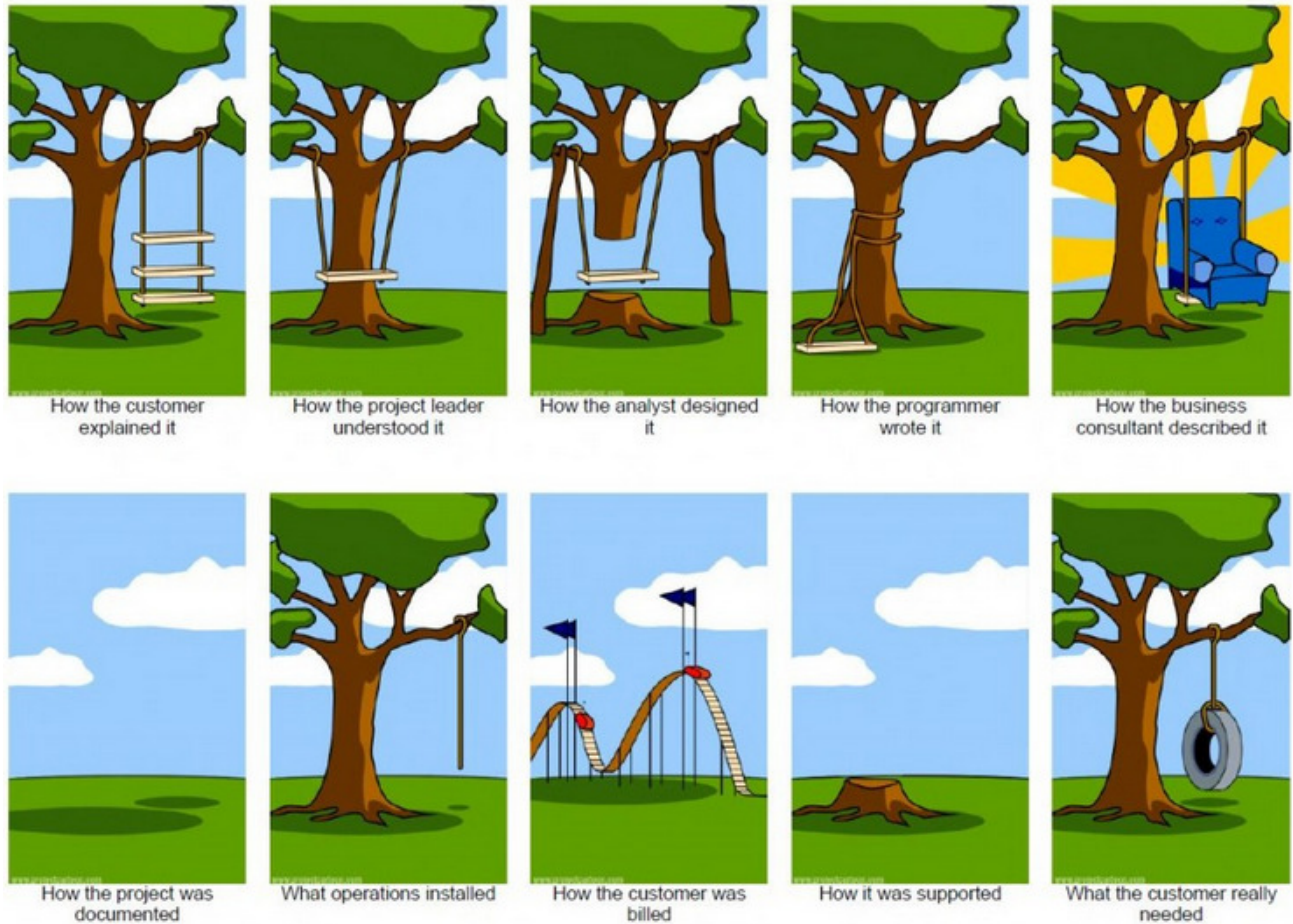


Wie Projektarbeit wirklich funktioniert



1. Wie es vom Kunde beschrieben wurde
2. Wie der Projektleiter es verstanden hat
3. Wie es designt wurde
4. Wie es programmiert wurde
5. Wie der Berater es beschrieben hat
6. Wie das Projekt dokumentiert wurde
7. Was installiert wurde
8. Was dem Kunden in Rechnung gestellt wurde
9. Was an Support geboten wurde
10. Was der Kunde eigentlich gebraucht hätte

(Quelle: projectcartoon.com)

