



Arbeitsaufgaben in Abstimmung mit kundenspezifischen Geschäfts- und Leistungsprozessen

AP1 PRÜFUNGSVORBEREITUNG

PROJEKTMANAGEMENT

In der heutigen IT-Landschaft ist strukturiertes Projektmanagement keine Option, sondern eine Notwendigkeit. Projekte in der IT-Infrastruktur – ob Firewall-Implementierung, Netzwerkmodernisierung oder Cloud-Migration – erfordern präzise Planung, klare Kommunikation und methodisches Vorgehen. Diese Präsentation führt euch durch die fundamentalen Konzepte des IT-Projektmanagements mit einem praktischen Durchlaufbeispiel: die Implementierung eines neuen Firewall-Systems mit High-Availability-Konfiguration.

Ihr werdet lernen, wie Projekte strukturiert, terminiert und dokumentiert werden, welche Vorgehensmodelle existieren und wann sie einzusetzen sind, sowie wie Teams effektiv arbeiten und aus Projekten lernen. Der Fokus liegt dabei auf prüfungsrelevanten Inhalten der Abschlussprüfung Teil 1 – mit konkreten Arbeitsaufträgen, die euch auf typische Aufgabenstellungen vorbereiten.

Merkmale und Methoden des Projektmanagements

Was macht ein Projekt zum Projekt?

Ein Projekt unterscheidet sich fundamental von der Linienarbeit durch vier zentrale Merkmale, die ihr in der Prüfung sicher benennen können müsst. Erstens ist ein Projekt **einmalig und nicht repetitiv** – eine Standard-Benutzererstellung ist kein Projekt, die Migration einer kompletten Active-Directory-Struktur schon. Zweitens hat jedes Projekt ein **klar definiertes Ziel oder Ergebnis**, das messbar und überprüfbar sein muss. Im Fall unseres Firewall-Projekts: „Drei Firewall-Geräte in HA-Konfiguration (Active/Passive) produktiv im Einsatz, getestet und dokumentiert“.

Drittens sind **Zeit, Budget und Ressourcen begrenzt** – es gibt einen Projektrahmen, innerhalb dessen gearbeitet werden muss. Viertens ist jedes Projekt mit **Risiken und Unsicherheiten** behaftet, die identifiziert, bewertet und gesteuert werden müssen. Ein Hardwareausfall während der Implementierung, unerwartete Kompatibilitätsprobleme oder Personalausfälle sind typische Risiken in IT-Projekten.

Prüfungsfokus: Planen, Begründen, Dokumentieren

In der AP1-Prüfung geht es nicht darum, etwas „irgendwie“ zu machen. Die Prüfer erwarten von euch strukturiertes Vorgehen mit nachvollziehbarer Begründung. Jede Entscheidung muss dokumentiert, jeder Schritt geplant und jedes Ergebnis nachgewiesen werden können. Dies spiegelt die Realität wider: Ohne saubere Planung und Dokumentation entstehen Chaos, Mehrkosten und Fehler, die später teuer werden.

Einmaligkeit

Nicht-repetitiv, individuelles Vorhaben mit spezifischen Anforderungen

Klares Ziel

Messbare Ergebnisse und Deliverables, überprüfbare Abnahmekriterien

Begrenzte Ressourcen

Zeit, Budget, Personal und Hardware sind limitiert und müssen geplant werden

Risiko & Unsicherheit

Unbekannte Faktoren müssen identifiziert und gemanagt werden

☐ **Unser Durchlaufbeispiel:** Ein Kunde benötigt ein neues Firewall-System mit Einbau von drei Geräten in High-Availability-Konfiguration (Active/Passive). Das Projekt umfasst Planung, Installation, Konfiguration, umfangreiche Tests der Failover-Funktionalität sowie die vollständige Dokumentation und Übergabe an den Betrieb. Dieses realistische Szenario wird uns durch alle folgenden Themen begleiten.

