

Claudio Franzetti

Essenz der Informatik

Was Benutzer alles wissen sollten

2. Auflage

Essenz der Informatik

Claudio Franzetti

Essenz der Informatik

Was Benutzer alles wissen sollten

2. Auflage

Claudio Franzetti
Meilen, Schweiz

ISBN 978-3-662-67153-5 ISBN 978-3-662-67154-2 (eBook)
<https://doi.org/10.1007/978-3-662-67154-2>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <https://portal.dnb.de> abrufbar.

© Springer-Verlag GmbH Deutschland, ein Teil von Springer Nature 2019, 2023

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung/Lektorat: Petra Steinmueller

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer-Verlag GmbH, DE und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Heidelberger Platz 3, 14197 Berlin, Germany

Nichts ist vom Erfolg her zweifelhafter und von der Durchführung her gefährlicher als der Wille, sich zum Neuerer aufzuschwingen. Denn wer dies tut, hat die Nutznießer des alten Zustandes zu Feinden, während er in den möglichen Nutznießern des neuen Zustandes nur lasche Verteidiger findet.

Niccolò Machiavelli

Je planmäßiger die Menschen vorgehen, desto wirksamer vermag sie der Zufall zu treffen.

Friedrich Dürrenmatt

Nullus propheta in patria: Der Prophet gilt nichts im eigenen Land.

Spruchwort

© 2022, *Claudio Franzetti, Im Koller 34, 8706
Meilen*

Vorwort zur 2. Auflage

Die vorliegende zweite Auflage dieses Buches ist nicht so sehr eine Korrektur oder Anpassung des schon Bestehenden, sondern viel mehr eine Erweiterung. Der Zahn der Zeit nagt nicht so schnell am Thema, es gibt aber immer wieder Inhalte, die an Bedeutung gewinnen. Ein solches Thema ist das Metaversum, das als neues Internet propagiert wird.

Treiber der Entwicklung sind vor allem zwei: der technologische Fortschritt, der bisher Rechen- und Speicherkapazität sowie die Transistorendichte kontinuierlich unaufhaltsam hat wachsen lassen. Zum anderen werden die jugendlichen Tech-Giganten ständig herausgefordert und müssen neue Geschäftsfelder erfinden, um ihr unheimliches Wachstum fortzusetzen.

Die Gesellschaft kann nur teilweise Schritt halten, wobei eine typische Altersabhängigkeit der Anpassungsfähigkeit erkennbar ist. Im Bereich der digitalen Transformation, besonders im Hinblick auf die öffentliche Verwaltung, sind in den vergangenen vier Jahren keine großen Fortschritte festzustellen gewesen, zumindest was bis zum Benutzer durchdrang. Wieso die Verwaltung nicht mitkommt, habe ich etwas ausführlicher illustrieren wollen. Man darf nicht vergessen, dass die Verwaltung im Wesentlichen die Gesetze umsetzt und diese immer differenzierter und damit aufwendiger werden.

Aufgrund der geopolitischen Verwerfungen im Jahr 2022 fällt das Augenmerk vermehrt auf die Einflussphären der technologischen Großmächte. Europa, die Alte Welt, scheint im Verhältnis zum Silicon Valley wenig innovativ, wenn man von der Regulierung absieht. Um selber die Technologie besser zu fördern, ist ein Verständnis vom Valley und seiner Weltsicht wichtig und deshalb hier stärker thematisiert. Das Valley in Europa nachzubilden, funktioniert aus grundlegenden Verschiedenheiten nicht.

Die Erweiterung des Buchs musste mit einer Umstellung der Kapitel einhergehen, um das Neue in einen konsistenten Ablauf einzubetten. Die Kennzeichnung der Abschnitte mit † habe ich aufgegeben, den Stern * behalten.

Zürich
März 2023

Claudio Franzetti

Vorwort

Dieses Buch richtet sich an Anwender und Benutzer, die ich einerseits als Schüler und Lehrer der Sekundarstufe und andererseits als typische Mitarbeiter von Nicht-IT-Unternehmen verstehe. Das Publikum ist also sehr breit.

Meine Kinder sind etwa in diesem Alter. Ich muss allerdings feststellen, dass sie zwar ein Fach „Informatik“ belegen, aber dass sich die Themen eher am Maschinenschreiben orientieren. Bei Gruppenarbeiten (kollaboratives Arbeiten) oder beim Vokabellernen, z. B. am Smartphone, vermisse ich jede effiziente IT-Unterstützung.

In meiner aktuellen Firma höre ich ständig vonseiten der Informatiker, die Mitarbeiter verstanden sie nicht, man würde aneinander vorbeireden. Bei einer Funktionsstörung will der Benutzer diese so schnell wie möglich behoben sehen, die Erklärungen werden erduldet und nicht genutzt, sich weiterzubilden.

Das wahrgenommene Desinteresse wird als Inkompetenz interpretiert. Dadurch verringert sich die Anstrengung, systematisch zu informieren. Dadurch erhöhen sich die widersprüchlichen Mitteilungen und verstärken die unklaren Äußerungen der Benutzer. Dadurch erhöht sich die wahrgenommene Inkompetenz der Benutzer durch die Techniker ... und weiter so.

Es scheint mir äußerst wichtig, den Mitarbeitenden die Möglichkeit zu eröffnen, sich stressfrei ein wenig Informatikwissen anzueignen und damit Anhaltspunkte zu gewinnen, um mit den Technikern zu reden.

Besonders in kleineren Firmen ist die Wahrscheinlichkeit groß, als Fachperson mit IT-Entwicklern kommunizieren und in Projekten zusammenarbeiten zu müssen. Deshalb ist es wichtig, gewisse Konzepte, Prinzipien und gewisse Zusammenhänge verstehen zu können.

Somit habe ich die Gelegenheit wahrgenommen, dieses Buch herauszugeben. Wie in der IT üblich, ist dem Vorhaben eine „Make-or-Buy“-Entscheidung vorangegangen. Ich habe das Angebot studiert, wobei auch und vor allem Publikationen für den Schulgebrauch im Brennpunkt standen. Keines hat mich davon überzeugt, kein eigenes Buch schreiben zu müssen. Deshalb habe ich die Herausforderung – *façon de parler* – angenommen.