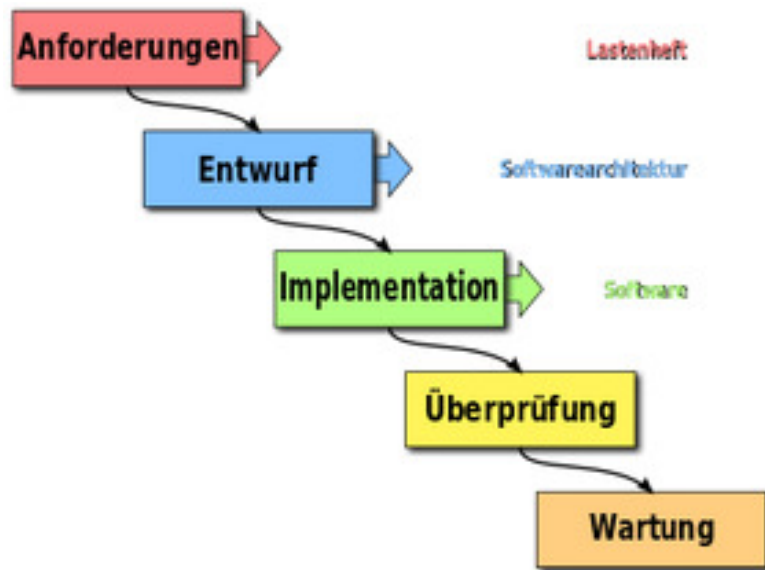
Eine etwas überspitzte Darstellung, die aber doch einiges an Wahrheitsgehalt aus der Praxis beinhaltet. Denn bei der Erstellung von Softwarekomponenten, bei der Integration von Systemen oder der Erstellung von Portalen gibt es zahlreiche Anforderungen, die berücksichtigt werden wollen. Oftmals stehen Kunden vor einer Problemstellung und haben vielleicht eine grobe Vorstellung von der gewünschten Lösung – es fehlt jedoch an etwas “Greif- und Erlebbarem” um die Anforderungen konkret zu machen.

Software - Entwicklungsmodell

Professionelle Softwareentwicklung kennt verschiedene Verfahren, um von einer Idee zu einem lauffähigen Softwareprodukt zu kommen. Ausgangspunkt ist der Software Life Cycle, der linear, zyklisch oder spiralförmig durchlaufen werden kann. Die entsprechenden Ansätze (Treppen-, Wasserfall-, Spiralmodell u. a.) erfordern im Idealfall eine vollständige Anforderungsdefinition und Spezifikation bevor die nächsten Phasen begonnen werden. Sie sind deshalb dokumentationslastig.

Für den Informatik-Unterricht ist primär ein Vorgehensmodell von Bedeutung:

Das Wasserfall-Modell



Das Wasserfallmodell ist eines der ältesten und beschreibt den Entwicklungsprozess als sequentiellen Ablauf von Phasen

- Jede Phase wird vollständig abgeschlossen, bevor die nächste begonnen werden darf.
- Jede Phase produziert genau definierte Ergebnisse, die als Input für die nächste Phase dienen.
- Rücksprünge und Iterationen zwischen aufeinanderfolgenden Phasen sind aber möglich.
- Am Ende jeder Phase steht ein Prüfschritt.

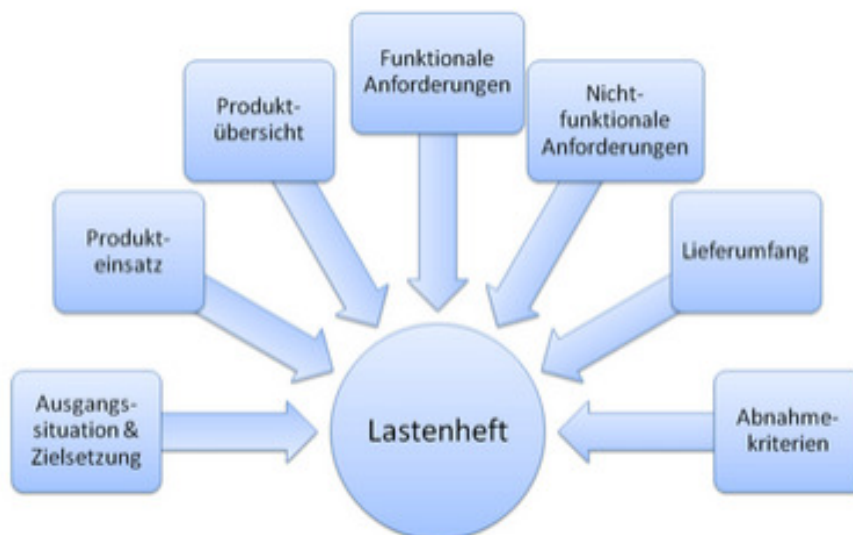
Dieses Vorgehen eignet sich für größere (halbjährliche) Softwareprojekte, bei denen alle Phasen komplett durchlaufen und erfahrbar gemacht werden sollen. Die notwendige begleitende Dokumentation und Delegation von Verantwortung verhindern weitgehend, dass ein schulisches Projekt im Chaos versinkt

