Bedarfsanalyse (IT-Systeme konzeptionieren)

Prüfungswissen

Ziel: Ist-Situation + Ziele + Einschränkungen → klare Anforderungen
Inhalte: Stakeholder, Use-Cases, Muss/Kann, Prioritäten, Budget/Termin, Risiken
Methoden: Interviews, Workshops, Beobachtung, Dokumentenanalyse

Ergebnis: Anforderungsliste + Abnahmekriterien (messbar)

Prüfungsfallen & Tipps

Zu früh Lösung nennen → Bedarf bleibt unscharf

„Alle wollen alles“ ohne Priorisierung → Scope-Creep

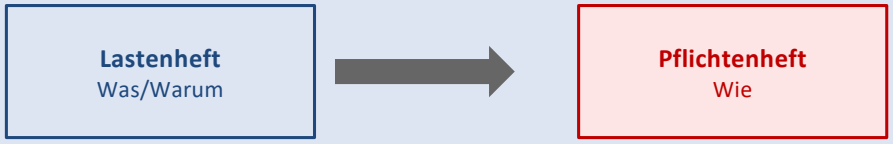
Keine Abnahmekriterien = Streit am Ende

Mini-Beispiel/Merksatz: *Muss: „VPN-Zugriff“, Kann: „Self-Service-Portal“*

Lastenheft vs Pflichtenheft

Prüfungswissen

Lastenheft: WAS und WARUM aus Kundensicht (Ziele, Anforderungen, Rahmen)
Pflichtenheft: WIE aus Auftragnehmersicht (Umsetzung, Architektur, Tests)
Gemeinsam: Scope, Schnittstellen, Qualitätsanforderungen, Abnahme



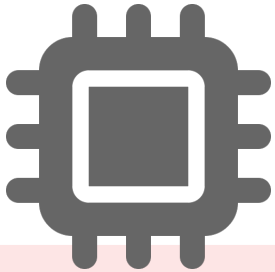
Prüfungsfallen & Tipps

Lastenheft zu technisch (verwechselt Pflichtenheft)

Pflichtenheft ohne Tests/Abnahmekriterien ist lückenhaft

Mini-Beispiel/Merksatz: *Merker: Lasten = „Was will der Kunde?“ / Pflichten = „Wie machen wir’s?“*

Installation & Konfiguration: OS (Überblick)



Prüfungswissen

Ablauf: Boot-Medium → Ziel-Datenträger → Partition/FS → Installation → Treiber/Updates
Basis-Härtung: Admin/Passwort, Firewall, Updates, Standarduser, AV/EDR
Dokumentation: Versionen, Keys/Lizenzen, Konfig, IPs, Passwörter (sicher)

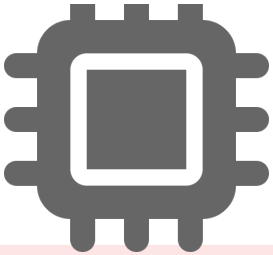
Prüfungsfallen & Tipps

Falsches Installationsziel → Datenverlust

Treiber/Updates auslassen → instabil/unsicher

Mini-Beispiel/Merksatz: *Reihenfolge: installieren → updaten → absichern*

BIOS/UEFI (Installation vorbereiten)



Prüfungswissen

UEFI: GPT, Secure Boot; moderner Standard

BIOS/Legacy: älter: oft MBR

Boot-Order: USB/SSD/Netzwerk; UEFI/Legacy-Modus passend zum Medium

Secure Boot/TPM: können Installationen beeinflussen

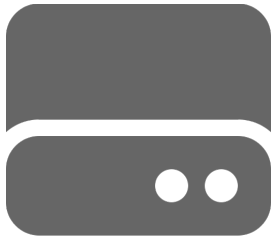
Prüfungsfallen & Tipps

Stick bootet nicht: falscher Modus/Boot-Order

Secure Boot blockt unsignierte Medien

Mini-Beispiel/Merksatz: *Merker: UEFI + GPT ist Standard*

Partitionierung & Formatierung (Basics)



