

Produktanforderungen

Unter dem Begriff Produktanforderungen (bzw. einer Anforderungsanalyse) geht es um die strukturierte und geplante Erfassung der Wünsche der Kunden auf ein spezifisches Produkt/Projekt. Das Ziel ist es hierbei, die Anforderungen des Auftraggebers/Kunden an das zu entwickelnde System zu ermitteln, zu strukturieren und zu prüfen. Das Ergebnis resultiert in ein Lastenheft bzw. bei agilen Projektmanagement in ein Produkt-Backlog.

Allgemein ist die Anforderungsanalyse in folgende Schritte zu unterteilen:

- ➔ Ermittlung
 - *Ermittlung der Anforderungen des Stakeholder (Kunden/Auftraggebers) und Ermittlung deren Interessen*
- ➔ Dokumentation / Spezifikation
- ➔ Strukturierung / Verwaltung / Management
- ➔ Analyse / Überprüfung

- **Vorgehen**

Allgemein besteht der Ablauf im Sammeln der Anforderungen, der Dokumentation (in Form von Text oder Modellen), der Spezifikation der Anforderungen, der Strukturierung und abschließenden Überprüfung auf die gesamte Stimmigkeit.

- **Ermittlung**
Besondere Bedeutung liegt hierbei auf der Übersetzung der Wünsche vom Kunden zu den Anforderungen für das Fachpersonal bzw. den Entwicklern. Hierbei ist auf verschiedenen Faktoren zu achten:
 - Vollständigkeit
-> *explizite Beschreibung aller Kundenwünsche, es darf hierbei keine umformulierten Annahmen des Kunden geben.*
 - Abgrenzung
-> *präzise Definition, um Missverständnisse vorzubeugen*
 - Verständliche Beschreibung
-> *Auftraggeber, so wie Entwickler versteht beim alleinigen Lesen die Anforderung*
 - Atomar
-> *pro Absatz nur eine Anforderung -> Aufteilung*
 - Identifizierbar
-> *Vergabe von Kennnummern*
 - Einheitliche Dokumentation
-> *einheitliche Schriftform und Aufbau*
 - Nachprüfbar
-> *Verknüpfung mit Abnahme Kriterien*
 - Konsistenz
-> *Anforderungen sind untereinander widerspruchsfrei*
- ⇒ **SMART+ Formulierung – Spezifisch | Messbar | Attraktiv | Realistisch | Terminiert**

- **Strukturierung und Abnahme**

Die Strukturierung und Klassifizierung der Anforderungen dient zur Schaffung von einer Übersichtlichkeit -> Erhöhung Verständnis der Beziehungen zwischen den Anforderungen.

- Abhängig
-> Ermittlung der Abhängigkeit unter den Anforderungen und Reihung dieser in einem nötigen zeitlichen Ablauf
- Zusammengehörig
-> fachlich-logische Zusammengehörigkeiten sollten gemeinsam umgesetzt werden
- Rollenbezogen
-> Jede Benutzergruppe hat ihre eigene Sicht auf die Anforderungen, die damit unterstützt werden.

- **Prüfung und Bewertung**

Bei der Prüfung und Bewertung geht es um die Qualitätssicherung der Anforderungen

- Korrektheit
-> untereinander widerspruchsfrei
- Machbarkeit
-> müssen realisierbar sein
- Notwendigkeit
-> was nicht vom Auftraggeber gefordert wird, ist keine Anforderung
- Priorisierung
-> Erkennung der wichtigsten Aufgaben -> frühere Umsetzung der wichtigeren Funktionen
- Nutzbarkeit
-> auch bei teilweiser Realisierung soll System funktionieren

➔ Aus dem Ergebnis der Ermittlung resultiert sich ein Lastenheft, welches sich im Weiteren nach Strukturierung und Prüfung, in angepasster Form zum Pflichtenheft entwickelt.

