

07

SYSTEMATISCHE QUALITÄTSSICHERUNG VON ANFORDERUNGEN, ARCHITEKTUREN UND PROZESSEN

- Aktivitäten zur Qualitätssicherung von Anforderungen
- Qualität von Architekturen
- Reifegrade für Softwareprozesse

- „Im Mittelpunkt des umfassenden Qualitätsmanagements stehen die:
 - Kunden mit ihren Aufgaben, Prozessen, Erfordernissen und Erwartungen und
 - das Bemühen der Lieferanten, mit ihren Leistungen stets zu entsprechen.“

- „Qualitätsmanagement wird nach ISO 9000:2005 als ‚aufeinander abgestimmte Tätigkeiten zum leiten und lenken einer Organisation bezüglich Qualität‘ definiert.“
- „Qualitätsmanagement dient dabei als Oberbegriff für eine Vielzahl von Managementaufgaben und –aktivitäten zur Sicherstellung von Qualität in Prozessen, Projekten und Produkten.“

- „Norm zum Thema Qualitätsmanagement“
- „DIN ISO 9000-Reihe. Diese Normen sind auf eine Fülle von stark unterschiedlichen Unternehmen anwendbar, und können daher keine spezifischen Techniken fordern.“
- ISO 9000: Diese Norm definiert Grundlagen und Begriffe zu Qualitätsmanagementsystemen.
- ISO 9001 legt die Mindestanforderungen an ein Qualitätsmanagementsystem (QM-System)

- „ISO 9001 ist im Wesentlichen frei von konkreten technischen Inhalten. Sie fordert die Durchführung bestimmter Tätigkeiten, z. B. die Beurteilung von Unterlieferanten oder die Existenz von Eingangsprüfungen. Sie legt nicht fest, wie diese Tätigkeiten zu realisieren sind. Im Wesentlichen definiert die DIN EN ISO 9001 einen organisatorischen Rahmen für das Qualitätsmanagement.“
- ISO 9004 → Leitfaden für die Implementierung von QM bzw. TQM

- Bekannte ISO Norm zur Bewertung von Softwarequalität:
 - Funktionalität: Erfüllt die Software die geforderten Funktionen?
 - Zuverlässigkeit: Kann sie ein bestimmtes Leistungsniveau über Zeit aufrechterhalten?
 - Benutzbarkeit: Wie aufwändig ist der Einsatz für den Benutzer?
 - Effizienz: Wie ist das Verhältnis von Leistung zu eingesetzten Betriebsmitteln?
 - Änderbarkeit/Wartbarkeit: Welchen Aufwand erfordern Änderungen an der Software?
 - Übertragbarkeit: Wie leicht lässt sich die Software in eine andere Umgebung übertragen?

- Die ISO 9126 ist in der neueren ISO/IEC 25000-Normenreihe aufgegangen.
- Erweiterungen und Neuerungen in ISO 25000/2500n-2504n
 - Einführung eines allgemeinen Referenzmodells
 - Einführung von Leitfäden für jede Teilnorm
 - Einbeziehung der Systemproduktqualität neben der Softwarequalität
 - Einführung eines Datenqualitätsmodells
 - Einführung von "Quality Measure Elements" in der Qualitätsmessungs-Teilnorm
 - Einführung einer Teilnorm für Qualitätsanforderungen
 - Überarbeitung und Einbindung der Evaluierungsprozesse
 - Praxisbeispiele zur Anwendung der Normen
 - Harmonisierung mit ISO/IEC 15939 (Software Measurement Process)

- Die ISO 9126 ist in der neueren ISO/IEC 25000-Normenreihe aufgegangen.
- Neue Qualitätsmerkmale:
 - Sicherheit und Kompatibilität als neue Hauptmerkmale hinzugefügt
 - Funktionalität umbenannt in "funktionale Eignung"
 - Effizienz umbenannt in "Leistungseffizienz"
 - Gebrauchstauglichkeit erweitert um "Bedienbarkeit" und "Zugänglichkeit"

- „[Heute] geht man zunehmend dazu über, die Qualität des Software Produktes indirekt über die Qualität des Entwicklungsprozesses zu verbessern. Prüfungen und Tests sind dann Teilaufgaben dieses Prozesses und als solche nur Teil des gesamten Qualitätsmanagements. Dieser Ansatz basiert auf der Hypothese, dass man durch einen qualitativ hochwertigen, reifen Prozess auch qualitativ hochwertige Produkte herstellen kann.“

In der Praxis kann eine Aufgabenverteilung wie folgt organisiert sein:

- Softwareentwickler/innen: Zuständig für den Aufbau der Softwarelösung; Sicherstellung der Qualität der Komponente, für die sie verantwortlich sind
- Software-Tester: Bewertet die Qualität der Software und verringert Risiko
- Projektleiter/in / Manager/in: Verwaltet Softwareprojekte & leitet Softwareabteilungen

Reflexion für euer Projekt (Blitzlicht): Wer übernimmt bei euch das Qualitätsmanagement?

Beachten: Es besteht eine Verbindung zwischen Qualitätsgrad (Wie hoch ist meine Qualität) und den Kosten, die für die Erstellung oder Erbringung notwendig sind.

- Zur Erbringung von Qualität werden Prüf- und Vorbeugekosten in Kauf genommen.
- Für Abweichungen in der Qualität werden Fehlerkosten in Kauf genommen.

- „Im Schadensfall ergibt sich z. B. aus dem Produkthaftungsgesetz die Verpflichtung Ersatz eines Schadens, der durch ein fehlerhaftes Produkt entstanden ist.“
- „Ein Haftungsausschluss setzt voraus, dass der Fehler zum Zeitpunkt des in Verkehr bringen noch nicht vorlag, oder dass er nach dem Stand von Wissenschaft und Technik nicht erkennbar war, Die Beweislast für diesen Sachverhalt liegt im wesentlichen beim Hersteller.“

- Softwarefehler können Schadensersatzforderungen nach sich ziehen
- „Besonders Firmen, die sicherheitskritische Systeme entwickeln, müssen im Schadensfall erhebliche juristische Konsequenzen fürchten.“
- Der Softwarehersteller trägt die Beweislast, nachzuweisen, dass sein Produkt fehlerfrei geliefert wurde oder ein Fehler nicht hätte erkannt werden können.

- „Durch Einhaltung der jeweils relevanten Normen kann ein Hersteller sicherstellen, dass der Stand der Technik erreicht ist, und er damit seine Sorgfaltspflicht erfüllt hat.“

Kernaktivität:

- Requirements Engineering (Anforderungsmanagement), auch entwicklungsbegleitend
- fachliche und technische Korrektheit
- Abstimmung mit allen relevanten Stakeholdern
- Achtung: Für Anforderungen gilt → vollständiges Testen ist nicht möglich

- Was hat Produktqualität mit Prozessqualität zu tun (Insbesondere bei Software)?
- Was sind eure Erkenntnisse aus den Unternehmen?
- Was sind eure Erkenntnisse aus euren Projekten?