Die Qualifikation für den Inhalt dieses Buches habe ich mir in langjähriger Beschäftigung mit Computern redlich erworben: Anfang der 1980er-Jahre im Ingenieurstudium an der ETH mit Einführung in Pascal und Rechnen auf der CDC7600 mit dem exotischen Betriebssystem „Venus“, dann als Forschungsingenieur mit Strömungssimulationen in Fortran auf der IBM 7600VF, Vax 3000, Silicon Graphics IRIS und dem damaligen Superrechner Cray-1 in Lausanne. Als Finanzfachmann entwickelte ich Datenbanken und Derivatepreise für Banken. In höheren Führungspositionen blieb IT mehr als ein Hobby. Wieder bei einer Kleinfirma als Geschäftsführer entwickelten wir Software für die Versicherungsindustrie. Und schließlich war ich für die IT-Entwicklung in der jetzigen Firma verantwortlich.

Immer war es mein Ziel, produktive Systeme einzuführen, beispielsweise das Wiki „Confluence“ oder das Issue-Tracking namens „Jira“.

Es ist mein lebhafter Wunsch, dass diese Schrift zu Diskussionen Anlass geben wird. Denn dies ist ein Zeichen, dass sie gebraucht wird.

Im Vorspann sind mehrere Zitate aufgeführt. Die einen sollen Skepsis bezüglich der Planbarkeit von IT-Entwicklungen ausdrücken, denn meiner Meinung nach sind diese ingenieurmäßigen Vorgehensweisen nicht immer problemadäquat. Dies wird in einem Kapitel thematisiert. Das Machiavelli-Zitat soll auf die Schwierigkeit hinweisen, der IT-Verantwortliche begegnen, wenn sie ihrer natürlichen Rolle als Neuerer nachkommen. IT-Entwicklung ist ein riskantes Geschäft.

Der Text ist gespickt mit Beispielen und Illustrationen. Hier versuche ich, möglichst keine proprietären Systeme zu zeigen, sondern auf frei verfügbare Ressourcen zu setzen. Damit kann die oder der Lesende selber Methoden nachvollziehen, damit spielen oder erweitern.

Der Text ist nicht primär darauf ausgelegt, das Programmieren zu vermitteln. Er möchte aber grundlegende Sachverhalte anhand einer Blocksprache wie Scratch aufzeigen, die dem Publikum den Einstieg in die Vertiefung ermöglichen. Die Absicht geht also viel mehr in die Breite als in die Tiefe.

Einige Kapitel sind mit einem * versehen. Diese sind im Vergleich zum Haupttext möglicherweise etwas anspruchsvoller oder setzen gewisse Kenntnisse voraus, die für die IT aber nicht ausschlaggebend sind. Abschnitte mit † sind für Mitarbeiter gedacht und weniger für Schüler, die den betrieblichen Alltag noch nicht kennen.

Zuletzt möchte ich allen danken, die an dieser Publikation mitgewirkt haben. Mein spezieller Dank geht an meinen Kollegen Dipl.-Ing. Claus Frauenheim, der sowohl gute inhaltliche Anregungen gegeben als auch das Manuskript genau gelesen hat. Fehler sind ausschließlich dem Autor anzulasten.

Inhaltsverzeichnis

1 Einführung	1
Literatur.....	2
2 Ganz kurze Geschichte der IT	3
2.1 Rechner.....	3
2.1.1 Erste Generation.....	4
2.1.2 Zweite Computergeneration.....	5
2.1.3 Dritte Computergeneration.....	5
2.1.4 Vierte Computergeneration.....	6
2.2 Software.....	7
Literatur.....	9
3 Alles ist Zahl	11
3.1 Zweier- oder Binärsystem.....	11
3.2 Sechzehner-System.....	14
3.3 Gleitkommazahlen*.....	14
3.4 Rohe Daten, Alphabete.....	14
3.5 Komprimierung und Verifizierung.....	15
3.6 Einfache Formallogik*.....	19
3.7 Rechenschaltungen*.....	20
Literatur.....	24
4 Computer als Hardware	25
4.1 Zentrale Einheiten.....	26
4.1.1 Halbleitertechnik*.....	26
4.1.2 Hauptplatine.....	27
4.1.3 Prozessor.....	29
4.1.4 Random Access Memory.....	30
4.1.5 Anschlüsse und Kabel.....	30
4.2 Smartphones.....	32
4.2.1 System-on-Chip.....	33

4.2.2	Eingebaute Sensoren	33
4.3	Periphere Geräte, Eingabe und Ausgabe	35
4.4	Sensoren, IoT	35
4.5	Körpergeräte, Wearables	37
4.6	Quantencomputer	39
	Literatur	40
5	Betriebssystem und Benutzeroberfläche	43
5.1	Operating Systems	44
5.2	Oberfläche – Benutzerkommunikation	45
5.3	Benutzer-Interaktion	46
5.4	Dateiverwaltung – Filesystem	48
5.5	Geräteverwaltung	49
5.6	Virtuelle Maschinen	50
5.7	Eingebettete Systeme	51
6	Netzwerke	53
6.1	Typen	54
6.2	Client-Server	55
6.3	Schichtenmodell und Protokolle	58
6.4	Internet, World Wide Web	59
6.5	Webservices, APIs	62
6.6	Deep Web und Darknet	64
	Literatur	66
7	Datenorganisation	67
7.1	Datentypen und -strukturen	68
7.2	Versionierung	69
7.3	Relationale Datenbanken	71
7.4	NoSQL-Datenbanken	76
7.5	Markup-Sprachen	77
7.5.1	XML, HTML und JSON	78
7.5.2	LaTeX	85
7.6	Datensicherung, Backup	87
	Literatur	89
8	IT-Entwicklung	91
8.1	Herausforderungen	91
8.2	Geschäftsprozesse und IT-Entwicklung	93
8.3	Problemadäquate Methoden	95
8.4	Monolithisch versus Service	96
8.5	Planmethoden, Wasserfall	98
8.6	Agile Methoden	102
8.7	Open Source	105