Test Driven Development

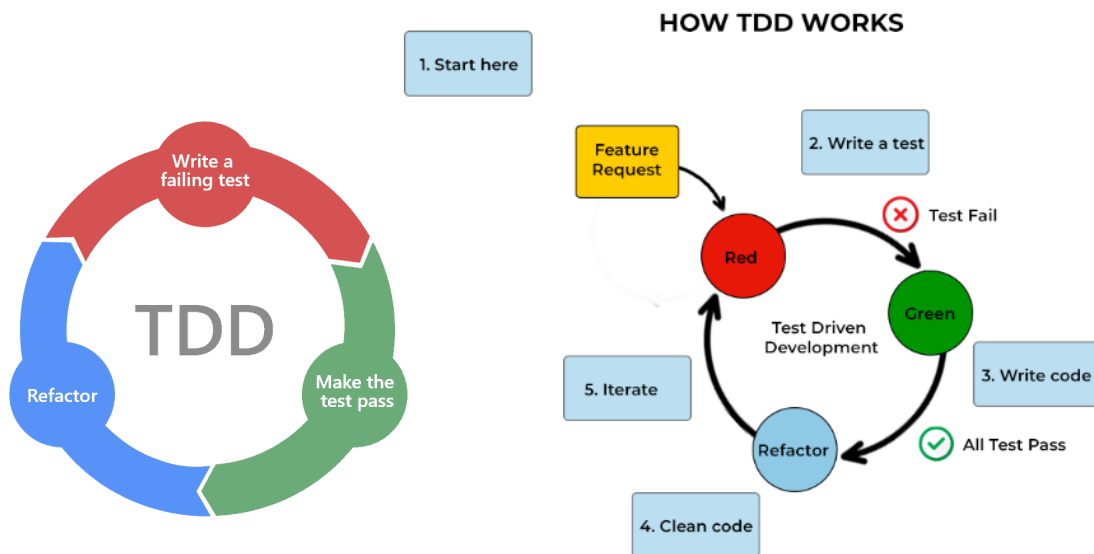
Test Driven Development ist ein inkrementelles Vorgehensmodell der Software-Entwicklung, welches es zum Ziel hat das Code erst ins Produktivsystem übergeht, wenn alle dazu gehörigen Anforderungen erfüllt sind.

Um dies zu Realisieren wird beim TDD der übliche Entwicklungsablauf von Implementierung zu Testung umgekehrt. Dies bedeutet das man zuerst Testfälle schreibt, welche die Anforderungen repräsentieren/überprüfen, und erst im Anschluss, die hierfür benötigten Code-Komponenten implementiert, welche die Testfälle erfüllen können.

Dieses Prozedere ist hierbei im sogenannten **Red-Green-Refactor-Zyklus** abgebildet, welcher wie folgt funktioniert:

1. **Rote Phase:** In dieser Phase denken Sie sich in die Rolle des Nutzers hinein. Dieser möchte den Code auf einfache Weise verwenden. Sie schreiben also einen Test, der Komponenten enthält, die noch nicht implementiert wurden. So müssen Sie eine Entscheidung darüber treffen, welche Elemente für das Funktionieren eines Codes wirklich notwendig sind.

2. **Grüne Phase:** Angenommen, der Test schlägt fehl und wird rot markiert. Nun nehmen Sie die Rolle eines Programmiers ein, der versucht, eine simple Lösung zu finden. Wichtig: Sie schreiben nur so viel Code wie nötig. Diesen integrieren Sie in den Produktivcode, sodass der Test mit grün markiert wird.
3. **Refactoring:** In diesem Schritt wird der Produktivcode regelrecht „aufgeräumt“ und in seiner Struktur perfektioniert. Das heißt, Sie sollten ihn ergänzen und so umstrukturieren, dass er aus Entwicklersicht elegant und verständlich ist. Entfernen Sie z. B. Code-Duplizierungen und bringen Sie ihn so auf ein professionelles Niveau.



Vorteile	Nachteile
Software ist auf einem hohen Niveau und enthält weniger Bugs.	Setzt Codeverständnis voraus und erfordert längere Einarbeitungszeit.
Systemarchitektur und Produktivcode sind verständlich und gut strukturiert.	Testet nur die Richtigkeit des Codes und nicht die Gebrauchstauglichkeit der Software.
Die Fehleranalyse verläuft schneller und Wartungsarbeiten reduzieren sich.	Muss eventuell durch andere Testverfahren ergänzt werden.
Entfernen von Redundanzen im Code vermeidet Over-Engineering.	

