

Projektplanung: PSP, Netzplan & Gantt

Prüfungswissen

- PSP zerlegt das Projekt hierarchisch in Arbeitspakete – Struktur ohne Zeit

- Arbeitspaket: kleinste plan- und kontrollierbare Einheit mit Verantwortlichem

- Netzplan zeigt Vorgänge und Abhängigkeiten; Vorwärts-/Rückwärtsrechnung und Puffer

- Kritischer Weg: längste Kette mit Gesamtpuffer = 0

- Pufferarten: Gesamt- und freier Puffer geben Spielräume

- Gantt-Chart visualisiert Termine, Meilensteine und Soll/Ist

- Meilenstein: Entscheidungspunkt mit Dauer = 0

Prüfungsfallen & Tipps

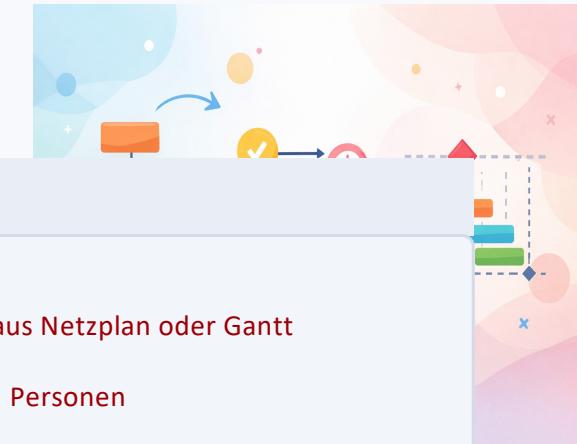
- PSP ist kein Zeitplan; Zeiten kommen aus Netzplan oder Gantt

- Puffer gehören zu Vorgängen, nicht zu Personen

- Meilenstein \neq Aufgabe – kein Aufwand

- Netzplan ohne Abhängigkeiten liefert keine Terminierung

- Kritischen Pfad überwachen, Verzögerung verschiebt das Projektende



Mini-Beispiel: A(2T) \rightarrow B(3T) \rightarrow C(1T) \rightarrow kritischer Pfad = 6T; +1T bei B verzögert Projekt um 1T.

Projektphasen (klassisch)

Prüfungswissen

- Initierung: Ziele, Scope, Stakeholder; Projektauftrag/Go-NoGo
- Planung: PSP, Aufwand-/Kosten- und Terminplan, Ressourcen, Kommunikation, Risiken
- Durchführung: Arbeitspakete umsetzen, Team steuern, Qualität sichern
- Überwachung/Steuerung: Soll-Ist-Vergleich, Abweichungen managen, Changes steuern
- Abschluss: Abnahme, Übergabe, Dokumentation, Lessons Learned

Prüfungsfallen & Tipps

- Abschluss umfasst auch Abrechnung und Übergabe an den Betrieb
- Ohne Messgrößen (Zeit, Kosten, Scope, Qualität) kein Controlling
- Scope-Creep vermeiden: Änderungen nur via Change-Prozess
- Projektauftrag entsteht in der Initiierungsphase
- Lessons Learned sichern Wissen für nächste Projekte

Mini-Check: Der Projektauftrag wird in der Initiierung erstellt; Controlling benötigt definierte Messgrößen.

Vorgehensmodelle: Wasserfall, V-Modell & Scrum



Prüfungswissen

- Wasserfall: sequentiell; geeignet bei stabilen Anforderungen
- V-Modell: koppelt Entwicklungs- und Testphasen; hohe QS/Dokumentation
- Phasen im V-Modell: Anforderung, Spezifikation, Architektur, Komponenten, Implementation & zugehörige Tests
- Scrum: iterativ; Product Owner priorisiert Backlog, Scrum Master moderiert; Team liefert Inkremente
- Auswahlkriterien: Änderungsrate, Risiko, Regulatorik, Teamreife, Feedback-Geschwindigkeit

Prüfungsfallen & Tipps

- Agil ≠ planlos: Backlog, DoD und Timeboxing sind Pflicht
- Wasserfall nur bei wirklich stabilen Anforderungen verwenden
- V-Modell ist nicht nur Wasserfall mit Tests, sondern strukturiert QS
- Scrum erfordert Disziplin: Rollen & Events einhalten
- Je unsicherer die Anforderungen, desto kürzer die Feedback-Schleifen

Merksatz: Je höher die Unsicherheit, desto kürzer sollten Feedback-Schleifen und Iterationen sein.