Prüfungswissen

Partition: logische Aufteilung; Dateisystem = Struktur/Rechte/Journaling

MBR vs GPT: GPT unterstützt große Disks + viele Partitionen

Dateisysteme: NTFS (Windows), ext4 (Linux); FAT/exFAT für Kompatibilität

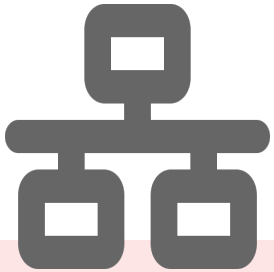
Prüfungsfallen & Tipps

Falsches Dateisystem → Rechte/Features fehlen

Partitionierung ohne Plan erschwert Dual-Boot/Recovery

Mini-Beispiel/Merksatz: *Beispiel: System (C:) + Daten (D:) + Recovery*

Netzwerk-Anbindung & IP-Konfiguration



Prüfungswissen

IP-Konfig: IP, Maske/Prefix, Gateway, DNS

DHCP vs statisch: DHCP für Clients; statisch für Server/Netzgeräte (oder Reservierung)

Test: ping Gateway, DNS-Name, traceroute

Prüfungsfallen & Tipps

Gateway falsch/fehlt → kein Internet

DNS falsch → „Internet kaputt“ obwohl IP ok

Mini-Beispiel/Merksatz: *Check: IP ok? Gateway ok? DNS ok?*

Remotedesktop (RDP/SSH/VNC – Einordnung)

Prüfungswissen

RDP: Windows Remote Desktop

SSH: sichere CLI

VNC: Bildschirmübertragung

Sicherheit: MFA/VPN, starke Passwörter/Keys, nur notwendige Ports, Logging

Zugriffskontrolle: Rollen/Rechte, getrennte Admin-Accounts

Prüfungsfallen & Tipps

RDP/SSH offen ins Internet ohne Schutz = No-Go

Keine Logs → keine Nachvollziehbarkeit

Mini-Beispiel/Merksatz: *Merker: „Remote nur über VPN + MFA“*

Geräteklassen: Desktop • Notebook • Tablet • Smartphone

Prüfungswissen

Desktop: günstig pro Leistung, gut erweiterbar, stationär

Notebook: mobil, Akku, Docking; oft weniger erweiterbar

Tablet: leicht, Touch, Apps; begrenzte Profi-Tools

Smartphone: Kommunikation/MFA; kleiner Screen, eingeschränkte Produktivität

Prüfungsfallen & Tipps

Einsatzfall entscheidet, nicht „Leistung um jeden Preis“

Reparierbarkeit/Upgrades bei Notebooks oft eingeschränkt

Mini-Beispiel/Merksatz: *Merker: Mobilität ↔ Erweiterbarkeit*

Mobile vs stationäre Arbeitsplatzsysteme (Netz-Anbindung)

Prüfungswissen

Stationär: LAN (stabil), ggf. WAN via Standortnetz

Mobil: WLAN, mobiler Datenfunk (LTE/5G), VPN für Firmennetz

Kriterien: Sicherheit, Bandbreite/Latenz, Verfügbarkeit, Kosten

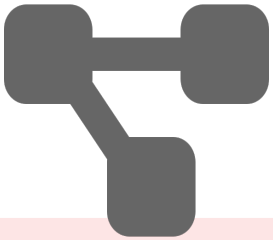
Prüfungsfallen & Tipps

WLAN = geteiltes Medium; bei Dichte wird’s langsam

LTE/5G: Volumen/Abdeckung beachten

Mini-Beispiel/Merksatz: *Merker: „Kabel für Stabilität, Funk für Flexibilität“*

LAN vs WAN vs mobiler Datenfunk (Kurzvergleich)



Prüfungswissen

LAN: lokal, schnell, geringe Latenz

WAN: verbindet Standorte/Internet, meist höhere Latenz

LTE/5G: mobil, variabel, abhängig von Netz/Abdeckung

Prüfungsfallen & Tipps

WAN-Probleme wirken wie „Server langsam“ → Latenz prüfen

Mobilfunk kann NAT/Restriktionen haben

Mini-Beispiel/Merksatz: *Beispiel: Videocall braucht stabile Upstream-Bandbreite*

Barrierefreiheit am Arbeitsplatz (praktisch)

Prüfungswissen

Ziel: Zugang für alle (Sehen/Hören/Motorik/Kognition)

Maßnahmen: 2. Monitor, größere Schrift/Zoom, Screenreader, Tastaturbedienung, gute Audio-Hardware