8.8	Diskussion	108
	Literatur	112
9	IT-Organisation	113
9.1	Strategie der IT	113
9.1.1	IT als Unterstützungsfunktion	114
9.1.2	IT als Erfolgsfaktor	114
9.1.3	IT als Disruptor	115
9.2	Management der Informationssysteme	115
9.3	IT-Service-Management	117
	Literatur	120
10	Programmierung	121
10.1	Voraussetzungen	123
10.2	Philosophie	125
10.3	Compiler und Interpreter	126
10.3.1	Grammatik*	127
10.4	Einfache Blocksprache – Einführung Programmieren	128
10.4.1	Schleifen, Parameter	129
10.4.2	Variablen, Unterprogramme	131
10.4.3	Listen, Operatoren	133
10.4.4	Bedingte Anweisungen	134
10.4.5	Weitere Beispiele	136
10.5	Hoch-Sprachen	138
10.5.1	Prozedurale Sprachen	143
10.5.2	Objektorientierte Sprachen	145
10.5.3	Neuere Trends	148
10.6	Skript-Sprachen	150
10.6.1	Kommandosprachen	150
10.6.2	Eingebettete Sprachen	151
10.7	Und Tabellenkalkulatoren?	153
10.8	Umgebung und Tools	154
10.8.1	Integrierte Entwicklungsumgebung	154
10.8.2	Entwicklungs-Tools	155
10.9	Architektur und Muster	158
10.10	Composable Model, MACH-Architektur	161
	Literatur	165
11	Applikationen	167
11.1	Systematik	167
11.2	Office: Schreibmaschine 4.0	168
11.3	Generische Anwendungen	173
11.3.1	Webbrowser	173
11.3.2	Suchmaschinen	175

11.3.3	E-Mail	177
11.3.4	PDF-Reader	182
11.3.5	Wiki	183
11.3.6	VoIP, Video-Telefonie, Instant Messaging	183
11.4	Geschäfts-Anwendungen	183
11.4.1	Desktop Publishing	184
11.4.2	Issue-Tracking-System	185
11.4.3	Customer Relationship Management, CRM	187
11.4.4	Content und Data Management Systeme	187
11.4.5	Analytische Informationssysteme	188
11.4.6	E-Commerce-Systeme	189
11.5	Hobby-Anwendungen	190
11.5.1	Gimp, PhotoShop	190
11.5.2	Audacity, LMMS, GarageBand	191
	Literatur	194
12	Cloud Computing	195
12.1	Infrastruktur	197
12.2	Plattform	198
12.3	Services	199
12.4	Architektur: Cloud, Fog, Edge	199
12.5	Elastizität	200
	Literatur	201
13	Sicherheit	203
13.1	Verschlüsselung	203
13.2	Identität und Echtheit	205
13.2.1	Identität	205
13.2.2	Echtheit	207
13.3	Zugang	208
13.3.1	Passwörter	209
13.3.2	Firewall	210
13.4	Hacking und Malware	211
13.4.1	Hacking	211
13.4.2	Malware, Botnets	214
13.4.3	„Epidemiologie“	214
	Literatur	216
14	Information und Medien	217
14.1	Kommunikation	217
14.1.1	Mensch zu Mensch	218
14.1.2	Mensch-Maschine	220
14.1.3	Maschine zu Maschine	221

14.1.4	Entropie*	221
14.2	Desinformation, Propaganda, Fake News	223
14.3	Soziale Medien	224
14.4	Medienkompetenz	226
14.4.1	Gefahren	227
14.4.2	Chancen	230
14.5	Werbung, Influencing, Monetarisierung	230
14.5.1	Werbung, Marketing	231
14.5.2	Influencing	233
14.5.3	Monetarisierung	233
	Literatur	235
15	Künstliche Intelligenz	237
15.1	Roboter	238
15.2	Virtual und Augmented Reality	239
15.3	Arten von Künstlicher Intelligenz	243
15.3.1	Eine bittere Erkenntnis	244
15.4	Algorithmen und Machine Learning	246
15.4.1	Intelligenz und Lernen	248
15.4.2	Interview mit einem Chatbot	251
15.4.3	KI der Emotionen	255
15.4.4	Algorithmen und Bionik	256
15.4.5	Genetischer Algorithmus*	258
15.4.6	Neuronale Netze	260
15.5	Super-Intelligenz	265
	Literatur	266
16	Virtuelle Welten	267
16.1	Blockchain	267
16.1.1	Hashing	268
16.1.2	Konsens und Prüfung	270
16.1.3	Smart Contracts	274
16.1.4	Vertrauens-Business oder „Distributed Ledger“	276
16.1.5	Kryptowährungen	278
16.2	Gaming	279
16.2.1	Hardware	279
16.2.2	Spieleentwicklung	281
16.3	Simulation	282
16.4	Tokens, NFT	285
16.5	Dezentrale Autonome Organisationen	288
16.6	Digitale Zwillinge	289
16.6.1	Dinge, Things	289
16.6.2	Smart Cities, Regions	291

16.6.3	Human Digital Twins	292
16.7	Metaversum	293
16.7.1	Geschichte	293
16.7.2	Problematiken	302
	Literatur	303
17	IT, Digitalisierung und Gesellschaft	305
17.1	Die verwaltete Welt	305
17.2	Ökonomische Transformation	307
17.2.1	Plattformen, Ökosysteme	308
17.3	Einfluss auf verschiedene Bereiche	309
17.3.1	Arbeitswelt	309
17.3.2	Finanzwesen, Banken	310
17.3.3	Behörden	311
17.3.4	Schule	317
17.3.5	Wissenschaft	319
17.3.6	Gesundheitswesen	322
17.3.7	Gerichtswesen	322
17.3.8	Tägliche Verrichtungen	323
17.4	Soziale Aspekte	325
17.4.1	Gesundheit	325
17.4.2	Aus- und Weiterbildung	326
17.4.3	Gesetzgebung	328
17.4.4	Überwachung	331
17.5	Ökonomische Aspekte	334
17.5.1	Produktivität	334
17.5.2	Qualität	335
17.5.3	Arbeit und Steuern	335
17.6	Ökologische Aspekte	338
17.6.1	Rohstoffe	339
17.6.2	Energie	340
17.7	Technologische Aspekte	341
17.7.1	Rechenleistung	341
17.7.2	Big Data, Datenanalyse	342
17.7.3	Quantenrechner*	343
17.8	Warnungstafeln	346
17.8.1	Weltanschauung	346
17.8.2	Gegengewichte	352
	Literatur	354
	Installation Scratch	357
	Stichwortverzeichnis	361

