rencana Pendukung:
 - Tim & Peran (Scrum Master, PO, Developer).
 - Komunikasi (Slack, Jira, Confluence).
 - Risiko & Mitigasi.

Teknik Pengembangan PMP: Collaboration & Tailoring

- Collaboration: PMP tidak dibuat sendirian. Libatkan tim teknis (developer, architect) untuk membuat rencana yang realistis. (Atlassian, 2023)
- Tailoring: Menyesuaikan proses, template, dan praktik manajemen proyek dengan konteks proyek (ukuran, kompleksitas, metodologi). (Wysocki, 2019)

Komponen Teknis PMP: Rencana Kualitas (Quality Management Plan)

- Strategi Testing: Unit Test, Integration Test, End-to-End Test, Performance Test.
- Proses Code Review: Wajibkah? Tools yang digunakan (GitHub PR, Gerrit).
- Definition of Done (DoD): Kriteria agar sebuah task/user story dianggap selesai. Contoh: Kode ditulis, di-review, di-test, dideploy ke staging environment, dan dokumentasi diperbarui.

Komponen Teknis PMP: Rencana Komunikasi (Communications Management Plan)

- Tabel Rencana Komunikasi:
 - Apa yang dikomunikasikan? (Laporan Bug, Status Sprint)
 - Kepada Siapa? (Tim Dev, Product Owner)
 - Kapan/Frekuensi? (Saat ditemukan, Setiap Hari)
 - Media/Kanal? (Jira, Slack, Email)
 - Siapa yang Bertanggung Jawab? (QA, Project Manager)

Komponen Teknis PMP: Rencana Risiko (Risk Management Plan)

- Identifikasi Risiko Teknis:Contoh: Ketergantungan pada library open source yang tidak ter-maintain, skalabilitas arsitektur, keamanan data.
- Strategi Mitigasi & Kontingensi:Contoh: Lakukan audit keamanan, gunakan load testing, siapkan library alternatif.

Studi Kasus Mini: Membuat Rencana Konfigurasi

- Scenario: Tim 5 developer akan membangun REST API. Buat garis besar Configuration Management Plan.
- Jawaban Contoh:
 - Version Control: GitHub Repository.
 - Branching Strategy: GitFlow (master, develop, feature/x, hotfix).
 - CI/CD: GitHub Actions akan menjalankan unit test dan build Docker image setiap ada push ke `develop`.
 - Database: Semua perubahan schema dilakukan melalui script migrasi yang versioned (menggunakan Flyway).

Baseline: Tolok Ukur Kinerja

- Apa itu Baseline? Versi yang disetujui dari rencana proyek (Scope, Jadwal, Biaya) yang digunakan sebagai tolok ukur untuk perbandingan.
- Fungsi: Untuk mengukur kinerja dan mengontrol perubahan. Perubahan pada baseline harus melalui proses change control.

Direct and Manage Project Work - Eksekusi & Penciptaan Nilai

- Definisi: Memimpin, melaksanakan, dan melakukan pekerjaan yang tercantum dalam PMP. (PMI, 2017)
- Ini adalah "inti" dari eksekusi proyek.
- Fokus pada creating deliverables dan managing stakeholders.

Aktivitas Utama dalam Direct and Manage Project Work

- Melakukan tugas untuk menghasilkan deliverable.
- Memperoleh, mengelola, dan memimpin tim proyek (people management).
- Mengelola proses procurement (jika ada).
- Berkomunikasi dengan stakeholder secara efektif.
- Mengelola risiko yang muncul.
- Mengimplementasikan perubahan yang telah disetujui.

Peran Project Manager sebagai Pemimpin Teknis

- Bukan Micro-manager: Memberikan arahan, bukan memerintah setiap langkah.
- Memfasilitasi Tim: Menghilangkan hambatan (blockers) bagi tim developer. (Cohn,2020; Sutherland, 2019)
- Memastikan Alur Kerja (Workflow): Memastikan proses dari development ke deployment berjalan lancar. (Kim,*et.al*,2021)
- Membuat Keputusan Teknis (dengan tim): Membantu tim dalam membuat trade-off teknis (misal: build vs buy, pilihan teknologi).

Tools & Teknik: Project Management Information System (PMIS)

- Definisi: Sistem informasi yang digunakan untuk mendukung semua aspek proyek. (PMI,2017; Kerzner, 2022)
- Contoh untuk Proyek TI:
 - Task & Issue Tracking: Jira, Azure DevOps, Trello, Asana.
 - Version Control: GitHub, GitLab, Bitbucket.
 - CI/CD: Jenkins, GitLab CI, GitHub Actions, CircleCI.
 - Komunikasi & Kolaborasi: Slack, Microsoft Teams, Confluence.
 - Monitoring: Grafana, DataDog, Prometheus.

Integrasi dengan Siklus Hidup Pengembangan (Development Lifecycle)

- Waterfall: Direct and Manage Work berjalan fase demi fase.
- Agile (Scrum): Proses ini terjadi dalam setiap Sprint. (Cohn, 2020)
- Sprint Planning: Merencanakan kerja untuk sprint.
- Daily Stand-up: Memantau progress harian.
- Sprint Execution: Tim melakukan pekerjaan pengembangan.