Software: Kontrastmodus, Untertitel, klare UI, kurze Texte

Prüfungsfallen & Tipps

Barrierefreiheit = nicht nur „nice“, oft Pflicht/Standard

„Einmal einstellen“ reicht nicht: Nutzer testen

Mini-Beispiel/Merksatz: *Merker: „Ergonomie + Zugänglichkeit = Produktivität“*

Softwareauswahl: Anwendungssoftware & Betriebssysteme

Prüfungswissen

Auswahl nach Anforderungen: Funktionen, Sicherheit, Support, Kompatibilität

Betrieb: Updates, Lifecycle/EOL, Admin-Aufwand, Rollout

Daten: Formate, Migration, Backup/Restore

Prüfungsfallen & Tipps

Tool wählen ohne Prozesse zu kennen → passt nicht

EOL ignorieren → Sicherheitsrisiko

Mini-Beispiel/Merksatz: *Check: Muss-Funktionen + Support + Kompatibilität*

IDE (Integrierte Entwicklungsumgebung)

Prüfungswissen

IDE: Editor + Build + Debugger + Tests + Versionskontrolle-Integration
Nutzen: schnellere Entwicklung, weniger Fehler, bessere Navigation/Refactoring
Kriterien: Sprache/Framework-Support, Plugins, Debugging, Team-Standards

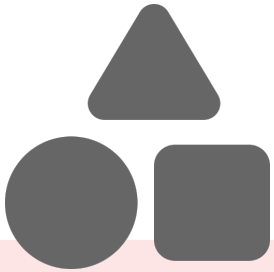
Prüfungsfallen & Tipps

“Lieblings-IDE” ≠ Teamstandard (Onboarding/CI beachten)

Plugins können Security/Performance beeinflussen

Mini-Beispiel/Merksatz: *Merker: Debugger spart Stunden*

Standard- vs Individualsoftware



Prüfungswissen

Standardsoftware: schneller verfügbar, günstiger, Updates vom Hersteller
Individualsoftware: passgenau, aber mehr Aufwand für Entwicklung/Wartung
Entscheidung: Prozess-Fit, Anpassbarkeit, Kosten, Abhängigkeit, Time-to-Market

Prüfungsfallen & Tipps

Individual = langfristige Pflegepflicht

Standard anpassen bis es teuerer als individuell wird

Mini-Beispiel/Merksatz: *Merker: „Build vs Buy“ immer mit Betriebskosten*

Branchensoftware

Prüfungswissen

Branchensoftware: Standardsoftware für spezielle Branchen (z.B. Arztpraxis, Handwerk)
Vorteile: Prozesse eingebaut, Compliance/Standards oft abgedeckt

Risiken: Lock-in, Schnittstellen, Datenexport/Migration

Prüfungsfallen & Tipps

Schnittstellen/Export nicht geprüft → später gefangen

Schulungsaufwand unterschätzt

Mini-Beispiel/Merksatz: *Check: Datenexport + API + Supportvertrag*

Open Source vs proprietär

Prüfungswissen

Open Source: Quellcode offen, Lizenzbedingungen; oft flexibel/anpassbar

Proprietär: geschlossen, Hersteller-Support, oft klare SLAs

Bewertung: Sicherheit/Updates, Community/Support, Kosten, Compliance

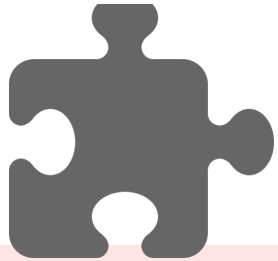
Prüfungsfallen & Tipps

Open Source ≠ kostenlos (Betrieb/Support zählt)

Proprietär ≠ sicher (Patchpolitik zählt)

Mini-Beispiel/Merksatz: *Merker: „Kosten = Lizenz + Betrieb + Risiko“*

Anpassbarkeit, Schnittstellen, Kompatibilität



Prüfungswissen

Anpassbarkeit: Konfiguration vs Customizing vs Code

Schnittstellen: API, Import/Export, Standardformate

Kompatibilität: OS, Browser, Datenbanken, Hardware, Protokolle

Prüfungsfallen & Tipps

Customizing ohne Update-Strategie → Update-Hölle

“Kompatibel” prüfen: echte Tests, nicht Marketing

Mini-Beispiel/Merksatz: *Check: API-Doku + Testintegration + Migrationspfad*

Urheberrecht (Basics für AP1)