Abbildungsverzeichnis

Abb. 1.1	Übersicht Geschäfts-IT	2
Abb. 2.1	Curta-Rechner	4
Abb. 2.2	UNIVAC I Röhrenrechner	5
Abb. 2.3	Personal Computer von IBM	6
Abb. 3.1	Datei-Erkennung	12
Abb. 3.2	Lydische Schrift	15
Abb. 3.3	PNG-Datei im Editor	16
Abb. 3.4	Kontaktschaltungen	22
Abb. 3.5	Addierer-Schaltungen	22
Abb. 3.6	XOR-Schaltung	23
Abb. 3.7	4-Bit-Volladdierer	23
Abb. 3.8	IC-Gehäuse	24
Abb. 4.1	Modell Zentraleinheit	26
Abb. 4.2	Diode und Transistor	28
Abb. 4.3	AND/NAND-Schaltung	28
Abb. 4.4	RAM-Speicher	30
Abb. 4.5	CPU-RAM-Verbindung	31
Abb. 4.6	Modell Zentraleinheit	31
Abb. 4.7	Offenes Smartphone	34
Abb. 4.8	Körpersensoren	38
Abb. 4.9	Beispiel Exoskelett	39
Abb. 5.1	Schichtenmodell	44
Abb. 5.2	Betriebssysteme	44
Abb. 5.3	Anteile der Betriebssysteme	45
Abb. 5.4	Eingabeaufforderung	46
Abb. 5.5	Desktop-Bildschirm	47
Abb. 5.6	Die drei typischen Eingabe-Akteure	48
Abb. 5.7	Die zwei Typen von virtuellen Maschinen	51
Abb. 6.1	Das ARPANET ganz zu Beginn im Jahr 1969	53
Abb. 6.2	ARPANET im Jahr 1977	54

Abb. 6.3	Schematische Darstellung von drei Clients mit einem Server	56
Abb. 6.4	Client-Server-Architekturen	57
Abb. 6.5	Zeitverlauf zur Client-Server-Architektur.	57
Abb. 6.6	OSI-7-Schichtenmodell.	58
Abb. 6.7	TCP/IP-Schichten	59
Abb. 6.8	Verpackungsmodell.	60
Abb. 6.9	Domain-Namen.	61
Abb. 6.10	Berechtigter Zugriff mit API mittels Authorization	63
Abb. 6.11	Struktur des Webs	64
Abb. 7.1	Ablagestruktur des Betriebssystem Windows.	68
Abb. 7.2	Vierstufiger Baum.	69
Abb. 7.3	Textverarbeitung in der Cloud	70
Abb. 7.4	Wiki und Versionierung	70
Abb. 7.5	Nicht normalisierte Verknüpfung	73
Abb. 7.6	Auflösung einer N:M-Beziehung	74
Abb. 7.7	Zuordnungstabelle bei N:M-Beziehung	74
Abb. 7.8	ER-Diagramm zum Bibliothekenverbund.	75
Abb. 7.9	Selbtsreferentielle Tabelle.	76
Abb. 7.10	Quellcode einer japanischen HTML-Seite	83
Abb. 7.11	Ausschnitt aus ShareLatex	86
Abb. 7.12	Schema einer betrieblichen Datensicherung.	88
Abb. 8.1	Wesentliche Funktionen für die IT-Entwicklung	92
Abb. 8.2	Ein BPMN-Prozess	93
Abb. 8.3	Ein BPMN-Prozess	97
Abb. 8.4	Verfahrensmethoden	98
Abb. 8.5	Phasenkonzept der NASA.	99
Abb. 8.6	Projekt-Abweichungsgrafik	100
Abb. 8.7	Phasenmodell von Siemens.	100
Abb. 8.8	Übersicht Projektmanagement	102
Abb. 8.9	Backlog	103
Abb. 8.10	Backlog im Issue-Tracker	104
Abb. 8.11	Beispiel Backlog.	105
Abb. 8.12	Open-Source-Entwicklungshierarchie	107
Abb. 8.13	Zielerreichung nach Methode	108
Abb. 8.14	Unterschiedliche Ziele der zwei Verfahrensmethoden	110
Abb. 9.1	Überblick FitSM.	118
Abb. 10.1	Schritte der Problemlösung.	122
Abb. 10.2	Abstraktion als Brücke	124
Abb. 10.3	Compiler und Linker.	126
Abb. 10.4	Flussdiagramm zum Programm rtbis	144
Abb. 10.5	Darstellung der Simulation mit Scratch	147
Abb. 10.6	VBA-Programmierungsumgebung.	153

Abb. 10.7	Beispiel Atom-IDE	155
Abb. 10.8	Beispiel Anwendungsfalldiagramm	156
Abb. 10.9	Beispiel Sequenzdiagramm.	157
Abb. 10.10	Beispiel Zustandsdiagramm	157
Abb. 10.11	Zustandsdiagramm: Theologie	158
Abb. 10.12	Übersicht MVC.	161
Abb. 10.13	Das Headless-Muster mit APIs.	164
Abb. 11.1	Synopsis Software.	168
Abb. 11.2	Typisches Aussehen eines Tabellenkalkulations-Programms.	169
Abb. 11.3	Typisches Aussehen eines Textprogramms.	169
Abb. 11.4	Spracherkennung von Google Drive.	170
Abb. 11.5	Spracherkennung auch in Fremdsprachen	171
Abb. 11.6	Formatvorlagen für Word in der Entwurfsansicht.	171
Abb. 11.7	Vergleich von Raster- und Vektordarstellung	172
Abb. 11.8	Screenshot Business-App	173
Abb. 11.9	Die Architektur eines Browsers	174
Abb. 11.10	Einstellungen des Opera-Browsers.	175
Abb. 11.11	Desktop-Publishing mit der freien Software „Scribus“	185
Abb. 11.12	Editieren einer Pdf-Datei mit Okular	186
Abb. 11.13	Zustandsdiagramm	186
Abb. 11.14	Schematischer Handel, direkt und über Marktplatz	190
Abb. 11.15	Manipulierte Fotografie	191
Abb. 11.16	Spielereien an einem Foto mit GIMP.	192
Abb. 11.17	Benutzer-Interface von Audacity, einem Audio-Editor.	192
Abb. 11.18	Ausschnitt aus dem User Interface von LMMS	193
Abb. 12.1	Cloud-Computing	196
Abb. 12.2	Cloud Konfiguration	197
Abb. 12.3	Cloud, Fog und Edge	200
Abb. 13.1	Asymmetrische Verschlüsselung.	204
Abb. 13.2	Daten und Identität	205
Abb. 13.3	Signieren von Dokumenten.	208
Abb. 13.4	Firewall	210
Abb. 14.1	Selbstvertrauen und Wissen	220
Abb. 14.2	Das Kommunikationsmodell von Shannon, erweitert	221
Abb. 14.3	Benutzer-Rennliste der Sozialen Medien	225
Abb. 14.4	Behaviorismus schematisch	232
Abb. 15.1	Kleiner Lernroboter Ozobot	239
Abb. 15.2	Realität als Kontinuum	240
Abb. 15.3	Video-Anprobe einer Brille.	241
Abb. 15.4	AR-Tourismus-Applikation	242
Abb. 15.5	Frühe VR-Brille	242
Abb. 15.6	Die Grenze zwischen Mensch und Maschine.	244