Output Kunci: Deliverables, Data Kinerja, dan Change Requests

- Deliverables: Fitur yang sudah di-coding, dokumentasi, laporan.
- Work Performance Data: Data mentah (misal: % task selesai, jumlah bug ditemukan, jam kerja yang dihabiskan).
- Change Requests: Setiap permintaan untuk mengubah baseline (scope, jadwal, biaya) akan diajukan di sini.

Manage Project Knowledge - Mengelola Aset Tak Berwujud

- Definisi: Menggunakan pengetahuan yang ada dan menciptakan pengetahuan baru untuk mencapai tujuan proyek dan mendukung organisasi belajar. (PMI, 2021)
- Konsep Kunci: Tacit Knowledge (di kepala orang) vs Explicit Knowledge (terdokumentasi).

Teknik Knowledge Management di Proyek TI

- Community of Practice (CoP): Forum diskusi antar developer backend, frontend, dll.
- Pair Programming & Mob Programming: Transfer pengetahuan tacit secara langsung. (Atlassian, 2023)
- Kode Review (Code Review): Selain meningkatkan kualitas, juga menyebarkan pengetahuan.
- Dokumentasi yang Baik: README.md, API Documentation (Swagger/OpenAPI), Architectural Decision Records (ADRs).
- Retrospectives: Ruang aman untuk membahas apa yang berjalan baik dan tidak. (Cohn, 2020; Sutherland, 2019)
- Knowledge Repositori: Wiki (Confluence), Shared Drives.

Contoh: Architectural Decision Record (ADR)

- Apa itu? Dokumen pendek yang mencatat keputusan arsitektural penting, konteks, dan konsekuensinya.
- Template ADR:
 - Title: [Keputusan yang diambil, e.g., "Menggunakan GraphQL daripada REST"]
 - Status: [Diusulkan, Diterima, Disetujui]
 - Context: [Masalah atau peluang yang dihadapi, e.g., "Kebutuhan frontend untuk data yang fleksibel dan minim over-fetching"]
 - Decision: ["Kami akan menggunakan GraphQL sebagai layer API utama."]
 - Consequences: ["Positif: Frontend lebih efisien. Negatif: Kurva belajar untuk tim, kebutuhan tooling tambahan."]

Monitor and Control Project Work - "Pilot yang Mengemudikan Pesawat"

- Definisi: Melacak, meninjau, dan melaporkan progress proyek untuk memenuhi tujuan kinerja yang ditetapkan dalam PMP. (PMI, 2017)
- Tujuan: Memastikan proyek tetap pada jalurnya (on track) dan mengambil tindakan korektif jika menyimpang.

Input: Work Performance Data -> Information -> Reports

- Data Kinerja Kerja (Work Performance Data): Data mentah (50% task selesai, 10 bug critical).
- Analisis -> Informasi Kinerja Kerja (Work Performance Information): Data yang telah dianalisis ("Kita berada di belakang jadwal 3 hari karena bug critical yang tidak terduga").
- Laporan Kinerja Kerja (Work Performance Reports): Laporan formal untuk stakeholder ("Status Report: Proyek BERISIKO, rekomendasi: tambahkan resource QA").

Teknik Analisis Kunci: EVM dan Metrik Agile

- Earned Value Management (EVM) - untuk proyek prediktif: Mengintegrasikan scope, cost, dan schedule. (PMI,2017; Kerzner,2022)
- Metrics: PV, EV, AC, CV, SV, CPI, SPI.
- Burndown/Burnup Charts - untuk proyek agile: Melihat progress sprint/release. (Cohn,2020)
- Cumulative Flow Diagram (CFD): Untuk melihat bottleneck dalam alur kerja.
- Lead Time & Cycle Time: Mengukur efisiensi tim. (Forsgren,*et.al*, 2018)

Contoh: Membaca Burndown Chart dan Cumulative Flow Diagram

- Tampilkan gambar Burndown Chart.
- Garis Ideal vs Garis Aktual.
- Interpretasi: "Jika garis aktual di atas garis ideal, berarti tim lebih lambat dari yang direncanakan."
- Tampilkan gambar Cumulative Flow Diagram (CFD).
- Tunjukkan bagaimana lebar band yang membesar di satu kolom (e.g., "In Progress") menandakan bottleneck.

Pertemuan Status (Status Meetings) yang Efektif

- Untuk Tim (Daily Stand-up):15 menit. Apa yang dikerjakan kemarin? Hari ini? Hambatan? (Sutherland,2019)
- Untuk Sponsor/Manajemen (Weekly/Monthly Review):Fokus pada pencapaian milestone, metrik kinerja, risiko, dan keputusan yang dibutuhkan.
- Hindari "Status Theater":Presentasi yang hanya indah tetapi tidak mencerminkan realitas.

Teknik: Akar Penyebab (Root Cause Analysis)

- Digunakan ketika ada masalah atau deviasi.
- Teknik: 5 Why's:Menanyakan "Mengapa?" secara berulang hingga menemukan akar penyebab.
- Contoh: Deployment gagal (Why 1?) -> Script gagal (Why 2?) -> Environment staging & production berbeda (Why 3?) -> Tidak ada dokumentasi yang memadai tentang environment requirement (Akar Penyebab).