20 Minuten Mixed-Groups → Danach Ergebnisübersicht der einzelnen Gruppen (Findings)

REMINDER: QUALITÄTSSICHERUNG VON ANFORDERUNGEN

Prüfkriterien festlegen



Prüfprinzipien und Prüftechniken auswählen



Prüfungen durchführen und Ergebnisse dokumentieren



Abstimmen der Anforderungen / Konfliktmanagement

REMINDER: QUALITÄTSSICHERUNG VON ANFORDERUNGEN

Prüfkriterien für Anforderungen

Prüfen des Inhalts

Prüfen der Dokumentation

Prüfkriterien zur Abgestimmtheit

Prüfprinzipien

Beteiligung der richtigen Stakeholder

Trennung von Fehlersuche und Fehlerkorrektur

Prüfung aus unterschiedlicher Sicht

Geeigneter Wechsel von
Entwicklungsartefakten

Konstruktion von Entwicklungsartefakten

Wiederholte Prüfung

Was versteht man noch einmal unter Softwarearchitektur?

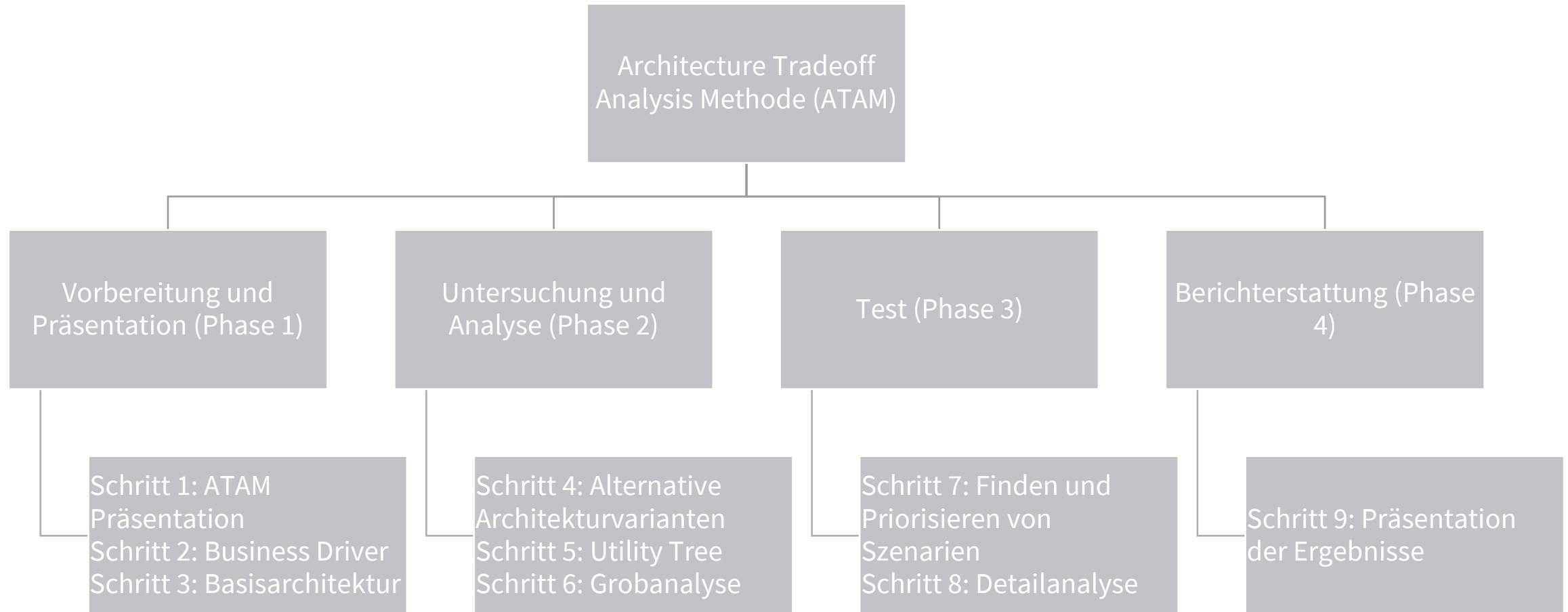
„Eine Softwarearchitektur ist einer der Architekturtypen in der Informatik und beschreibt die grundlegenden Komponenten und deren Zusammenspiel innerhalb eines Softwaresystems.“

Prüfung, der Eignung der Architektur zur Erfüllung der Anforderungen

Ex ante oder ex Post?

- **Ex Ante (vor der Implementierung):** Szenariobasiertes Vorgehen: Bewertung anhand von definierten Anwendungsszenarien => Architecture Trade off Analysis Methode (ATAM)
- **Ex Post (nach der Implementierung):** Prüfen der Implementierung gegen Architekturbeschreibung (Architecture Compliance Checking)

- Szenariobasiertes Vorgehen, d.h. Bewertung anhand von definierten Anwendungsszenarien (Ex ante)
 - Erkennt Trends
 - Früh anwendbar
- “When evaluating an architecture using the ATAM, the goal is to understand the consequences of architectural decisions with respect to the quality attribute requirements of the system.”
(inkl. des Trade-Offs der Ziele Architekturentscheidungen)