Projektstrukturplan (PSP) – Planung in Arbeitspakete zerlegen

Zweck und Aufbau des PSP

Der Projektstrukturplan ist das zentrale Werkzeug zur Zerlegung eines Projekts in überschaubare, steuerbare Einheiten. Sein Hauptzweck besteht darin, den **Projektumfang sauber zu schneiden** und in **Arbeitspakete (AP)** mit konkreten **Ergebnissen (Deliverables)** zu gliedern. Der PSP ist **hierarchisch aufgebaut** (z.B. 1.0 → 1.1 → 1.1.1), bildet aber keine Zeitachse ab – das ist Aufgabe des Gantt-Diagramms.

Ein Arbeitspaket gilt als „fertig“, wenn sein **Ergebnis prüfbar vorliegt**. Beispiel: Das Arbeitspaket „HA-Failover-Test“ ist abgeschlossen, wenn der Testplan durchgeführt wurde, das Failover funktioniert hat und die Ergebnisse dokumentiert sind. Diese Prüfbarkeit ist entscheidend für die Projektsteuerung und Qualitätssicherung.

Praktisches Vorgehen

Beginnt immer mit der Frage: „Welche Hauptphasen hat mein Projekt?“ Im IT-Bereich sind das typischerweise: Analyse, Konzeption, Umsetzung, Test, Abschluss. Jede Hauptphase wird dann in konkrete Arbeitspakete unterteilt. Achtet darauf, dass jedes Arbeitspaket ein tangibles Ergebnis liefert – nicht nur eine Aktivität beschreibt.

□ **Arbeitsauftrag:** Formuliere für das Firewall-Projekt fünf Arbeitspakete mit jeweils einem messbaren Ergebnis. Achte darauf, dass jedes Paket eine klare Abgrenzung hat und das Ergebnis prüfbar ist (z.B. „Interface-Konfiguration dokumentiert“, nicht nur „Interfaces konfigurieren“).

Beispiel-PSP: Firewall-Projekt

1	Analyse <ul style="list-style-type: none"> 1.1 Ist-Aufnahme Netzwerk, Rack-Infrastruktur und Verkabelung 1.2 Anforderungsanalyse: DHCP/DNS/Proxy-Services, Security-Funktionen, Performance-Anforderungen
2	Konzeption <ul style="list-style-type: none"> 2.1 Zielbild für Interfaces, VLANs und Routing-Struktur 2.2 HA-Design inkl. Failover-Verkabelung und Synchronisations-Konzept 2.3 Security-Policy-Design (IPS, DoS-Protection, Application Control)
3	Umsetzung <ul style="list-style-type: none"> 3.1 Physischer Einbau, Verkabelung und Initial-Boot 3.2 Interface-Konfiguration, VLAN-Setup und Routing 3.3 Implementierung Security-Policies und Dienste 3.4 HA-Konfiguration und Cluster-Setup
4	Test & Validierung <ul style="list-style-type: none"> 4.1 Erstellung und Durchführung HA-Failover-Testplan 4.2 Performance- und Lasttests 4.3 Security-Policy-Validierung
5	Abschluss <ul style="list-style-type: none"> 5.1 Erstellung Betriebs- und Notfalldokumentation 5.2 Abnahme mit Kunde gemäß Abnahmekriterien 5.3 Übergabe an Betrieb inkl. Knowledge Transfer

Netzplan und Kritischer Weg – Dauer, Abhängigkeiten, Puffer

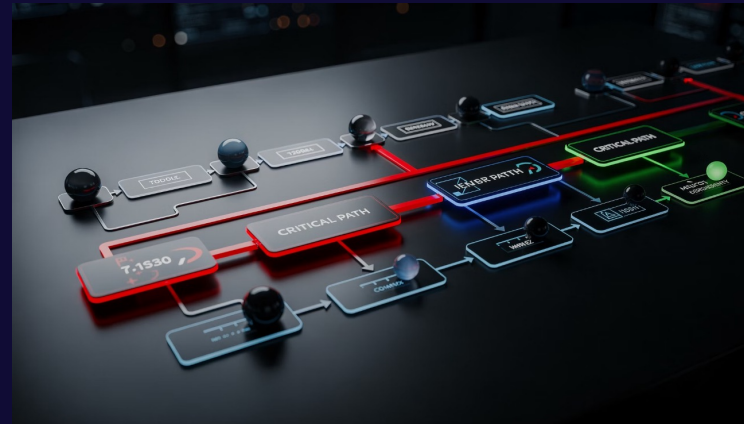
Während der Projektstrukturplan die *Was-Frage* beantwortet, klärt der Netzplan die *Wann-Frage*. Er zeigt die **Reihenfolge und Abhängigkeiten** zwischen Arbeitspaketen: Was muss vor was passieren? Welche Aktivitäten können parallel laufen? Wo entstehen Engpässe? Diese Informationen sind entscheidend für die realistische Terminplanung und Ressourcensteuerung.