Teambildung & -entwicklung (Tuckman)



Prüfungswissen

- Team: gemeinsames Ziel, Abhängigkeiten und klare Verantwortlichkeiten
- Phasen: Forming (Orientierung), Storming (Konflikte), Norming (Regeln), Performing (Leistung)
- Führung: Orientierung geben, Konflikte moderieren, Regeln stabilisieren, Hindernisse beseitigen
- Werkzeuge: Rollen klären, Arbeitsregeln definieren, regelmäßige Abstimmungen, Task-Board

Prüfungsfallen & Tipps

- Storming ist normal – Konflikte aktiv adressieren
- Unklare Verantwortlichkeiten führen zu „niemand zuständig“
- Kommunikation planen: Kanal, Rhythmus, Verantwortliche
- Teamentwicklung braucht Zeit; Geduld und Feedback wichtig
- Rollen regelmäßig überprüfen und anpassen

Mini-Beispiel: Ein neues Team durchläuft zuerst Orientierung, dann Reibung, dann Regelbildung und erreicht schließlich die Leistungsphase.

Reflexionsmethoden



Prüfungswissen

- Retrospektive: Team reflektiert regelmäßig, was gut/schlecht lief und leitet Verbesserungen ab
- Lessons Learned: Erfahrungen systematisch sammeln und in Wissensbasis überführen
- After Action Review: Erwartet? Passiert? Warum? Was lernen wir?
- 5-Why-Methode: Ursache durch mehrfaches „Warum?“ finden

Prüfungsfallen & Tipps

- Ohne konkrete Maßnahmen (SMART, Owner, Termin) bleibt es beim Reden
- Schuldzuweisungen blockieren Lernen – Fokus auf Ursachen und Systeme
- Maßnahmen verfolgen und Ergebnisse sichtbar machen
- Offenheit fördern: nur so werden Probleme angesprochen

Mini-Format: Start / Stop / Continue – Was anfangen, was aufhören, was fortführen?

Machbarkeitsanalyse



Prüfungswissen

- Zweck: Prüft Realisierbarkeit und Sinnhaftigkeit; Grundlage für Go/NoGo-Entscheidung
- Dimensionen: technisch, wirtschaftlich, organisatorisch, rechtlich, zeitlich
- Ergebnis: Varianten, Annahmen, grobe Kosten/Nutzen, Hauptrisiken, Empfehlung
- Varianten vergleichen: Make-or-Buy, Outsourcing usw.

Prüfungsfallen & Tipps

- Machbar bedeutet nicht automatisch wirtschaftlich sinnvoll
- Ohne klaren Scope wird die Analyse wertlos
- Annahmen dokumentieren: Ressourcen, Termine, Abhängigkeiten
- Frühzeitige Einbindung von Experten sichert Qualität

Mini-Check: Zwei bis drei Varianten (Make/Buy/Outsource) vergleichen und Nutzen bewerten.