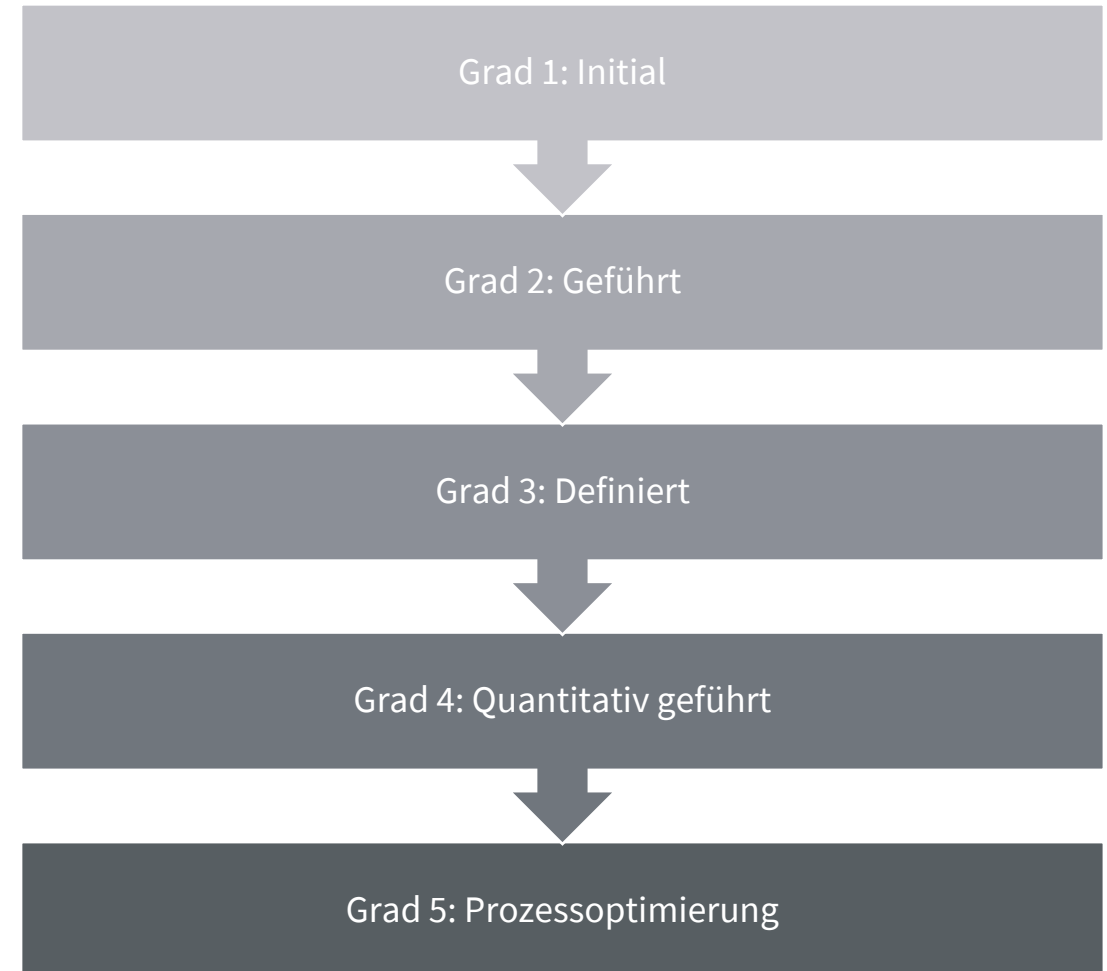


- Durch Stakeholderbefragung mit Hilfe von Prozessmerkmalen
- Reifegradmodelle z.B. Capability Maturity Model Integration

- Verständlichkeit
- Standardisierung
- Sichtbarkeit
- Messbarkeit
- Unterstützbarkeit
- Akzeptanz
- Zuverlässigkeit
- Stabilität
- Wartungsfreundlichkeit
- Schnelligkeit

QUALITÄTSSICHERUNG: CAPABILITY MATURITY MODEL INTEGRATION

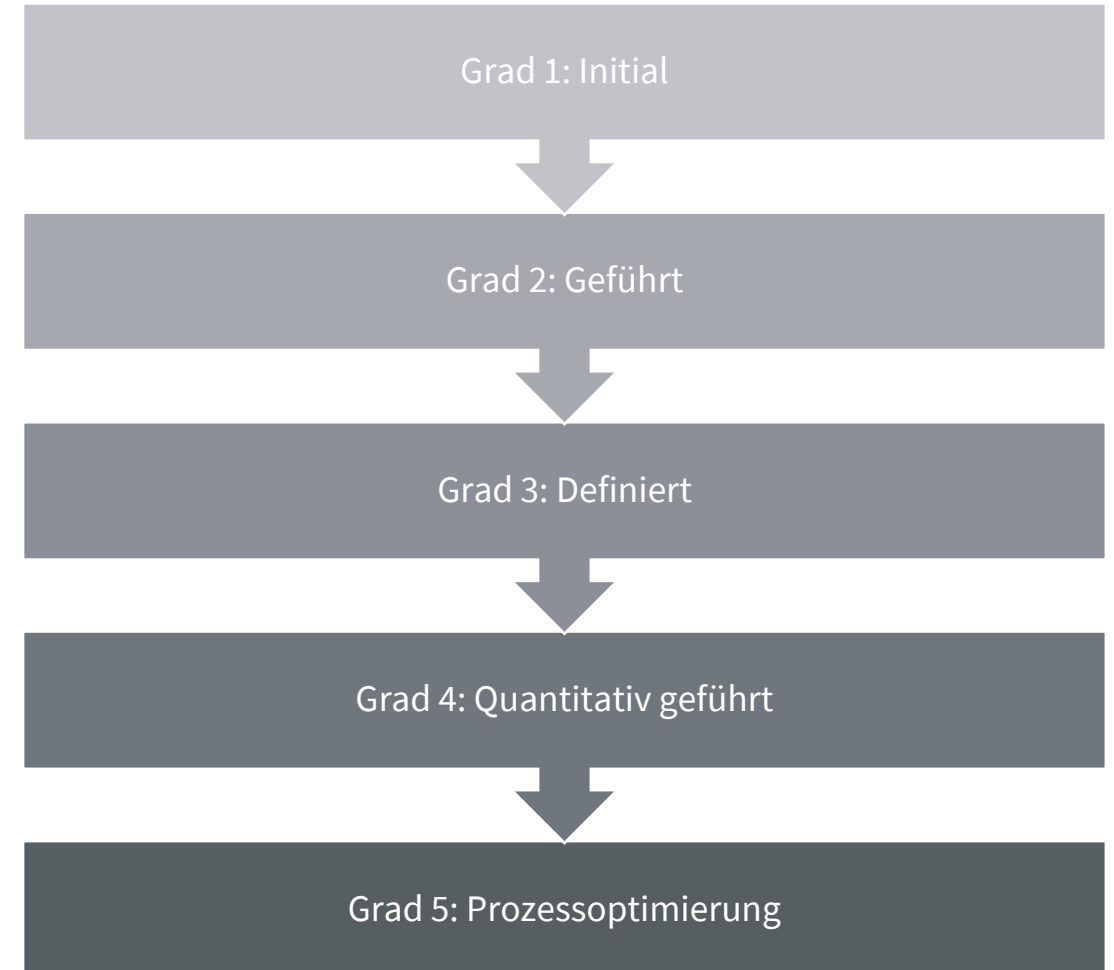
- Rahmenwerk für Prozessverbesserungen
- Stufenmodell
- 22 Prozessgebiete aus dem Software Engineering
- Ziele, die spezifisch für jedes Prozessgebiet formuliert werden; sowie
- Vorgehensweisen, mit denen die Ziele erreicht werden können.