Prüfungswissen

Software: urheberrechtlich geschützt (Quellcode, ggf. UI/Assets)

Nutzung: nur mit Lizenz/Rechtseinräumung; sonst Verstoß

Kopieren/Weitergeben/Ändern: abhängig von Lizenz

Prüfungsfallen & Tipps

„Im Internet gefunden“ ist keine Lizenz

Screens/Icons/Fonts sind auch geschützt

Mini-Beispiel/Merksatz: *Merker: „Ohne Lizenz keine Nutzung“*

Lizenzarten: EULA • OEM • GNU (Überblick)

Prüfungswissen

EULA: Endnutzer-Lizenzvertrag; regelt Nutzung/Installationen/Weitergabe

OEM: an Hardware gebunden, oft nicht übertragbar

GNU (z.B. GPL): Open-Source-Lizenz mit Bedingungen (Weitergabe/Quellcode je nach Nutzung)

Prüfungsfallen & Tipps

OEM bei Gerätewechsel oft problematisch

GPL kann “Copyleft”-Pflichten auslösen (je nach Verteilung)

Mini-Beispiel/Merksatz: *Merker: Lizenzbedingungen lesen, nicht raten*

Pay-per-Use / Pay by Use

Prüfungswissen

Abrechnung: nach Nutzung (z.B. pro User/Monat, pro Stunde, pro GB, pro Request)

Vorteile: flexibel, skalierbar, geringe Einstiegskosten

Risiken: Kosten steigen bei hoher Nutzung; Forecast/Monitoring nötig

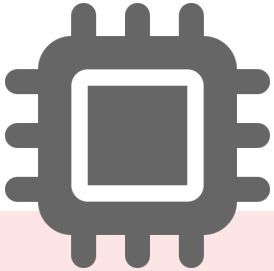
Prüfungsfallen & Tipps

Ohne Limits/Budgets → Kostenexplosion

Nutzungsdaten = Datenschutz/Transparenz beachten

Mini-Beispiel/Merksatz: *Merker: „Pay-per-use braucht Monitoring“*

Installation & Konfiguration: Hardware + OS (Praxisablauf)



Prüfungswissen

Hardware prüfen: Kompatibilität, Firmware, BIOS/UEFI-Settings

OS installieren: Partition/FS, Treiber, Updates

Security-Basics: Benutzer/Rechte, Firewall, AV, Verschlüsselung

Übergabe: Doku + Abnahme + Backup-Plan

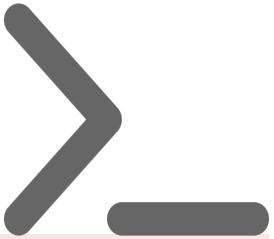
Prüfungsfallen & Tipps

"Fertig" ohne Updates/Hardening ist nicht fertig

Keine Doku → Support-Hölle

Mini-Beispiel/Merksatz: *Checkliste: Install → Update → Secure → Document*

Kommandozeile: Befehlssyntax & Parameter



Prüfungswissen

Schema: Befehl + Optionen/Parameter + Pfad/Argumente

Hilfe: man/help/--help (je nach System)

Pfad: relative vs absolute; Wildcards (*, ?)

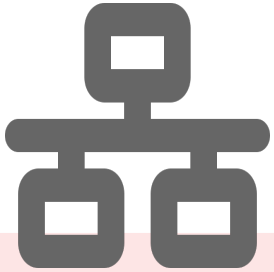
Prüfungsfallen & Tipps

Falsches Verzeichnis = “Befehl geht nicht”

Groß-/Kleinschreibung (Linux) beachten

Mini-Beispiel/Merksatz: *Merker: „erst pwd/dir prüfen, dann handeln“*

Netzwerk konfigurieren: IP, DHCP, WLAN, VPN (Basics)



Prüfungswissen

IP: statisch oder DHCP; DNS/Gateway prüfen

DHCP: Lease, Reservierung, Scope

WLAN: SSID, WPA2/WPA3, PSK vs Enterprise

VPN: Tunnel für sicheren Zugriff; Auth (MFA), Split-Tunnel-Regeln

Prüfungsfallen & Tipps

Zwei DHCP-Server → Chaos

WPA2-PSK schwach geteilt → Sicherheitsloch

Mini-Beispiel/Merksatz: *Check: IP→DNS→Route→VPN→Service*

Troubleshooting Netzwerk (Systematisch)