Lasten- und Pflichtenheft

Das Lastenheft

Das Lastenheft (teils auch Anforderungsspezifikation, Anforderungskatalog, Produktskizze, Kundenspezifikation oder Englisch Requirements Specification genannt) beschreibt die Gesamtheit der Anforderungen des Auftraggebers an die Lieferungen und Leistungen eines Auftragnehmers. Das Lastenheft beschreibt in der Regel somit, was und wofür etwas gemacht werden soll.

Die Anforderungen in einem Lastenheft sollten durch ihre Formulierung so allgemein wie möglich und so einschränkend wie nötig formuliert werden. Hierdurch hat der Auftragnehmer die Möglichkeit, optimale Lösungen zu erarbeiten, ohne durch zu konkrete Anforderungen in seiner Lösungskompetenz eingeschränkt zu sein. Der Auftraggeber sollte das Lastenheft formulieren. Hauptbestandteil des Lastenhefters sind somit die Funktionalen Anforderungen (Was soll das Produkt können?). Aber auch nichtfunktionale Anforderungen (Benutzbarkeit, Zuverlässigkeit, Effizienz, Änderbarkeit, Übertragbarkeit, Wartbarkeit) können aufgenommen werden.



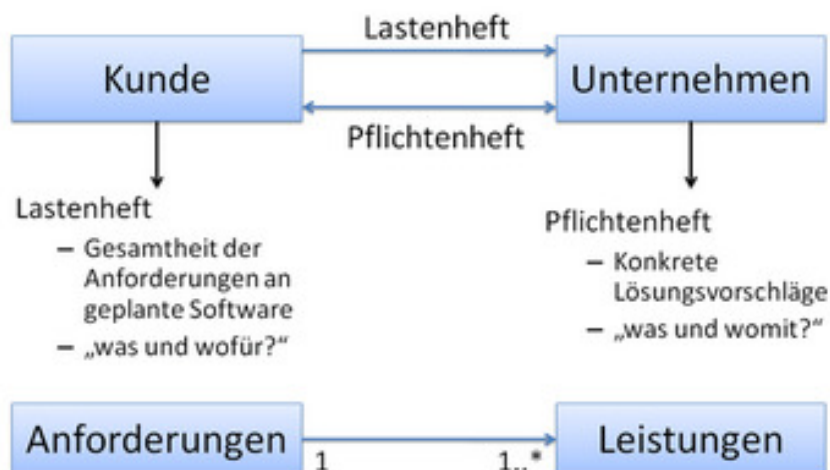
Die Forderungen im Lastenheft sind häufig in Muss- Forderungen, Soll- Forderungen und Kann- Forderungen unterteilt.

Das Pflichtenheft

Das Pflichtenheft beschreibt in konkreter Form, wie der Auftragnehmer die Anforderungen des Auftraggebers zu lösen gedenkt – das sogenannte wie und womit. Der Auftraggeber beschreibt vorher im Lastenheft möglichst präzise die Gesamtheit der Forderungen – was er entwickelt oder produziert haben möchte. Das Pflichtenheft wird vom Auftragnehmer formuliert. Erst wenn der Auftraggeber das Pflichtenheft akzeptiert, sollte die eigentliche Umsetzungsarbeit beim Auftragnehmer beginnen.



Die Abgrenzung von Pflichtenheft ist daher verkürzt so zu beschreiben, dass das Lastenheft eine Problemstellung vorgibt und das Pflichtenheft die geplante Lösung beschreibt.



Beispiel-Dokumente der Uni Paderborn, die auch als Vorlage dienen können:

Siehe im Gutsche-Ordner unter 10_02 Lastenheft Beispiel und 10_03 Pflichtenheft Beispiel.

Aufgabe 1: Wissen Prüfung Software-Entwicklung

1. Warum muss der Beginn einer Phase im Wasserfallmodell immer auf den Abschluss der vorausgehenden Phase warten?
2. Was passiert, wenn mitten in der Implementation neue Anforderungen hinzukommen.
3. Beschreibe in einem Satz den wesentlichen Unterschied zwischen Lasten- und Pflichtenheft
4. In welches von beiden Dokumenten würde das Design einer GUI-passen? Begründe!
5. In welcher Phase des Projekts würde man ein UML-Diagramm konzipieren? Begründe!