Stakeholderanalyse



Prüfungswissen

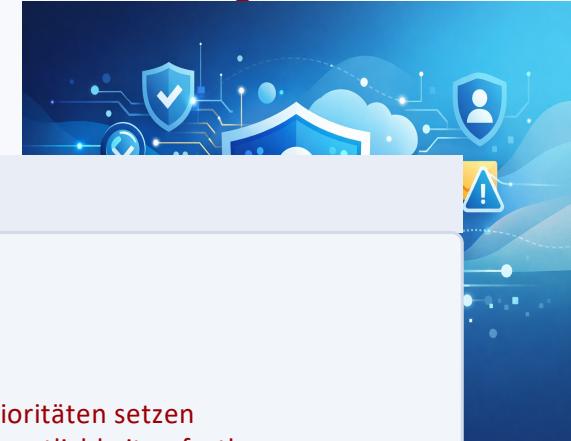
- Stakeholder identifizieren: intern/extern, direkt/indirekt betroffen
- Bewertung mit Einfluss-/Interesse-Matrix: vier Felder
- Strategien: hohe Macht + hohes Interesse → eng einbinden; hohe Macht + geringes Interesse → zufrieden halten; geringe Macht + hohes Interesse → informieren; geringe Macht + geringes Interesse → beobachten
- Kommunikationsplan: Wer? Welche Infos? Kanal? Frequenz? Owner?

Prüfungsfallen & Tipps

- Wichtigste Stakeholder sind nicht immer die lautesten
- Oft vergessen: Betrieb/Support, Datenschutz, Einkauf, Betriebsrat
- Frühe Einbindung reduziert Widerstände und späte Änderungen
- Bewertungen regelmäßig aktualisieren; Interessen ändern sich

Mini-Beispiel: Auftraggeber mit hoher Macht und hohem Interesse muss eng geführt werden.

Risikoanalyse

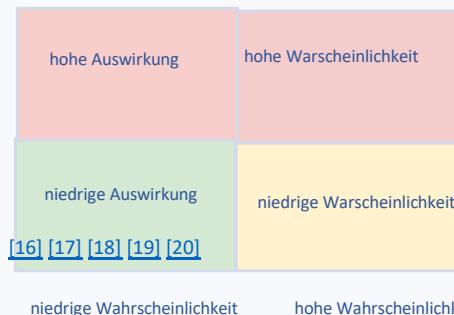


Prüfungswissen

- Risiko = unsicheres Ereignis mit Auswirkungen auf Zeit, Kosten, Qualität oder Scope
- Bewertung per Risikomatrix: Wahrscheinlichkeit × Auswirkung → grün/gelb/rot
- Risikowert = Eintrittswahrscheinlichkeit × Schadenshöhe → Priorisierung
- Strategien: vermeiden, vermindern, verlagern, akzeptieren
- Risikoregister: ID, Beschreibung, Bewertung, Maßnahmen, Owner, Status, Trigger

Prüfungsfallen & Tipps

- Liste ohne Bewertung ist wertlos – Prioritäten setzen
- Ohne Owner passiert nichts – Verantwortlichkeiten festlegen
- Trigger/Frühwarnindikatoren definieren; Risiken überwachen
- Maßnahmen regelmäßig überprüfen und anpassen



Mini-Beispiel: Lieferverzug → Puffer planen & Zweitlieferanten definieren.

Stamm- und Bewegungsdaten



Prüfungswissen

- Stammdaten: stabile Informationen zu Kunden, Produkten, Mitarbeitern etc. – seltene Änderungen
- Stammdaten werden unternehmensweit genutzt und sind Grundlage für Transaktionen
- Bewegungsdaten: dynamische Transaktionen wie Bestellungen, Rechnungen, Buchungen
- Bewegungsdaten referenzieren Stammdaten über Schlüssel; ohne Stammdaten keine Bewegungsdaten
- Stammdatenpflege: Dublettensuche, Pflichtfelder, Versionierung für Datenqualität

Prüfungsfallen & Tipps

- Schlechte Stammdaten ruinieren Auswertungen und Prozesse
- Bewegungsdaten ohne Zeitstempel/Status sind schwer steuerbar
- Datenschutz beachten: Stammdaten sind oft personenbezogen
- Verantwortlichkeiten für Stammdatenpflege klar definieren

Mini-Beispiel: Kunde (Stamm) → Bestellung (Bewegung) → Lieferstatus (Bewegung).

