

IT Support Fundamentals Tag 3

Bits, Bytes und das Binärsystem:

- Bits (b) und Bytes (B):
 - Der Begriff **Bit** bedeutet „Binary Digit“, auf Deutsch „Binärziffer“.
 - Ein Bit ist in der IT die kleinste Informationseinheit und kann die Werte „0“ oder „1“ annehmen.
 - Oft werden „0“ auch als „false“ oder „nein“ und „1“ als „true“ oder „ja“ bezeichnet.
 - Eine Bitfolge, bestehend aus acht Bits ergibt ein **Byte**.
 - Etwa 1.000 Bytes sind 1 Kilobyte(KB) (genau 1.024 Bytes), etwa 1 Million Bytes sind ein **Megabyte** (MB) (genau 1.048.576 Bytes). Etwa 1 Milliarde Bytes sind ein **Gigabyte** (GB) (genau 1.073.741.824 Bytes). Etwa 1 Billion Bytes sind ein **Terabyte** (TB) (genau 1.099.511.627.776 Bytes)
 - Physikalisch lässt sich ein Bit z.B. als Schaltungszustand darstellen. Lichtschalter aus „0“, Lichtschalter an „1“.
 - Ein Byte erlaubt die Darstellung von zwei hoch acht (=256) verschiedenen Zuständen.
 - Die Anzahl möglicher Zustände einer Bitfolge kann man mit der Formel **zwei hoch n** (n =Anzahl der Bits) berechnen.
 - Mittlerweile werden Speicherkapazitäten in Byte und Vielfachen davon, Übertragungsgeschwindigkeiten in Bit pro Sekunde und Vielfachen davon angegeben.
- Binärsystem:
 - Sowie das Dezimalsystem (10) oder das Hexadezimalsystem (16) ist das **Binärsystem** (2) auch ein **Stellenwert-Zahlensystem**.
 - Man nennt es auch Zweiersystem oder Dualsystem, wie der Name schon sagt, besteht dieses Zahlensystem aus nur zwei verschiedenen Ziffern, demnach ist die Basis 2. Somit die „0“ und „1“. (Die ersten zwei Zahlen)
 - Das Binärsystem ist Grundlage der Funktionsweise aller Computer.
 - Computer arbeiten mit Bits und deren Zustand lässt sich praktisch mit 2 physikalischen Zuständen beschreiben. Entweder es fließt Strom (ON) oder es fließt kein Strom (OFF).
 - Man kann ganz einfach eine Binärziffer ins Dezimalsystem umrechnen. Im Binärsystem ist die Basis 2. Basis hoch Stellen-Nr.

- **Einführung Programmiersprachen:**

- Beim Programmieren, schreibt man Anweisungen, welche das Programm nachher ausführen soll und das Programm gibt dann diese Anweisungen an den Computer weiter.
- Ziel vieler Programmierer ist es, Funktionalität durch möglichst wenig Zeilen Code zu erreichen.
- Es gibt verschiedene Programmiersprachen mit jeweils unterschiedlicher „Grammatik“ und Syntax.

- **Kompilierte Sprachen:**

- Eine kompilierte Programmiersprache ist eine, die die Verwendung eines Compilers (Ein Programm, welches den Quellcode einer Programmiersprache in Maschinencode, Bytecode oder eine andere Programmiersprache übersetzt) erfordert, um sie in Maschinencode zu übersetzen.
- Erstellen und Verwenden:
 - Man schreibt die Anwendung in einer Programmiersprache wie Java oder C++. Dies wird als Quellcode bezeichnet.
 - Man verwendet einen Compiler, um den Quellcode in Maschinencode zu übersetzen. Die meisten Softwareentwicklungsanwendungen verfügen über einen Compiler.
 - Die Programmdatei führt man aus, die (unter Windows) normalerweise die Erweiterung .exe hat.
- Die beliebtesten kompilierten Sprachen sind Java, C, C++ und C#.