Der Kritische Pfad

Der **Kritische Pfad** ist die längste Kette von abhängigen Aktivitäten durch das Projekt – jene Aktivitäten, die keinen zeitlichen Spielraum (Puffer) haben. Verzögert sich eine Aktivität auf dem kritischen Weg, verschiebt sich automatisch der gesamte Projektendtermin. Das macht diese Aktivitäten zu den risikoreichsten im Projekt und erfordert besondere Aufmerksamkeit.

Im Gegensatz dazu haben **nicht-kritische Aktivitäten** einen Puffer – sie können sich verzögern, ohne das Projektende zu gefährden. Man unterscheidet zwischen **Gesamtpuffer** (maximale Verzögerung ohne Projektende-Verschiebung) und **Freiem Puffer** (Verzögerung ohne Nachfolgeaktivitäten zu beeinflussen).

Typische Abhängigkeiten im Firewall-Fall



Prüfungstipp: Kritikalität begründen

In der Prüfung wird häufig gefragt: „Warum ist Aktivität X kritisch?“ Die korrekte Antwort folgt immer diesem Muster: „Aktivität X liegt auf dem kritischen Weg, weil sie **keinen Puffer hat**. Folgeaktivitäten blockiert und eine Verzögerung den Meilenstein Y verschiebt.“ Konkret für unser Firewall-Projekt: „Die HA-Verkabelung ist kritisch, weil sie keinen Puffer hat, die HA-Tests blockiert und eine Verzögerung den Abnahme-Meilenstein verschiebt.“

Arbeitsauftrag: Nenne drei Aktivitäten aus dem Firewall-Projekt, die realistisch auf dem kritischen Weg liegen würden, und begründe jeweils kurz, warum (Hinweis: Denke an Abhängigkeitsketten und Aktivitäten, die nicht parallel laufen können).

1

Rack & Verkabelung

Muss vor Boot/Inbetriebnahme abgeschlossen sein

2

Interface-Konfiguration

Voraussetzung für Security-Policies

3

HA-Setup

Verkabelung + Konfig vor HA-Tests

4

Tests durchführen

Alle Tests vor Abnahme

Gantt-Diagramm – Terminierung, Meilensteine, Überblick

Visualisierung mit dem Gantt-Diagramm

Das Gantt-Diagramm ist das meistgenutzte Werkzeug zur visuellen Darstellung von Projektzeitplänen. Es zeigt auf einer **Zeitachse** Start- und Enddaten, Dauern, Parallelitäten und Meilensteine. Im Gegensatz zum Netzplan, der Abhängigkeiten fokussiert, liegt die Stärke des Gantt-Diagramms in der intuitiven Darstellung des zeitlichen Ablaufs – wann passiert was, wie lange dauert es, und was läuft gleichzeitig?

Besonders wichtig sind **Meilensteine** – spezielle Markierungen mit Null-Dauer, die einen erreichten Projektzustand kennzeichnen. Ein Meilenstein ist **kein Arbeitspaket**, sondern ein Prüfpunkt, an dem etwas Wesentliches erreicht wurde. Die Verwechslung von Arbeitspaketen und Meilensteinen ist eine häufige Prüfungsfalle!

Meilenstein vs. Arbeitspaket

❏ **Arbeitsauftrag:** Definiere vier Meilensteine für das Firewall-Projekt und gib jeweils ein konkretes Kriterium an, woran man „erreicht“ eindeutig prüfen kann (z.B. „M2: Konzeption freigegeben“ → Kriterium: „Unterschrift Kunde auf Konzeptdokument vorhanden“).