Datenschutz & IT-Sicherheit

Prüfungswissen

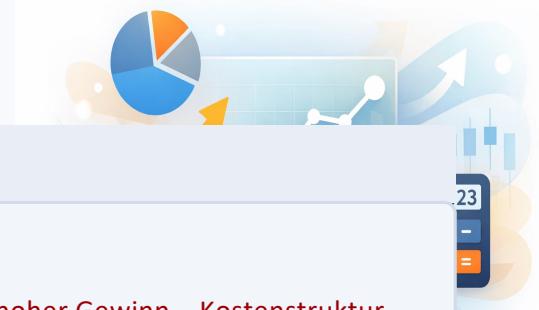
- DSGVO-Grundsätze: Rechtmäßigkeit, Treu und Glauben, Transparenz; Zweckbindung; Datenminimierung; Richtigkeit; Speicherbegrenzung; Integrität & Vertraulichkeit
- Schutzziele (CIA): Vertraulichkeit, Integrität, Verfügbarkeit
- TOMs: Zugriffskontrollen, Verschlüsselung, Backup/Restore, Patch-Management, Logging, Rechtekonzept (Least Privilege)
- Praxis: Zweck & Rechtsgrundlage dokumentieren; Löschkonzept; Berechtigungen konsequent umsetzen

Prüfungsfallen & Tipps

- IT-Sicherheit ist mehr als Antivirus: Rechte, Prozesse und Updates sind entscheidend
- Backup ohne Restore-Test hat keinen Wert
- Nur notwendige Daten erheben („Need to know“)
- Datenschutz und Informationssicherheit frühzeitig im Projekt berücksichtigen

Mini-Check: Personenbezogene Daten? → Zweck, Rechtsgrundlage, TOMs und Löschung prüfen.

Kennzahlen: Umsatz, Gewinn & Deckungsbeitrag



Prüfungswissen

- Umsatz: Gesamterlöse in einem Zeitraum; Menge \times Preis
- Gewinn = Umsatz minus alle Kosten (Personal, Produktion, Marketing etc.)
- Deckungsbeitrag (DB) = Umsatz – variable Kosten; deckt Fixkosten und trägt zum Gewinn bei
- Break-even: Fixkosten / DB-Quote (oder DB je Stück) \rightarrow ab diesem Punkt entsteht Gewinn
- DB hilft bei Preis- und Produktentscheidungen; höherer DB = besser

Prüfungsfallen & Tipps

- Hoher Umsatz heißt nicht automatisch hoher Gewinn – Kostenstruktur analysieren
- DB unterstützt Sortiments- und Produktpriorisierung
- Fix- und variable Kosten sauber trennen
- Break-even-Analyse regelmäßig durchführen

Mini-Beispiel: Preis 100 €, variable Kosten 60 € \rightarrow DB 40 €; Fixkosten 400 € \rightarrow Break-even bei 10 Stück.

Kundenkommunikation (IT-Support)

Prüfungswissen

- Ziel: Problem verstehen, Erwartung klären, nächsten Schritt & Zeitrahmen abstimmen
- Techniken: aktives Zuhören & Paraphrasieren; offene W-Fragen
- Ablauf: Begrüßung, Problem aufnehmen, Zusammenfassen, Priorität/SLA klären, Lösung/Workaround, Abschluss
- Dokumentation: Ticketnotizen klar (Wer? Was? Wann? Wirkung?)
- Emotionsmanagement: ruhig bleiben, empathisch reagieren
- Proaktive Kommunikation: regelmäßige Updates, sich aktiv melden
- Klare Sprache: Fachbegriffe vermeiden; Kunden abholen



Prüfungsfallen & Tipps

- Fachwörter ohne Übersetzung verursachen Missverständnisse
- Keine Versprechen ohne Plan – lieber Schritte & Termine nennen
- Status aktiv kommunizieren, auch wenn es keine Neuigkeiten gibt
- Gesamten Kontaktprozess gestalten: vom ersten Eindruck bis zur Nachverfolgung

Satzbaustein: „Ich fasse zusammen ... Nächster Schritt ... bis ...“

Fehlermanagement (Bugs)