- **Markup Sprachen:**

- Programmiersprache, die Texte "mit Anmerkungen versieht" (=to mark up)
- Codifizierte Regeln vermitteln dem Prozessor, was er mit marked-up Text machen soll
- Am weitesten verbreitet: Hypertext Markup Language (HTML); wird von Webbrowsern verwendet
- Neuste Version: HTML 5
- Eine ebenfalls weitverbreitete mark-up Programmiersprache ist die Extensible Markup Language (XML)
- Seiten, die HTML enthalten, werden (extern) auf einem Server gespeichert, bei Abruf auf einen Client heruntergeladen und anschließend von einer speziellen Software verarbeitet.
- HTML beruht auf sog. "Tags": <> ist dabei der Öffnungstag, </> der schließende Tag

- **Query Sprachen:**

- Query bedeutet so viel wie "Fragen" oder "Suchen"
- Query-Programmiersprachen sind darauf ausgelegt, Daten aus Datenbanken abzurufen
- Die (bei weiten) am meisten verbreitete Programmiersprache ist die Structured Query Language (SQL)
- Die Syntax von SQL sieht wie folgt aus:
 - SELECT column_name
 - FROM table_name
 - WHERE condition

- SQL erlaubt weiterhin, Daten in Tabellen einzufügen, Datensätze von verschiedenen Tabellen zu verbinden sowie verschiedene Operatoren zu verwenden, wie beispielsweise das Finden des Minimums und Maximum Values zählen, das Errechnen von Mittelwerten und das Aufsummieren von Werten.
- **Kompilierte Sprachen (Team A):**
 - Eine kompilierte Programmiersprache ist eine, die die Verwendung eines Compilers (Ein Programm, welches den Quellcode einer Programmiersprache in Maschinencode, Bytecode oder eine andere Programmiersprache übersetzt) erfordert, um sie in Maschinencode zu übersetzen.
 - Erstellen und Verwenden:
 - Man schreibt die Anwendung in einer Programmiersprache wie Java oder C++. Dies wird als Quellcode bezeichnet.
 - Man verwendet einen Compiler, um den Quellcode in Maschinencode zu übersetzen. Die meisten Softwareentwicklungsanwendungen verfügen über einen Compiler.
 - Die Programmdatei führt man aus, die (unter Windows) normalerweise die Erweiterung .exe hat.
 - Die beliebtesten kompilierten Sprachen sind Java, C, C++ und C#.
- **Programmierlogik:**
 - Prozessoren führen mathematische und logische Operationen durch.
 - Alle Programme bestehen daher aus Logik und Arithmetik.
 - Die Aufgabe eines Programmierers besteht darin, dem Prozessor Anweisungen auf Grundlage von Logikrätseln zu geben.
 - Die beiden wesentlichen Arten von Programmen, "logisch zu denken" sind *branching* und *looping*
- **Branching:**
 - "Verzweigungslogik" basieren auf "wenn... dann"-Aussagen (sowie "wenn...dann, ansonsten..."; sowie "wenn...dann, ansonsten wenn... (ansonsten...)")
 - Z.B. if light = red, then stop else if light = green, then go else if light = amber, then slow down
 - Verzweigungslogische Aussagen werden verwendet um Variablen und Konstanten mit anderen Variablen und Konstanten über Datentypen hinweg zu vergleichen. Der Boolische Ausgabewert (richtig oder falsch) werden verwendet, um den Pfad zu bestimmen, den das Programm nimmt. Zweige (*Branches*) eignen sich für einfache logische Vergleiche.

- **Common Data Types:**

- siehe Tabelle:

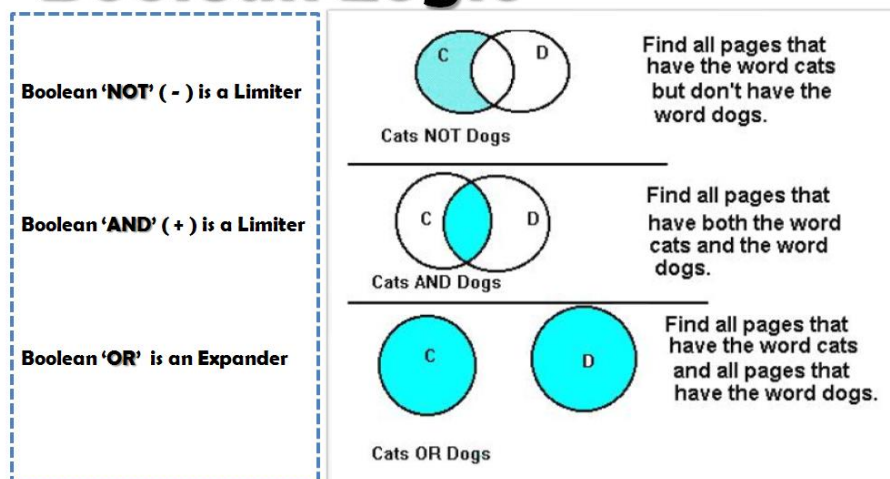
TABLE 6.4 Common data types

Type	Explanation	Examples
Char	One character, such as a UTF-16 or UTF-32 character	A or a
String	Zero or more characters	"This is a string" and "S0 is th1\$."
Integer	Whole number with no decimal point	5 or 500000
Floats	Any number with a decimal place	5.2 or 5.000001
Boolean	A true or false condition, usually represented by a 1 (true) or 0 (false)	1 or 0