Beispiel-Meilensteine: Firewall-Projekt

M1: Anforderungen bestätigt

Kriterium: Anforderungsdokument vom Kunden unterzeichnet, alle Security-Requirements definiert und priorisiert

M3: Hardware produktiv

Kriterium: Alle drei Firewall-Geräte eingebaut, verkabelt, mit Grundkonfiguration versehen und erfolgreich gebootet

M5: Projekt abgeschlossen

Kriterium: Kundenabnahme erfolgt, Betriebsdokumentation übergeben, System im Produktivbetrieb, Projektabschlussbericht erstellt

1

2

3

4

5

M2: Konzeption freigegeben

Kriterium: Netzwerkdesign, HA-Architektur und Security-Policies vom Kunden abgenommen, Go für Umsetzung erteilt

M4: HA erfolgreich validiert

Kriterium: Failover-Tests gemäß Testplan durchgeführt, alle Szenarien erfolgreich, Active/Passive-Betrieb stabil, Ergebnisse dokumentiert

❌ Falsch

„Meilenstein: Dokument schreiben“

Das ist ein Arbeitspaket, keine Zustandsänderung

✓ Richtig

„Meilenstein: Konzeption freigegeben“

Markiert einen erreichten Projektzustand

Projektphasen – was passiert wann?

Jedes Projekt durchläuft typischerweise definierte Phasen, die einen logischen Ablauf von der Idee bis zum Abschluss bilden. In der klassischen Projektmanagement-Theorie spricht man von **Initiierung** → **Planung** → **Durchführung/Steuerung** → **Abschluss**. In der IHK-Prüfung und in IT-Projekten wird dies meist praxisnäher formuliert: **Analyse** → **Konzept** → **Umsetzung** → **Test** → **Abnahme/Übergabe**.

Phasen und ihre Artefakte

Jede Phase hat spezifische Ziele und produziert definierte **Artefakte (Deliverables)** – dokumentierte Ergebnisse, die die Basis für die nächste Phase bilden und später die Projektarbeit nachweisen. Diese Artefakte sind in der Prüfung von zentraler Bedeutung, da sie zeigen, dass ihr systematisch und nachvollziehbar gearbeitet habt.

Beispiel-Deliverables zuordnen

Die korrekte Zuordnung von Artefakten zu Projektphasen ist eine beliebte Prüfungsaufgabe. Ihr müsst verstehen, in welcher Phase welches Dokument entsteht und warum.

- **Testplan** → **Test-Phase** (definiert, WAS und WIE getestet wird)
- **Abnahmeprotokoll** → **Abschluss-Phase** (dokumentiert erfolgreiche Übergabe)
- **Netzdesign** → **Konzept-Phase** (Soll-Architektur wird definiert)
- **Anforderungsliste** → **Analyse-Phase** (Was soll das System können?)
- **Betriebsdoku** → **Abschluss-Phase** (How-to für den laufenden Betrieb)

☐ **Arbeitsauftrag:** Ordne die folgenden Deliverables den korrekten Projektphasen zu und begründe kurz: Testplan, Abnahmeprotokoll, Netzdesign, Anforderungsliste, Betriebsdokumentation. Überlege dabei: Wann im Projektverlauf entsteht das Dokument logischerweise?

01

Analyse

Ist-Zustand erfassen, Anforderungen erheben

Artefakte: Anforderungskatalog, Ist-Analyse, Stakeholder-Übersicht

02

Konzept

Soll-Zustand definieren, Lösungsweg planen

Artefakte: Netzwerkdesign, Architektur-Dokument, Projektplan

03

Umsetzung

Lösung implementieren, konfigurieren

Artefakte: Konfigurationsdateien, Build-Dokumentation, Change-Protokolle

04

Test

Funktionalität prüfen, Qualität sichern

Artefakte: Testplan, Testergebnisse, Fehlerberichte

05

Abschluss

Übergeben, dokumentieren, abnehmen

Artefakte: Betriebsdoku, Abnahmeprotokoll, Projektabschlussbericht

Vorgehensmodelle – welches passt wann?