QUALITÄTSSICHERUNG: CAPABILITY MATURITY MODEL INTEGRATION

Stufe 1 –Initial:

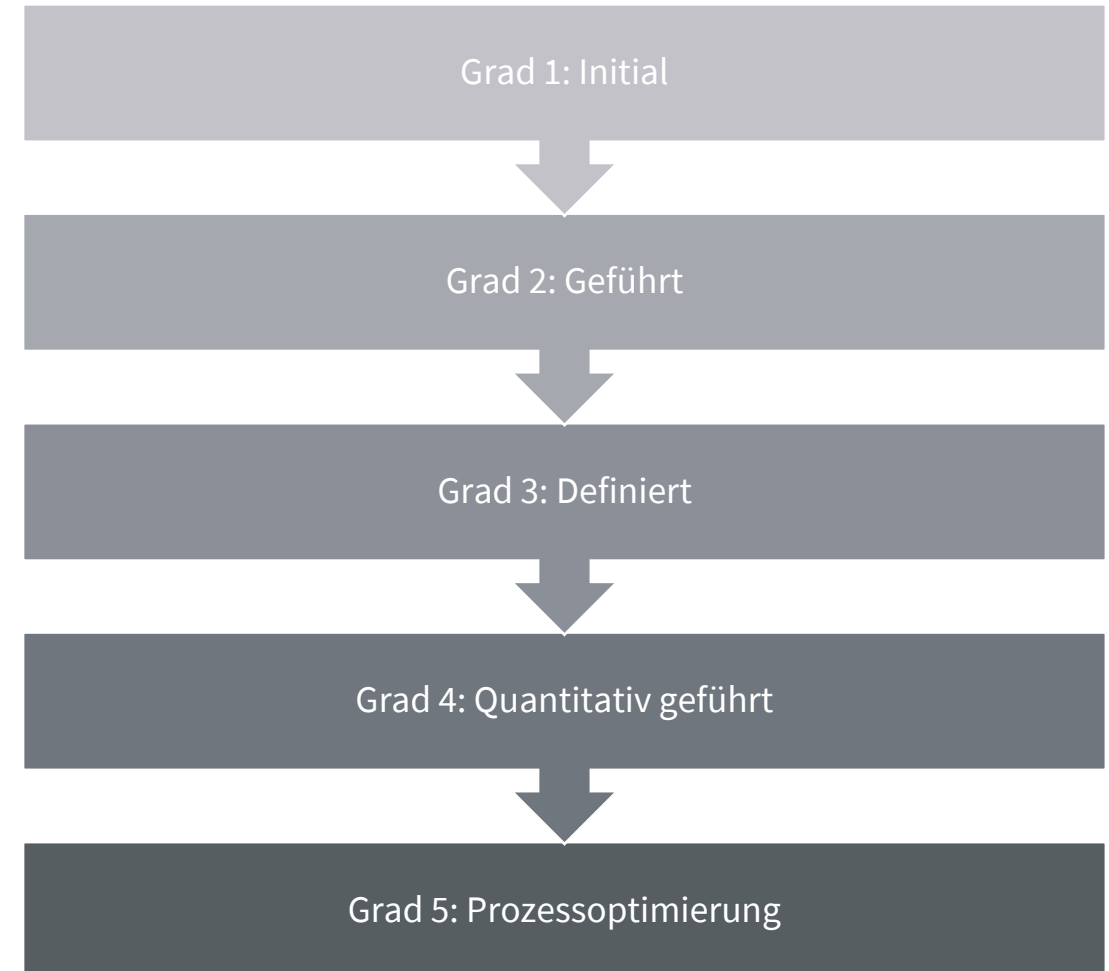
- Es gibt keine Prozessdefinition, die Aktivitäten finden ad hoc statt bzw. werden chaotisch durchgeführt. Der Prozess ist in der Regel nicht wiederholbar.



QUALITÄTSSICHERUNG: CAPABILITY MATURITY MODEL INTEGRATION

Stufe 2 –Geführt:

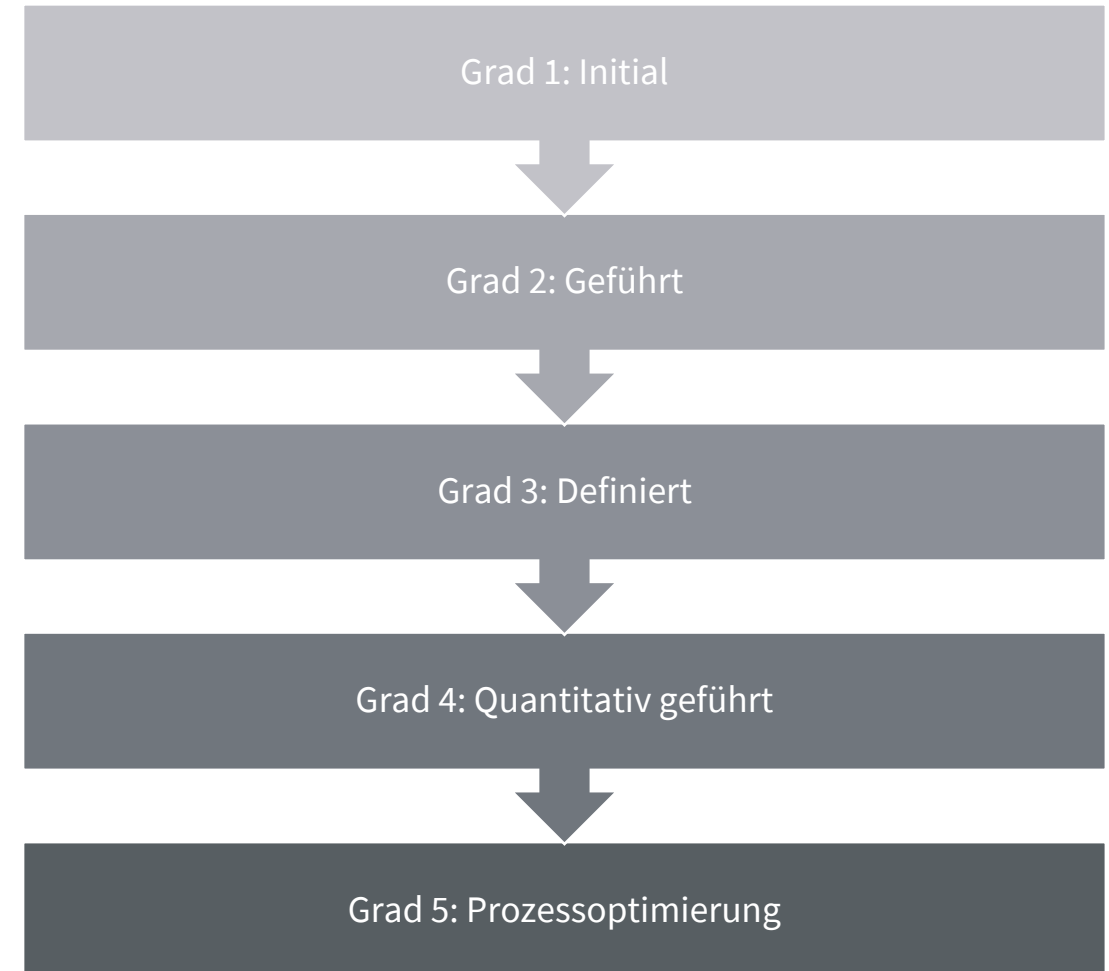
- Der Prozess ist wiederholbar. Umfang und Ergebnis der zu erledigenden Aktivitäten ist den Teammitgliedern bekannt. Es gibt aber keinen detaillierten Prozess, der Aktivitäten, Rollen, Ergebnisse und deren Abhängigkeiten explizit definiert.