Abb. 15.7	Maschinelles Lernen im Kontext	247
Abb. 15.8	Lernen in zwei Kontexten	247
Abb. 15.9	KI-Bilder	249
Abb. 15.10	Dell-e-2-Beispiele	250
Abb. 15.11	Koenigsberger Brücken	257
Abb. 15.12	Lernen in zwei Kontexten	258
Abb. 15.13	Künstliches Chromosom	259
Abb. 15.14	Crossing-Over	259
Abb. 15.15	Generelle Fragestellung als Black-Box	260
Abb. 15.16	Künstliches Neuronales Netz	261
Abb. 15.17	Aktivierungsfunktionen	262
Abb. 15.18	Input-Probieren	262
Abb. 15.19	Körpererkennung	263
Abb. 15.20	Neuronales Netz zum Probieren	264
Abb. 16.1	Hashfunktion	269
Abb. 16.2	Ein Merkle-Baum oder Hash-Tree	270
Abb. 16.3	Blockchain	271
Abb. 16.4	Mining-Algorithmus als Flowchart	272
Abb. 16.5	Technologieverlauf	276
Abb. 16.6	Verteilte Buchhaltung im Vergleich	277
Abb. 16.7	Transaktion mit Bitcoin	279
Abb. 16.8	Beschleunigtes Rechnen dank der Grafikkarte	280
Abb. 16.9	Raumleiter Hermes	283
Abb. 16.10	Die CRAY-1 von Lausanne	283
Abb. 16.11	Monte Carlo Simulation	284
Abb. 16.12	Geldarten und ihre Grundlage	286
Abb. 16.13	Schematische Darstellung einer realen Komponente und seinem Zwilling	290
Abb. 16.14	Erstellung eines digitalen Zwillings	290
Abb. 16.15	Smart City als Digital twin	291
Abb. 16.16	Ad-hoc-Avatar in Decentraland	296
Abb. 16.17	Graphischen Übersicht der Zutaten des Metaversums	297
Abb. 16.18	Zwilling der Erde	299
Abb. 16.19	Modell eines Kopfes in Blender	300
Abb. 16.20	Körperbewegungen am Modell	301
Abb. 16.21	Eine Komposition mit Blender	301
Abb. 16.22	Einfaches Modell der „Realität“	302
Abb. 17.1	Ausdifferenzierung und Digitalisierung	306
Abb. 17.2	Digitale Transformation von Unternehmen	307
Abb. 17.3	Sternarchitektur der Behördenprozesse	312
Abb. 17.4	Gekammerte Amtsdaten	313
Abb. 17.5	Behördengänge automatisiert	313

Abb. 17.6	Alte Aktenvorschrift	314
Abb. 17.7	Geoinformationssystem mit Lärmbelastung.	316
Abb. 17.8	Shapefiles darstellen	316
Abb. 17.9	Digitalisierte Bücher, Beispiel	321
Abb. 17.10	Ein möglicher Patientenpfad.	323
Abb. 17.11	MOOC-Angebote	327
Abb. 17.12	Profiling.	329
Abb. 17.13	Spuren von Fitness-Geräten	334
Abb. 17.14	Moore'sches Gesetz	341
Abb. 17.15	Big-Data schematisch	342
Abb. 17.16	Auto als Datenquelle.	343
Abb. 17.17	Quantenrechner-Architektur	345
Abb. A.1	Anmeldung Scratch. Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu , abgerufen am: 1. November 2018	358
Abb. A.2	Entwicklungsumgebung von Scratch	358

Tabellenverzeichnis

Tab. 3.1	Bitmuster und Interpretation.	13
Tab. 5.1	Treiberliste	49
Tab. 6.1	Die wichtigsten Arten von Servern.	56
Tab. 7.1	Einfache Datentypen.	69
Tab. 8.1	Projektsituationen.	99
Tab. 8.2	Größten Open-Source-Projekte	106
Tab. 8.3	Erfolgsquote nach Verfahrensmodell	108
Tab. 8.4	Das Bimodalmodell von Gartner	110
Tab. 10.1	Drei Fischobjekte von Scratch	149
Tab. 12.1	Die 10 größten Cloudanbieter.	196
Tab. 14.1	Influencer im Bereich Mode	233
Tab. 15.1	Mit KI erzeugter Text nach Maßgabe von Stichworten	251
Tab. 16.1	Die schnellsten Computer der Welt	285
Tab. 16.2	Größte Anteile an verschiedenen Metaverse-Aktienfonds	298
Tab. 17.1	Auszug aus dem API-Katalog des britischen Zolls	315
Tab. 17.2	Rangliste der grössten Unternehmen 2022.	336
Tab. 17.3	Zukunft der Berufe	351



Zu Beginn eine Begriffsklärung (oder das pure Gegenteil): Im Titel dieser Broschüre steht „Informatik“. Viele denken aber automatisch an „IT“, das für Informationstechnologie steht, aber auch „Informationstechnik“ abkürzt. In letzter Zeit hat sich noch der Begriff „IKT“ für Informations- und Kommunikationstechnologie eingeschlichen.