Die Wahl des richtigen Vorgehensmodells ist eine strategische Entscheidung, die den Projekterfolg maßgeblich beeinflusst. Es gibt kein „bestes“ Modell für alle Situationen – die Kunst liegt darin, das Modell an die spezifischen Rahmenbedingungen anzupassen. In der AP1-Prüfung müsst ihr die Charakteristika der wichtigsten Modelle kennen und begründen können, wann welches Modell sinnvoll ist.



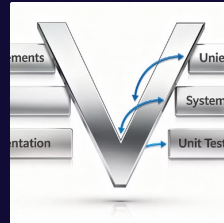
Wasserfall (Klassisch)

Prinzip: Sequenzielle Phasen, eine nach der anderen. Jede Phase muss vollständig abgeschlossen sein, bevor die nächste beginnt.

Stärken: Klare Struktur, planbar, gut dokumentierbar, hohe Transparenz für Stakeholder

Schwächen: Unflexibel bei Änderungen, späte Fehler sind teuer, Kunde sieht Ergebnis erst spät

Passt bei: Stablen Anforderungen, hohem Dokumentationsbedarf, regulierten Umfeldern



V-Modell (Vereinfacht)

Prinzip: Erweiterung des Wasserfalls – jeder Planungsschritt links hat einen entsprechenden Testschritt rechts gegenüber.

Stärken: Systematische Qualitätssicherung, zu jedem Design-Schritt passende Teststufe, Fehler werden früh erkannt

Schwächen: Ähnlich unflexibel wie Wasserfall, aufwendig in der Planung

Passt bei: Projekten mit hohen Qualitätsanforderungen, klaren Abnahmekriterien, kritischer Infrastruktur



Agil (Scrum/Kanban)

Prinzip: Iterative Entwicklung in kurzen Zyklen (Sprints), kontinuierliches Feedback, anpassungsfähig

Stärken: Flexibel bei unklaren/wechselnden Anforderungen, schnelles Feedback, frühe Wertschöpfung

Schwächen: Benötigt aktiven Kunden/Product Owner, Disziplin im Team, schwerer planbar für Management

Passt bei: Unscharfen Anforderungen, innovativen Projekten, wenn schnelles Feedback wichtig ist

AP1-Entscheidungslogik (Kurzregel)

Anforderungen stabil?

Sind die Anforderungen klar, vollständig und ändern sich voraussichtlich nicht? → **Eher klassisch/V-Modell**

Dokumentation wichtig?

Gibt es hohe Doku-/Abnahmeanforderungen (z.B. regulierte Branchen, Audits)? → **Eher klassisch/V-Modell**

Änderungsrisiko hoch?

Sind Anforderungen unklar oder erwartbar dynamisch? → **Eher agil**

Entscheidung für das Firewall-Projekt

Für unser Firewall-Projekt würde ich ein **klassisches/V-Modell-Vorgehen** empfehlen. Begründung:

- Stabile Anforderungen:** Die Anforderungen an eine Firewall-Installation sind typischerweise zu Projektbeginn klar definierbar – welche Interfaces, welche VLANs, welche Security-Policies benötigt werden, ändert sich während der Implementierung kaum.
- Hoher Abnahme-/Dokumentationsbedarf:** Security-Infrastruktur erfordert detaillierte Dokumentation für Audits, Compliance und Betrieb. Jeder Schritt muss nachvollziehbar und testbar sein. Das V-Modell mit seinen systematischen Teststufen passt perfekt zu den notwendigen HA-Tests, Security-Validierungen und Abnahmetests.

Arbeitsauftrag: Wähle ein Vorgehensmodell für das Firewall-Projekt und gib zwei konkrete Gründe an, die sich auf Anforderungsstabilität, Abnahmebedarf oder Änderungsrisiko beziehen.

Teambuilding und -entwicklung – warum Teams nicht sofort „laufen“

Ein häufiger Irrtum in Projekten: „Wir haben gute Leute, also läuft das Team automatisch.“ Die Realität ist komplexer. Teams durchlaufen vorhersagbare Entwicklungsphasen, und das Verständnis dieser Phasen hilft, typische Probleme zu antizipieren und gezielt zu steuern. Das **Tuckman-Modell** ist das prüfungsrelevante Standardmodell zur Teamentwicklung.