QUALITÄTSSICHERUNG: CAPABILITY MATURITY MODEL INTEGRATION

Stufe 3 –Definiert:

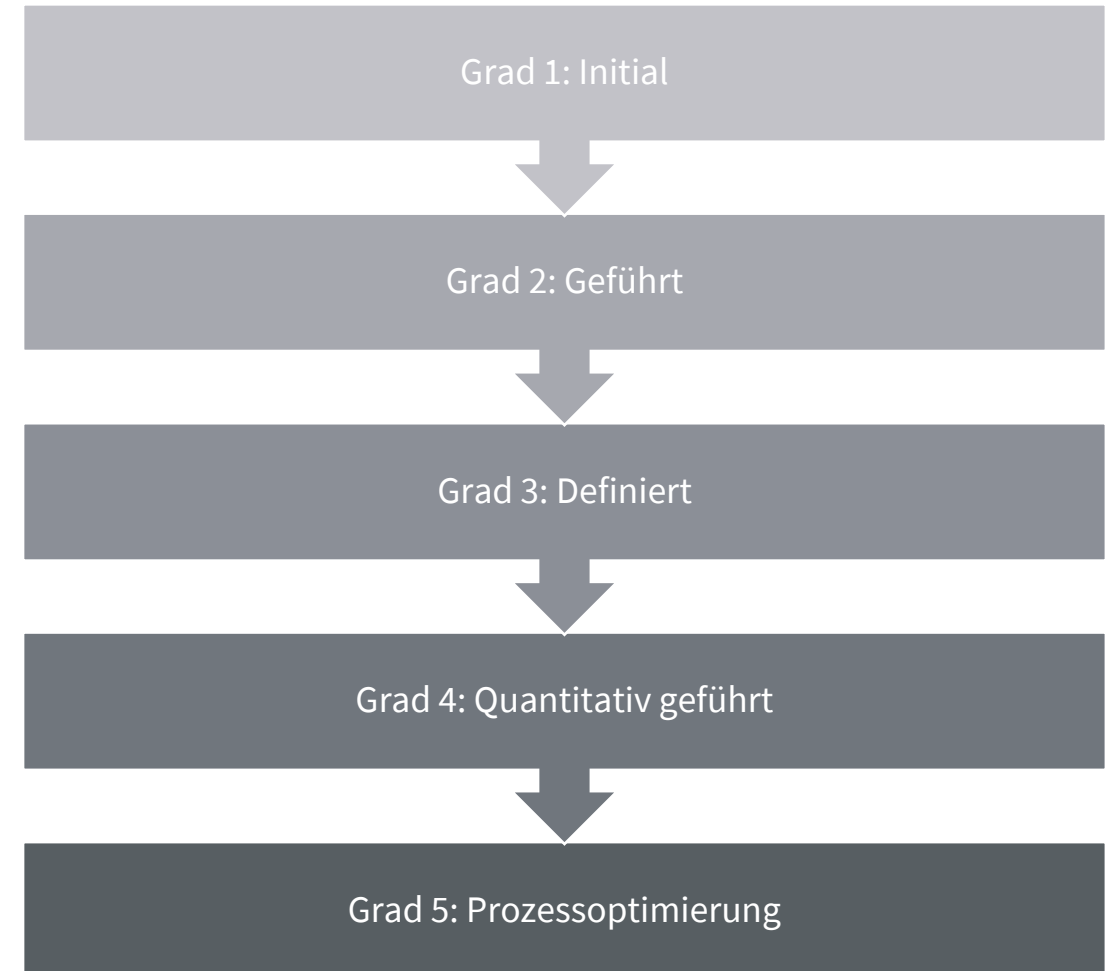
- Es gibt einen definierten Prozess, in dem Aktivitäten, Rollen und Ergebnisse verbindlich dokumentiert sind. Der Prozess enthält konkrete Methoden und Vorgehensweisen zu den einzelnen Software Engineering Aktivitäten.



QUALITÄTSSICHERUNG: CAPABILITY MATURITY MODEL INTEGRATION

Stufe 4 –Quantitativ geführt:

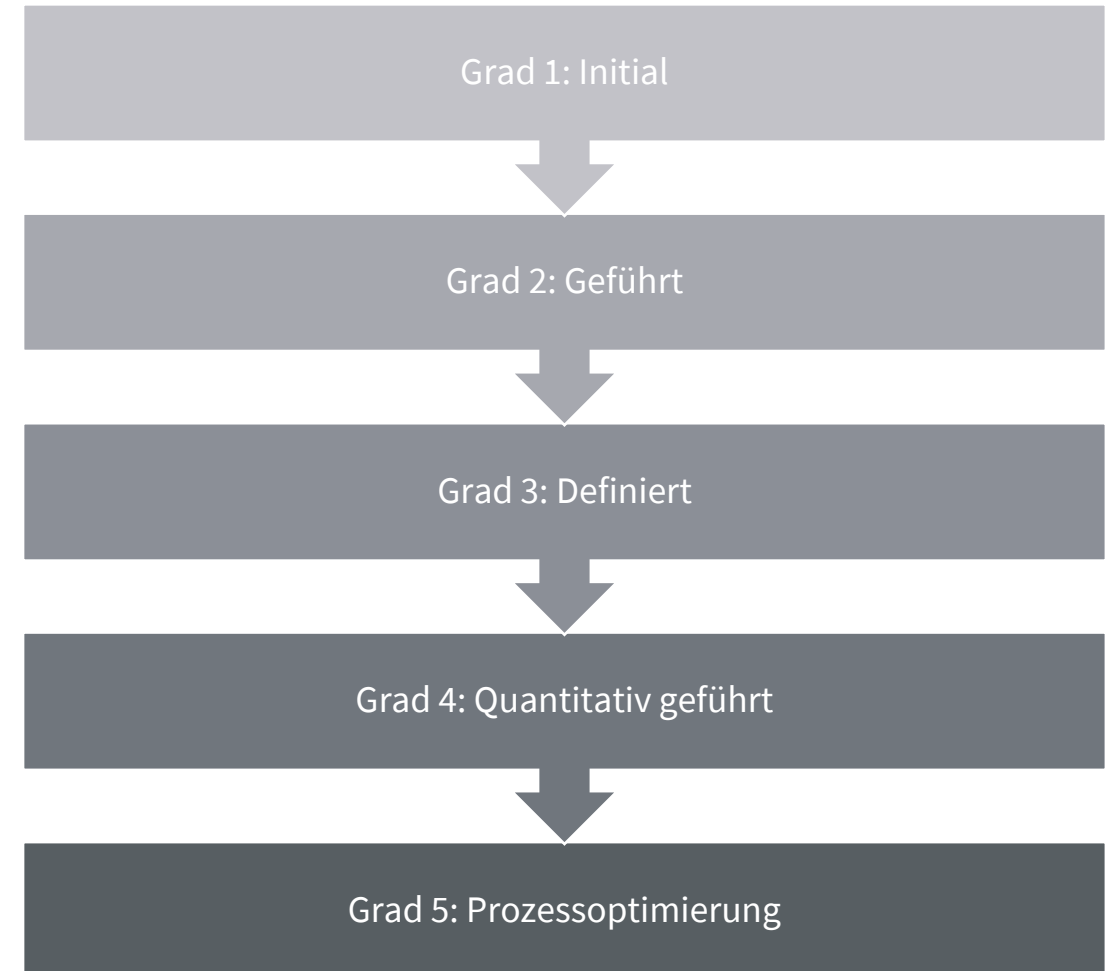
- Die Prozessqualität kann über Messung quantitativer Prozesseigenschaften festgestellt und gesteuert werden. Über erhobene Kennzahlen zu Aktivitäten der Prozesse sowie zu den erzeugten Ergebnissen können Havarien identifiziert und entsprechend gegengesteuert werden.

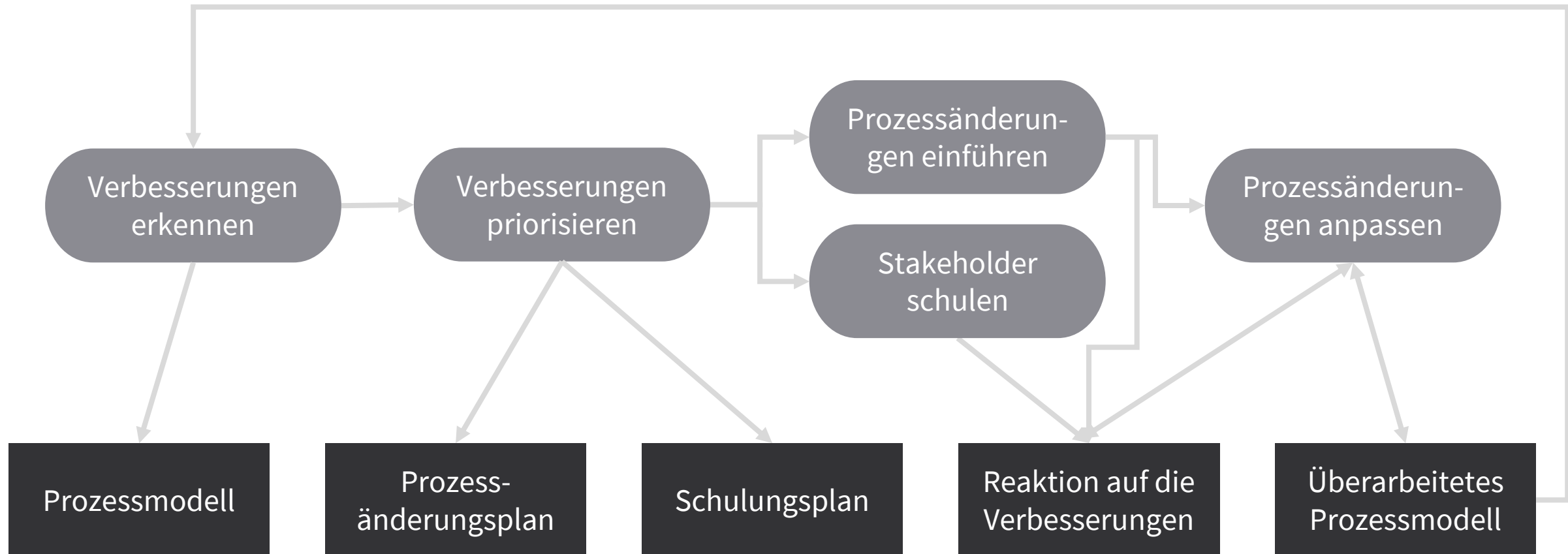


QUALITÄTSSICHERUNG: CAPABILITY MATURITY MODEL INTEGRATION

Stufe 5 – Prozessoptimierung:

- Prozess- und Produktmessungen werden kontinuierlich erhoben und zur Prozessverbesserung eingesetzt. Je nach Messwert kann der Prozess auf den aktuellen Bedarf hin angepasst werden.





