**Komponententest**

-Bei diesem Test werden die einzelnen Module der Software getestet. Der Test bezieht sich dabei auf die Funktionalität einzelner Teile. Dieser Test ist der aller erste Test, danach kommt der Integrationstest.

-Dieser Test könnte nach jedem Anlegen einer Komponente oder eines Objektes ausgeführt werden. Oder man teste an einem bestimmten Punkt mehrere Komponenten oder Objekte jeweils einzeln.

-Je nach Komponente muss man mehr oder weniger wissen. Außerdem kommt es darauf an, wie der Code kommentiert ist. Aber dennoch wäre es gut alle Einzelheiten einer Komponente aus der Spezifikation zu kennen. Es kann auch Komponenten geben, die nicht in der Spezifikation auftreten, zum Beispiel Prozeduren, die notwendig sind, aber noch nicht bei der Spezifikation bekannt waren.

----

-Der Test basiert auf der Spezifikation. Bei dem Komponententest kann man statisch und dynamisch vorgehen. Beim statischen Vorgehen vergleicht man den Code mit der Spezifikation. Beim dynamischen Vorgehen vergleicht man die Ergebnisse der Ausführung mit der Spezifikation.

-Vorteil:

-man kann testen, ob eine Komponente erfolgreich implementiert wurde

-genau (testet jede einzelne Komponente)

-Nachteil:

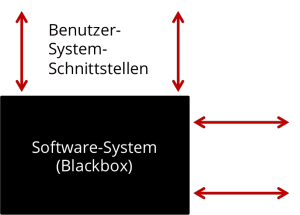
-sehr zeitaufwendig alles zu testen

-unabhängig von anderen Komponenten (nicht alle Fehler entdeckt)

**Integrationstest**

* eine Reihe von Einzeltest um voneinander Abhängige Komponenten im Zusammenspiel miteinander zu testen
  + für jede Abhängigkeit von zwei Komponenten wird ein Szenario definiert, das überprüft ob diese Komponenten spezifikationsmäßig miteinander funktionieren
  + es wird im Voraus des Integrationstestes immer ein isolierter Test der Komponenten durchgeführt 🡪 um sicherzugehen das dort keine Fehler sind
* dieser Test sollten nach der Fertigstellung von zwei miteinander agierenden Komponenten durchgeführt werden
* für diese Testart ist die Spezifikation als Informationsstand notwendig, da dort die Interaktion zwischen den einzelnen Komponenten beschrieben ist
* wenn man die Funktionen schon miteinander agieren lässt (z.B. Knopf drücken 🡪 Tür öffnet sich) ist es schon quasi in Inform7 implementiert
* **basiert auf der Spezifikation**
* Vorteil:
  + Kann genau prüfen ob die Komponente funktioniert/ sich in den Code einfügt
  + Ist sehr genau
* Nachteil:
  + Ist nicht für die komplette Geschichte geeignet 🡪 da der zeitliche Aufwand überproportional zu der Komponentenzahl ansteigt und es mit der kompletten Geschichte nicht mehr effizient wäre
* Einsatzzweck:
  + Nur kleine Teile der Geschichte werden überprüft
  + Überprüfung der Funktionsweise von zwei miteinander interagierenden Komponenten
* Einsatzzeitpunkt:
  + Nach der Vollendigung von zwei miteinander agierenden Komponenten
* Fehlerentdeckungsrate:
  + 95% 🡪 wird sicherlich die größten Fehler finden, wenn es sich nicht laut der Spezifikation verhält 🡪 falsch

**Systemtest**

-Beim Systemtest wird das gesamte Software-System gegen die gesamten Anforderungen geprüft. Das Software-System wird hierbei als eine Art „Blackbox“ angesehen. Es wird über die Nutzer-System- und System-System-Schnittstelle geprüft. Dabei wird eine Testumgebung geschaffen, die das Verhalten des späteren Users simuliert. Der Systemtest kann manuell oder automatisiert durchgeführt werden. Er dient darüber hinaus detaillierten Planung des weiteren Projektverlaufs im Hinblick auf eventuelle Änderungen des Zeitplans, des Budgets etc. je nach Ausgang des Tests.

-Zeitplans, des Budgets etc. je nach Ausgang des Tests.

-Der Systemtest ist die vorletzte Teststufe und reiht sich nach dem Integrationstest und vor dem Abnahmetest ein.

-Der Systemtest wird oft unabhängig vom erstellenden Entwicklerteam durchgeführt, um eine möglichst realistische Einschätzung des Produkts zu generieren. Somit muss das testende Team in erster Instanz die Anforderungen des Lastenhefts kennen, um dagegen zu testen.

-Der Systemtest testet alle funktionalen und nicht-funktionalen Anforderungen, d.h. er testet gegen die Anforderungen im Lastenheft.