- Boolische Operatoren: AND, OR, NOT sind die wichtigsten
- "AND" verknüpft mehrere Aussagen miteinander
- "OR" nicht alle Elemente der Aussagen müssen wahr sein, mindestens ein (kein "exklusives" oder)
- "not" zum Verneinen von Aussagen (a oder b; nicht a --> b)
- Neben den grundlegenden Boolischen Operatoren gibt es noch komposierte B. Operatoren: (*Material conditionals*: --> (wenn dann) ($x \rightarrow y$); *Exclusive OR*, *Logical Equivalence* (=))

(vgl. Wikipedia Boolean algebra_Boolean operations)

Boolean Logic



- **Looping:**

- "Durchlaufen einer Schleife" ist ein kreisförmiger (und nicht wie beim *branching* linearer) Prozess.
- Zentral: das *while*-statement
- Ein Befehl wird ausgeführt, bis eine *exit condition* erreicht wurde
- Wichtig: der *loop* sollte nicht unendlich weiterlaufen
- Beispiel: "while x<10 is true, count x+1" (exit condition: x<10)
- *Loops* werden häufig zusammen mit *Branches* verwendet, um komplexere Operationen durchzuführen.

- **Organizing Code:**

- Aufgliederung als entscheidendes Organisationsprinzip
- Größere Aufgaben werden in überschaubare Stücke eingeteilt
- "Außerhalb" des Codes ist es für einen Entwickler sinnvoll, einen "Bauplan" für die Funktionsweise des Programms zu erstellen und ggf. Aufgaben an Teammitglieder zu verteilen

- **Fehlertoleranz:**

- Die Fähigkeit, auch dann noch korrekt zu arbeiten, wenn Teile der Hardware oder Software ausfallen.
- Der Begriff Toleranz bezieht sich hierbei darauf, wie viel Risiko man bereit ist zu akzeptieren.
- Wenn gewisse Teile einer Software ausfallen, und genau diese aber notwendig sind zum Arbeiten, ist es sinnvoll einen Notfallplan zu erstellen für solche Fälle.
- Es gibt verschiedene Varianten von fehlertoleranten Plänen, die implementiert werden können, von Vollzeit-, Always-On-Lösungen bis hin zu Offline-Backups, deren Wiederherstellung ein oder zwei Tage dauern kann.

- **Notfallpläne:**

- Der Notfallplan ist ein Aktionsplan für den Fall, dass es ein Problem gibt.
- Notfallpläne variieren zum Teil aber im Allgemeinen sind es die folgenden sechs Schritte, die man unternehmen muss, um einen zu erstellen:
 - **Schritt 1: Führe eine Analyse der geschäftlichen Auswirkungen durch:**
Zuerst sollte man prüfen, welche Auswirkungen, wenn überhaupt, Ausfälle von Komponenten oder Datenverlust auf das Geschäft haben.
 - **Schritt 2: Identifiziere präventive Systeme:**
Welche Systeme verfügt das Unternehmen, um Probleme zu vermeiden? (bspw. Rauchmelder, Sprinkleranlagen, Energiesicherungen, etc.)
 - **Schritt 3: Entwickle einen Sanierungsplan:**
Was ist der Plan? Wer ist wofür verantwortlich? Werden alle Arten von Problemen berücksichtigt?
 - **Schritt 4: Teste den Wiederherstellungsplan:**
Durchlaufe einmal den Ernstfall. Funktioniert der Plan? Gibt es Verbesserungsvorschläge? Sind alle Personen eingebunden?
 - **Schritt 5: Richte einen Wartungs- und Überprüfungszeitplan ein.**
Setze regelmäßig Termine an, um zu prüfen, ob der Plan noch zu den aktuellen Gegebenheiten passt, damit dieser auch immer aktuell ist.
 - **Schritt 6: Implementiere eine Schulung:**
In den Wartungs- und Überprüfungsterminen sollte man regelmäßig auch Schulungen zum Notfallplan machen, damit jede beteiligte Person immer auf aktuellem Stand ist und weiß was zu tun ist.