Wir finden, dass die Kommunikation, das Zur-Verfügung-Stellen und Übermitteln von Informationen, eigentlich im Überbegriff Information schon enthalten ist. Eine Definition von Kommunikation lautet: Austausch von Information über Netzwerke.

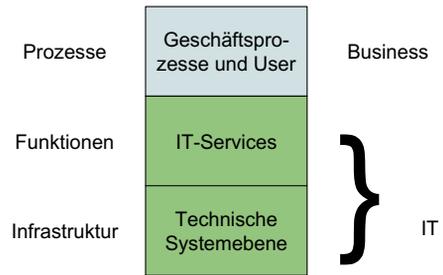
Der Unterschied zwischen Technik und Technologie ist der Unterschied zwischen Anwendung (Technik) und Forschung, Entwicklung und Wissenschaft (Technologie). Letztere kann man wiederum mit Informatik, auf englisch „Computer Science“ gleichsetzen.

Da man auf dem Papier eher Theorie denn Praxis üben kann, sollte der Begriff Informatik eigentlich recht gut passen. Denn es soll der Zweck des Buches sein, ein paar grundlegende Tatsachen in Erinnerung zu rufen, die für den täglichen Gebrauch der Informationstechnik hilfreich sein können. Zudem soll sie helfen, das Verständnis zwischen IT-Spezialisten und IT-Anwendern zu verbessern.

Die Abb. 1.1 zeigt schematisch den Zusammenhang zwischen Geschäftstätigkeit und IT, wobei Letztere in Infrastruktur und Funktionen oder Dienste aufgeteilt ist. Die Schnittstelle zwischen Business und IT ist eine Problemzone. Denn bei der Weiterentwicklung der Dienste sind es ja Menschen, die miteinander eine Lösung erarbeiten müssen, auch wenn sie in ganz unterschiedlichen Modellen und Vorstellungen geschult worden sind.

Dieses Hindernis kann man durch Kennenlernen der jeweils anderen „Welt“ senken. Für die Geschäftsseite heißt dies, sich über IT einen gewissen Überblick zu verschaffen.

Abb. 1.1 Übersicht Geschäfts-IT. Die IT kann unterstützend oder gestaltend sein. Zweites ist besonders in disruptiven Zeiten der Fall, in denen der Fortschritt das Business bestimmt



Im Marketing gilt der Spruch: „People don’t know what they want until they know what they can get.“¹ Dies gilt häufig auch für die Geschäftsseite einer Unternehmung. Meistens bleibt man beim Bekannten haften. Vom Autopionier Henry Ford ist der Spruch überliefert: „Wenn ich die Menschen gefragt hätte, was sie wollen, hätten sie gesagt: schnellere Pferde.“ Neue Situationen, neue Technologien, neue Geschäftsmodelle sind problematisch. Der Physiker Max Planck sagte über die Physik (Planck, 1955, 22):

„Eine neue wissenschaftliche Wahrheit pflügt sich nicht in der Weise durchzusetzen, daß ihre Gegner überzeugt werden und sich als belehrt erklären, sondern vielmehr dadurch, daß ihre Gegner allmählich aussterben und daß die heranwachsende Generation von vornherein mit der Wahrheit vertraut gemacht ist.“

Wer sich mit Innovation im Geschäftsumfeld befasst, muss die Persönlichkeit besitzen, gegen Widerstände anzukämpfen. Schon Niccolò Machiavelli, ein Philosoph und Dichter, hat im fünfzehnten Jahrhundert die Problematik der Innovatoren beschrieben, wie es im Zitat ganz am Anfang angedeutet ist.

Literatur

Planck, M. (1955). *Wissenschaftliche Selbstbiographie mit der von Max von Laue gehaltenen Traueransprache*. Lebensdarstellungen deutscher Naturforscher. Berlin: J.A. Barth.

¹ Die Leute wissen erst was sie wollen, wenn sie erkennen, was sie haben können.



Die Geschichte der IT ist aus einer technologischen Sicht vor allem die Geschichte der Computer (Rechner, Hardware) und der Programme (Software). Notwendige Vorarbeiten in Mathematik und Logik waren etwa die Zahlendarstellung im Zweiersystem und die entsprechende Algebra der Logik von George Boole. Diese ist ebenfalls zweiwertig, denn Aussagen sind entweder wahr oder falsch. In den 1930er-Jahren stellte Alan Turing ein vollständiges (theoretisches) Modell eines Universalrechners vor.

An der Zweiteilung in Hard- und Software halten wir fest.

2.1 Rechner

Das Rechnen, bestehend aus den vier Grundrechenarten Addition, Subtraktion, Multiplikation und Division, wurde schon seit Tausenden von Jahren im Osten und im Westen durch den Zählrahmen oder Abakus unterstützt. Mit der Einführung der arabischen Zahlen, die allerdings aus Indien stammen, und der Erfindung der Logarithmen im 17. Jahrhundert kam der Rechenschieber auf. Die nächste Etappe bildeten die mechanischen Rechenmaschinen, deren erste von Wilhelm Schickard 1632 entwickelt wurde, gefolgt von der Pascaline von Blaise Pascal von 1642. Im Jahr 1676 legte Gottfried Wilhelm Leibniz seinen Rechner mit der Staffelwalze vor. Vor allem Uhrmacher wie Jean-Baptiste Schwilgué vom Straßburger Dom besaßen die feinmechanischen Fertigkeiten, um komplizierte Rechner (und Astrolabien) zu bauen. Um 1850 wurde der erste mechanische Rechner von Charles Xavier Thomas industriell gefertigt. Der letzte mechanische Rechner ist die Curta von Curt Herzstark, der bis 1945 gebaut wurde und ein 15-stelliges Resultatwerk aufwies. Er sah wie eine kleine Kaffeemühle aus (siehe Abb. 2.1).