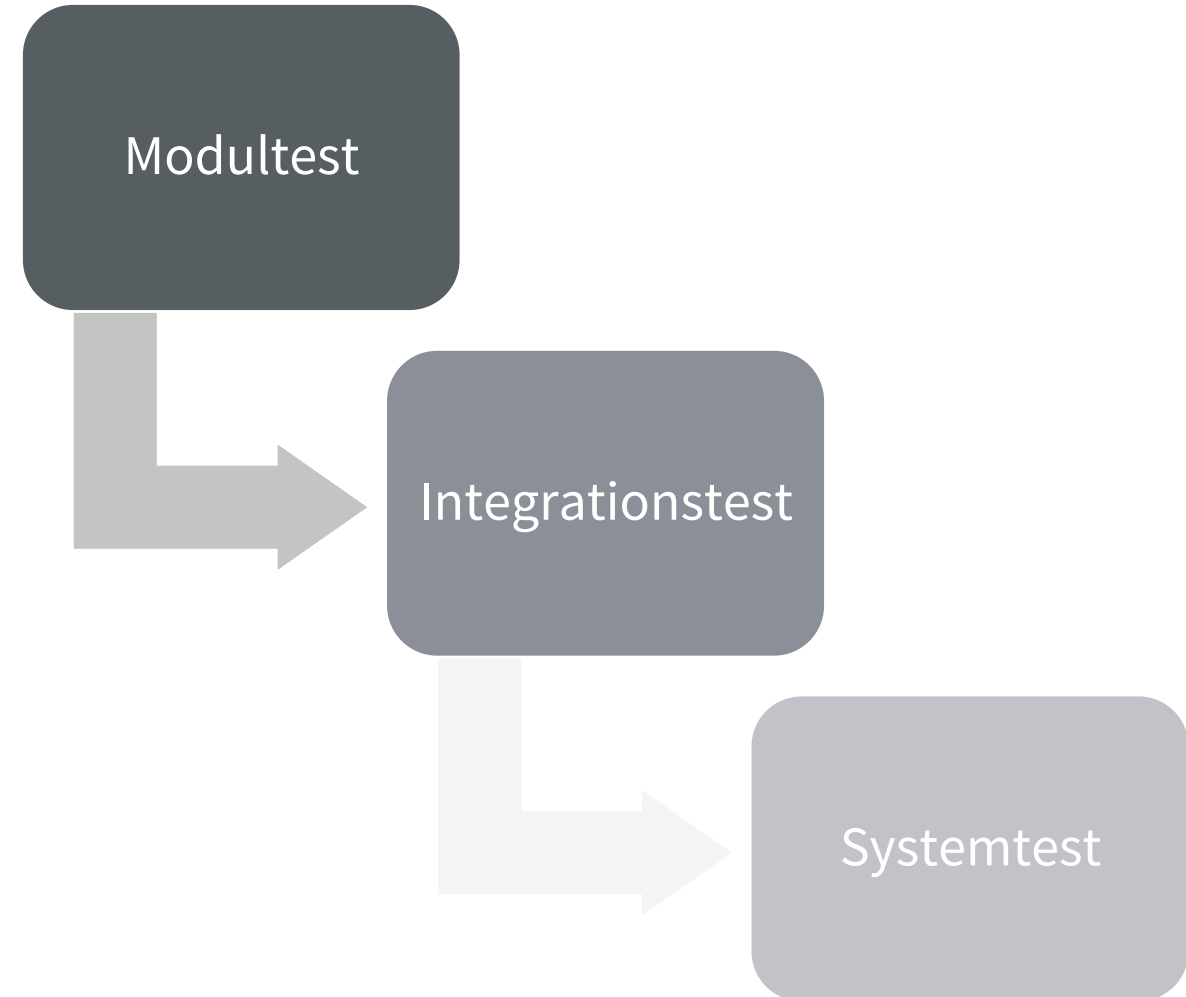
Gruppendiskurs/Reflektion:

- Warum ist es nicht möglich Anforderungen komplett zu testen?
- Welches sind vernünftige Größen von Software- oder Architekturelementen zum testen?

REMINDER: ARTEN DER QUALITÄTSSICHERUNG IN DER SOFTWAREENTWICKLUNG

Modultest:

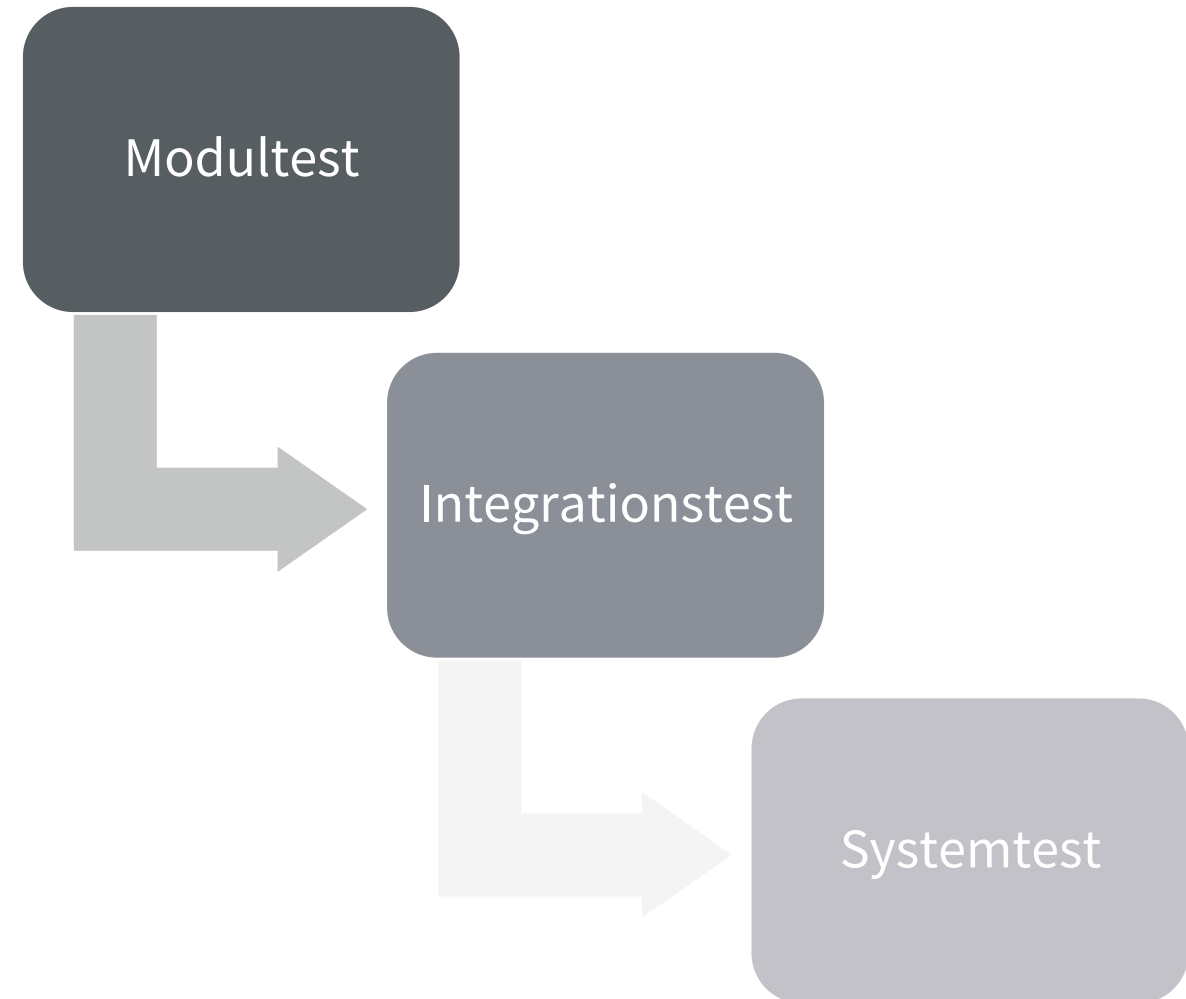
„Das Ziel des Modultests ist die Überprüfung einzelner Komponenten (Module) gegen ihre Spezifikation.“