Prüfungswissen

- Bug-Report: reproduzierbare Schritte, erwartetes vs. aktuelles Verhalten und Umgebungsdetails
- Triage: Dublette? reproduzierbar? Severity (Auswirkung) & Priority (Dringlichkeit) bestimmen; Zuweisung
- Lebenszyklus: New → Confirmed → In Progress → Resolved → Verified → Closed
- Severity vs. Priority: Severity = wie schlimm; Priority = wie schnell zu fixen
- Ziel: entscheiden (fixen, Workaround, später, ablehnen) und dokumentieren

Prüfungsfallen & Tipps

- Severity ≠ Priority – Auswirkung und Dringlichkeit trennen
- „Nicht reproduzierbar“ deutet auf fehlende Informationen
- Fix ohne Test oder Regression erzeugt neue Fehler – Regressionstests einplanen
- Klare Dokumentation erleichtert spätere Analysen

Merker: Severity = wie schlimm, Priority = wie schnell.

Störungsmanagement (Incident)



Prüfungswissen

- Incident: ungeplante Unterbrechung oder Qualitätsminderung eines IT-Services
- Ziel: Service schnell wiederherstellen; Auswirkungen minimieren
- Prozess: Erfassen → Klassifizieren → Priorisieren → Zuweisen → Lösen → Dokumentieren → Schließen
- Eskalation: funktional (2nd/3rd Level) und hierarchisch (Management)
- SLAs: Reaktionszeit vs. Lösungszeit; Priorität = Impact × Dringlichkeit

Prüfungsfallen & Tipps

- Erst Wiederherstellung, dann Ursachenanalyse (Problem Management)
- Kommunikation während des Incidents ist Pflicht: Status, Workaround, ETA
- Priorität begründen – Impact & Urgency statt Bauchgefühl
- Dokumentation: Wer, Was, Wann – für Nachvollziehbarkeit

Mini-Beispiel: VPN-Ausfall → Workaround bereitstellen, eskalieren; Ursachenanalyse im Nachgang.

Supportanfragen (Service Requests)



Prüfungswissen

- Service Request: Standardleistung oder Anfrage nach Neuem (Zugang, Gerät, Info)
- Abgrenzung: Incident = Störung; Request = Erfüllung ohne Störung
- Request-Fulfilment: Katalog, Genehmigungen, Standard-Workflows
- Automatisierung: Self-Service, Vorlagen, klare Zuständigkeiten
- Genehmigungen abhängig von Sensitivität; Compliance & Sicherheit beachten
- Saubere Dokumentation: Prozesse, Status, Zeitrahmen

Prüfungsfallen & Tipps

- Alles als Incident melden verfälscht Berichte/SLA-Auswertung
- Genehmigungen nicht abkürzen – Sicherheitsrisiko
- Standard-Requests zuerst automatisieren für schnellen ROI
- Servicekatalog regelmäßig aktualisieren

Mini-Beispiel: „Rechte auf Ordner X“ (Request) vs. „Ordner X nicht erreichbar“ (Incident).

Bearbeitungsstatus (Ticketsystem)



Prüfungswissen

- Minimal-Workflow: Open/New → In Progress → Pending/Waiting → Resolved → Closed
- Status zeigt, wer gerade am Zug ist (Team, Kunde, Dritter)
- Kennzahlen: First Response Time, Resolution Time, Reopen Rate
- Gute Praxis: Status + nächste Aktion + Termin immer sichtbar
- Differenzierung Resolved vs. Closed: gelöst vs. bestätigt/abgenommen

Prüfungsfallen & Tipps

- Pending ohne Grund/Termin wird zum Ticket-Friedhof
- „Resolved“ erst setzen, wenn Lösung geliefert; „Closed“ erst nach Bestätigung
- Dokumentation: Wer, Was, Wann – unverzichtbar für Nachvollziehbarkeit
- Klare SLA-Definition und Eskalation bei Überschreitung

Mini-Beispiel: Kunde liefert Info → Pending; Info da → In Progress; Lösung → Resolved; Bestätigung → Closed.