Programmierbare Rechenmaschinen brauchen einen Speicher, der als gestanzte Metallplatte vom Weber und Erfinder Joseph-Marie Jacquard eingeführt wurde, um komplizierte Muster zu weben. Charles Babbage stellte 1833 das funktionsfähige Konzept vor, das er

Abb. 2.1 Curta-Rechner
(Wikipedia, 2018), gemeinfrei



in der „Analytical Engine“ zu realisieren versuchte, allerdings an Geld und Zeit scheiterte. Diese Maschine sollte die Anweisungen aus den gestanzten Platten einlesen. Ada Lovelace, Tochter des Dichters Lord Byron, beschäftigte sich schon früh mit Mathematik. Sie begeisterte sich für die technischen Erfindungen und lernte mit 16 Jahren Babbage kennen. Für seine Maschine, Vorläufer des modernen Computers, entwickelte sie ein komplexes Programm und gilt so als erste Programmiererin der Welt. Sie gab der Programmiersprache ADA den Namen.

Herman Hollerith verwendete als Statistiker der US-Volkszählung 1890 Lochkarten und entsprechende Auswertemaschinen, um dem wachsenden Datenanfall Herr zu werden. Aus seiner Firma ging später IBM hervor, kurz für International Business Machines.

2.1.1 Erste Generation

Die Rechner der 1940er-Jahre fußten auf elektromechanischen Relais oder Elektronenröhren. Ihre Anzahl konnte man an einer Hand abzählen. Sie waren riesig, schwer, störanfällig und teuer, sodass die Einsatzmöglichkeiten sehr eingeschränkt blieben. Ein Repräsentant ist ENIAC (Electronic Numerical Integrator And Computer), der 1945/46 in den USA fertiggestellt wurde. Den ersten elektronischen Rechner zu nennen, fällt schwer. Im Jahr 1941 allerdings brachte Konrad Zuse die Z3 zum Laufen und auch in den USA nahmen John Atanasoff und sein Doktorand einen Computer in Betrieb.

Dieser brauchte ca. 18.000 Röhren, benötigte eine Fläche von 150 qm und wog 30 t. Die Steuerung der Maschine (Programmierung) wurde durch das Stellen von Hunderten von Schaltern und das Stecken von Kabelverbindungen erreicht. Zur selben Zeit entwickelte John von Neumann (1903–1957) das Grundkonzept moderner Rechner.

Der erste kommerzielle Computer der USA wurde 1951 in Betrieb genommen, die sogenannte UNIVAC I (UNIVersal Automatic Computer I) von Abb. 2.2.

Abb. 2.2 UNIVAC I
Röhrenrechner. Foto: Matthias
Kirschner, gemeinfrei,
(Wikipedia, 2016)



Ein wesentliches Prinzip ist, dass das Programm genauso wie die Daten im Speicher abgelegt wird. Damit ist nicht mehr nur ein starrer Ablauf möglich, sondern das Programm kann geändert werden und im Extremfall sogar sich selbst verändern.

2.1.2 Zweite Computergeneration

Elektronenröhren schalteten zwar viel schneller als elektromechanische Relais, doch sie waren teuer, produzierten einen Haufen Wärme, nahmen viel Platz weg und hatten eine geringe Lebensdauer. Im Jahr 1948 erfand William Shockley den Transistor und wurde später mit dem Nobelpreis für Physik geehrt.

Etwa von 1955 bis 1965 wurden Elektronenröhren zunehmend durch Transistoren ersetzt, einem Schaltelement aus Silizium, Gallium und Arsen. Als Hauptspeicher dienten Magnetkernspeicher, erste Magnetbänder und Magnetplatten. In den Jahren 1958/59 entwickelte die Firma Texas Instruments den ersten integrierten Schaltkreis. Und 1960 wurde die Programmiersprache ALGOL-60 vorgestellt.

2.1.3 Dritte Computergeneration

In den frühen 1970er-Jahren wurden mehrere Transistoren und Bauelemente zu Schaltgruppen zusammengefasst. Die Rechner wurden durch *Betriebssysteme* gesteuert. Es entstanden erste Rechenzentren.

Computerbenutzer gaben ihren Auftrag (Job) auf und mussten auf die Aushändigung der Resultate warten. Typisch war die Eingabe eines Programms auf Lochkarten. Später wurden Terminals angeschlossen, an denen Benutzer Befehle direkt eingeben konnten. Da dann mehrere Programme gleichzeitig abgearbeitet wurden, mussten die Betriebssysteme das *Timesharing* beherrschen.

2.1.4 Vierte Computergeneration

Riesige Fortschritte in der Halbleitertechnik, besonders der Miniaturisierung, ermöglichten eine immer höhere Integrationsdichte, d. h. Schalter pro Fläche. Dadurch wurden sogenannte „Mikrocomputer“ möglich und für Private langsam erschwinglich. Im Jahr 1977 kam der sehr preiswerte Apple II auf den Markt, der von einer einzigen Person, Steve Wozniak, entworfen wurde. IBM konterte 1981 mit dem als Personal Computer PC bezeichneten Modell (Abb. 2.3), das für einige Jahre als der „PC“ galt. Viele kleinere Firmen bauten PC-kompatible Modelle. Es verwendete DOS 1.0 von Microsoft als Betriebssystem. Das ursprüngliche Betriebssystem kaufte Microsoft für schätzbare 50.000 USD und machte Millionen, wenn nicht Milliarden damit..

Nebeneinander existieren heute:

- Großrechner (Mainframes),
- Workstations und
- Personal Computers (PC) sowie
- Personal Devices (Smartphone, Spielkonsolen, Tablets etc.).

Abb. 2.3 Personal Computer von IBM (Wikipedia contributors, 2022). ©Ruben de Rijcke, CC-BY-SA 3.0



Durch die nun mögliche Vernetzung wurden Aufgabenteilungen wie im Client-Server-Konzept möglich, die neben die Rechenzentren traten. Letztere haben dank der „Cloud“-Entwicklung wieder Morgenluft bekommen.

2.2 Software

Es scheint nicht klar zu sein, wann der Begriff „Software“ Eingang in den Sprachgebrauch gefunden hat; vermutlich Ende der 1950er-Jahre und nachdem der Begriff „Hardware“ gebräuchlich wurde. Das Paar Hard- und Software wurde von Howard Aiken, der die Mark I entwickelt hatte, und Grace M. Hopper, die als Mathematikerin die Anweisungen auf Lochkarten brachte, verkörpert. Grace merkte schnell, dass wiederkehrende Nebenrechnungen häufig auftraten, die aber erst mit einem Speicher wiederverwertet werden konnten. Diese Nebenrechnungen nannte man dann „Subroutines“. Hopper stellte ein Programm zusammen, das in der Lage war, diese gespeicherten Subroutinen wiederholt einzulesen. Sie nannte dieses Programm „Compiler“. Verbesserte Versionen des Compilers wurden dann den Käufern mit der Hardware ausgeliefert.