REMINDER: ARTEN DER QUALITÄTSSICHERUNG IN DER SOFTWAREENTWICKLUNG

Integrationstest:

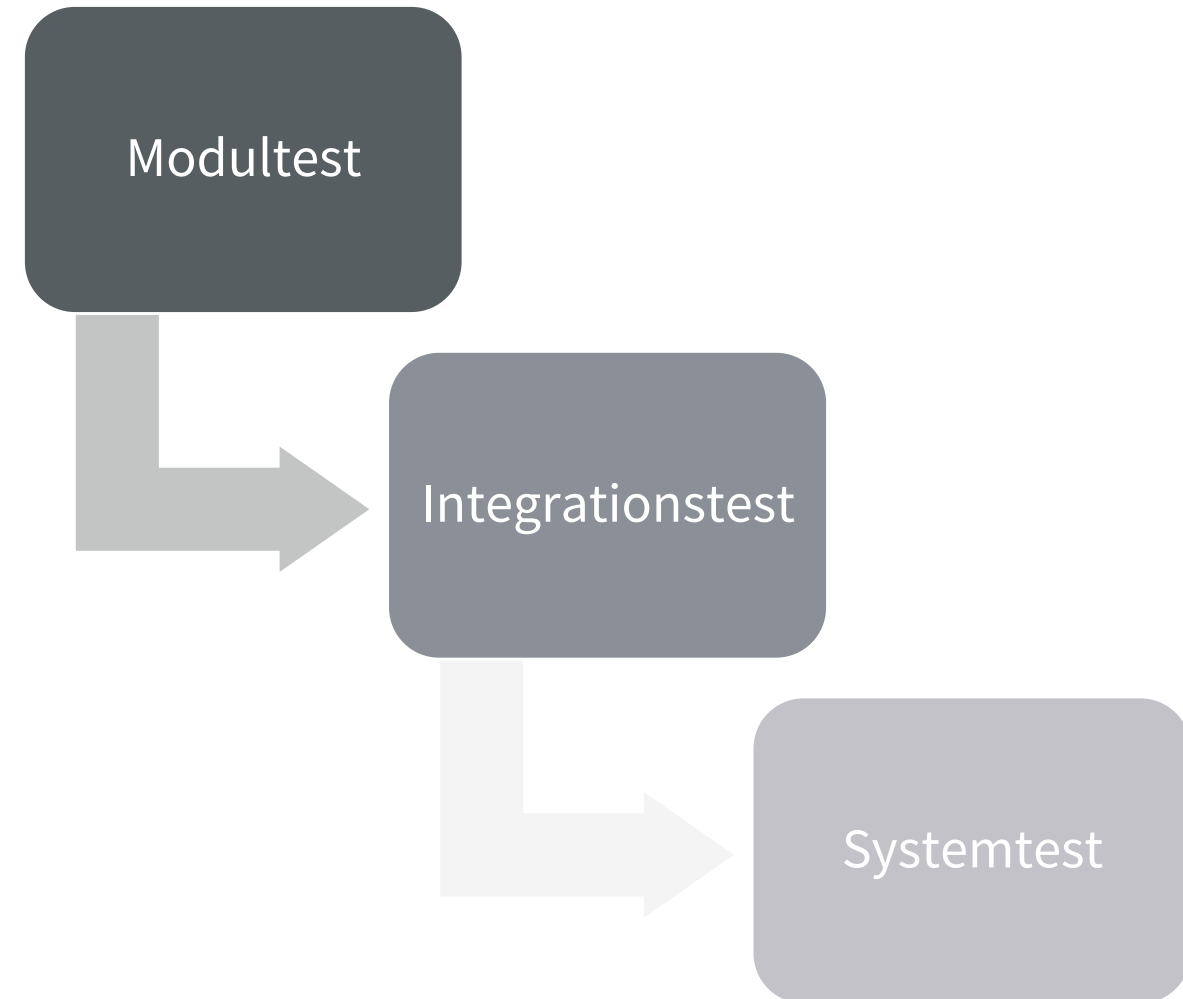
„Das Ziel des Integrationstests ist die Überprüfung des Interagierens verschiedener Module über Schnittstellen. In dieser Phase sind der innere Aufbau von Modulen und die von der derzeit betrachteten Schnittstelle weit entfernten Schnittstellen nicht relevant.“



REMINDER: ARTEN DER QUALITÄTSSICHERUNG IN DER SOFTWAREENTWICKLUNG

Systemtest:

„Die Ziele des Systemtests sind die Prüfung der Funktion, Leistung und Qualität des fertig integrierten Systems gegen die Anforderungsdokumente. Im Systemtest werden die inneren Strukturen von Systemen nicht betrachtet.“



BONUS: CODE-REVIEW-ÜBUNG

```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    List<Product> products;
    int input = 0; int i = 0;
    do {
        i++;
        System.out.println("Was möchten Sie tun? 1 = Produkt anlegen,
        2 = kommentieren, 3 = beenden");
        input = scan.nextInt();
        if(input == 1) {
            System.out.println("Name: ");
            String n = scan.next();
            products.add(new Product(i, n, ""));
        }
        if(input == 2) {
            System.out.println("Welches Produkt? (ID)");
            long id = scan.nextLong();
            Product p = getProduct(products, id);
            System.out.println("Kommentar: ");
            String c = scan.next();
            p.comments.add(new Comment(c));
        }
    }while(input != 3);
}
```

Comment.java

```
public class Comment {
    public Comment(String c) {
        comment = c;
    }
    public long ProduktID;
    public long commentID;
    public String comment;
    public String autor;
}
```

Product.java

```
public class Product {
    private long id;
    private String name;
    private String description;
    public List<Comment> comments;
    public Product(long pId, String pName, String description) {
        setId(pId);
        name = pName;
        description = description;
    }
    public void setId(long id) {
        this.id = id;
    }
}
```

BONUS: CODE-REVIEW-ÜBUNG – GRUPPENARBEIT (40 MINUTEN)

1. Führt eine Inspektion/Walkthrough in einer 4-Personen-Gruppe mit dem Quelltext durch: **Eine Person übernimmt jeweils eine der folgenden Rollen:**

- Moderator,
- Gutachter,
- Autor des Quellcodes,
- Protokollführer

2. Lass im Anschluss Chat GPT o. ä. Systeme den Quellcode prüfen und vergleicht die Ergebnisse mit euren Ergebnissen

DANKE

IU International Hochschule
Duales Studium
Siemensstraße 10
30173 Hannover

Prof. Dr. Knut Linke

 0511 310109 35

 knut.linke@iu.org