Prüfungswissen

Schichten denken (OSI): Kabel/WLAN → IP → DNS → Ports → Anwendung

Tools: ping, traceroute, nslookup/dig, ipconfig/ifconfig, arp

Dokumentation: Symptome, Tests, Ergebnisse, Lösung

Prüfungsfallen & Tipps

Random klicken statt Messpunkte setzen

DNS wird zu spät geprüft

Mini-Beispiel/Merksatz: *Merker: „Erst Netzwerk, dann Dienst“*

Konsolenbefehle: Dateisystem (dir/ls, mkdir, del/rm, cp/copy)

Prüfungswissen

dir/ls: Dateien anzeigen

mkdir: Ordner erstellen

del/rm: löschen (vorsichtig!)

cp/copy: kopieren; Pfade/Wildcards nutzen

Prüfungsfallen & Tipps

rm kann irreversibel sein → Pfad doppelt prüfen

Windows vs Linux Befehle unterscheiden

Mini-Beispiel/Merksatz: *Merker: „erst anzeigen, dann löschen“*

Konsolenbefehle: Netzwerk (ipconfig/ifconfig, iproute2, arp, ping, traceroute)

Prüfungswissen

ipconfig/ifconfig/ip: IP-Konfig anzeigen/setzen

arp: ARP-Cache

ping: Erreichbarkeit

traceroute: Pfad/Router-Hops

alias: Kurzbefehle (Shell)

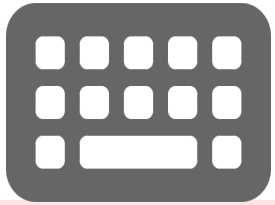
Prüfungsfallen & Tipps

ping kann durch Firewall blockiert sein → nicht sofort “Host down”

traceroute zeigt, wo es hängt (Routing/Firewall)

Mini-Beispiel/Merksatz: *Check: ping Gateway → ping DNS-IP → ping Domain*

Programmiersprachen: Compiler, Linker, Interpreter



Prüfungswissen

Compiler: übersetzt Quellcode → Maschinencode/Binary

Linker: verbindet Objektdateien/Bibliotheken → ausführbares Programm
Interpreter: führt Code direkt aus (oder Bytecode), oft dynamischer

Prüfungsfallen & Tipps

“Interpreter = langsam” ist zu pauschal

Build-Fehler vs Laufzeitfehler unterscheiden

Mini-Beispiel/Merksatz: *Merker: Compile = vorab, Interpret = zur Laufzeit*

Programmiergrundlagen (prozedural, OOP, Strukturen, Debugging)

Prüfungswissen

Prozedural: Funktionen/Prozeduren, Daten getrennt

OOP: Klassen/Objekte, Methoden, Kapselung, Vererbung (Basics)

Grundbausteine: Variablen, Datentypen, Arrays/Listen, Kontrollstrukturen (if/for/while)

Libraries/Frameworks: wiederverwendbarer Code + Struktur

Debugging: Breakpoints, Step, Watch, Logs

Prüfungsfallen & Tipps

Vererbung falsch nutzen (Komposition oft besser, aber AP1: Basics kennen)

Debugging nur mit “print” → ineffizient

Mini-Beispiel/Merksatz: *Merker: „Bug finden = reproduzieren, eingrenzen, fixen, testen“*

Datenbanken: ER-Modell (einfach) & SELECT (1 Tabelle)

Prüfungswissen

ER: Entitäten (z.B. Kunde), Attribute, Beziehungen (1:1, 1:n, n:m)

Schlüssel: Primärschlüssel identifiziert Datensatz eindeutig

SELECT Basics: SELECT Spalten FROM Tabelle WHERE Bedingung

Filter: AND/OR, LIKE, BETWEEN; Sortierung: ORDER BY; Aggregat: COUNT (Basics)

Prüfungsfallen & Tipps

WHERE vergessen → zu viele Daten

LIKE mit % und _ verwechseln

Mini-Beispiel/Merksatz: *Beispiel: SELECT name FROM kunde WHERE ort='Berlin' ORDER BY name*