- **Troubleshooting:**

- **Schritt 1. Identifiziere das Problem:**

- Informationen vom Nutzer sammeln und fragen, ob der Nutzer etwas geändert hat.
- Oftmals liegt die Fehlerursache bei der Bedienung des Nutzers.
- Schritt-für-Schritt Demonstration des Problems vom Nutzer anfordern.
 - Nichts annehmen (oftmals gehen Nutzer von falschen Problemursachen aus oder verwenden Fachbegriffe falsch).
 - Den Nutzer befragen (Haben Sie etwas geändert? In freundlichem (nicht-drohenden) Ton. Zu Anfang offene Fragen verwenden, später enger gefasste (Ja/Nein) Fragen).
 - Frage nach äußeren oder Infrastrukturveränderungen (Wurde kürzlich etwas hinzugefügt zum Computer oder Netzwerk? Wurden Instandhaltungsarbeiten geleistet?)
 - Offensichtliche Möglichkeiten abklären, z.B. ist der Bildschirm eingestöpselt? Ist das Mikrofon stummgeschaltet?
 - Nicht annehmen es gäbe kein Problem, wenn der Nutzer das Problem nicht demonstrieren kann. Manche Probleme treten nur zeitweise auf.
 - Benutze alle fünf Sinne: Geräusche vom Stromstecker/ Gehäuse oder CPU Ventilatoren, Kopfhörerschall, Festplatte oder des optischen Laufwerks. Stromversorgung unterbrechen, wenn verbrannter Geruch auftritt!
 - Daten-Sicherung (Back-Up) erstellen, bevor Änderungen vorgenommen werden.

- **Schritt 2. Etabliere eine Theorie der wahrscheinlichen Ursache:**

- Zunächst muss man das Problem des Nutzers gehört oder gesehen haben.
- Oft macht es Sinn zu fragen, ob das Problem mit Software, Hardware oder beidem zu tun hat.
- Benutze alle deine Sinne (Sicht, Geräusche, Geruch).
- System- und Anwendungslogs überprüfen. Betriebssysteme dokumentieren in den Logs was passiert und können bereits über Probleme Auskunft geben.
- In Windows gehe dafür in das Windows Systemkonfigurationsfenster > Boot tab und klicke auf die *Boot log* Kästchen > Anwenden > OK > Neustarten.
- Im Einstellungsfenster gehe auf *Boot Logging aktivieren* > Reboot > und überprüfe die Loggingdatei ntbtlg.txt (Unter "Ausführen" nach C:\Windows\ntbtlog.txt suchen)
- Interne Ressourcen nutzen: Kollegen oder Vorgesetzten fragen.
- Externe Ressourcen nutzen: Z.B. Internetrecherche

- Vergiss nicht, dass Offensichtliche zu erfragen (Z.B. Gerät an Strom angeschlossen?)

Der Boot-Prozess im Detail (Mitaufnehmen?)

- Wird eingeleitet wie oben beschrieben
- Der Power-on self-test (POST), wird vom BIOS/UEFI (Binary Input-Output System/ Unified Extensible Firmware Interface) beim Start des Computers ausgeführt. Hilfreich beim Aufdecken von Hardware-Problemen.
- Audiosignale des POSTs geben Auskunft über Gerätefehler (verschiedene Codes für verschiedene Hersteller)
- Weiter können Fehlermeldungen auf dem Bildschirm erscheinen. Je nach BIOS/UEFI Chiphersteller erscheinen bestimmte Fehlercodes/-signale. Ein einzelner Piep Ton ist meist das Signal für einen erfolgreichen Test ohne Hardwarefehler.
- Für eine ausführliche Beschreibung des BIOS-Vorgangs (v.a. auf älteren Computern), Audio- und visuelle Fehlercodes ("POST Codes and Error messages") siehe Seiten 24-28 im Buch.

Weitere Diagnosewerkzeuge

- Manche IT-Mitarbeiter besitzen POST-Karten. Hierbei handelt es sich um einen PCI/PCIe Adapter oder einer Karte mit USB-Anschluss, die Hardwarediagnosen durchführen und die Ergebnisse auf einem LED-Display in Form von Codes oder durch LED-Lichter darstellt
- Sind heutzutage eher weniger gebräuchlich, da die meisten UEFI-basierten Motherboards selbst über wirkungsstarke Diagnostiken verfügen,
- Eine POST-Karte kann nützlich sein, wenn ein System nicht bootet und keine anderen Symptome auftreten.