Die Tuckman-Phasen

Typische Stolpersteine

- **Unklare Rollen:** Wer ist wofür verantwortlich? Wer entscheidet was?
- **Fehlende Regeln:** Wie kommunizieren wir? Wie treffen wir Entscheidungen?
- **Kommunikationslücken:** Informationen kommen nicht an, Missverständnisse häufen sich
- **Unterschiedliche Erwartungen:** Was bedeutet „fertig“? Welche Qualität erwarten wir?

Praktische Rollen im IT-Projekt

Ohne in Overkill zu verfallen, braucht jedes Projekt klare Rollenverteilungen. Für unser Firewall-Projekt würden typischerweise folgende Rollen sinnvoll sein:

Storming meistern: Konflikte produktiv nutzen

Die Storming-Phase ist unvermeidlich und sogar notwendig – hier werden unterschiedliche Perspektiven sichtbar und im besten Fall zu besseren Lösungen integriert. Entscheidend ist, wie ihr damit umgeht.

- **Arbeitsauftrag:** Nenne zwei typische Konflikte, die in der Storming-Phase eines Firewall-Projekts auftreten könnten (z.B. unterschiedliche Meinungen zum HA-Design, Zeitdruck vs. Qualitätsanspruch), und beschreibe je eine konkrete Maßnahme, wie du sie einfangen würdest (z.B. strukturierte Entscheidungsfindung, Rollen klären, Eskalationsprozess).



Forming (Orientierung)

Team kommt zusammen, höflich und vorsichtig, Rollen unklar, hohe Abhängigkeit vom Projektleiter



Storming (Konflikt)

Konflikte entstehen, Machtkämpfe, unterschiedliche Arbeitsweisen prallen aufeinander – kritische Phase!



Norming (Regelbildung)

Team entwickelt gemeinsame Regeln, Rollen klären sich, Zusammenarbeit wird effektiver



Performing (Leistung)

Team arbeitet hochproduktiv, selbstorganisiert, Fokus auf Zielerreichung, gegenseitige Unterstützung



Adjourning (Auflösung)

Projekt endet, Team löst sich auf – wichtig für emotionalen Abschluss und Lessons Learned

Projektleitung

Verantwortlich für Plan, Status-Reporting, Risikomanagement, Stakeholder-Kommunikation, Ressourcensteuerung

Technical Lead

Technische Entscheidungen, Architektur-Design, Code-/Konfig-Reviews, technische Eskalationen klären

Umsetzung/Build

Implementierung der Konfiguration, Installation, Verkabelung, Setup gemäß Konzept

Test & Dokumentation

Testplanung und -durchführung, Erstellung von Betriebs- und Projektdokumentation, Nachweise sammeln

Reflexionsmethoden – damit Fehler nicht recyceln wie ein schlechter Meme

Projekte sind Lernchancen – aber nur, wenn wir systematisch reflektieren, was gut und was schlecht lief. Ohne strukturierte Reflexion wiederholen wir dieselben Fehler in jedem neuen Projekt. In der AP1-Prüfung wird erwartet, dass ihr Reflexionsmethoden kennt und anwenden könnt. Der Schlüssel: Reflexion muss **konkret und umsetzbar** sein, nicht vage und allgemein.

Bewährte Reflexionsmethoden



Lessons Learned (Projektende)

Strukturierte Reflexion am Projektende: Was lief gut? Was lief schlecht? Was ändern wir beim nächsten Mal? Wichtig: Konkrete Maßnahmen formulieren, nicht nur Probleme benennen.

Beispiel: „Problem: Testplan zu spät erstellt, Zeitdruck in Testphase“ → **Maßnahme:** „Nächstes Projekt: Testplan parallel zur Konzeptphase starten, Review vor Umsetzungsbeginn“



Retrospektive (Agil/Iterativ)

Regelmäßige, kurze Reflexion nach jedem Sprint oder in festgelegten Intervallen. Fokus auf kontinuierliche Verbesserung der Zusammenarbeit und Prozesse.

Typische Fragen: Was sollen wir beibehalten? Was sollen wir ändern? Was sollen wir ausprobieren?



After Action Review (AAR)

Ursprünglich aus