Im Jahr 1957 wurde FORTRAN von IBM vorgestellt, eine höhere Programmiersprache. Der Name leitet sich von Formula-Translator ab, womit auch schon etwas über die Verwendung als eher mathematischer Problemlöser gesagt ist. LISP, heute noch verwendet, und ALGOL sind weitere, bei Akademikern damals beliebte Sprachen. Auf Bestreben des U.S. Department of Defense wurde COBOL 1959 entwickelt, das dem Aufbau der englischen Sprache glich.

Mit der wachsenden Komplexität und Leistungsfähigkeit der Rechner wurde es für den Benutzer zusehends schwieriger, alle Verwaltungs- und Überwachungsaufgaben im Betrieb zu erledigen. Hardwareproduzenten und Benutzergruppen stellten kleinere und größere Programme zusammen, um den Betrieb zu vereinfachen. Solche „Systemsoftware“ verwandelte sich dann in die sogenannten Betriebssysteme oder englisch „Operating Systems“. Ende der 1960er-Jahre entstand das Betriebssystem UNIX, zuerst für den internen Gebrauch der Bell Labs, dann mit der zusammen entwickelten Sprache C neu implementiert. Durch die Nähe von UNIX mit C eignet sich diese Sprache sowohl für systemnahe als auch für problemnahe Programme.

UNIX ist modular und damit nicht sehr einheitlich aufgebaut. Dieses Betriebssystem erlaubte es mit dem Timesharing, der Zuteilung von Rechenleistung auf mehrere, an Terminalen sitzende Benutzer, beinahe moderne Verhältnisse zu erlangen. Jeder hatte den Eindruck, an einem eigenen Gerät zu arbeiten. C ist eine der am weitesten verbreiteten Programmiersprachen. UNIX blieb eine lizenzierte Software von AT&T, während an der Universität von Berkeley eine freie Distribution unter dem Kürzel BSD für „Berkeley Systems Distribution“ entstand. Wie man sich leicht vorstellen kann, kamen sich die kommerziellen Interessen in die Quere, was zu gerichtlichen Verfahren führte. Um die Geschichte kurz zu halten: Es entstanden mehrere proprietäre Systeme (AIX, Solaris etc.). Im Jahr 1991 ent-

wickelte der Finne Linus Torvalds ein Programm, das sich als Betriebssystem herausstellte. Im Namen ist ein Verweis auf UNIX impliziert, da es ähnlich aufgebaut ist. Um den Kern des Programms sind verschiedene Distributionen entstanden.

Zurück in der Zeit: 1968 kündigte IBM an, künftig Software unabhängig von der Hardware zu verkaufen. Damit startet der Software-Markt und die Nachfrage nach Portabilität.

Die frühen Sprachen sind typischerweise *Top-Down-Ansätze*. Ein Problem wird in kleinere Portionen mit Wiederverwertbarkeit oder Bibliotheken zerlegt. Das liegt auch der prozeduralen, ablaufgesteuerten Programmierung zugrunde.

Im Laufe der 1980er-Jahre gewannen *Bottom-up-Ansätze* für die Programmiersprachen an Bedeutung. Dies war ebenfalls der gestiegenen Komplexität geschuldet. Ausgangspunkt war damals das Objekt, das die Eigenschaften einer sogenannten Klasse aufweist. Das Objekt zeichnet sich durch Strukturelemente wie *Daten* und funktionale Elemente wie *Methoden* aus. Ein Fahrzeug beispielsweise besitzt einen Fahrer, eine Farbe und Leistungsmerkmale und kann fahren, stoppen, zuladen etc. Die Objekte sind der Ausgangspunkt für die Programmierung.

Die grundlegenden Ideen der Objektorientierung stammen von der Sprache SIMULA und deren Nachfolgerin SMALLTALK und flossen um 1983 in einer Vermengung von C in die verbreitete Sprache C++ und `Objective C` ein. Die Referenz auf C ist eindeutig. Im Jahr 1995 entstand dann daraus die Sprache JAVA. Etwa gleichzeitig entstand Python, das eine recht breite Anwendergemeinschaft gefunden hat. Die Applikationen in Android werden mit JAVA realisiert, Apple verwendet für das iPhone `Objective C`.

In den letzten Jahren haben sich auch wieder neue Sprachen entwickelt. Zu nennen sind *Go* oder Golang von Google, *Swift* von Apple und *Hack* von Facebook. Go ist nicht objektorientiert, aber es kann gut mit gleichzeitig ausführbaren Elementen (Concurrency) umgehen.

Weitere Details findet man bei Petzold (1999).

Quiz zu Kap. 2

Quiz

Wie steht es mit einem Selbsttest?

1. Wann wurde der erste elektronische Computer in Betrieb genommen?
2. Wie wurden die Schaltungen realisiert?
3. Welche Bedeutung kommt den Transistoren und der Miniaturisierung zu?
4. Wie entstand die Unterscheidung in Hard- und Software?
5. Wann kam der PC auf den Markt?
6. Wofür steht FORTRAN?
7. Welches sind aktuell die gebräuchlichsten Programmiersprachen?
8. Was ist Objektorientierung?

9. Was unterscheidet C++ von C?
10. Welche Sprachen kommen in Android und im iPhone zur Anwendung?

Literatur

- Petzold, C. (1999). *Code : the hidden language of computer hardware and software*. Microsoft Press.
- Wikipedia. (2016). UNIVAC I – Wikipedia, Die freie Enzyklopädie. https://de.wikipedia.org/w/index.php?title=UNIVAC_I&oldid=154306265. Zugegriffen: 25. Aug. 2018.
- Wikipedia. (2018). Curta – Wikipedia, Die freie Enzyklopädie. <https://de.wikipedia.org/w/index.php?title=Curta&oldid=178429295>. Zugegriffen: 1. Okt. 2022.
- Wikipedia contributors. (2022). IBM Personal Computer – Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=IBM_Personal_Computer&oldid=861836470. Zugegriffen: 1. Okt. 2022.