- ➔ Mit dieser Methode lässt sich eine Software schrittweise mit neuen Funktionen anreichern, wobei der hierfür benötigte Code erst eingebunden wird, wenn eine Anforderung vollständig durch einen Testfall erfüllt wurde.

Quelle:

<https://www.ionos.at/digitalguide/websites/web-entwicklung/was-ist-test-driven-development/>

Kano Modell – Modell zur Analyse von Kundenwünschen



Das Kano-Modell (von Noriaki Kano (emeritierter Professor der Tokyo University of Science)) beschreibt den Zusammenhang zwischen der Erfüllung von Kundenanforderungen und der Kundenzufriedenheit. Einflussfaktoren sind hierbei:

- **Basismerkmale.**
Sie gelten als selbstverständlich, werden dem Kunden aber erst bewusst, sofern sie nicht vorhanden sind. Basismerkmale sind implizite Muss-Kriterien, die von Kunden nicht direkt artikuliert, sondern stillschweigend vorausgesetzt werden. Fehlen Basismerkmale, sind Kunden unzufrieden, sind sie vorhanden, entsteht jedoch keine zusätzliche Zufriedenheit. Basismerkmale werden auch als "expected requirements" bezeichnet. (Bsp.: Suchfunktion, Möglichkeit Passwort Änderung)
- **Leistungsmerkmale.**
Sie werden von Kunden explizit verlangt und haben Einfluss auf die Zufriedenheit. Werden Leistungsmerkmale nicht erfüllt, entsteht Unzufriedenheit bei Kunden. Werden Leistungsmerkmale übertroffen, steigt entsprechend die Zufriedenheit. Leistungsmerkmale – manchmal auch als Qualitätsmerkmale beschrieben – lassen sich durch Marktuntersuchungen, Marktbeobachtungen und Marktbefragungen ermitteln. Sie werden auch als "normal requirements" bezeichnet. (Bsp.: Verwaltung von Stücken)

- **Begeisterungsmerkmale.**
Diese Merkmale sind in der Lage, Kunden zu begeistern. Sie stiften tatsächlichen oder zumindest gefühlten Nutzen. Begeisterungsmerkmale werden nicht erwartet und ein Fehlen entsprechender Merkmale schafft auch keine Unzufriedenheit. Ist aber ein Begeisterungsmerkmal vorhanden, kann bereits eine kleine Leistungssteigerung zu einem überproportionalen Nutzen führen. Begeisterungsmerkmale werden auch als “delightful requirements” bezeichnet.
(Bsp.: Verlinkungen; Ansicht der letzten Änderungen)
 - **Unerhebliche Merkmale.**
Sie führen weder zur Zufriedenheit noch zur Unzufriedenheit, unabhängig ob sie vorhanden sind oder nicht.
(Bsp.: Eintragung der Tel-Nummer und des Geb.-Datums)
 - **Rückweisungsmerkmale.**
Existieren diese Merkmale führen sie zu Unzufriedenheit, sind sie hingegen nicht vorhanden, schaffen sie dennoch keine Zufriedenheit.
(Bsp.: unübersichtliche Tabelle, aufgrund zu vieler Daten)
- ➔ Hierbei ist zu beachten, dass sie diese Merkmale mit dem zeitlichen Verlauf verschieben können und somit, nicht mehr zur Zufriedenheit der Kunden beitragen können.
(Bsp.: Farbfernsehen, Bluetooth)
- Ebenso ist festzustellen, dass sich aus dem Kano-Modell keine Gesamtzufriedenheit ableiten lässt, sondern nur eine relative Kundenzufriedenheit.
- (Vergleich zu ITPP 3.Klasse -> Muss/Nicht-Ziele und Optionale-Ziele)

Quelle:

<https://www.microtool.de/wissen-online/was-ist-das-kano-modell/>