Hardwarefehler

- Schwarzer Bildschirm, optisches Laufwerk ohne Licht, Fehlermeldung des Druckers, ...
- Wenn ein physisches Anschlussproblem vermutet wird, verwende einen *loopback plug* (sendet elektrische Signale aus und erlaubt diesen zurückzukommen)
- Meist offensichtlich durch POST Fehlermeldungen oder Probleme die beim Benutzen eines Gerätes auftauchen.
- Bei Festplatten, Druckern wird eine Fehlerdiagnose beim Start des Gerätes meist automatisch durchgeführt. Die Diagnosen lassen sich unter dem Windowsfenster des entsprechenden Geräts unter *Eigenschaften* einsehen.

Intermittierender Gerätefehler

- Wenn keine der bisher beschriebenen Hardware-Fehlerbehebungsaktionen funktioniert, kann es sich auch um ein Erdungsproblem handeln. Zu den Symptomen hiervon gehören intermittierende und unerklärliche Abschaltungen

- Um einen intermittierenden Geräte Fehler zu beheben, baue den Computer außerhalb des Computergehäuses auf einer antistatischen Matte auf.
- Schließe zunächst nur das Netzteil, die Hauptplatine und den Lautsprecher an.
- Stelle sicher, dass sich der Netzteillüfter dreht,
- Überprüfe die Spannung der Stromversorgung
- Wenn sich der Lüfter dreht und die Spannungen korrekt sind, schalte das Gerät aus und füge Grafikkarte und Monitor hinzu. Wenn das Gerät nicht funktioniert, setze Grafikkarte in einen anderen Erweiterungssteckplatz ein. Wenn es dann immer noch nicht funktioniert, tausche die Grafikkarte aus.
- Wenn die Grafikkarte funktioniert, füge nacheinander Geräte hinzu und überprüfe die Spannungen,
- Wenn ein Adapter nicht funktioniert, wähle einen anderen Erweiterungssteckplatz, bevor du einen anderen Adapter ausprobiert.
- Wenn sich der Erweiterungssteckplatz als Problem erweist, überprüfe ihn auf Fremdkörper. Wenn keine gefunden werden, aber das Problem weiterhin auftritt, platziere eine Notiz darauf.
- Ein intermittierender Geräteausfall ist eine der am schwierigsten zu behebenden Probleme.
- Häufig tritt das Problem beim Motherboard, der RAM, dem Prozessor oder der Stromversorgung auf.
- RAM Fehlerbehebung ...
- Intermittierende Stromversorgungsprobleme: ...
- Motherboard oder Prozessor ...

Softwarefehler

- Entstehen beim Zugriff auf eine bestimmte Anwendung oder Datei oder beim Systemstart
- Anwendungen zeigen manchmal Fehlermeldungen an
- Manchmal startet sich eine Anwendung selbst neu oder muss manuell über den Task-Manager gestoppt werden.
- Manche Softwareprobleme beruhen eigentlich auf Hardware-Problemen, z.B. eine mit einem USB-Stick verwendete Anwendung funktioniert nicht mehr. Hier hilft häufig ein Aus- und erneutes Einstöpseln des Gerätes
- Manche Softwareprobleme lassen sich durch einen *warm boot* lösen.
- Alle vorgenommenen Änderungen werden dann übernommen.
- Z.B. durch Neustart
- Benutze Windows' Gerätemanager um festzustellen, ob ein Problem auf Soft- oder Hardware beruht

- Jede Software hat Bugs. Softwarehersteller bieten *patches* oder *service releases* um bekannte Problem zu lösen.
- Anwendungen und das Betriebssystem müssen regelmäßig "*gepatched*" werden
- Ein *Servicepaket* enthält mehrere Patches und installiert sie gemeinsam statt, dass der Nutzer sie einzeln (von der Webseite des Softwareherstellers) downloaden muss.