Perform Integrated Change Control - Penjaga Baseline

- Definisi: Proses meninjau, menyetujui/menolak, dan mengelola perubahan pada baseline proyek (scope, jadwal, biaya). (PMI,2017)
- Tujuan Utama: Mencegah "Scope Creep" - perubahan tidak terkendali yang membunuh proyek.

Alur Kerja Change Control yang Khas

- 1. Permintaan Perubahan (Change Request): Diajukan secara formal (misal: melalui Jira).
- 2. Analisis Dampak (Impact Analysis): Tim teknis menganalisis dampak perubahan terhadap:
 - Scope (fitur lain?).
 - Jadwal (berapa hari delay?).
 - Biaya (biaya tambahan?).
 - Kualitas & Risiko.
- 3. Keputusan oleh CCB/Sponsor: Change Control Board (atau Sponsor untuk proyek kecil) menyetujui, menolak, atau menunda.
- 4. Komunikasi & Implementasi: Jika disetujui, update PMP dan baseline, lalu kerjakan.

Change Control di Lingkungan Agile

- Prinsip Agile: "Welcome changing requirements, even late in development." (Ries,2021)
- Bagaimana Caranya? (Referensi #4)
- Perubahan tidak dilakukan di tengah sprint. Sprint goal dilindungi.
- Perubahan ditambahkan ke Product Backlog.
- Product Owner yang memprioritaskan ulang backlog untuk sprint berikutnya.
- Perubahan besar dapat mempengaruhi Release Plan.

Close Project or Phase - Mengakhiri dengan Elegansi

- Definisi: Menyelesaikan semua aktivitas untuk menutup proyek, fase, atau kontrak secara formal. (PMI,2017)
- Bukan Hanya untuk Proyek yang Sukses! Proyek yang dihentikan juga perlu ditutup secara formal.

Aktivitas Penutupan Proyek TI

- Administratif:

Mendapatkan acceptance formal dari stakeholder/client (Sign-off).

Menyelesaikan semua kontrak dan pembayaran.

- Teknis:

Release Final & Deployment to Production.

Handover System: Serahkan kode sumber, dokumentasi, credential, dan akses ke tim operasi (Ops).

Archiving Repository: Pastikan semua code, dokumentasi, dan artifact proyek diarsipkan dengan baik.

- Manajerial:

Melepas tim dan sumber daya.

Mengukur kepuasan stakeholder.

Lessons Learned Workshop - Investasi untuk Masa Depan

- Tujuan: Mengidentifikasi apa yang berjalan baik dan apa yang dapat ditingkatkan. (PMI,2021;Cohn,2020)
- Format: Retrospective (untuk Agile) atau Workshop formal.
- Pertanyaan Panduan:
 - Apa yang berjalan dengan baik? (Proses, tools, komunikasi)
 - Apa yang bisa diperbaiki?
 - Apa yang akan kita lakukan berbeda di proyek berikutnya?
- Pastikan hasilnya terdokumentasi dan dibagikan!

Tren Masa Depan dalam Project Integration Management

AI dan Machine Learning dalam Manajemen Proyek: (Kerzner,2022)

AI untuk prediksi risiko (dari data historis).

Chatbot untuk otomatisasi pelaporan status.

Tools seperti Jira sekarang sudah memiliki kemampuan AI.

Hyper-Automation: Mengotomasi sebanyak mungkin aspek eksekusi dan monitoring proyek (CI/CD adalah contohnya). (Kim,*et.al*,2021; Forsgren,*et.al*,2018)

Project Management di Lingkungan Remote/Hybrid: Alat kolaborasi asynchronous menjadi kunci integrasi.

Ringkasan Utama Seluruh Materi

- Integration Management adalah benang merah yang menyatukan semua aspek proyek.
- Dimulai dengan Charter yang kuat, diikuti dengan PMP yang komprehensif dan disesuaikan (tailored).
- Eksekusi (Direct and Manage Work) dipantau secara terus menerus menggunakan metrik yang relevan (EVM/Agile).
- Perubahan harus dikelola melalui proses yang jelas, bukan ditolak atau diterima begitu saja.
- Penutupan yang baik, termasuk Lessons Learned, sangat berharga untuk perbaikan berkelanjutan.

DAFTAR PUSTAKA

- Atlassian. (2023). *Agile Coach Documentation & Guides*. [Online] Tersedia di: <https://www.atlassian.com/agile>
- Cohn, M. (2020). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High-Performing Technology Organizations*. IT Revolution Press.
- Kerzner, H. (2022). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. 13th Edition. Wiley.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. 2nd Edition. IT Revolution Press.
- Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition*. Project Management Institute, Inc.
- Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition*. Project Management Institute, Inc.
- Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*.
- Sutherland, J. (2019). *Scrum: The Art of Doing Twice the Work in Half the Time*. Currency.
- Wisoeki, R. K. (2019). *Effective Project Management: Traditional, Agile, Extreme, Hybrid*. 8th Edition. Wiley.