- **Schritt 3: Teste die Theorie, um die Ursache zu bestimmen:**

- Wenn man eine Theorie hat oder einen Bereich vermutet, muss man die nächsten Schritte bestimmen, die zur Lösung des Problems erforderlich sind.
- Wenn man eine Reihe von Schritten ausgeführt hat, diese aber nicht zur Lösung beigetragen haben, sollte man eventuell einen Schritt zurückgehen und versuchen das Problem neu zu bewerten, um von dort aus eine neue Theorie aufzustellen.
- Teile das Problem in logische Bereiche auf und teile diese dann weiter auf, bis man das Problem isoliert hat. *(z.B. jedes Mal, wenn der Benutzer versucht, Daten auf ein Flashlaufwerk zu schreiben und ein Fehler auftritt. Sind logischen Bereiche, der Anschluss, an den das Flashlaufwerk angeschlossen ist, möglicherweise ein Kabel, das den Anschluss mit dem Motherboard verbindet, die Motherboard-Schaltkreise, die diesen Port steuern, die BIOS/UEFI-Einstellungen, bei denen der Port deaktiviert werden könnte, und das Softwareprogramm, das derzeit verwendet wird. Jeder von ihnen kann die Ursache des Problems sein.)*
- Das Teilen der Bereiche dient so als Ausschlussverfahren.
- Prüfe zuerst die einfachsten Dinge, bevor direkt komplexe Lösungswege probiert werden. *(Wenn ein Display fehlerhaft ist, tauschen Sie das Display durch ein anderes aus, bevor Sie den Computer öffnen und die Grafikkarte austauschen.)*
- Wenn mehrere Dinge ein Problem verursachen könnten, kann eine Liste zur Übersicht mit den Möglichkeiten helfen, diese kann dann einfach abgearbeitet werden, somit kann man potenzielle Probleme nacheinander beseitigen.
- Wichtig ist auch, sich beim Benutzer zu erkundigen ob in letzter Zeit oder seitdem das Problem vorhanden ist, irgendwelche Änderungen vorgenommen wurden. (z.B. ob etwas Neues installiert oder deinstalliert wurde)
- Falls es ein Hardware Problem ist, kann man auch den Geräte-Manager und den Windows-Problembehandlungsassistenten verwenden, um es weiter einzugrenzen.
- Wenn ein Softwarefehler vermutet wird, man aber keine ungewöhnlichen Audiosignale hören oder POST-Fehlercodes sehen kann, starte den Computer neu und gehe dabei in die Erweiterten Startoptionen. Dort kann man bestimmte Optionen wählen, die einem helfen könnten (z.B. Computer reparieren, Abgesicherter Modus oder Letzte als funktionierend bekannte Konfiguration.)
- Bei Windows kommt man wie folgt in das Erweiterte Startmenü = „Einstellungen“ > „Update & Sicherheit“ > „Wiederherstellung zu“ > „Erweiterter Start“ > „jetzt neu starten“. **Oder** = Drücke vor dem Start von Windows die Taste „•“.
- Die Bestimmung, was das Problem ist, dauert in der Regel länger als die Behebung.
- Das Internet ist ein tolles Hilfsmittel für eventuelle Empfehlungen des Anbieters.
- Sollte nichts helfen, um das Problem zu lösen, gehe immer mal wieder einen Schritt zurück und fange neue an.

- **Schritt 4: Erstelle einen Aktionsplan, um das Problem zu lösen und die Lösung zu implementieren:**
 - Jede Reparatur sollte einen Aktionsplan beinhalten.
 - Dieser hilft einem, durch die Lösung des Problems und die Implementierung der Lösung zu führen.
 - Einen Plan zu haben, anstatt Dinge in einer zufälligen Reihenfolge zu versuchen, spart Zeit.

- **Schritt 5: Überprüfe die volle Systemfunktionalität und implementiere gegebenenfalls vorbeugende Maßnahmen:**
 - Wenn ein Problem gelöst wurde, heißt das nicht, dass dies der einzige Fehler war.
 - Lass den Kunden den Computer unter normalen Bedingungen testen, um sicherzustellen, dass das Problem tatsächlich gelöst wurde. Stelle damit die vollständige Systemfunktionalität sicher.
 - Eventuell müssen noch vorbeugende Maßnahmen unternommen werden (z.B. Computer oder Gerät reinigen, Installation einer Antivirensoftware, Aktualisieren der Software, Wartung der Festplatte, etc.).

- **Schritt 6: Dokumentiere Anhaltspunkte, Aktionen und Ergebnisse:**
 - Es ist wichtig alle Schritte, die zur Lösung eines Problems unternommen wurden detailliert und übersichtlich zu dokumentieren.
 - Häufig wird die Dokumentation gebraucht, um nachher eine Rechnung zu erstellen oder sie wird in der Akte des Kunden aufbewahrt.
 - Die Dokumentation sollte leicht lesbar und leicht verständlich sein, so dass nicht-technische-Benutzer diese auch lesen und verstehen können.